Configuring T1/E1 Interfaces

This chapter provides information about configuring the T1/E1 interface module on the chassis. It includes the following sections:

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

For more information about the commands used in this chapter, refer to the Cisco IOS Command Reference publication for your Cisco IOS software release.

- Configuration Tasks, on page 1
- Verifying the Interface Configuration, on page 18
- Configuration Examples, on page 19

Configuration Tasks

This section describes how to configure the T1/E1 interface module for the chassis and includes the following topics:

Limitations

This section describes the software limitations that apply when configuring the T1/E1 interface module.

- The following interface modules are not supported on the RSP3 module:
  - 16-port T1/E1 interface module
  - 8-portT1/E1 interface module
  - 32-portT1/E1 interface module
- The `configure replace` command is not supported on the T1/E1 interface modules.
- The chassis does not support more than 16 IMA groups on each T1/E1 interface module.
- The chassis only supports the following BERT patterns: $2^{11}$, $2^{15}$, $2^{20}$-O153, and $2^{20}$-QRSS.
- L2TPv3 encapsulation is not supported.
- Replacing a configured interface module with a different interface module in the same slot is not supported.
• Mixed configurations of features are not supported on the same port.
• The Payload calculation per unit for T1/E1 interface module is:
  • Framed E1 / T1 with no. of time Slots less than 4 -> Payload = 4 x no. of time slots.
  • Framed E1 / T1 with no. of Time Slots greater than or equal 4 -> Payload = 2 x no. of time slots.
  • Unframed T1, C11 → Payload = 48 (2 x 24 (all slots)).
  • Unframed E1, C12 → Payload = 64 (2 x 32 (all slots)).
• Channelization is not supported for serial interfaces. However, channelization is supported for CEM at the DS0 level.

Required Configuration Tasks

This section lists the required configuration steps to configure the T1/E1 interface module. Some of the required configuration commands implement default values that might be appropriate for your network. If the default value is correct for your network, then you do not need to configure the command.

Setting the Card Type

The interface module is not functional until the card type is set. Information about the interface module is not indicated in the output of any show commands until the card type has been set. There is no default card type.

Mixing of T1 and E1 interface types is not supported. All ports on the interface module must be of the same type.

To set the card type for the T1/E1 interface module, complete these steps:

SUMMARY STEPS

1. `configure terminal`
2. `card type {e1 | t1} slot/subslot`
3. `exit`

DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td>Sets the serial mode for the interface module:</td>
</tr>
<tr>
<td>`card type {e1</td>
<td>t1} slot/subslot`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# card type e1 0/3</code></td>
<td></td>
</tr>
</tbody>
</table>
### Configuring T1/E1 Interfaces

#### Enabling T1 Controller

T1/T3 or E1/E3 does not require any license.

To enable T1 controller:

```
enable
cfg-term
controller mediatype 0/4/0
mode t1
end
```

#### Configuring the Controller

To create the interfaces for the T1/E1 interface module, complete these steps:

**SUMMARY STEPS**

1. configure terminal
2. controller {t1 | e1} slot/subslot/port
3. clock source {internal | line}
4. linecode {ami | b8zs | hdb3}
5. For T1 Controllers:
6. cablelength {long | short}
7. exit

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
</tbody>
</table>

Example:

```
Router# configure terminal
```
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 2</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>controller {t1</td>
<td>e1} slot/subslot/port</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>Router(config)# controller t1 0/3/0</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>clock source {internal</td>
<td>line}</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>Router(config-controller)# clock source internal</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><strong>Step 4</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>linecode {ami</td>
<td>b8zs</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>Router(config-controller)# linecode ami</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td><strong>Step 5</strong></td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>For T1 Controllers:</td>
<td>Selects the framing type.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>Router(config-controller)# framing sf</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>For E1 Controllers:</td>
<td>Selects the framing type.</td>
</tr>
<tr>
<td>Example:</td>
<td><strong>Purpose</strong></td>
</tr>
<tr>
<td>Router(config-controller)# framing crc4</td>
<td><strong>Purpose</strong></td>
</tr>
</tbody>
</table>
### Purpose

**Command or Action**

| Step 6 | cablelength {long | short}  
**Example:**  
Router(config-controller)# cablelength long |
|---|---|
| Step 7 | exit  
**Example:**  
Router(config)# exit |

<table>
<thead>
<tr>
<th>Purpose</th>
</tr>
</thead>
</table>
| To fine-tune the pulse of a signal at the receiver for an E1 cable, use the `cablelength` command in controller configuration mode.  
Exits configuration mode and returns to the EXEC command interpreter prompt. |

### Verifying Controller Configuration

To verify the controller configuration, use the `show controllers` command:

```
Router# show controllers t1 0/3/0 brief
T1 0/3/0 is up.  
Applique type is A900-IMA16D  
Cablelength is long gain36 0db  
No alarms detected.  
alarm-trigger is not set  
Soaking time: 3, Clearance time: 10  
AIS State:Clear LOS State:Clear LOF State:Clear  
Framing is ESF, Line Code is B8ZS, Clock Source is Internal.  
Data in current interval (230 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 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs  
Total Data (last 24 hours):  
  136 Line Code Violations, 63 Path Code Violations,  
  0 Slip Secs, 6 Fr Loss Secs, 4 Line Err Secs, 0 Degraded Mins,  
  7 Errored Secs, 1 Bursty Err Secs, 6 Severely Err Secs, 458 Unavail Secs  
  2 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs |
```

### Optional Configurations

There are several standard, but optional, configurations that might be necessary to complete the configuration of your T1/E1 interface module.

### Configuring Framing

Framing is used to synchronize data transmission on the line. Framing allows the hardware to determine when each packet starts and ends. To configure framing, use the following commands.

**SUMMARY STEPS**

1. `configure terminal`
2. `controller {t1 | e1} slot/subslot/port`
3. For T1 controllers  
4. `exit`
# DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example: Router# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller {t1</td>
<td>e1} slot/subslot/port</td>
</tr>
<tr>
<td>Example: Router(config)# controller t1 0/3/0</td>
<td></td>
</tr>
<tr>
<td>· t1—Specifies the T1 controller.</td>
<td></td>
</tr>
<tr>
<td>· e1—Specifies the E1 controller.</td>
<td></td>
</tr>
<tr>
<td>· slot/subslot/port—Specifies the location of the controller.</td>
<td></td>
</tr>
<tr>
<td><strong>Note</strong> The slot number is always 0.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> For T1 controllers</td>
<td>Sets the framing on the interface.</td>
</tr>
<tr>
<td>Example: Router(config-controller)# framing sf</td>
<td></td>
</tr>
<tr>
<td>· sf—Specifies Super Frame as the T1 frame type.</td>
<td></td>
</tr>
<tr>
<td>· esf—Specifies Extended Super Frame as the T1 frame type. This is the default for T1.</td>
<td></td>
</tr>
<tr>
<td>For E1 controllers Example: Router(config-controller)# framing crc4</td>
<td></td>
</tr>
<tr>
<td>· crc4—Specifies CRC4 frame as the E1 frame type. This is the default for E1.</td>
<td></td>
</tr>
<tr>
<td>· no-crc4—Specifies no CRC4 as the E1 frame type.</td>
<td></td>
</tr>
<tr>
<td><strong>Step 4</strong> exit</td>
<td>Exits configuration mode and returns to the EXEC command interpreter prompt.</td>
</tr>
<tr>
<td>Example: Router(config)# exit</td>
<td></td>
</tr>
</tbody>
</table>

## Verifying Framing Configuration

Use the show controllers command to verify the framing configuration:

```
Router# show controllers t1 0/3/0 brief
T1 0/3/0 is up.
   Appliance type is A900-IMA16D
   Cablelength is long gain36 0db
   No alarms detected.
   alarm-trigger is not set
```
Soaking time: 3, Clearance time: 10
AIS State:Clear  LOS State:Clear  LOF State:Clear
Framing is ESF, Line Code is B8ZS
, Clock Source is Line.
Data in current interval (740 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 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs
Total Data (last 24 hours):
- 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 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs

Setting an IP Address

To set an IP address for the serial interface, complete these steps:

You can also set an IP address using an IMA or CEM configuration.

**SUMMARY STEPS**

1. `interface serial 0/subslot/port:channel-group`
2. `ip address address mask`
3. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td></td>
</tr>
<tr>
<td><code>interface serial 0/subslot/port:channel-group</code></td>
<td>Selects the interface to configure from global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# interface serial 0/0/1:0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong></td>
<td></td>
</tr>
<tr>
<td><code>ip address address mask</code></td>
<td>Sets the IP address and subnet mask.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config-if)# ip address 192.0.2.1 255.255.255.0</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong></td>
<td></td>
</tr>
<tr>
<td><code>exit</code></td>
<td>Exits configuration mode and returns to the EXEC command interpreter prompt.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td><code>Router(config)# exit</code></td>
<td></td>
</tr>
</tbody>
</table>
### What to do next

**Note**
IPV4 routing protocols, such as `eigrp`, `ospf`, `bgp`, and `rip`, are supported on serial interfaces.

## Configuring Encapsulation

When traffic crosses a WAN link, the connection needs a Layer 2 protocol to encapsulate traffic.

**Note**
L2TPv3 encapsulation is *not* supported.

To set the encapsulation method, use the following commands:

### SUMMARY STEPS

1. `configure terminal`
2. `interface serial 0/subslot/port:channel-group`
3. `encapsulation {hdlc | ppp}`
4. `exit`

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong></td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td><code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
<tr>
<td><code>Router# configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td></td>
</tr>
</tbody>
</table>

| **Step 2** | Selects the interface to configure from global configuration mode. |
| `interface serial 0/subslot/port:channel-group` | |
| **Example:** | |
| `Router(config)# interface serial 0/0/1:0` |  |
| **Example:** | |

- **subslot** — Specifies the subslot in which the T1/E1 interface module is installed.
- **port** — Specifies the location of the controller. The port range for T1 and E1 is 1 to 16.
- **channel-group** — Specifies the channel group number configured on the controller. For example: interface serial 0/0/1:1.

<p>| <strong>Step 3</strong> | Set the encapsulation method on the interface. |
| <code>encapsulation {hdlc | ppp}</code> | |
| <strong>Example:</strong> | |
| <code>Router(config-if)# encapsulation hdlc</code> | |</p>
<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>retransmission. This is the default for synchronous serial interfaces.</td>
</tr>
<tr>
<td></td>
<td>• <strong>ppp</strong>—Described in RFC 1661, PPP encapsulates network layer protocol information over point-to-point links.</td>
</tr>
</tbody>
</table>

**Step 4**

**exit**

**Example:**

Router(config)# exit

**Verifying Encapsulation**

Use the **show interfaces serial** command to verify encapsulation on the interface:

```
Router# show interfaces serial 0/0/1:0
Serial0/0/1:0 is up, line protocol is up
  Hardware is Multichannel T1
  MTU 1500 bytes, BW 1536 Kbit/sec, DLX 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC
    , crc 16, loopback not set
  Keepalive set (10 sec)
  Last input 00:00:01, output 00:00:02, 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
    60 packets input, 8197 bytes, 0 no buffer
    Received 39 broadcasts (0 IP multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
    64 packets output, 8357 bytes, 0 underruns
    0 output errors, 0 collisions, 0 interface resets
    0 unknown protocol drops
    0 output buffer failures, 0 output buffers swapped out
    1 carrier transitions
```

**Configuring the CRC Size for T1 Interfaces**

All T1/E1 serial interfaces use a 16-bit cyclic redundancy check (CRC) by default, but also support a 32-bit CRC. CRC is an error-checking technique that uses a calculated numeric value to detect errors in transmitted data. The designators 16 and 32 indicate the length (in bits) of the frame check sequence (FCS). A CRC of 32 bits provides more powerful error detection, but adds overhead. Both the sender and receiver must use the same setting.

CRC-16, the most widely used CRC throughout the United States and Europe, is used extensively with WANs. CRC-32 is specified by IEEE 802 and as an option by some point-to-point transmission standards.

To set the length of the cyclic redundancy check (CRC) on a T1 interface, use these commands:
### SUMMARY STEPS

1. configure terminal
2. interface serial 0/subslot/port:channel-group
3. crc {16 | 32}
4. exit

### DETAILED STEPS

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# configure terminal</td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> interface serial 0/subslot/port:channel-group</td>
<td>Selects the interface to configure from global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
</tbody>
</table>
| Router(config)# interface serial 0/0/1:0 | • number — Specifies the location of the controller. The number range for T1 and E1 is 1 to 16.  
  • channel-group — Specifies the channel group number configured on the controller. For example: interface serial 0/1:1. |
| **Step 3** crc {16 | 32} | Selects the CRC size in bits. |
| Example: | |
| Router(config-if)# crc 16 | • 16—16-bit CRC. This is the default.  
  • 32—32-bit CRC. |
| **Step 4** exit | Exits configuration mode and returns to the EXEC command interpreter prompt. |
| Example: | |
| Router(config)# exit | |

### Verifying the CRC Size

Use the `show interfaces serial` command to verify the CRC size set on the interface:

```
Router# show interfaces serial 0/0/1:0  
Serial10/0/1:0 is up, line protocol is up  
Hardware is Multichannel T1  
MTU 1500 bytes, BW 1536 Kbit/sec, 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 00:00:01, output 00:00:02, output hang never  
Last clearing of "show interface" counters never
```
Configuring a Channel Group

Follow these steps to configure a channel group:

**SUMMARY STEPS**

1. `configure terminal`
2. `controller {t1 | e1} slot/subslot/port`
3. `channel-group {t1 | e1} number {timeslots range | unframed} [speed {56 | 64}]`
4. `exit`

**DETAILED STEPS**

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Step 1</strong> configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router# <code>configure terminal</code></td>
<td></td>
</tr>
<tr>
<td><strong>Step 2</strong> controller `{t1</td>
<td>e1} slot/subslot/port`</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config)# controller t1 0/3/0</td>
<td></td>
</tr>
<tr>
<td><strong>Step 3</strong> channel-group `{t1</td>
<td>e1} number {timeslots range</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>Router(config-controller)# channel-group t1 1	imeslots 1</td>
<td>number—Channel-group number. When configuring a T1 data line, channel-group numbers can be values from 1 to 28. When configuring an E1 data line, channel-group numbers can be values from 0 to 30.</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>timeslots range—One or more time slots or ranges of time slots belonging to the channel group. The first time slot is numbered 1. For a T1 controller, the time slot range is from 1 to 24. For an E1 controller, the time slot range is from 1 to 31.</td>
</tr>
<tr>
<td></td>
<td>unframed—Unframed mode (G.703) uses all 32 time slots for data. None of the 32 time slots are used for framing signals.</td>
</tr>
</tbody>
</table>
### Purpose

<table>
<thead>
<tr>
<th>Command or Action</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>speed—(Optional)</td>
<td>Specifies the speed of the underlying DS0s in kilobits per second. Valid values are 56 and 64.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The default is 64. Speed is not mentioned in the configuration.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Each channel group is presented to the system as a serial interface that can be configured individually.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Once a channel group has been created with the channel-group command, the channel group cannot be changed without removing the channel group. To remove a channel group, use the no form of the channel-group command.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>The unframed option is not currently supported.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>DS0-level channelization is not currently supported.</td>
</tr>
</tbody>
</table>

### Step 4

**Example:**

```
Router(config)# exit
```

### Exit

Exits configuration mode and returns to the EXEC command interpreter prompt.

### Saving the Configuration

To save your running configuration to nonvolatile random-access memory (NVRAM), use the following command in privileged EXEC configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>copy running-config startup-config</code></td>
<td>Writes the new configuration to NVRAM.</td>
</tr>
</tbody>
</table>

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

### Troubleshooting E1 and T1 Controllers

You can use the following methods to troubleshoot the E1 and T1 controllers using Cisco IOS software:

- Setting Loopbacks, on page 12
- Running Bit Error Rate Testing, on page 14

### Setting Loopbacks

The following sections describe how to set loopbacks:

---

Configuring T1/E1 Interfaces
Setting a Loopback on the E1 Controller

To set a loopback on the E1 controller, perform the first task followed by any of the following tasks beginning in global configuration mode:

<table>
<thead>
<tr>
<th>Command</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>configure terminal</td>
<td>Enters global configuration mode.</td>
</tr>
<tr>
<td>controller e1 slot/subslot/port</td>
<td>Select the E1 controller and enter controller configuration mode. The slot number is always 0.</td>
</tr>
<tr>
<td>loopback diag</td>
<td>Set a diagnostic loopback on the E1 line.</td>
</tr>
<tr>
<td>loopback network {line</td>
<td>payload}</td>
</tr>
<tr>
<td>end</td>
<td>Exit configuration mode when you have finished configuring the controller.</td>
</tr>
</tbody>
</table>

Setting a Loopback on the T1 Controller

You can use the following loopback commands on the T1 controller in global configuration mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>controller t1 slot/subslot/port</td>
<td>Selects the T1 controller and enter controller configuration mode. The slot number is always 0.</td>
<td></td>
</tr>
<tr>
<td>loopback diag</td>
<td>Sets a diagnostic loopback on the T1 line.</td>
<td></td>
</tr>
<tr>
<td>loopback local {line</td>
<td>payload}</td>
<td>Sets a local loopback on the T1 line. You can select to loopback the line or the payload.</td>
</tr>
<tr>
<td>loopback remote iboc</td>
<td>Sets a remote loopback on the T1 line. This loopback setting will loopback the far end at line or payload, using IBOC (in band bit-orientated code) or the Extended Super Frame (ESF) loopback codes to communicate the request to the far end.</td>
<td></td>
</tr>
<tr>
<td>end</td>
<td>Exits configuration mode when you have finished configuring the controller.</td>
<td></td>
</tr>
</tbody>
</table>

Note

To remove a loopback, use the **no loopback** command.

Table 1: Loopback Descriptions

<table>
<thead>
<tr>
<th>Loopback</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loopback diag</td>
<td>Loops the outgoing transmit signal back to the receive signal. This is done using the diagnostic loopback feature in the interface module’s PMC framer. The interface module transmits AIS in this mode. Set the <strong>clock source</strong> command to <strong>internal</strong> for this loopback mode.</td>
</tr>
</tbody>
</table>
### Loopback

<table>
<thead>
<tr>
<th>Loopback</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>loopback local</td>
<td>Loops the incoming receive signal back out to the transmitter. You can specify whether to use the line or payload.</td>
</tr>
<tr>
<td>local line</td>
<td>The incoming signal is looped back in the interface module using the framer’s line loopback mode. The framer does not reclock or reframe the incoming data. All incoming data is received by the interface module driver.</td>
</tr>
<tr>
<td>local payload</td>
<td>Loops the incoming signal back in the interface module using the payload loopback mode of the framer. The framer reclocks and reframes the incoming data before sending it back out to the network. When in payload loopback mode, an all 1s data pattern is received by the local HDLC receiver and the clock source is automatically set to line (overriding the clock source command). When the payload loopback is ended, the clock source returns to the last setting selected by the clock source command.</td>
</tr>
<tr>
<td>loopback remote iboc</td>
<td>Attempts to set the far-end T1 interface into line loopback. This command sends an in-band bit-oriented code to the far-end to cause it to go into line loopback. This command is available when using ESF or SF framing mode.</td>
</tr>
<tr>
<td>network line</td>
<td>Loops the incoming signal back in the interface module using the line loopback mode of the framer. The framer does not reclock or reframe the incoming data. All incoming data is received by the interface module driver.</td>
</tr>
<tr>
<td>network payload</td>
<td>Loops the incoming signal back using the payload loopback mode of the framer. The framer reclocks and reframes the incoming data before sending it back out to the network. When in payload loopback mode, an all 1s data pattern is received by the local HDLC receiver, and the clock source is automatically set to line (overriding the clock source command). When the payload loopback is ended, the clock source returns to the last setting selected by the clock source command.</td>
</tr>
</tbody>
</table>

---

### Running Bit Error Rate Testing

Bit error rate testing (BERT) is supported on each of the E1 or T1 links. The BERT testing is done only over a framed E1 or T1 signal and can be run only on one port at a time.

The interface modules contain onboard BERT circuitry. With this, the interface module software can send and detect a programmable pattern that is compliant with CCITT/ITU O.151, O.152, and O.153 pseudo-random and repetitive test patterns. BERTs allows you to test cables and signal problems in the field.

When running a BER test, your system expects to receive the same pattern that it is transmitting. To help ensure this, two common options are available:

- Use a loopback somewhere in the link or network
- Configure remote testing equipment to transmit the same BERT test pattern at the same time

To run a BERT on an E1 or T1 controller, perform the following optional tasks beginning in global configuration mode:
### Task

| controller \{e1 | t1\} slot/subslot/port |
|--------------------------------------------------|
| Selects the E1 or T1 controller and enters controller configuration mode. The slot number is always 0. |

| bert pattern 0s | 1s | 2^11 | 2^15 | 2^20-O153 | 2^20-QRSS | 2^23 | alt-0-1 \{ interval minutes \} |
|--------------------------------------------------|
| Specifies the BERT pattern for the E1 or T1 line and the duration of the test in minutes. The valid range is 1 to 1440 minutes. |
| Note Only the 2^11, 2^15, 2^20-O153, and 2^20-QRSS patterns are supported. |

<table>
<thead>
<tr>
<th>end</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exit configuration mode when you have finished configuring the controller.</td>
</tr>
</tbody>
</table>

| show controllers \{e1 | t1\} slot/subslot/port |
|--------------------------------------------------|
| Displays the BERT results. |

The following keywords list different BERT keywords and their descriptions.

---

**Caution**

Currently only the 2^11, 2^15, 2^20-O153, and 2^20-QRSS patterns are supported.

### Table 2: BERT Pattern Descriptions

<table>
<thead>
<tr>
<th>Keyword</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0s</td>
<td>Repeating pattern of zeros (...000...).</td>
</tr>
<tr>
<td>1s</td>
<td>Repeating pattern of ones (...111...).</td>
</tr>
<tr>
<td>2^11</td>
<td>Pseudo-random test pattern that is 2,048 bits in length.</td>
</tr>
<tr>
<td>2^15</td>
<td>Pseudo-random O.151 test pattern that is 32,768 bits in length.</td>
</tr>
<tr>
<td>2^20-O153</td>
<td>Pseudo-random O.153 test pattern that is 1,048,575 bits in length.</td>
</tr>
<tr>
<td>2^20-QRSS</td>
<td>Pseudo-random QRSS O.151 test pattern that is 1,048,575 bits in length.</td>
</tr>
<tr>
<td>2^23</td>
<td>Pseudo-random 0.151 test pattern that is 8,388,607 bits in length.</td>
</tr>
<tr>
<td>alt-0-1</td>
<td>Repeating alternating pattern of zeros and ones (...01010...).</td>
</tr>
</tbody>
</table>

Both the total number of error bits received and the total number of bits received are available for analysis. You can select the testing period from 1 minute to 24 hours, and you can also retrieve the error statistics anytime during the BER test.
To terminate a BERT test during the specified test period, use the `no bert` command.

You can view the results of a BERT test at the following times:

- After you terminate the test using the `no bert` command
- After the test runs completely
- Anytime during the test (in real time)

## Monitoring and Maintaining the T1/E1 Interface Module

After configuring the new interface, you can monitor the status and maintain the interface module by using `show` commands. To display the status of any interface, complete any of the following tasks in EXEC mode:

<table>
<thead>
<tr>
<th>Task</th>
<th>Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>`show controllers {e1</td>
<td>t1} [slot/port-adapter/port/e1-line] [brief]`</td>
</tr>
<tr>
<td><code>show interface serial slot/subslot/port</code></td>
<td>Displays statistics about the serial information for a specific E1 or T1 channel group. Valid values are 0 to 30 for E1 and 0 to 23 for T1.</td>
</tr>
<tr>
<td><code>clear counters serial slot/subslot/port</code></td>
<td>Clears the interface counters</td>
</tr>
</tbody>
</table>

To change the T1/E1 card type configuration, use the `no card type` command and reload the router.

## AIS on Core Failure

AIS stands for Alarm Indication Signal. Prior to Cisco IOS XE Fuji Release 16.7.1, the PDH AIS alarms were generated only when the CE would go down and an event was set in the CEM control-word by the remote provider edge (PE). AIS alarms were not generated when the pseudowire went down. Now, AIS alarm are generated when the pseudowire goes down.

This feature is only supported on the Cisco ASR 900 RSP2 module, for 8-port T1/E1 and 16-port T1/E1 interface modules and only for unframed E1 mode (SAToP) type.

### Limitations of AIS

- AIS is not supported on CESoP and CEM over UDP.
- AIS is not supported on T1 mode. It is only supported on E1 mode.
- AIS is not supported on the 4-port OC3/STM-1 (OC-3) interface module (IM) and 32-port T1/E1 IM.
- AIS is supported only for MPLS core.
- AIS is not supported in pseudowire HSPW mode, when `graceful-restart` command is enabled.
• Removing the MPLS IP address from the core interfaces results in a delay of 10-12 minutes to notify the peer end. This depends on the negotiated forwarding hold timer between the routers, which is the least value of the configured LDP GR forwarding hold timer of the two routers.

• Supported CEM class range of de-jitter buffer size is between 1 to 32 ms.

• If the `shutdown unpowered` command is used to shut down the IM, an OIR must be performed to trigger the AIS alarms..

### Core Failure Event Detection

AIS configuration is used to detect core defects. The core failure is detected in the following events:

- Shutdown of the PE controller or tug level.
- Removing the cross-connect feature.
- Removal of Gigabit Ethernet configuration, CEM configuration, controller configuration, or OSPF configuration.
- Shut on OSPF, CEM group, cross-connect, or Gigabit Ethernet interface.
- CE1 controller shut—AIS alarm is seen on the remote CE.
- PE1 controller shut—AIS alarm is seen on the remote CE.
- PE1 core shut—AIS alarm is seen on both the CEs.
- PE2 core shut—AIS alarm is seen on both the CEs.
- Pesudowire down—AIS alarm is seen on both the CEs.
- Core IGP down—AIS alarm is seen on both the CEs.
- Core LDP down—AIS alarm is seen on both the CEs.

### Configuring AIS for Core Failure

When you enable the AIS, Plesiochronous Digital Hierarchy (PDH) AIS alarm is supported for core failure events on the 8-port T1/E1 and 16-port T1/E1 interface modules. When a core failure is detected due to any event, core flap flag is updated and the core flap event sends an event, which asserts an AIS. When the AIS is not enabled, core failure events are ignored.

Use the following procedure to enable AIS:

```
Router> enable
Router#configure terminal
Router(config)#controller t1 0/1/2
Router(config-controller)#ais-core-failure
```

### Verifying AIS Configuration

Use the `show run | sec` command to verify the configuration of AIS:

```
Router(config-controller)#show run | sec 0/3/0
controller E1 0/3/0
ais-core-failure
framing unframed
```
cem-group 30 unframed
interface CEM0/3/0

Example: AIS Trigger

The following example shows a sample configuration of a controller O/P when an AIS is triggered:

Router# show controller e1 0/2/1
E1 0/2/1 is down.
Applique type is A900-IMA16D
Cable length is Unknown
Transmitter is sending remote alarm.
Receiver is getting AIS. <<<<<<<<<< This is AIS alarm received
ais-shut is not set
alarm-trigger is not set
Framing is crc4, Line Code is HDB3, Clock Source is Line.
BER thresholds: SF = 10e-5 SD = 10e-5
International Bit: 1, National Bits: 1111
Data in current interval (0 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
1 Near-end path failures, 0 Far-end path failures, 0 SEF/AIS Secs

Verifying the Interface Configuration

Besides using the show running-configuration command to display the configuration settings, use the show interfaces serial and the show controllers serial commands to get detailed information on a per-port basis for your T1/E1 interface module.

Verifying Per-Port Interface Status

To view detailed interface information on a per-port basis for the T1/E1 interface module, use the show interfaces serial command.

Router# show interfaces serial 0/0/1:0
Serial0/0/1:0 is up, line protocol is up
Hardware is SPA-8XCHT1/E1
Internet address is 79.1.1.2/16
MTU 1500 bytes, BW 1984 Kbit, DLY 20000 usec,
reliability 255/255, txload 224/255
Encapsulation HDLC, crc 16, loopback not set
Keepalive not set
Last input 3d21h, output 3d21h, output hang never
Last clearing of ''show interface'' counters never
Input queue: 0/375/0 (size/max/drops/flushes); Total output drops: 2998712
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 1744000 bits/sec, 644 packets/sec
5 minute output rate 1874000 bits/sec, 690 packets/sec
180817311 packets input, 61438815508 bytes, 0 no buffer
Received 0 broadcasts (0 IP multicasts)
0 runts, 0 giants, 0 throttles
2 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored, 2 abort
180845200 packets output, 61438125092 bytes, 0 underruns
0 output errors, 0 collisions, 2 interface resets
0 output buffer failures, 0 output buffers swapped out
1 carrier transitions no alarm present
Timeslot(s) Used:1-31, subrate: 64Kb/s, transmit delay is 0 flags 2

**Configuration Examples**

This section includes the following configuration examples:

**Example: Framing and Encapsulation Configuration**

The following example sets the framing and encapsulation for the controller and interface:

```
! Specify the controller and enter controller configuration mode
! Router(config)# controller t1 2/0/0
! ! Specify the framing method
! Router(config-controller)# framing esf
! ! Exit controller configuration mode and return to global configuration mode
! Router(config-controller)# exit
! ! Specify the interface and enter interface configuration mode
! Router(config)# interface serial 2/0/0:0
! ! Specify the encapsulation protocol
! Router(config-if)# encapsulation ppp
! ! Exit interface configuration mode
! Router(config-if)# exit
! ! Exit global configuration mode
! Router(config)# exit
```

**Example: CRC Configuration**

The following example sets the CRC size for the interface:

```
! Specify the interface and enter interface configuration mode
! Router(config)# interface serial 2/0/0:0
! ! Specify the CRC size
! Router(config-if)# crc 32
! ! Exit interface configuration mode and return to global configuration mode
! Router(config-if)# exit
! ! Exit global configuration mode
! Router(config)# exit
```
Example: Facility Data Link Configuration

The following example configures Facility Data Link:

```bash
! Specify the controller and enter controller configuration mode
! Router(config)# controller t1 2/0/0
! ! Specify the FDL specification
! Router(config-controller)#
fdl ansi
! ! Exit controller configuration mode and return to global configuration mode
! Router(config-controller)# exit
! ! Exit global configuration mode
! Router(config)# exit
```

Example: Invert Data on the T1/E1 Interface

The following example inverts the data on the serial interface:

```bash
! Enter global configuration mode
! Router# configure terminal
! ! Specify the serial interface and enter interface configuration mode
! Router(config)# interface serial 2/1/3:0
! ! Configure invert data
! Router(config-if)# invert data
! ! Exit interface configuration mode and return to global configuration mode
! Router(config-if)# exit
! ! Exit global configuration mode
! Router(config)# exit
```