Configuring the Channelized 12-port CT3/T1 Optical Services Modules

This chapter describes how to configure the 12-port channelized/unchannelized DS3 Optical Services Modules (OSM-12CT3/T1).

The chapter consists of these sections:

Understanding the Channelized/Unchannelized CT3/T1 Modules

The OSM-12CT3/T1 module provides 12DS3 interface connections using mini-SMB connectors.

Each DS3 port can be independently configured all the way from a clear channel DS3 down to and including DS1, E1,and DS0 connections on the same port, with a maximum of 128 channels per DS3 port.

Each OSM-12CT3/T1 can support any combination of DS3/DS1/E1/T1s up to 1023 channels. For example, each port can support the following:

- 1 DS3 at 44.7 Mbps
- 21 E1s at 2.048Mbps
- 28 DS1s at 1.5 Mbps
- 128 DS0s at 64 Kbps

An unchannelized DS3 connection provides a single serial interface data channel that may be configured to use all of the DS3 bandwidth or a fractional portion of it. This mode is compatible with several vendors of fractional (subrate) DS3 data service units (DSUs).

A cost, depending on the bandwidth, is associated with a serial interface. The total cost possible for 12 DS3 ports is 4096. That is, the sum of the costs of all the serial interfaces configured on 12 DS3 ports must be no more than 4096. The **show controller t3** command displays the VC cost, which indicates the available cost of 12 DS3 ports. Table 7-1 shows the cost associated with each link type.

Table 7-1 Serial Interface Cost

Link Type	Cost
DS3	336
Unframed E1	16
1-6 DS-/TS	4
7-9 DS0/TS	6
10-16 DS0/TS	8
17-24 DS0/TS	12
25-31 TS	16

Channelized DS3 Overview

In the channelized mode of operation, an OSM-12CT3/T1 link is channelized into 28 DS1 or 21 E1 data lines in an industry standard multiplexing format.

Each of the DS1 lines contain 24 timeslots of 64 or 56 kbps each, and each of the E1 lines contain 32 timeslots of 64 or 56 kbps each. The DS1and E1 lines can support one or more user data channels which appear to the system as serial interfaces. Each serial interface is assigned one or more of the timeslots giving the serial interface a bandwidth of n x 56 kbps or n x 64 kbps, where n is the number of timeslots assigned. Any unused timeslots of the lines are filled with an idle channel pattern.

Unchannelized DS3 Overview

In the unchannelized mode of operation, a DS3 link provides a single high speed user data channel, rather than being multiplexed into 28 DS1 or 21 E1 lines. The data channel appears to the system as a serial interface that may be configured to use the full DS3 bandwidth or a smaller portion of the DS3 bandwidth.

In unchannelized DS3 mode, the OSM-12CT3/T1 supports the maintenance data link (MDL) channel when using c-bit parity framing as well as local and network loopbacks.

Supported Features

The 12-port channelized/unchannelized DS3 OSMs support the following features:

General Features

The general features are as follows:

- Channelized DS3 with 28 DS1 lines or 21 E1 lines multiplexed into a DS3 line.
- C-bit parity, M23, and auto-detect framing.
- Internal clocking configurable.
- Bit error rate test (BERT) diagnostics for each DS-3 channel
- Local, line, and remote loopback.
- Generation and termination of DS-3 maintenance data link (MDL) in C-bit framing.
- Compliance with ANSI T1.231-1993, Bellcore GR820, ANSI T1.107, ANSI T1.403 1989, G.703 Section 2, G.704 Section 3, AT&T Pub. 62411 1990, and Facilities Data Link (ANSI T1.403 and AT&T 54016).

Serial Encapsulation Protocols

The OSMs support for the following serial encapsulation protocols:

- Point-to-Point Protocol (PPP)
- Distributed Multilink PPP (dMLPPP)

  The OSM-12CT3/T1 module supports dMLPPP, which means that the MLPPP encapsulation is performed on the OSM-12CT3/T1 module rather than the route processor. The following dMLPP features are supported:

  – 336 bundles per module
  – A maximum of 12 DS1/E1 channels per bundle
  – All links in a bundle should operate at the same speed
  – 100 ms of differential delay
- High-Level Data Link Control (HDLC)
- Frame Relay

  Configure the OSM-12CT3/T1 interfaces for Frame Relay as described in the *Cisco IOS Wide-Area Networking Configuration Guide, Release 12.1* under “Configuring Frame Relay” and in the *Cisco IOS Wide-Area Networking Command Reference, Release 12.1* at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_c/wcdfrely.htm

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/wan_r/wrdfrely.htm

  Configure traffic shaping for Frame Relay as described in the *Cisco IOS Quality of Service Solutions Configuration Guide* under “Configuring Distributed Traffic Shaping” at:

  http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_c/fqcprt4/qcfdts.htm.

  For information on Multilink Frame Relay (FRF.16), see Multilink Frame Relay (FRF.16), page 1-6.
- MPLS/VPN

  For MPLS/VPN configuration information, see Chapter 11, “Configuring Multiprotocol Label Switching on the Optical Services Modules.”

DSU Mode

Unchannelized DS-3 supports subrate and scrambling formats for the following DSUs.

- Digital Link
- Verilink
- Adtran
- Larscom
- Kentrox

T1 Configuration Options

The T1 lines have the following configuration options:

- Local, line, and remote loopback.
- Generation and termination of DS-3 maintenance data link (MDL) in C-bit framing.
- In T1 mode, the channelized CT3/T1 OSMs support facilities data link (FDL) in Extended Superframe (ESF) framing, as well as various loopbacks. Bit error rate testing (BERT) is supported on each of the T1 lines. It can be done over a framed or unframed DS-1 signal. The OSM-ChOC12C/T1 module has four framers, each connected to three contiguous ports. BER testing can be done simultenously on 6 T1 interfaces in addition to any DS3 interfaces configured on the three ports to which a framer is connected.
- In the channelized mode of operation, a channelized CT3/T1 OSM link can be channelized into maximum of 336 T1 data lines in an industry standard multiplexing format. Each of the T1 lines contains 24 timeslots of 64 or 56 kbps each. The T1 lines can support one or more user data channels, which appear to the system as serial interfaces. Each serial interface is assigned one or more of the timeslots giving the serial interface a bandwidth of *n* x 56 kbps or *n* x 64 kbps, where *n* is the number of timeslots assigned. Any unused timeslots of the T1 are filled with an idle channel pattern.

E1 Configuration Options

The E1 lines have the following configuration options:

- Channelized E1—Any of the 21 E1 lines can be configured as channelized E1 lines, but you are limited to a total of 128 logical channels. You can group the time slots in these E1 lines into several individual logical channel groups, each of which carries data with different data link layer protocol encapsulations. You can configure timeslot 16 as a data channel, although it is typically used for common channel signaling. (Channel associated signaling (CAS) for voice channels and E1 Facilities Data Link [FDL] on timeslot 16 are not supported.)
- Each logical channel group can be composed of individual 64-kbps timeslots and/or ranges of timeslots, for example, 1, 9, 12-14. Each logical channel group can contain from 1-31 timeslots maximum; the same timeslot cannot be used in more than one logical channel group. Any unused timeslots are filled with programmable idle-channel data.
- Fractional E1— Any of the 21 E1 lines can be configured as fractional E1 lines, each of which can be either E1 frames or E1 cyclic redundancy check (CRC) multiframes, as specified by CCITT/ITU G.704 and G.706. A fractional E1 line is a subset of the full E1 bandwidth, which uses *n* x 64 kbps; where *n* is a timeslot in the range of 1-31.

Fractional E1 lines contain only a single logical channel group that can be either a single 64-kbps timeslot or a range of timeslots; for example timeslot 1, or timeslots 15-23. Any unused timeslots are filled with programmable idle-channel data.

Note If you assign only one channel group to an E1 line, it is a fractional E1 line. If you assign more than one channel group to an E1 line, it is a channelized E1 line.

- Unframed E1—Any of the 21 E1 lines can be configured as unframed E1 data lines. Each unframed E1 line contains no framing overhead and is not timeslot divided.

Note For channelized, fractional, and unframed configurations each configured channel group, which might contain individual timeslots and/or ranges of timeslots, uses only one of the 128 available logical channels. For example, if you assign the range of timeslots 3-7 to a channel group, only one logical channel is used. Likewise, if you assign just timeslot 3 to a channel group, only one logical channel is used.

DS3 Alarms

This section lists the supported DS3 alarms:

- Alarm indication signal (AIS)
- Out of frame (OOF)
- Far End Remote Failure (FERF)
- Loss of Signal (LOS)
- Far End Block Errors (FEBE)
- Far-End Alarm and Control (FEAC) support

Network Management

This section lists the supported network management features:

- Local Loopback.
- Network Loopback.
- RFC 1407 MIB support.

Quality of Service Protocols

For information on configuring QoS on the Channelized OSMs, refer to Chapter 9, “Configuring QoS on the Optical Services Modules.”

The following QoS features are supported on the Channelized OSMs:

- Hierarchical traffic shaping for Frame Relay, HDLC, and PPP encapsulations
- PFC2 QoS on the LAN and WAN ports
- Differentiated Services Control Point (DSCP)
- IP Precedence classification

Configure class-based marking as described in the *Class-Based Marking* Feature Module at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121t/121t5/cbpmark2.htm

- Classification and priority marking based on the following:
  - Ethertype
  - IP Source Address (SA)
  - IP Destintation Address (DA)
  - TCP port number
  - UDP port number
  - IP SA + TCP/UDP port number + IP DA + TCP/UDP port number
- Class-based weighted fair queuing (CBWFQ) on the WAN ports.
- Low latency queuing (LLQ) on the WAN ports.

For general information on classification, marking, and queuing in IOS, refer to the "Classificiation" chapter of the *Cisco IOS Quality of Service Solutions Configuration Guide*, Release 12.1 at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_c/index.htm

For information about platform-independent IOS QoS commands, refer to the *Cisco IOS Quality of Service Solutions Command Reference*, Release 12.1at this URL:

http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/qos_r/index.htm

Configuring the Interfaces

This chapter includes these sections:

Configuring the T3 Controller

To configure the controller, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller t3` *slot/port*	Selects a port and enters controller configuration mode.
Step 3	`Router(config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. **no channelized**—configures the interface as unchannelized. **channelized mode t1**—configures the interfaces for T1 channelized mode **channelized mode e1**—configures the interfaces for E1 channelized mode Default is **t1**.
Step 4	`Router(config-controller)# [no] bert pattern [2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1] interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 5	`Router(config-controller)# [no] mdl string {eic | fic | generator | lic | pfi | port | unit}`	(Optional) Specifies the maintenance data link (MDL) messages. **eic**—equipment ID code **fic**—frame ID code **generator**—generator number in MDL test signal **lic**—location ID code **pfi**—facility ID code in MDL path message **port**—port number in MDL idle string message **unit**—unit code Default is **no mdl string**.
Step 6	`Router(config-controller)# [no] mdl transmit {path | idle-signal | test-signal}`	(Optional) Enables MDL message transmission. Default is **no mdl transmit**.
Step 7	`Router(config-controller)# t1 1 bert channel-group 0 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on channel-group 0 (all timeslots under channel group 0).
Step 8	`Router(config-controller)# t1 1 bert timeslots 21,24 pattern 2^11 interval 1`	(Optional) Configures bit error rate testing on timeslots 21 and 24 only.

This example shows how to configure the controller:

```
Router# configure terminal
Router(config)# controller T3 3/0
Router(config-controller)# no channelized
Router(config-controller)# exit
Router(config)#
```

Configuring the Unchannelized DS3 Interface

After you verify the controller configuration, you can configure the associated DS3 interfaces.

Note IP addresses can not be assigned to main interfaces configured for Frame Relay.

To configure the unchannelized DS3 interfaces, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# interface serial slot/port`	Specifies the serial port to configure.
Step 3	`Router(config-if)# framing {c-bit | m23}`	Specifies the framing.
Step 4	`Router(config-if)# [no] dsu bandwidth Kilobits/sec`	Sets the DSU subrate bandwidth.
	`Router(config-if)# [no] dsu mode {0|1|2|3|4}`	Specifies the DSU mode: 0–Digital-Link 1–Kentrox 2–Larscom 3–Adtran 4–Verilink
Step 5	`Router(config-if)# [no] dsu remote [accept | fullrate]`	Specifies if the local (near-end) interface will accept incoming requests from the remote (far-end) interface, or if the local interface will request that the remote interface set its bandwidth to fullrate.
Step 6	`Router(config-if)# encapsulation hdlc | ppp`	Specifies the encapsulation type. HDLC is the default encapsulation.
Step 7	`Router(config-if)# [no] clock source {internal | line}`	(Optional) Specifies the clock source. Default is **line**.
Step 8	`Router(config-if)# [no] loopback {local | network | remote {line | payload}}`	(Optional) Sets the loopback mode. Default is **no loopback**.
Step 9	`Router(config-if)# [no] cablelength feet`	Specifies the cable length. Default is 224.
Step 10	`Router(config-if)# [no] keepalive`	Turns on and off keepalive messages.
Step 11	`Router(config-if)# no shutdown`	Enables the interface.

This is an example of an unchannelized DS3 interface configuration:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller t3 3/0
Router (config-controller)# no channelized
Router (config-controller)# exit
Router (config)# interface serial 1/0
Router (config-if)# dsu bandwidth 16000
Router (config-if)# encapsulation ppp
Router (config-if)# no shutdown
Router (config-if)# exit
Router(config)#
```

Configuring the Channelized DS3 Interface

The DS3 Interface can be channelized down to T1 and E1 links. These sections provide configuration information for T1 and E1 link configuration:

- Configuring the T3 Controller for Channelization, page 7-9
- Configuring the T1/NxDS0 Lines, page 7-9
- Configuring the E1 Lines, page 7-10

Note Bit error rate testing (BERT) is supported on each of the T1 and E1 lines. It can be done over a framed or unframed DS-1 signal. The OSM-12CT3/T1 module has four framers, each connected to three contiguous ports. BER testing can be done simultenously on 6 T1 or E1 interfaces in addition to any DS3 interfaces configured on the three ports to which a framer is connected.

Note BERT channel-groups and timeslots are also supported.

Configuring the T3 Controller for Channelization

To configure the T3 controller for T1 of E1 channelization, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller` *slot/port*	Selects the controller.
Step 3	`Router (config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. Default is **t1**.

This example configures the T3 controller for T1 channelization mode:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller T3 2/0
Router(config-controller)# channelized mode t1
Router(config-controller)# CNTL/Z
Router#
```

Configuring the T1/NxDS0 Lines

To configure T1/NxDS0 lines, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller t3 slot/port`	Enters controller configuration mode.
Step 3	`Router(config-controller)# [no] t1 t1-number channel-group channel-group-number timeslots list-of-timeslots [speed 56 | 64]`	Configures the channel group.
Step 4	`Router (config-if)# [no] t1 t1-number framing {sf | esf}`	Specifies the framing.
Step 5	`Router (config-controller)# [no] t1 t1-number fdl ansi`	(Optional) Enables the once-second transmission of the remote performance reports via facility data link (FDL).
Step 6	`Router (config-controller)# [no] t1 t1-number yellow {detection | generation}`	(Optional) Enables detection and generation of T1 yellow alarms.
Step 7	`Router (config-controller)# [no] t1 t1-number clock source {internal | line}`	(Optional) Defines clock source for the specified T1 line.
Step 8	`Router (config-controller)# [no] t1 t1-number loopback {local | network | remote {line fdl {ansi | bellcore}| payload fdl ansi}}`	(Optional) Sets the loopback mode.
Step 9	`Router (config-controller)# [no] [e1 | t1] [e1-number | t1 number] bert pattern {2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1} interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 10	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 11	`Router(config)# [no] interface serial slot/port/t1 number: channel-group number`	Selects the interface and configures the channel group.

This example configures a T1/NxDS0 line:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# controller T3 3/1
Router(config-controller)# t1 6 channel-group 0 timeslots 1-5, 20-23
Router(config-controller)# t1 6 framing sf
Router(config-controller)# t1 6 clock source line
Router(config-controller)# t1 6 loopback remote line
Router(config-controller)# t1 6 bert pattern 2^23 interval 5 unframed
Router(config-controller)# exit
Router(config)# interface serial t1 3/1/6:0
Router (config-if)# CNTL/Z
Router#
```

Configuring the E1 Lines

To configure E1 lines, perform this task:

	Command	Purpose
Step 1	`Router# configure terminal`	Enters configuration mode and specifies that the console terminal is the source of the configuration subcommands.
Step 2	`Router(config)# controller slot/port`	Selects the controller.
Step 3	`Router (config-controller)# [no] channelized [mode {t1 | e1}]`	Specifies the channelization mode. Default is **t1**.
Step 4	`Router (config-controller)# [no] e1 e1number clock source {internal | line}`	Defines clock source for the specified E1 line.
Step 5	`Router (config-controller)# [no] e1 e1 number unframed`	Configures the E1 line as unframed.
Step 6	`Router (config-controller)# [no] e1 e1 number framing {crc4 | no-crc4}`	Specifies the framing.
Step 7	`Router (config-controller)# [no] e1 e1 number national bits pattern`	Specifies the national bits reserved for country-specific control information.
Step 8	`Router (config-controller)# [no] e1 e1 number channel-group channel-group number timeslots list-of-timeslots speed [56 | 64]`	Configures a channel-group.
Step 9	`Router (config-controller)# [no] e1 e1 number loopback {local | network}`	(Optional) Sets the loopback mode.
Step 10	`Router (config-controller)# [no] [e1 | t1] [e1 number | t1 number] bert pattern {2^11 | 2^15 | 2^20 O.153 | 2^20 QRSS | 2^23 | 0s | 1s | alt-0-1} interval [1-1440]`	(Optional) Configures bit-error-rate (BER) testing.
Step 11	`Router(config-controller)# exit`	Exits controller configuration mode.
Step 12	`Router(config)# [no] interface serial slot/port/t1 number: channel-group number`	Selects the interface and configures the channel group.

This example configures an E1 line:

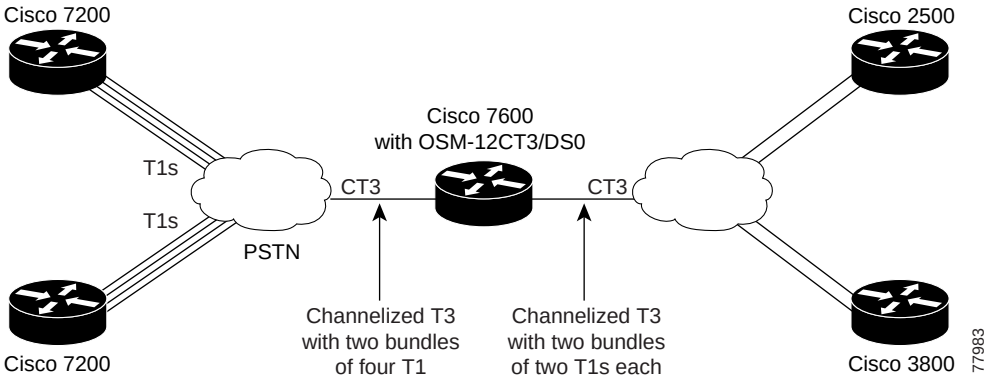
```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router)# controller T3 4/1
Router(config-controller)# channelized mode e1
Router(config-controller)# e1 10 channel-group 0 timeslots 1-5, 20-23
Router(config-controller)# e1 10 framing crc4
Router(config-controller)# e1 10 national bits
Router(config-controller)# e1 10 clock source line
Router(config-controller)# e1 10 loopback network line
Router(config-controller)# e1 10 bert pattern 2^23 interval 5 unframed
Router (config-controller)# exit
Router(config)# interface serial e1 4/0/10:0
Router(config-if)# CNTL/Z
Router#
```

Configuring Distributed MLPPP

Distributed MLPPP allows you to combine T1 or E1 lines into a bundle that has the combined bandwidth of multiple T1 or E1 lines. Bundling T1 or E1 lines enables you to increase the bandwidth of your network links beyond that of a single T1 or E1 line without having to purchase a T3 line.

Figure 7-1 shows a network using an MLPPP link. The Cisco 7600 Series router is connected to the network with an OSM-12CT3/T1 module. An OSM-12CT3/DS interface has been configured with MLPPP to carry two bundles of four T1 lines each. Each of these bundles goes out to separate remote Cisco 7200 series routers, which each have one MLPPP bundle of four T1 lines. Another OSM-12CT3/DS interface has been configured with MLPPP to carry two bundles of two T1 lines. One of these bundles goes out to a Cisco 2500 series router and the other goes out to a Cisco 3800 series router.

Figure 7-1 MLPPP Topology



Creating a Multilink Bundle

To create a multilink bundle, perform this task beginning in global configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface multilink` *bundle-id*	Statically configures link bundles. **Note** The Multilink PPP command is available on serial interfaces only.
Step 2	`Router(config-if)# ip address` *address mask*	Assigns an IP address to the multilink interface.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# ppp multilink`	(Optional) Enables Multilink PPP.
Step 5	`Router(config-if)# [no] multilink-group` *bundle-id*	(Optional) Designates an interface as part of a multilink bundle. *bundle-id* is the bundle number created with the **interface multilink** *bundle-id* command.

The following example shows how to create a multilink bundle:

```
Router(config)# interface multilink 1
Router(config-if)# ip address 10.0.0.0 10.255.255.255
Router(config-if)# encapsulation ppp
Router(config-if)# ppp multilink
Router(config-if)# multilink-group 1
Router(config-if)# CNTL/Z
Router#
```

Assigning an Interface to a Multilink Bundle

To assign an interface to a multilink bundle, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config)# interface serial slot/port/DS0 number: channel-group number`	Selects the interface.
Step 2	`Router(config-if)# no ip address`	Removes any specified IP address.
Step 3	`Router(config-if)# encapsulation ppp`	Enables PPP encapsulation.
Step 4	`Router(config-if)# multilink-group bundle-id`	Designates an interface as part of a multilink bundle. *bundle-id* is the bundle number created with the **interface multilink** *bundle-id* command.
Step 5	`Router(config-if)# ppp authentication chap`	(Optional) Enables Challenge Handshake Authentication Protocol (CHAP) authentication.
Step 6	`Router(config-if)# ppp chap hostname name`	(Optional) Enables the presence of multiple bundles between two hosts.

The following example shows how to create a multilink interface and add it to a multilink bundle:

```
Router(config)# interface serial 1/0/0:1
Router(config-if)# no ip address
Router(config-if)# encapsulation ppp
Router(config-if)# multilink-group 1
Router(config-if)# CNTL/Z
Router#
```

Enabling PPP Multilink Fragmentation

By default, PPP multilink fragmentation is enabled. To enable PPP multilink fragmentation after it has been disabled, perform this task in interface configuration mode:

Command	Purpose
`Router(config-if)# [no] ppp multilink fragmentation`	Enables fragmentation. By default fragmentation is enabled.
`Router(config-if)# [no] multilink fragment-delay milliseconds`	Specifies the tolerable delay in sending the fragment on the multilink. This delay is used to compute fragment size of the packet. Three supported fragment sizes are 128, 256, 512. Default delay is 30ms.

Enabling fragmentation reduces the delay latency among bundle links, but adds some load to the CPU. Disabling fragmentation may result in better throughput.

If your data traffic is consistently of a similar size, we recommend disabling fragmentation. In this case, the benefits of fragmentation may be outweighed by the added load on the CPU.

Configuring Multilink PPP Minimum Links Mandatory

To configure the Multilink PPP Minimum Links Mandatory feature, use the following commands in interface configuration mode:

	Command	Purpose
Step 1	`Router(config-if)# ppp multilink`	Enables Multilink PPP.
Step 2	`Router(config-if)# multilink min-links` *links* `mandatory`	Specifies the required minimum number of links in a Multilink PPP (MLP) bundle. If the minimum number of links in the MLP bundle falls below the number specified by the *links* attribute, the MLP bundle is disabled. • *links*—Minimum number of links, in the range 0 to 12. • The **mandatory** keyword is required on the OSM-12CT3/T1.

This example shows how to configure the Multilink PPP Minimum Links Mandatory feature:

```
Router(config-if)# ppp multilink
Router(config-if)# multilink min-links 5 mandatory
```

For additional information about the Multilink PPP Minimum Links Mandatory feature, refer to this URL:

```
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121newft/121limit/121e/121e11/12e_mlp.htm#xtocid0
```

The commands listed at this URL are all supported on the OSM-12CT3/T1, except the following:

- **multilink max-link**
- **multilink load-threshold**
- **multilink max-fragments**
- **debug ppp multilink fragments**

Verifying the Configuration

Use the **show ppp multilink** command to display information about the newly created multilink bundle:

```
Router# show ppp multilink

Multilink1, bundle name is group1
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 not set, min not set)
 Serial1/0/0:1
```

Creating a Multilink Interface Example

```
Interface multilink1
 ip address 100.0.0.1 255.255.255.0

interface serial6/12/1:0
 no ip address
 encapsulation ppp
```

```
 ppp chap hostname mul1
 multilink-group 1

interface serial6/12/2:0
 no ip address
 encapsulation ppp
 ppp chap hostname mul1
 multilink-group 1

interface serial6/12/3:0
 no ip address
 encapsulation ppp
 ppp chap hostname mul1
 multilink-group 1

show ppp multilink

Multilink1, bundle name is mul1
  Bundle up for 19:58:37
  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:3 active, 0 inactive (max not set, min not set)
    Serial6/12/1:0, since 19:58:37, no frags rcvd  384 weight
    Serial6/12/2:0, since 19:58:37, no frags rcvd  384 weight
    Serial6/12/3:0, since 19:58:37, no frags rcvd  384 weight
```