CHAPTER 17

Configuring the 2-Port and 4-Port Channelized T3 SPAs

This chapter provides information about configuring the 2-Port and 4-Port Channelized T3 Shared Port Adapters (SPAs) on the Catalyst 6500 Series switch. It includes the following sections:

For information about managing your system images and configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

For more information about the commands used in this chapter, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication. Also refer to the related Cisco IOS Release 12.2 software command reference and master index publications. For more information about accessing these publications, see the "Related Documentation" section on page xliv.

Configuration Tasks

This section describes how to configure the serial SPAs for the Catalyst 6500 Series switch and includes information about verifying the configuration.

It includes the following topics:

Required Configuration Tasks

This section lists the required configuration steps to configure the 2-Port and 4-Port Channelized T3 SPA. Some of the required configuration commands implement default values that might be appropriate for your network.

Note To better understand the address format used to specify the physical location of the SPA Interface Processor (SIP), SPA, and interfaces, see the "Specifying the Interface Address on a SPA" section on page 17-7.

Configuring the T3 Controller

To configure the T3 controller for the 2-Port and 4-Port Channelized T3 SPA, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t3` *slot/subslot/port*	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the CT3 SPA port. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)# [no] channelized`	(Optional) Specifies the channelization mode. • **channelized**—In channelized mode, the T3 link can be channelized into 28 T1s, and each T1 can be further channelized into 24 DS0s. This is the default. • **no channelized**—In the unchannelized mode, the T3 link provides a single high-speed data channel of 44210 kbps.

	Command	Purpose
Step 4	`Router(config-controller)# framing {auto-detect | c-bit | m23}`	(Optional) Specifies the framing type in channelized mode. • **auto-detect**—Detects the framing type at the device at the end of the line and switches to that framing type. If both devices are set to auto-detect, c-bit framing is used. • **c-bit**—Specifies C-bit parity framing. This is the default. • **m23**—Specifies M23 framing. **Note** To set the framing type for an unchannelized T3, see the "Configuring T3 Framing" section on page 17-13.
Step 5	`Router(config-controller)# clock source {internal | line}`	(Optional) Specifies the clock source. • **internal**—Specifies that the internal clock source is used. Default for channelized mode. • **line**—Specifies that the network clock source is used. Default for unchannelized mode.
Step 6	`Router(config-controller)# cablelength length`	(Optional) Specifies the cable length. • *length*—Range is 0-450 feet. The default is 224 feet.

Configuring the Logical T1 Interfaces

If channelized mode is configured for the T3 controller, perform this task to configure the logical T1 interfaces:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t3 slot/subslot/port`	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the CT3 SPA port. See the "Specifying the Interface Address on a SPA" section on page 17-7.

	Command	Purpose
Step 3	`Router(config-controller)# t1 t1-number channel-group channel-number timeslots range [speed {56 | 64}]`	Specifies the T1 channel and timeslots to be mapped to each channel. • *t1-number*—T1 number from 1–28. • *channel-number*—Specifies a channel-group mapping(0–23) under the designated T1. • *range*—List of timeslots under the channel-group. Timeslots assigned to this T1 can be 1–24 or a combination of subranges within 1– 24. You can indicate a range using a hyphen, commas, or a combination of both. One timeslot equals one DS0. • **speed 56** or **64**— Specifies the speed of a timeslot as either 56 or 64 kbps. The default speed of 64 kbps is not mentioned in the configuration.
Step 4	`Router(config-controller)# t1 t1-number framing {esf | sf [hdlc-idle {0x7e | 0xff}] [mode {j1}]}`	(Optional) Specifies the T1 framing type using the **framing** command. • **sf**—Specifies Super Frame as the T1 frame type. **Note** If you select sf framing, you should consider disabling yellow alarm detection because the yellow alarm can be incorrectly detected with sf framing. • **esf**—Specifies Extended Super Frame as the T1 frame type. This is the default. • **hdlc-idle**— The hdlc-idle option allows you to set the idle pattern for the T1 interface to either **0x7e** (the default) or **0xff**.
Step 5	`Router(config-controller)# t1 channel-number clock source {internal | line}`	(Optional) Specifies the T1 clock source. • **internal**—Specifies that the internal clock source is used. This is the default. • **line**—Specifies that the network clock source is used.

After configuring a logical T1 interface, configure the serial interfaces. For detailed interface configuration information, see the *Cisco IOS Interface Configuration Guide, Release 12.2*.

Note After a T1 channel is configured, it appears to the Cisco IOS software as a serial interface; therefore, all the configuration commands for a serial interface are available. However, not all commands are applicable to the T1 interface. All the encapsulation formats, such as PPP, HDLC, and Frame Relay are applicable to the configured T1. Encapsulation can be set via the serial interface configuration commands.

Verifying T3 Controller Configuration

Use the **show controllers** command to verify the controller configuration:

```
Router# show controllers t3
T3 3/1/0 is administratively down.
T3 3/1/1 is administratively down.
T3 3/1/2 is up.  Hardware is 4 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Channelized T3
  No alarms detected.
  Framing is M23, Line Code is B3ZS, Clock Source is Internal
  Equipment customer loopback
  Data in current interval (746 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 0 LOS Defect Secs

  T1 1 is up
  timeslots: 1-24
  FDL per AT&T 54016 spec.
  No alarms detected.
  Framing is ESF, Clock Source is Internal
  Data in current interval (177 seconds elapsed):
     0 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, 0 Stuffed Secs
     0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
  Total Data (last 2 15 minute intervals):
     0 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, 0 Stuffed Secs
     0 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs

  T1 2
    Not configured.

  T1 3
    Not configured.
.
.
```

```
.
T3 3/1/3 is up.  Hardware is 4 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  No alarms detected.
  MDL transmission is disabled

  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (657 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     0 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 0 LOS Defect Secs
```

Verifying Interface Configuration

Use the **show interface serial** command to verify the interface configuration. The following example shows the ouput for the serial interface for an unchannelized T3:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions alarm present
  DSU mode 0, bandwidth 44210 Kbit, scramble 0, VC 0
```

The following example shows the output for a serial interface for the first T1 on a channelized T3:

```
Router# show interface serial3/0/1/1:0
Serial3/0/1/1:0 is administratively down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 1500 bytes, BW 832 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
```

```
Keepalive set (10 sec)
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
   0 packets input, 0 bytes, 0 no buffer
   Received 0 broadcasts (0 IP multicast)
   0 runts, 0 giants, 0 throttles
   0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
   0 packets output, 0 bytes, 0 underruns
   0 output errors, 0 collisions, 1 interface resets
   0 output buffer failures, 0 output buffers swapped out
   0 carrier transitions alarm present
VC 1: timeslot(s): 2-14, Transmitter delay 0, non-inverted data
```

Specifying the Interface Address on a SPA

SPA interface ports begin numbering with "0" from left to right. Single-port SPAs use only the port number 0. To configure or monitor SPA interfaces, you need to specify the physical location of the SIP, SPA, and interface in the CLI. The interface address format is *slot*/*subslot*/*port*, where:

- *slot*—Specifies the chassis slot number in the Catalyst 6500 Series switch where the SIP is installed.
- *subslot*—Specifies the secondary slot of the SIP where the SPA is installed.
- *port*—Specifies the number of the individual interface port on a SPA.

The following example shows how to specify the first interface (0) on a SPA installed in the first subslot of a SIP (0) installed in chassis slot 3:

```
Router(config)# interface serial 3/0/0
```

This command shows a serial SPA as a representative example, however the same *slot*/*subslot*/*port* format is similarly used for other SPAs (such as ATM and POS) and other non-channelized SPAs.

For the 4-Port Channelized T3 SPA, the interface address format is *slot/subslot/port/t1-number*:*channel-group*, where:

- *t1-number*—Specifies the logical T1 number in channelized mode.
- *channel-group*—Specifies the logical channel group assigned to the timeslots within the T1 link.

For more information about identifying slots and subslots, see the "Identifying Slots and Subslots for SIPs, SSCs, and SPAs" section on page 4-2.

Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your serial SPA.

Note For additional command output details, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

Configuring the Data Service Unit Mode

Configure the SPA to connect with customer premise Data Service Units (DSUs) by setting the DSU mode. Subrating a T3 or E3 interface reduces the peak access rate by limiting the data transfer rate. To configure the Data Service Unit (DSU) mode, perform this task:

z

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port*	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the controller. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# dsu mode {0 | 1 | 2 | 3 | 4}`	Specifies the interoperability mode used by the T3 controller. • **0**—Connects a T3 controller to another T3 controller or to a Digital Link DSU. Bandwidth range is from 300 to 44210 kbps. This is the default. • **1**—Connects a T3 controller to a Kentrox DSU. Bandwidth range is from 1500 to 35000, or 44210 kbps. **Note** If the bandwidth is set between 35000–44210 kbps, an error message is displayed. • **2**—Connects a T3 controller to a Larscom DSU. Bandwidth range is from 3100 to 44210 kbps. • **3**—Connects a T3 controller to an Adtran T3SU 300. Bandwidth range is from 75 to 44210 kbps. • **4**—Connects a T3 controller to a Verilink HDM 2182. Bandwidth range is from 1500 to 44210 kbps.
Step 4	`Router(config-if)# dsu bandwidth` *kbps*	Specifies the maximum allowable bandwidth. • *kbps*—Bandwidth range is from 1 to 44210 kbps.

Verifying DSU Mode

To display the DSU mode of the controller, enter the **show controllers serial** command:

```
Router# show controllers serial
Serial3/1/0 -
   Framing is c-bit, Clock Source is Internal
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (0 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     0 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
```

```
     0 AIS Defect Secs, 0 LOS Defect Secs

   Transmitter is sending AIS.
.
.
```

Configuring Maintenance Data Link

MDL messages are used to communicate identification information between local and remote ports. The type of information included in MDL messages includes the equipment identification code (EIC), location identification code (LIC), frame identification code (FIC), unit, Path Facility Identification (PFI), port number, and Generator Identification numbers. To configure Maintenance Data Link (MDL), perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# controller t3` *slot/subslot/port*	Selects the controller to configure and enters controller configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)# mdl [string {eic | fic | generator | lic | pfi | port | unit} string}] | [transmit {idle-signal | path | test-signal}]`	Configures the MDL message. • **string eic**—Specifies the Equipment Identification Code; can be up to 10 characters. • **string fic**—Specifies the Frame Identification Code; can be up to 10 characters. • **string generator**—Specifies the Generator number string sent in the MDL Test Signal message; can be up to 38 characters. • **string lic**— Specifies the Location Identification Code; can be up to 11 characters. • **string pfi**—Specifies the Path Facility Identification Code sent in the MDL Path message; can be up to 38 characters. • **string port**—Specifies the Port number string sent in the MDL Idle Signal message; can be up to 38 characters. • **string unit**—Specifies the Unit Identification Code; can be up to 6 characters. • **transmit idle-signal**—Enable MDL Idle-Signal message transmission • **transmit path**—Enable MDL Path message transmission. • **transmit test-signal**—Enable MDL Test-Signal message transmission.

Verifying MDL

To display the MDL settings, enter the **show controller** command:

```
Router# show controller t3 3/0/0
T3 3/0/0 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  Receiver has loss of signal.
  MDL transmission is enabled
     EIC: new, LIC: US, FIC: 23, UNIT: myunit
     Path FI: test pfi
     Idle Signal PORT_NO: New-port
     Test Signal GEN_NO: test-message
  FEAC code received: No code is being received
  Framing is C-BIT Parity, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (869 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     869 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     869 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 870 LOS Defect Secs
```

Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic. To set the encapsulation method, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

Step 2	`For channelized:`
	
	`Router(config)# interface serial` *slot/subslot/port/t1-number:channel-group*
	
	`For unchannelized:`
	
	`Router(config)# interface serial` *slot/subslot/port*
		Selects the interface to configure and enters interface configuration mode.
		• Channelized:
		slot/subslot/port/t1-number:channel-group—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
		• Unchannelized:
		slot/subslot/port—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.

Step 3	`Router(config-if)# encapsulation {hdlc | ppp | frame-relay}`	Set the encapsulation method on the interface. • **hdlc**—High-Level Data Link Control (HDLC) protocol for serial interface. This is the default. • **ppp**—Point-to-Point Protocol (PPP) (for serial interface). • **frame-relay**—Frame Relay (for serial interface).

Verifying Encapsulation

To display the encapsulation method, enter the **show interface serial** command:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 2 interface resets
```

```
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions alarm present
  DSU mode 0, bandwidth 44210 Kbit, scramble 0, VC 0
```

Configuring T3 Framing

To set the T3 framing type, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# framing {c-bit | m13}`	Specifies the framing type in unchannelized mode. • **c-bit**—Specifies C-bit parity framing. This is the default. • **m13**—Specifies DS3 Framing M13 (same as M23).

Verifying Framing

To display the framing type, enter the **show controller** command:

```
Router# show controller t3 3/0/0
T3 3/0/0 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Subrate T3
  Receiver has loss of signal.
  Framing is M13, Line Code is B3ZS, Clock Source is Line
  Equipment customer loopback
  Data in current interval (656 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     666 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 666 LOS Defect Secs
```

Configuring FDL

Facility Data Link (FDL) is a far-end performance reporting tool. In ANSI mode, you can enable 1-second transmissions of performance reports on both ends of the T1 connection. To configure FDL, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-controller)# t1` *number* `fdl {ansi}`	(Optional) Enables FDL. • *number*—Specifies the T1 channel number. • **ansi**—Specifies the FDL bit per the ANSI T1.403 specification.

Verifying FDL

To display the FDL setting, enter the **show controller** command:

```
Router# show controller t3 3/0/1/1
T3 3/0/1 is down.  Hardware is 2 ports CT3 SPA
  ATLAS FPGA version: 0, FREEDM336 version: 0
  TEMUX84(1) version: 0, TEMUX84(1) version: 0
  SUBRATE FPGA version: 0
  Applique type is Channelized T3
  Receiver has loss of signal.
  Framing is M23, Line Code is B3ZS, Clock Source is Internal
  Equipment customer loopback
  Data in current interval (456 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation, 0 P-bit Err Secs
     0 P-bit Severely Err Secs, 0 Severely Err Framing Secs
     456 Unavailable Secs, 0 Line Errored Secs
     0 C-bit Errored Secs, 0 C-bit Severely Errored Secs
     0 Severely Errored Line Secs
     0 Far-End Errored Secs, 0 Far-End Severely Errored Secs
     0 CP-bit Far-end Unavailable Secs
     0 Near-end path failures, 0 Far-end path failures
     0 Far-end code violations, 0 FERF Defect Secs
     0 AIS Defect Secs, 456 LOS Defect Secs

  T1 1 is down
  timeslots: 2-14
  FDL per ANSI T1.403 and AT&T 54016 spec.
  Configured for FDL remotely line looped (bell)
  Transmitter is sending LOF Indication.
  Receiver is getting AIS.
  Framing is ESF, Clock Source is Line
  BERT running on timeslots 2,3,4,5,6,7,8,9,10,11,12,13,14,
  BERT test result (running)
     Test Pattern : All 1's, Status : Not Sync, Sync Detected : 0
     Interval : 2 minute(s), Time Remain : 2 minute(s)
     Bit Errors (since BERT started): 0 bits,
```

```
    Bits Received (since BERT started): 0 Kbits
    Bit Errors (since last sync): 0 bits
    Bits Received (since last sync): 0 Kbits
  Data in current interval (703 seconds elapsed):
    0 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
    713 Unavail Secs, 0 Stuffed Secs
    357 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
```

Configuring Scramble

T3 scrambling is used to assist clock recovery on the receiving end. Scrambling is designed to randomize the pattern of 1s and 0s carried in the physical layer frame. Randomizing the digital bits can prevent continuous, nonvariable bit patterns (long strings of all 1s or all 0s). Several physical layer protocols rely on transitions between 1s and 0s to maintain clocking.

Scrambling can prevent some bit patterns from being mistakenly interpreted as alarms by switches placed between the Data Service Units (DSUs).

To configure scrambling, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port*	Selects the interface to configure and enters interface configuration mode. • *slot/subslot/port*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# scramble [0 | 1]`	Enables scrambling. Scrambling is disabled by default. • Scramble settings: **1**—enabled **0**—disabled

Verifying Scrambling

To display the scramble setting, enter the **show interface serial** command:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
```

```
    0 runts, 0 giants, 0 throttles
             0 parity
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 applique, 4 interface resets
    0 output buffer failures, 0 output buffers swapped out
    1 carrier transitions alarm present
DSU mode 0, bandwidth 44210 Kbit, scramble 1, VC 0
```

Configuring Multilink Point-to-Point Protocol (Hardware-Based)

Multilink Point-to-Point Protocol (MLPPP) allows you to combine T1 or E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1 or E1 lines in each bundle.

MLPPP for T1/E1 Configuration Guidelines

The required conditions are as follows:

- Only T1 or E1 links in a bundle.
- All links on the same SPA.
- Maximum of 12 links in a bundle.

Consider these guidelines about hardware-based MLPPP:

- Only three fragmentation sizes are supported: 128, 256, and 512 bytes.
- Fragmentation is enabled by default, with a default size of 512 bytes.
- Fragmentation size is configured using the **ppp multilink fragment-delay** command after using the **interface multilink** command. Among the three possible fragmentationsizes, the least size satisfying the delay criteria is configured. For example, a 192 byte packet causes a delay of 1 millisecond on a T1 link, so the nearest fragmentation size is 128 bytes.

  Use the **show ppp multilink** command to indicate the MLPPP type and the fragmentation size:

```
Router# show ppp multilink
Multilink1, bundle name is Patriot2
Bundle up for 00:00:13
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned
0 discarded, 0 lost received, 206/255 load
0x0 received sequence, 0x0 sent sequence
Member links: 2 active, 0 inactive (max not set, min not set)
Se4/2/0/1:0, since 00:00:13, no frags rcvd
Se4/2/0/2:0, since 00:00:10, no frags rcvd
Distributed fragmentation on. Fragment size 512.  Multilink in Hardware.
```

Fragmentation is disabled explicitly by using the **no ppp multilink fragmentation** command after using the **interface multilink** command.

Creating a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface multilink` *group-number*	Creates a multilink interface and enter multilink interface mode. • *group-number*—The group number for the multilink bundle.
Step 3	`Router(config-if)# ip address` *address mask*	Sets the IP address for the multilink group. • *address*—The IP address. • *mask*—The IP netmask.

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port/t1-number:channel-group*	Selects the interface to configure and enters interface configuration mode. See the "Specifying the Interface Address on a SPA" section on page 17-7. • *slot/subslot/port/t1-number:channel-group*—Select the interface to configure.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# multilink-group` *group-number*	Assigns the interface to a multilink bundle. • *group-number*—The multilink group number for the T1 or E1 bundle.
Step 5	`Router(config-if)# ppp multilink`	Enables multilink PPP on the interface.
	Repeat these commands for each interface you want to assign to the multilink bundle.	

Configuring Fragmentation Size on an MLPPP Bundle (optional)

To configure the fragmentation size on a multilink ppp bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.

	Command	Purpose
Step 2	`Router(config)# interface multilink` *slot/subslot/port/t1-number:channel-group*	Creates a multilink interface and enters multilink interface mode. • *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	`Router(config-if)# ppp multilink fragment-delay` *delay*	Sets the fragmentation size satisfying the configured delay on the multilink bundle. • *delay*—delay in milliseconds

Disabling the Fragmentation on an MLPPP Bundle (optional)

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface multilink` *group-number*	Creates a multilink interface and enters multilink interface mode. • *group-number*—The group number for the multilink bundle. The range is 1 to 2147483647.
Step 3	`Router(config-if)# no ppp multilink fragmentation`	Disables the fragmentation on the multilink bundle.

Verifying Multilink PPP

To verify the PPP multilinks, enter the **show ppp multilink** command:

```
Router# show ppp multilink
Multilink1, bundle name is mybundle
    Bundle up for 01:40:50
    Bundle is Distributed
    0 lost fragments, 0 reordered, 0 unassigned
    0 discarded, 0 lost received, 1/255 load
    0x0 received sequence, 0x0 sent sequence
Member links: 5 active, 0 inactive (max not set, min not set)
    Se6/0/0/1:0, since 01:40:50, no frags rcvd
    Se6/0/1/1:0, since 01:40:09, no frags rcvd
    Se6/0/3/1:0, since 01:15:44, no frags rcvd
    Se6/0/4/1:0, since 01:03:17, no frags rcvd
    Se6/0/6/1:0, since 01:01:06, no frags rcvd
    Se6/0/6:0, since 01:01:06, no frags rcvd
```

Configuring MLFR for T1/E1

Multilink Frame Relay (MLFR) allows you to combine T1/E1 lines into a bundle that has the combined bandwidth of multiple T1/E1 lines. You choose the number of bundles and the number of T1/E1 lines in each bundle. This allows you to increase the bandwidth of your network links beyond that of a single T1/E1 line.

MLFR for T1/E1 Configuration Guidelines

MLFR will function in hardware if all of the following conditions are met:

- Only T1 or E1 member links.
- All links are on the same SPA.
- Maximum of 12 links in a bundle.

Create a Multilink Bundle

To create a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface mfr` *number*	Configures a multilink Frame Relay bundle interface. • *number*—The number for the Frame Relay bundle.
Step 3	`Router(config-if)# frame-relay multilink bid` *name*	(Optional) Assigns a bundle identification name to a multilink Frame Relay bundle. • *name*—The name for the Frame Relay bundle. **Note** The bundle identification (BID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shutdown** and **no shutdown** commands in interface configuration mode.

Assign an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters global configuration mode.
Step 2	`Router(config)# interface serial` *slot/subslot/port:channel-group*	Selects the interface to assign. • *slot/subslot/port:channel-group*—Specifies the location of the interface. See the "Specifying the Interface Address on a SPA" section on page 17-7.
Step 3	`Router(config-if)# encapsulation frame-relay mfr` *number* [*name*]	Creates a multilink Frame Relay bundle link and associates the link with a bundle. • *number*—The number for the Frame Relay bundle. • *name*—The name for the Frame Relay bundle.

	Command	Purpose
Step 4	`Router(config-if)# frame-relay multilink lid` *name*	(Optional) Assigns a bundle link identification name with a multilink Frame Relay bundle link. • *name*—The name for the Frame Relay bundle. **Note** The bundle link identification (LID) will not go into effect until the interface has gone from the down state to the up state. One way to bring the interface down and back up again is by using the **shut** and **no shut** commands in interface configuration mode.
Step 5	`Router(config-if)# frame-relay multilink hello` *seconds*	(Optional) Configures the interval at which a bundle link will send out hello messages. The default value is 10 seconds. • *seconds*—Number of seconds between hello messages sent out over the multilink bundle.
Step 6	`Router(config-if)# frame-relay multilink ack` *seconds*	(Optional) Configures the number of seconds that a bundle link will wait for a hello message acknowledgment before resending the hello message. The default value is 4 seconds. • *seconds*—Number of seconds a bundle link will wait for a hello message acknowledgment before resending the hello message.
Step 7	`Router(config-if)# frame-relay multilink retry` *number*	(Optional) Configures the maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment. The default value is 2 tries. • *number*—Maximum number of times a bundle link will resend a hello message while waiting for an acknowledgment.

Verifying Multilink Frame Relay

To verify the Frame Relay multilinks, enter the **show frame-relay multilink detailed** command:

```
Router# show frame-relay multilink detailed
Bundle: MFR49, State = down, class = A, fragmentation disabled
 BID = MFR49
 No. of bundle links = 1, Peer's bundle-id =
 Bundle links:

  Serial6/0/0:0, HW state = up, link state = Add_sent, LID = test
    Cause code = none, Ack timer = 4, Hello timer = 10,
    Max retry count = 2, Current count = 0,
    Peer LID = , RTT = 0 ms
    Statistics:
    Add_link sent = 21, Add_link rcv'd = 0,
    Add_link ack sent = 0, Add_link ack rcv'd = 0,
    Add_link rej sent = 0, Add_link rej rcv'd = 0,
    Remove_link sent = 0, Remove_link rcv'd = 0,
    Remove_link_ack sent = 0, Remove_link_ack rcv'd = 0,
    Hello sent = 0, Hello rcv'd = 0,
```

```
Hello_ack sent = 0, Hello_ack rcv'd = 0,
outgoing pak dropped = 0, incoming pak dropped = 0
```

Configuring Multipoint Bridging

Multipoint bridging (MPB) enables the connection of multiple ATM PVCs, Frame Relay PVCs, BCP ports, and WAN Gigabit Ethernet subinterfaces into a single broadcast domain (virtual LAN), together with the LAN ports on that VLAN. This feature enables service providers to add support for Ethernet-based Layer 2 services to the proven technology of their existing ATM and Frame Relay legacy networks. Customers can then use their current VLAN-based networks over the ATM or Frame Relay cloud. This feature also allows service providers to gradually update their core networks to the latest Gigabit Ethernet optical technologies, while still supporting their existing customer base.

For MPB configuration guidelines and restrictions and feature compatibility tables, see the "Configuring Multipoint Bridging" section on page 4-17 of Chapter 4, "Configuring the SIPs and SSC."

Configuring Bridging Control Protocol Support

The Bridging Control Protocol (BCP) enables forwarding of Ethernet frames over SONET networks and provides a high-speed extension of enterprise LAN backbone traffic through a metropolitan area. The implementation of BCP on the SPAs includes support for IEEE 802.1D, IEEE 802.1Q Virtual LAN (VLAN), and high-speed switched LANs.

For BCP configuration guidelines and restrictions and feature compatibility tables, see the "Configuring PPP Bridging Control Protocol Support" section on page 4-18 of Chapter 4, "Configuring the SIPs and SSC."

Configuring BCP on MLPPP

Consider the following guidelines when configuring BCP on MLPPP:

- Only Distributed MLPPP is supported.
- Only channelized interfaces are allowed, and member links must be from the same controller card.
- Only trunk port BCP is supported on MLPPP.
- Bridging can be configured only on the bundle interface.

Note BCP on MLPPP operates only in trunk mode.

Configuring BCP on MLPPP Trunk Mode

To configure BCP on MLPPP trunk mode, perform this task:

	Command	Purpose
Step 1	`Router(config)#interface multilink`	Selects the multilink interface.
Step 2	`Router(config-if)#switchport`	Puts an interface that is in Layer 3 mode into Layer 2 mode for Layer 2 configuration.
Step 3	`Router(config-if)#switchport trunk allowed vlan` *vlan-list*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.

	Command	Purpose
Step 4	`Router(config-if)#`**`switchport mode trunk`**	Configures the router port connected to the switch as a VLAN trunk port.
Step 5	`Router(config-if)#`**`switchport nonegotiate`**	Puts the LAN port into permanent trunking mode but prevents the port from generating DTP frames
Step 6	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 7	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 8	`Router(config-if)#`**`multilink-group 1`**	Assigns this interface to the multilink group.
Step 9	`Router(config-if)#`**`interface Serial1/0/0.1/1/1/1:0`**	Designates a serial interface as a multilink bundle.
Step 10	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 11	`Router(config-if)#`**`encapsulation ppp`**	Enables PPP encapsulation.
Step 12	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 13	`Router(config-if)#`**`multilink-group 1`**	Assigns this interface to the multilink group 1.
Step 14	`Router(config-if)#`**`interface Serial1/0/0.1/1/1/2:0`**	Designates a serial interface as a multilink bundle.
Step 15	`Router(config-if)#`**`no ip address`**	Unassigns the IP address.
Step 16	`Router(config-if)#`**`encapsulation ppp`**	Enables PPP encapsulation.
Step 17	`Router(config-if)#`**`ppp multilink`**	Enables this interface to support MLP.
Step 18	`Router(config-if)#`**`multilink-group 2`**	Assigns this interface to the multilink group 2.
Step 19	`Router(config-if)#`**`shutdown`**	Shuts down an interface.
Step 20	`Router(config-if)#`**`no shutdown`**	Reopens an interface.
Step 21	`Router(config-if)#`**`switchport trunk allowed vlan`** *`vlan-list`*	By default, no VLANs are allowed. Use this command to explicitly allow VLANs; valid values for *vlan-list* are from 1 to 4094.

Verifying BCP on MLPPP Trunk Mode

To display information about Multilink PPP, perform this task in EXEC mode.

Command	Purpose
`Router(config-if)#` **`show ppp multilink`**	Displays information on a multilink group.

The following example provides sample output of the **show ppp multilink** command:

```
Router# show ppp multilink

Multilink1, bundle name is group 1
Bundle is Distributed
0 lost fragments, 0 reordered, 0 unassigned, sequence 0x0/0x0 rcvd/sent
0 discarded, 0 lost received, 1/255 load
Member links: 4 active, 0 inactive (max no set, min not set)
Serial1/0/0/:1
Serial1/0/0/:2
Serial1/0/0/:3
Serial1/0/0/:4
```

Link Fragmentation and Interleaving (LFI) Guidelines

LFI can function by using either FRF.12 or MLPPP. MLPPP LFI operates in both hardware and software while FRF.12 LFI operates only in hardware.

Hardware MLPPP LFI Guidelines

LFI using MLPPP will function only in hardware if there is just one member link in the MLPPP bundle. The link can be a fractional T1 or full T1. Note the following guidelines:

- The **ppp multilink interleave** command must be configured to enable interleaving.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- Fragmentation is enabled by default, and the default size is 512 bytes.
- A policy map having a priority class must be applied to the main interface.
- When hardware-based LFI is enabled, fragmentation counters are not displayed.

FRF.12 LFI Guidelines

LFI using FRF.12 is always performed in hardware. Note the following guidelines:

- The fragmentation is configured at the main interface.
- Only three fragmentation sizes are supported: 128 bytes, 256 bytes, and 512 bytes.
- A policy map having a priority class must be applied to the main interface.

Configuring QoS Features on Serial SPAs

For information about the QoS features supported by the serial SPAs, see the "Configuring QoS Features on a SIP" section on page 4-33 of Chapter 4, "Configuring the SIPs and SSC."

Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), perform this task in privileged EXEC configuration mode:

Command	Purpose
`Router# copy running-config startup-config`	Writes the new configuration to NVRAM.

For more information about managing configuration files, refer to the *Cisco IOS Configuration Fundamentals Configuration Guide, Release 12.2* and *Cisco IOS Configuration Fundamentals Command Reference, Release 12.2* publications.

Verifying the Interface Configuration

Besides using the **show running-configuration** command to display your Catalyst 6500 Series switch configuration settings, you can use the **show interfaces serial** and the **show controllers serial** commands to get detailed information on a per-port basis for your 2-Port and 4-Port Clear Channel T3/E3 SPA.

Verifying Per-Port Interface Status

To find detailed interface information on a per-port basis for the 2-Port and 4-Port Channelized T3 SPA, use the **show interfaces serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

The following example provides sample output for the serial interface on an unchannelized T3:

```
Router# show interface serial3/0/0
Serial3/0/0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 4470 bytes, BW 44210 Kbit, DLY 200 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
              0 parity
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 applique, 4 interface resets
     0 output buffer failures, 0 output buffers swapped out
     1 carrier transitions alarm present
  DSU mode 0, bandwidth 44210 Kbit, scramble 1, VC 0
```

The following example provides sample output for the serial interface on a channelized T3:

```
Router# show interface serial3/0/1/1:0
Serial3/0/1/1:0 is down, line protocol is down
  Hardware is Channelized/ClearChannel CT3 SPA
  MTU 1500 bytes, BW 832 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, crc 16, loopback not set
  Keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
  Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
  Queueing strategy: fifo
  Output queue: 0/40 (size/max)
  5 minute input rate 0 bits/sec, 0 packets/sec
  5 minute output rate 0 bits/sec, 0 packets/sec
     0 packets input, 0 bytes, 0 no buffer
     Received 0 broadcasts (0 IP multicast)
     0 runts, 0 giants, 0 throttles
     0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
```

```
     0 packets output, 0 bytes, 0 underruns
     0 output errors, 0 collisions, 2 interface resets
     0 output buffer failures, 0 output buffers swapped out
     0 carrier transitions alarm present
  VC 1: timeslot(s): 2-14, Transmitter delay 0, non-inverted data
```

To find detailed status and statistical information on a per-port basis for the 2-Port and 4-Port Clear Channel T3/E3 SPA, use the **show controllers serial** command. For a description of the command output, see the *Catalyst 6500 Series Cisco IOS Command Reference, 12.2SX* publication.

The following example provides sample controller statistics for the third port on the SPA located in the first subslot of the SIP-200 that is installed in slot 5 of a Catalyst 6509 switch:

```
Router# show controller serial 5/0/2
Serial5/0/2 -
   Framing is c-bit, Clock Source is Line
   Bandwidth limit is 44210, DSU mode 0, Cable length is 10
   rx FEBE since last clear counter 0, since reset 0
   Data in current interval (807 seconds elapsed):
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 306 Unavailable Secs
     500 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 1:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 2:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     564 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 3:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     562 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
  Data in Interval 4:
     0 Line Code Violations, 0 P-bit Coding Violation
     0 C-bit Coding Violation
     0 P-bit Err Secs, 0 P-bit Sev Err Secs
     0 Sev Err Framing Secs, 0 Unavailable Secs
     560 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs
    .
    .
    .
    Total Data (last 44 15 minute intervals):
         0 Line Code Violations, 0 P-bit Coding Violation,
         0 C-bit Coding Violation,
         0 P-bit Err Secs, 0 P-bit Sev Err Secs,
         0 Sev Err Framing Secs, 0 Unavailable Secs,
         24750 Line Errored Secs, 0 C-bit Errored Secs, 0 C-bit Sev Err Secs

       Transmitter is sending AIS.

       Receiver has loss of signal.

        40434 Sev Err Line Secs, 0 Far-End Err Secs, 0 Far-End Sev Err Secs
        0 P-bit Unavailable Secs, 0 CP-bit Unavailable Secs
```

```
  0 CP-bit Far-end Unavailable Secs
  0 Near-end path failures, 0 Far-end path failures

 No FEAC code is being received
MDL transmission is disabled
```

Configuration Examples

This section includes the following configuration examples:

DSU Configuration Example

The following example sets the DSU mode on interface port 0 on slot 4, subslot 1:

```
! Specify the interface and enter interface configuration mode.
!
Router(config-int)# interface t3 4/1/0
!
!Specifies the interoperability mode used by the T3 interface.
!
Router(config-int)# dsu mode 2
!
!Specifies the maximum allowable bandwidth.

Router(config-int)# dsu bandwidth 23000
```

MDL Configuration Example

The following example configures the MDL strings on controller port 0 on slot 4, subslot 1:

```
! Enter controller configuration mode.
!
Router(config)# controller t3 4/1/0
!
! Specify the mdl strings.
!
Router(config-controller)# mdl string eic beic
Router(config-controller)# mdl string lic beic
Router(config-controller)# mdl string fic bfix
Router(config-controller)# mdl string unit bunit
Router(config-controller)# mdl string pfi bpfi
Router(config-controller)# mdl string port bport
Router(config-controller)# mdl string generator bgen
Router(config-controller)# mdl transmit path
```

```
Router(config-controller)# mdl transmit idle-signal
Router(config-controller)# mdl transmit test-signal
```

Encapsulation Configuration Example

The following example configures encapsulation on a channelized T1 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1/1:0
!
! Specify the encapsulation method.
!
Router(config-if)# encapsulation ppp
```

The following example configures encapsulation and framing on a unchannelized T3 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1
!
! Specify the encapsulation method.
!
Router(config-if)# encapsulation ppp
```

Framing—Unchannelized Mode Configuration Example

The following example configures framing on an unchannelized T3 interface:

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/1
!
! Specify the framing type.
!
Router(config-if)# framing m13
```

Facility Data Link Configuration Example

The following example configures FDL on a channelized T1 interface:

```
! Specify the controller to configure and enter controller configuration mode.
!
Router(config)# controller t3 3/1/0
!
! Specify the T1 controller and set the FDL bit.
!
Router(config-controller)# t1 1 fdl ansi
```

Scrambling Configuration Example

The following example configures scrambling on the T3 interface:

```
! Enter global configuration mode.
!
Router# configure terminal
!
```

```
! Specify the interface to configure and enter interface configuration mode.
!
Router(config)# interface serial 4/1/3
!
! Enable scrambling.
!
Router(config-if)# scrambling
```

Creating a Multilink Bundle Configuration Example

The following example creates a multilink bundle and assigns an IP address:

```
! ! Enter global configuration mode.
!
Router# configure terminal
!
! Create a multilink interface and enter interface configuration mode.
!
Router(config)# interface multilink 1
!
! Specify the IP address for the interface.
!
Router(config-if)# ip address 123.345.678.21 255.255.255.0
!
```

Assigning a T1 Interface to a Multilink Bundle Configuration Example

The following example assigns a T1 interface to a multilink bundle:

```
! ! Enter global configuration mode.
!
Router# configure terminal
!
! Specify the T1 interface and enter interface configuration mode.
!
Router(config)# interface serial 1/0/1/1:0
!
! Specify PPP encapsulation.
!
Router(config-if)# encapsulation ppp
!
! Specify the multilink bundle the T1 will belong to.
!
Router(config-if)# multilink-group 1
!
```