

Chapter 5

Configure Physical Interface Properties

For each network media type, the software driver for that media sets reasonable default values for general interface properties, such as the interface's MTU size, receive and transmit leaky bucket properties, link operational mode, and clock source. To modify any of the default general interface properties, include one or more statements in the [edit interfaces *interface-name*] hierarchy:

```
interfaces {
  interface-name {
    clocking clock-source;
    dce;
    disable;
    description text;
    encapsulation type;
    hold-time up milliseconds down milliseconds;
    keepalives <down-count number> <interval seconds> <up-count number>;
    link-mode mode;
    mac mac-address;
    mtu bytes;
    no-keepalives;
    no-traps;
    receive-bucket {
      overflow (tag | discard);
      rate percentage;
      threshold number;
    }
    speed (10m | 100m);
    transmit-bucket {
      overflow (tag | discard);
      rate percentage;
      threshold number;
    }
  }
}
```

This chapter discusses the following physical interface properties that you can configure:

Configure the Interface Name on page 28

Add an Interface Description to the Configuration on page 29

Configure the Link Characteristics on page 30

Configure the Media MTU on page 30

Configure Interface Encapsulation on page 32

- Configure the Interface's Speed on page 33
- Configure the MAC Address on the Management Ethernet Interface on page 34
- Configure Keepalives on page 34
- Configure the Clock Source on page 35
- Configure the Router as a DCE on page 35
- Configure Receive and Transmit Leaky Bucket Properties on page 36
- Damp Interface Transitions on page 37
- Disable SNMP Notifications on Physical Devices on page 37
- Disable a Physical Interface on page 37
- Configure ATM Physical Interface Properties on page 38
- Configure Channelized OC-12 Interface Properties on page 39
- Configure E1 and T1 Physical Interface Properties on page 40
- Configure E3 and T3 Physical Interface Properties on page 45
- Configure Fast Ethernet and Gigabit Ethernet Physical Interface Properties on page 51
- Configure SONET/SDH Physical Interface Properties on page 53
- Configure 802.1Q VLAN Tagging on page 63

Configure the Interface Name

Each interface has a name that identifies the physical interface type and the location of the interface card in the chassis. To configure the interface name, specify it at the [edit interfaces] hierarchy level:

```
[edit]
interfaces {
  interface-name {
    ...
  }
}
```

Specify the interface name in the following format:

```
physical<:channel>.logical
```

The physical part of an interface name has the following format:

```
type-fpc/pic/port
```

type is the media type and can be one of the following:

at—ATM interface.

e1—E1 interface.

e3—E3 interface.

fe—Fast Ethernet interface.

ge—Gigabit Ethernet interface.

gr—Generic Route Encapsulation tunnel interface.

ip—IP-over-IP encapsulation tunnel interface.

so—SONET interface.

t1—T1 interface. If the name does not include a channel identifier, it is assumed to be a copper-cable-based T1 interface. If the name includes a channel identifier, it is assumed to be a DS1 channel on a Channelized DS-3 interface.

t3—T3 interface. If the name does not include a channel identifier, it is assumed to be a copper-cable-based T3 interface. If the name includes a channel identifier, it is assumed to be a DS-3 channel on a Channelized OC-12 interface.

fxp—Management and internal Ethernet interfaces.

lo—Loopback interface.

fpc is the slot in which the FPC card is installed.

pic is the number of the PIC location in which the interface card is installed on the FPC.

port is the specific port on a PIC.

The logical unit part of the interface name corresponds to the logical unit number, which can be a number in the range 0 through 65535.

Add an Interface Description to the Configuration

You can include a text description of each physical interface in the configuration file. Any descriptive text you include is displayed in the output of the `show interfaces` commands. It has no impact on the interface's configuration. To add a text description, include the description statement at the `[edit interfaces interface-name]` hierarchy level:

```
[edit interfaces interface-name]  
description text;
```

The description can be a single line of text. If the text contains spaces, enclose it in quotation marks.

Example: Add an Interface Description to the Configuration

Add a description to a SONET interface:

```
[edit interfaces so-1/1/0]
user@host# set description "BB: ph101 P12/0/0 - local wire"
[edit interfaces so-1/1/0]
user@host# commit
[edit interfaces so-1/1/0]
user@host# exit configuration-mode
cli> show interfaces so-1/1/0
so-1/1/0 {
  physical-interface index 9 snmp-ifindex 10;
  enabled physical-link up;
  description "BB: ph101 P12/0/0 - local wire";
  encapsulation cisco-hdlc;
  ...
}
```

Configure the Link Characteristics

By default, the router’s management Ethernet interface, fxp0, and any installed Fast Ethernet interfaces, autonegotiate whether to operate in full-duplex or half-duplex mode. All other interfaces can operate only in full-duplex mode. For Gigabit Ethernet, the link partner must also be set to full duplex.

To explicitly configure a Fast Ethernet interface or the management Ethernet interface to operate in either full-duplex or half-duplex mode, include the link-mode statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex);
```

Configure the Media MTU

The default media MTU size used on a physical interface depends on the encapsulation being used on that interface. Table 1 and Table 2 list the media MTU size by interface type and Table 3 lists the encapsulation overhead by encapsulation type.

Table 1: Media MTU Sizes by Interface Type for M20 and M40 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
E1/T1	1504	9192	4470
E3/T3	4474	9192	4470
Fast Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	9192	4470

Table 2: Media MTU Sizes by Interface Type for M160 Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
E1/T1	1504	4500	4470
E3/T3	4474	4500	4470
Fast Ethernet	1514	4500	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	4500	4470

Table 3: Encapsulation Overhead by Encapsulation Type

Interface Encapsulation	Encapsulation Overhead (Bytes)
ATM PVC	12
Cisco HDLC	4
Frame Relay	4
Point-to-Point Protocol	4
Ethernet 802.3	14
Ethernet SNAP	17

The default media MTU is calculated as follows:

$$\text{Default media MTU} = \text{Default IP MTU} + \text{encapsulation overhead}$$

When you are configuring point-to-point connections, the MTU sizes on both sides of the connections must be the same. Also, when you are configuring point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

For information about configuring the encapsulation on an interface, see “Configure Interface Encapsulation” on page 32.

To modify the default media MTU size for a physical interface, include the `mtu` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
mtu bytes;
```

If you change the size of the media MTU, you must ensure that the size is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. You configure the protocol MTU by including the `mtu` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level, as discussed in “Set the Protocol MTU” on page 82.

Configure Interface Encapsulation

For each physical interface, you must configure an encapsulation to use for packets transmitted on the interface. You can optionally configure an encapsulation on a logical interface, which is the encapsulation used within certain packet types.

Configure the Encapsulation on a Physical Interface

The physical interface encapsulation can be one of the following:

Frame Relay—This encapsulation is defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E3, SONET, and T3 interfaces can use Frame Relay encapsulation.

Frame Relay Circuit Cross Connect (CCC)—This encapsulation is the same as standard Frame Relay for DLCIs 0 through 511. DLCIs 512 through 1022 are dedicated to CCC, and the logical interface must also have frame-relay-ccc encapsulation.

ATM PVC—ATM Permanent Virtual Circuit (PVC) encapsulation is defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. ATM interfaces can use ATM PVC encapsulation.

Point-to-Point Protocol (PPP)—PPP encapsulation is defined in RFC 1331, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. E3, SONET, and T3 interfaces can use PPP encapsulation. There is also a CCC version (ppp-ccc); the logical interfaces do not require an encapsulation statement, but they cannot have families.

Cisco HDLC—E3, SONET, and T3 interfaces can use Cisco HDLC encapsulation. There is also a CCC version (cisco-hdlc-ccc); the logical interfaces do not require an encapsulation statement, but they cannot have families.

VLAN Circuit Cross-Connect (CCC)—Ethernet interfaces with Virtual Local Area Network (VLAN) tagging enabled can use VLAN-CCC encapsulation.

To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
  encapsulation (atm-pvc | cisco-hdlc | cisco-hdlc-ccc | frame-relay | frame-relay-ccc | ppp | ppp-ccc |
    vlan-ccc);
```

When you configure a point-to-point encapsulation (such as PPP and Cisco HDLC) on a physical interface, the physical interface can have only one logical interface (that is, only one unit statement) associated with it. When you configure a multipoint encapsulation (such as Frame Relay), the physical interface can have multiple logical units and the units can be either point to point or multipoint.

Ethernet interfaces in VLAN mode can have multiple logical interfaces, but in CCC mode VLAN IDs from 0 through 511 are reserved for normal VLANs and VLAN IDs from 512 through 1023 are reserved for CCC VLANs. For more information, see “Configure 802.1Q VLANs” on page 140.

Example: Configure the Encapsulation on a Physical Interface

Configure PPP encapsulation on a SONET interface. The second two family statements allow IS-IS and MPLS to run on the interface.

```
[edit interfaces]
so-7/0/0 {
  encapsulation ppp;
  unit 0 {
    point-to-point;
    family inet {
      address 192.168.1.113/32 {
        destination 192.168.1.114;
      }
    }
    family iso;
    family mpls;
  }
}
```

Configure the Encapsulation on a Logical Interface

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for some encapsulation types, such as Frame Relay, ATM, and Ethernet VLAN encapsulations, you also can configure the encapsulation type that is used inside the Frame Relay, ATM, or VLAN circuit itself. To do this, include the encapsulation statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
encapsulation (atm-nlpid | atm-cisco-nlpid | atm-snap | atm-vc-mux | atm-ccc-vc-mux |
frame-relay-ccc | vlan-ccc);
```

The ATM encapsulations are defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*.

With the atm-nlpid, atm-cisco-nlpid, and atm-vc-mux encapsulations, you can configure the family inet only. With the circuit cross-connect (CCC) encapsulations, you cannot configure a family on the logical interface. A logical interface cannot have frame-relay-ccc encapsulation unless the physical device also has frame-relay-ccc encapsulation. In addition, you must assign this logical interface a DLCI in the range 512 through 1022 and configure it as point-to-point.

A logical interface cannot have vlan-ccc encapsulation unless the physical device also has vlan-ccc encapsulation. You must also assign this logical interface a VLAN ID in the range 512 through 1023; if the VLAN ID is 511 or lower, it is subject to the normal destination filter lookups in addition to source address filtering.

Configure the Interface's Speed

By default, the router's management Ethernet interface, fxp0, autonegotiates whether to operate at 10 Mbps or 100 Mbps. All other interfaces automatically choose the correct speed based on the PIC type and whether the PIC is configured to operate in multiplexed mode (using the no-concatenate statement in the [edit chassis] configuration hierarchy, as described in "Configure Channelized Interfaces" on page 121).

To configure the management Ethernet interface to operate at 10 Mbps or 100 Mbps, include the speed statement at the [edit interfaces fxp0] hierarchy level:

```
[edit interfaces fxp0]
speed (10m | 100m);
```

Configure the MAC Address on the Management Ethernet Interface

By default, the router's management Ethernet interface (fxp0) uses as its MAC address the MAC address that is burned into the Ethernet card. To display this address, enter the show chassis mac-address operational mode command.

To change the management Ethernet interface's MAC address, include the mac statement at the [edit interfaces fxp0] hierarchy level:

```
[edit interfaces fxp0]
mac mac-address;
```

Specify the MAC address as six hexadecimal bytes in one of the following formats: *nnnn.nnnn.nnnn* or *nn:nn:nn:nn:nn:nn*. For example, 0011.2233.4455 or 00:11:22:33:44:55.

Configure Keepalives

By default, physical interfaces configured with ATM, Cisco HDLC, or PPP encapsulation send keepalive packets at 10-second intervals. Frame Relay calls keepalives Local Management Interface (LMI) packets and ATM calls them Operation, Administration, and Maintenance (OAM) cells. (Note that the JUNOS software supports ANSI LMIs.)

To disable the sending of keepalives on a physical interface, include the no-keepalives statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
no-keepalives;
```

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a DTE (the default JUNOS configuration) and the other as a DCE.

To explicitly enable the sending of keepalives on a physical interface, include the keepalives statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
keepalives;
```

On interfaces configured with Cisco HDLC or PPP encapsulation, you can configure the following three keepalive parameters; note that Frame Relay encapsulation is not affected by these options:

interval seconds—The time in seconds between successive keepalive requests. The range is 10 seconds through 32767 seconds, with a default of 10 seconds.

down-count number—The number of keepalive packets a destination must fail to receive before the network takes a link down. The range is 1 through 255, with a default of 3.

up-count number—The number of keepalive packets a destination must receive to change a link's status from down to up. The range is 1 through 255, with a default of 1.

To change one or more of the default keepalive values, include the appropriate option at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
  keepalives <interval seconds> <down-count number> <up-count number>;
```

For interfaces using multipoint or multicast connections over Frame Relay encapsulation, if keepalives are enabled, the number of possible DLCI configurations is limited by the MTU size selected for the interface. To calculate the available DLCIs, use the formula $(MTU - 12) / 5$. To increase the number of possible DLCIs, disable keepalives on the interface.

Configure the Clock Source

For interfaces such as SONET that can use different clock sources, you can configure the source of the transmit clock on each interface. The source can be internal (also called line timing or normal) or external (also called loop timing). The default source is internal, which means that each interface uses the router's internal stratum 3 clock.

For DS-3 channels on a Channelized OC-12 interface, the clocking statement is supported only for channel 0; it is ignored if included in the configuration of channels 1 through 11. The clock source configured for channel 0 applies to all channels on the Channelized OC-12 interface. The individual DS-3 channels use a gapped 45-MHz clock as the transmit clock.

To configure loop timing on an interface, include the clocking external statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit]
user@host# set interfaces interface-name clocking external
[edit]
user@host# show
interfaces {
  interface-name {
    clocking external;
  }
}
```

To explicitly configure line timing on an interface, include the clocking internal statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
  clocking internal;
```

Configure the Router as a DCE

By default, when you configure an interface with Frame Relay encapsulation, the router is assumed to be data terminal equipment (DTE). That is, the router is assumed to be at a terminal point on the network. To configure the router to be data circuit-terminating equipment (DCE), include the dce statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
  dce;
```

When you configure the router to be a DCE, keepalives are disabled by default.

For back-to-back Frame Relay connections, either disable the sending of keepalives on both sides of the connection, or configure one side of the connection as a DTE (the default JUNOS configuration) and the other as a DCE.

Configure Receive and Transmit Leaky Bucket Properties

For all interface types except ATM, Fast Ethernet and Gigabit Ethernet, you can configure leaky bucket properties, which allow you to limit the amount of traffic received on and transmitted by a particular interface. You effectively specify what percentage of the interface's total capacity can be used to receive or transmit packets. You might want to set leaky bucket properties to limit the traffic flow from a link that is known to transmit high volumes of traffic.

By default, leaky buckets are disabled and the interface can receive and transmit packets at the maximum line rate.

For each DS-3 channel on a Channelized OC-12 interface, you can configure unique receive and transmit buckets.

To configure leaky bucket properties, include one or both of the receive-bucket and transmit-bucket statements at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
user@host# show
receive-bucket {
  overflow (tag | discard);
  rate percentage;
  threshold number;
}
transmit-bucket {
  overflow discard;
  rate percentage;
  threshold number;
}
```

In the rate option, specify the percentage of the interface line rate that is available to receive or transmit packets. The percentage can be a value from 0 (none of the interface line rate is available) to 100 (the maximum interface line rate is available). For example, when you set the line rate to 33, the interface receives or transmits at one third of the maximum line rate.

In the threshold option, specify the bucket threshold, which controls the burstiness of the leaky bucket mechanism. The larger the value, the more bursty the traffic, which means that over a very short amount of time the interface can receive or transmit close to line rate, but the average over a longer time is at the configured bucket rate. The threshold can be a value from 0 through 16777215 bytes. For ease of entry, you can enter *number* either as a complete decimal number or as a decimal number followed by the abbreviation k (1,000) or m (1,000,000). For example, the entry threshold 2m corresponds to a threshold of 2,000,000 bytes.

In the overflow option, specify how to handle packets that exceed the threshold:

discard—Discard received packets that exceed the threshold. No counting is done.

tag—(receive-bucket only) Tag, count, and process received packets that exceed the threshold.

Damp Interface Transitions

By default, when an interface transitions from being up to being down, or from down to up, this transition is advertised immediately to the router software and hardware. In some situations, for example, when an interface is connected to an ADM or WDM, or to protect against SONET framer holes, you might want to damp interface transitions, thereby not advertising the interface's transition until a certain period of time has transpired. When you have damped interface transitions and the interface goes from up to down, the interface is not advertised to the rest of the system as being down until it has remained down for the hold-time period. Similarly when an interface goes from down to up, it is not advertised as being up until it has remained up for the hold-time period.

To damp interface transitions, include the hold-time statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
hold-time up milliseconds down milliseconds;
```

The time can be a value from 0 through 65,535 milliseconds. The time value that you specify is rounded up to the nearest whole second. The default value is 0, which means that interface transitions are not damped.

Disable SNMP Notifications on Physical Devices

By default, SNMP notifications are sent when the state of interface or connection changes. To disable this notification on the physical interface, include the no-traps statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
no-traps;
```

Disable a Physical Interface

You can disable a physical interface, marking it as being down, without removing the interface configuration statements from the configuration. To do this, include the disable statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
disable;
```

Example: Disable a Physical Interface

Disable a physical interface:

```
[edit interfaces]
user@host# show so-1/1/0
so-1/1/0 {
  mtu 8000;
  clocking internal;
  encapsulation ppp;
  sonet-options {
    fcs 16;
  }
  unit 0 {
    family inet {
      address 12.12.12.21/32 {
        destination 12.12.12.22;
      }
    }
  }
}
[edit interfaces]
user@host# set so-1/1/0 disable
[edit interfaces]
user@host# show so-1/1/0
so-1/1/0 {
  disable;          # Interface is marked as disabled
  mtu 8000;
  clocking internal;
  encapsulation ppp;
  sonet-options {
    fcs 16;
  }
  unit 0 {
    family inet {
      address 12.12.12.21/32 {
        destination 12.12.12.22;
      }
    }
  }
}
```

Configure ATM Physical Interface Properties

For ATM physical interfaces, you can configure two ATM-specific physical device properties: the maximum number of virtual circuits (VCs) allowed on a virtual path (VP) and communication with directly attached ATM switches. You configure these properties by including the `atm-options` statement at the `[edit interfaces interface-name]` hierarchy level:

```
[edit interfaces interface-name]
atm-options {
  vpi vpi-identifier max-vcs maximum-vcs;
  ilmi;
}
```

You can configure the following ATM-specific properties:

Configure the Maximum Number of Virtual Circuits on a Virtual Path on page 39

Configure Communication with Directly Attached ATM Switches on page 39

Configure the Maximum Number of Virtual Circuits on a Virtual Path

You configure the maximum number of virtual circuits allows on a virtual path so that sufficient memory on the ATM PIC can be allocated for each VC. When configuring ATM interfaces on the router, you must include this statement.

To configure the largest numbered VCs on a VP, include the `vpi` statement at the [edit interfaces *interface-name atm-options*] hierarchy level:

```
[edit interfaces interface-name atm-options]
vpi vpi-identifier max-vcs maximum-vcs;
```

The VP identifier can be a value from 0 through 255. The maximum number of VCs you can configure per ATM interface is 4090. The largest numbered VC value you can configure is 4089.

All the VPIs that you configure in the `atm-options` statement are stored by the software in a single table. If you modify the VPIs, for example, by editing them in configuration mode or by issuing a load override command, all VCs on the interface are closed and then reopened, resulting in a temporary loss of connectivity for all the VCs on the interface.

You can also include some of the statements in the `sonet-options` statement to set SONET/SDH parameters on ATM interfaces as described in “Configure SONET/SDH Physical Interface Properties” on page 53.

Configure Communication with Directly Attached ATM Switches

You configure communication to directly attached ATM switches to enable querying of the IP addresses and port numbers of the switches. You query the switch by entering the following command:

```
user@host> show ilmi interface interface-name
```

The router uses VC 0.16 to communicate with the ATM switch.

To configure communication between the router and its directly attached ATM switches, include the `ilmi` statement at the [edit interfaces *interface-name atm-options*] hierarchy level:

```
[edit interfaces interface-name atm-options]
ilmi;
```

Configure Channelized OC-12 Interface Properties

To configure Channelized OC-12 interface properties, you can include the `sonet-options` and `t3-options` statements. Some of the SONET/SDH options are ignored and some can only be configured for channel 0, although they apply equally to all channels.

You can configure twelve channels per interface, and each interface can have logical interfaces. The `long-buildout` statement under `t3-options` is also ignored. For more information, see “Configure SONET/SDH Physical Interface Properties” on page 53 and “Configure E3 and T3 Physical Interface Properties” on page 45. You can configure twelve channels per interface. Each channel can have logical interfaces, the same as other physical interfaces. Table 4 summarizes the OC-12 to DS-3 numbering scheme.

Table 4: OC-12 to DS-3 Numbering Scheme

2-Level STS-1 Number (STS-3,STS-1)	1-Level STS Number	OC-12 to DS-3 PIC DS-3 Number
1,1	1	0
1,2	2	1
1,3	3	2
2,1	4	3
2,2	5	4
2,3	6	5
3,1	7	6
3,2	8	7
3,3	9	8
4,1	10	9
4,2	11	10
4,3	12	11

Configure E1 and T1 Physical Interface Properties

To configure E1-specific physical interface properties, include the `e1-options` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
e1-options {
  fcs (32 | 16);
  framing (g704 | unframed);
  idle-cycle-flag (flags | ones);
  loopback (local | remote);
  start-end-flag (shared | filler);
  timeslots slot-number;
}
```

To configure T1-specific physical interface properties, include the `t1-options` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
t1-options {
  buildout (0-133 | 133-266 | 266-399 | 399-532 | 532-655);
  byte-encoding (nx64 | nx56);
  fcs (32 | 16);
  framing (sf | esf);
  idle-cycle-flag (flags | ones);
  invert-data;
  line-encoding (ami | b8zs);
  loopback (local | remote);
  start-end-flag (shared | filler);
  timeslots slot-number;
}
```

You can configure the following E1-specific and T1-specific properties:

Configure T1 Buildout on page 41

Configure T1 Byte Encoding on page 41

Configure T1 Data Inversion on page 42

Configure the E1 and T1 Frame Checksum on page 42

Configure E1 Framing on page 42

Configure T1 Framing on page 43

Configure the E1 and T1 Idle Cycle Flag on page 43

Configure T1 Line Encoding on page 43

Configure E1 and T1 Loopback Capability on page 44

Configure the E1 and T1 Start-End Flag on page 44

Configure the E1 and T1 Timeslots on page 45

See also the following properties, which apply to a number of different interfaces:

Configure the Media MTU on page 30

Configure the Clock Source on page 35

Configure Receive and Transmit Leaky Bucket Properties on page 36

Configure T1 Buildout

A T1 interface has five possible setting ranges for the T1 line buildout: 0-133, 133-266, 266-399, 399-532, or 532-655 feet. By default, the T1 interface uses the shortest setting (0-133).

To have the interface support one of the longer distance ranges, include the buildout statement with the appropriate value at the [edit interfaces *interface-name* t1-options] hierarchy level:

```
[edit interfaces interface-name t1-options]
buildout 532-655;
```

Configure T1 Byte Encoding

By default, T1 interfaces use a byte encoding of 8 bits per byte (nx64). You can configure an alternative byte encoding of 7 bits per byte (nx56).

To have the interface use 7 bits per byte encoding, include the byte-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the nx56 option:

```
[edit interfaces interface-name t1-options]
byte-encoding nx56;
```

To explicitly configure nx64 byte encoding, include the byte-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the nx64 option:

```
[edit interfaces interface-name t1-options]
byte-encoding nx64;
```

Configure T1 Data Inversion

By default, data inversion is disabled. To enable data inversion at the HDLC level, include the `invert-data` statement at the [edit interfaces *interface-name* t1-options] hierarchy level:

```
[edit interfaces interface-name t1-options]
invert-data;
```

When you enable data inversion, all data bits in the data stream are transmitted inverted; that is, zeroes are transmitted as ones and ones as zeroes. Data inversion is normally used only in AMI mode to guarantee ones density in the transmitted stream.

Configure the E1 and T1 Frame Checksum

By default, E1 and T1 interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment might not support 32-bit checksums.

To configure a 32-bit checksum, include the `fcs 32` statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
[edit interfaces interface-name t1-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the `fcs 32` statement from the configuration:

```
[edit]
user@host# delete interfaces t1-fpc/pic/port t1-options fcs 32
```

To explicitly configure a 16-bit checksum, include the `fcs 16` statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level:

```
[edit interfaces interface-name t1-options]
fcs 16;
```

Configure E1 Framing

By default, E1 interfaces use the G704 framing mode. You can configure the alternative unframed mode if needed.

To have the interface use the unframed mode, include the `framing unframed` statement at the [edit interfaces *interface-name* e1-options] hierarchy level, specifying the unframed option:

```
[edit interfaces interface-name e1-options]
framing unframed;
```

To explicitly configure G704 framing, include the `framing g704` statement at the [edit interfaces *interface-name* e1-options] hierarchy level, specifying the g704 option:

```
[edit interfaces interface-name e1-options]
framing g704;
```

Configure T1 Framing

By default, T1 interfaces use ESF (extended super frame) framing format. You can configure SF (super frame) format as an alternative.

To have the interface use the SF framing format, include the framing statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the sf option:

```
[edit interfaces interface-name t1-options]
framing sf;
```

To explicitly configure ESF framing, include the framing statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the esf option:

```
[edit interfaces interface-name t1-options]
framing esf;
```

Configure the E1 and T1 Idle Cycle Flag

By default, E1 and T1 interfaces transmit the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead, include the idle-cycle-flag statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level, specifying the ones option:

```
[edit interfaces interface-name t1-options]
idle-cycle-flag ones;
```

To explicitly configure the default value of 0x7E, include the idle-cycle-flag statement with the flags option:

```
[edit interfaces interface-name t1-options]
idle-cycle-flag flags;
```

Configure T1 Line Encoding

By default, T1 interfaces use B8ZS line encoding. You can configure AMI line encoding if necessary.

To have the interface use AMI line encoding, include the line-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the ami option:

```
[edit interfaces interface-name t1-options]
line-encoding ami;
```

To explicitly configure B8ZS line encoding, include the line-encoding statement at the [edit interfaces *interface-name* t1-options] hierarchy level, specifying the b8zs option:

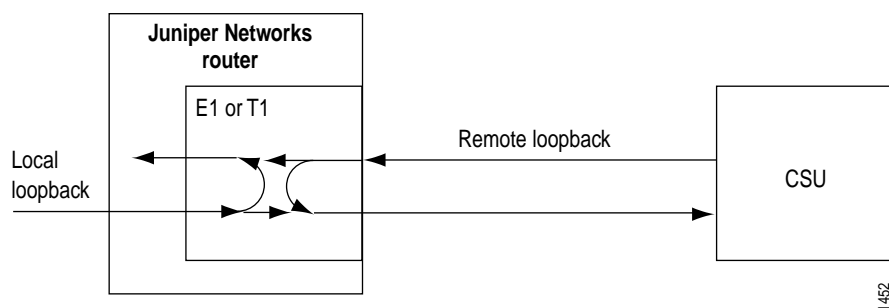
```
[edit interfaces interface-name t1-options]
line-encoding b8zs;
```

When setting the line encoding parameter, you must set the same value for paired ports. Ports 0 and 1 must share the same value, and likewise ports 2 and 3 must share the same value, but ports 0 and 1 can have a different value from that of ports 2 and 3.

Configure E1 and T1 Loopback Capability

You can configure loopback capability between the local E1 or T1 interface and the remote CSU. You can configure the loopback to be local or remote. With local loopback, the E1 or T1 interface can transmit packets to the CSU, but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the E1 or T1 interface but also are immediately retransmitted to the CSU (see Figure 4).

Figure 3: Remote and Local E1 or T1 Loopback



To configure loopback capability on an E1 or a T1 interface, include the loopback statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level. Packets can be looped on either the local router or the remote CSU.

```
[edit interfaces interface-name t1-options]
loopback (local | remote);
```

To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces t1-fpc/pic/port t1-options loopback
```

Configure the E1 and T1 Start-End Flag

By default, an E1 or a T1 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags, include the start-end-flag statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level, specifying the shared option. This configuration can result in better performance on the interface.

```
[edit interfaces interface-name t1-options]
start-end-flag shared;
```

To explicitly configure the default of waiting two idle cycles between the start and end flags, include the idle-cycle-flag statement with the filler option:

```
[edit interfaces interface-name t1-options]
start-end-flag filler;
```

Configure the E1 and T1 Timeslots

To configure the number of timeslots allocated to the interface, include the `timeslots` statement at the [edit interfaces *interface-name* e1-options] or [edit interfaces *interface-name* t1-options] hierarchy level, specifying the *slot-number*. The range is 1 through 24 for T1 interfaces and 1 through 32 for E1 interfaces.

```
[edit interfaces interface-name t1-options]
timeslots slot-number;
```

You can designate any combination of timeslots for usage. The default is to use all the timeslots.

To use timeslots 1 through 10, designate *slot-number* as follows:

```
[edit interfaces interface-name t1-options]
timeslots 1-10;
```

To use timeslots 1 through 5, timeslot 10, and timeslot 24, designate *slot-number* as follows:

```
[edit interfaces interface-name t1-options]
timeslots 1-5,10,24;
```

To use the first four odd-numbered timeslots, designate *slot-number* as follows:

```
[edit interfaces interface-name t1-options]
timeslots 1,3,5,7;
```

Note that spaces are not allowed in specifying timeslot numbers.

Configure E3 and T3 Physical Interface Properties

To configure T3-specific physical interface properties, include the `t3-options` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
t3-options {
  bert-algorithm algorithm;
  bert-error-rate rate;
  bert-period seconds;
  (cbit-parity | no-cbit-parity);
  compatibility-mode (digital-link | kentrox | larscom) <subrate value>;
  fcs (32 | 16);
  (feac-loop-respond | no-feac-loop-respond);
  idle-cycle-flag value;
  (long-buildout | no-long-buildout);
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
  start-end-flag value;
}
```

To configure E3-specific physical interface properties, include the `e3-options` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
e3-options {
  bert-algorithm algorithm;
  bert-error-rate rate;
```

```

    bert-period seconds;
    compatibility-mode (digital-link | kentrox | larscom);
    fcs (32 | 16);
    idle-cycle-flag value;
    loopback (local | remote);
    (payload-scrambler | no-payload-scrambler);
    start-end-flag value;
}

```

You can configure the following E3-specific and T3-specific properties:

Configure E3 and T3 CSU Compatibility Mode on page 46

Disable T3 C-Bit Parity Mode on page 47

Configure the E3 and T3 Frame Checksum on page 47

Configure E3 and T3 Loopback Capability on page 48

Configure T3 FEAC Response on page 48

Configure the T3 Line Buildout on page 49

Configure the E3 and T3 Idle Cycle Flag on page 49

Configure the E3 and T3 Start-End Flag on page 50

Configure E3 and T3 HDLC Payload Scrambling on page 50

Configure E3 and T3 BERT Properties on page 50

Configure E3 and T3 CSU Compatibility Mode

To configure an E3 or a T3 interface so that it is compatible with the CSU at the remote end of the line, include the compatibility statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level. For example:

```

[edit interfaces interface-name t3-options]
  compatibility-mode (digital-link | kentrox | larscom) <subrate value>;

```

You can configure the interface to be compatible with a Digital Link, Kentrox, or Larscom CSU.

The subrate of a T3 interface must exactly match that of the remote CSU. To specify the subrate, include the subrate option in the compatibility-mode statement:

For Digital Link CSUs, specify the subrate *value* as the data rate you configured on the CSU in the format *xKb* or *x.Mb*. For a list of specific rate values, use the command completion feature in the CLI. The range is 301 Kbps through 44.2 Mbps.

Kentrox CSUs do not support subrate.

For Larscom CSUs, specify the subrate *value* as a number from 1 through 14 that exactly matches the value configured on the CSU.

E3 interfaces do not support the subrate option.

Disable T3 C-Bit Parity Mode

On T3 interfaces only, C-bit parity mode controls the type of framing that is present on the transmitted T3 signal. When C-bit parity mode is enabled, the C-bit positions are used for the FEBE, FEAC, terminal data link, path parity, and mode indicator bits, as defined in ANSI T1.107a-1989. When C-bit parity mode is disabled, the basic T3 framing mode is used.

By default, C-bit parity mode is enabled. To disable C-bit parity mode and use the basic T3 framing mode, include the `no-cbit-parity` statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-cbit-parity;
```

To return to the default, enabling C-bit parity mode, delete the `no-cbit-parity` statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options no-cbit-parity
```

To explicitly enable C-bit parity mode, include the `cbit-parity` statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
cbit-parity;
```

Configure the E3 and T3 Frame Checksum

By default, E3 and T3 interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment may not support 32-bit checksums.

On a Channelized OC-12 interface, the `fcs` statement is not supported. To configure FCS on each DS-3 channel, you must include the `t3-options fcs` statement in the configuration for each channel. For SONET, the Channelized OC-12 interface supports DS-3 to STS-1 to OC-12. For SDH, the Channelized OC-12 interface supports *nxDS-3* to *nxVC3* to *nxAU3* to STM-*n*.

To configure a 32-bit checksum, include the `fcs 32` statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the `fcs 32` statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options fcs 32
```

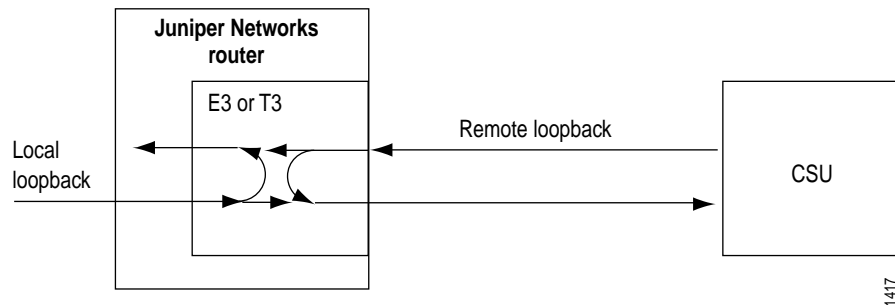
To explicitly configure a 16-bit checksum, include the `fcs 16` statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
fcs 16;
```

Configure E3 and T3 Loopback Capability

You can configure loopback capability between the local E3 or T3 interface and the remote CSU. You can configure the loopback to be local or remote. With local loopback, the E3 or T3 interface can transmit packets to the CSU, but receives its own transmission back again and ignores data from the CSU. With remote loopback, packets sent from the CSU are received by the E3 or T3 interface but also are immediately retransmitted to the CSU (see Figure 4).

Figure 4: Remote and Local E3 or T3 Loopback



To configure loopback capability on an E3 or a T3 interface, include the loopback statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level. Packets can be looped on either the local router or the remote CSU.

```
[edit interfaces interface-name t3-options]
loopback (local | remote);
```

To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options loopback
```

For DS-3 channels on a Channelized OC-12 interface, the SONET loopback statement is supported only for channel 0. It is ignored if included in the configuration for channels 1 through 11. The SONET loopback configured for channel 0 applies to all 12 channels equally. To configure loopbacks on the DS-3 channels, you must include the t3-options loopback statement in the configuration for each channel. Each DS-3 channel can be put in loopback mode independently.

Configure T3 FEAC Response

For T3 interfaces, the T3 far-end alarm and control (FEAC) signal is used to send alarm or status information from the far-end terminal back to the near-end terminal and to initiate T3 loopbacks at the far-end terminal from the near-end terminal. To allow the remote CSU to place the local router into loopback, you must configure the router to respond to the CSU's FEAC request by including the feac-loop-respond statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
feac-loop-respond;
```

By default, the router does not respond to FEAC requests.

If you have configured remote or local loopback with the T3 loopback statement, the router does not respond to FEAC requests from the CSU even if you have included the feac-loop-respond statement in the configuration. To have the router respond, you must delete the loopback statement from the configuration.

To explicitly configure the router not to respond to FEAC requests, include the no-feac-loop statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-feac-loop-respond;
```

Configure the T3 Line Buildout

A T3 interface has two settings for the T3 line buildout: a short setting, which is less than 225 feet (about 68 meters), and a long setting, which is greater than 225 feet. By default, the interface uses the short setting.

The long-buildout and no-long-buildout statements apply only to copper-cable-based T3 interfaces. You cannot configure a line buildout for a DS-3 channel on a Channelized OC-12 interface, which runs over fiber-optic cable. If you configure this statement on a Channelized OC12 interface, it is ignored.

To have the interface drive a line that is longer than 255 feet, include the long-buildout statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
long-buildout;
```

To explicitly configure the default short line buildout, include the no-long-buildout statement at the [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-long-buildout;
```

Configure the E3 and T3 Idle Cycle Flag

By default, an E3 or a T3 interface transmits the value 0x7E in the idle cycles. To have the interface transmit the value 0xFF (all ones) instead, include the idle-cycle-flag statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level, specifying the ones option:

```
[edit interfaces interface-name t3-options]
idle-cycle-flag ones;
```

To explicitly configure the default value of 0x7E, include the idle-cycle-flag statement with the flags option:

```
[edit interfaces interface-name t3-options]
idle-cycle-flag flags;
```

Configure the E3 and T3 Start-End Flag

By default, an E3 or a T3 interface waits two idle cycles between sending start and end flags. To configure the interface to share the transmission of start and end flags, include the start-end-flag statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level, specifying the shared option. This configuration can result in better performance on the interface.

```
[edit interfaces interface-name t3-options]
start-end-flag shared;
```

To explicitly configure the default of waiting two idle cycles between the start and end flags, include the idle-cycle-flag statement with the filler option:

```
[edit interfaces interface-name t3-options]
start-end-flag filler;
```

Configure E3 and T3 HDLC Payload Scrambling

E3 or T3 HDLC payload scrambling, which is disabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

On a Channelized OC-12 interface, the SONET payload-scrambler statement is ignored. To configure scrambling on the DS-3 channels on the interface, you can include the payload-scrambler statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level for each DS-3 channel:

```
[edit interfaces interface-name t3-options]
payload-scrambler;
```

To explicitly disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
no-payload-scrambler;
```

To disable payload scrambling again (return to the default), delete the payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces t3-fpc/pic/port t3-options payload-scrambler
```

Configure E3 and T3 BERT Properties

You can configure an E3 or a T3 interface to execute a bit error rate test (BERT) when the interface receives a request to run this test. You specify the duration of the test, the pattern to send in the bit stream, and the error rate to include in the bit stream by including the bert-period, bert-algorithm, and bert-error-rate statements, respectively, at the [edit interfaces *interface-name* e3-options] or [edit interfaces *interface-name* t3-options] hierarchy level:

```
[edit interfaces interface-name t3-options]
bert-algorithm algorithm;
bert-error-rate rate;
bert-period seconds;
```

period is the duration of BERT test, in seconds. The test can last from 1 to 240 seconds; the default is 10 seconds. *algorithm* is the pattern to send in the bit stream. For a list of supported patterns, see the CLI help text or the description of bert-algorithm on page 203. *rate* is the bit error rate. This can be an integer in the range 0 through 7, which corresponds to a bit error rate in the range 10^{-0} (that is, 0, which corresponds to no errors) to 10^{-7} (that is, 1 error per 10 million bits).

Configure Fast Ethernet and Gigabit Ethernet Physical Interface Properties

To configure Fast Ethernet-specific physical interface properties, include the fastether-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
link-mode (full-duplex | half-duplex)
fastether-options {
  (loopback | no-loopback);
  source-address-filter {
    mac-address;
  }
  (source-filtering | no-source-filtering);
}
```

To configure Gigabit Ethernet-specific physical interface properties, include the ggether-options statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
ggether-options {
  (flow-control | no-flow-control);
  (loopback | no-loopback);
  source-address-filter {
    mac-address;
  }
  (source-filtering | no-source-filtering);
}
```

You can configure the following properties specific to Fast Ethernet and Gigabit Ethernet interfaces:

Configure MAC Address Filtering on page 51

Configure Loopback Mode on page 52

Configure Flow Control on page 52

Configure MAC Address Filtering

On Fast Ethernet and Gigabit Ethernet interfaces, you can enable source address filtering, which blocks all incoming packets to that interface. To enable the filtering, include the source-filtering statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* ggether-options] hierarchy level:

```
source-filtering;
```

When source address filtering is enabled, you can configure the interface to receive packets from specific MAC addresses. To do this, specify the MAC addresses in the source-address-filter statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* ggether-options] hierarchy level:

```
source-address-filter {
  mac-address;
  <additional-mac-address;>
}
```

You can specify the MAC address as *nn:nn:nn:nn:nn:nn* or *nnnn.nnnn.nnnn*, where *n* is a decimal digit. To specify more than one address, include multiple *mac-address* options in the source-address-filter statement.

If the remote Ethernet card is changed, the interface will not be able to receive packets from the new card because it will have a different MAC address.

Configure Loopback Mode

By default, local Fast Ethernet or Gigabit Ethernet interfaces connect to a remote system. To place an interface in loopback mode, include the loopback statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* ggether-options] hierarchy level:

```
loopback;
```

To disable loopback mode, delete the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces fe-fpc/pic/port fastether-options loopback
```

To explicitly disable loopback mode, include the no-loopback statement at the [edit interfaces *interface-name* fastether-options] hierarchy level:

```
[edit interfaces interface-name fastether-options]
no-loopback;
```

Configure Flow Control

By default, the router imposes flow control to regulate the amount of traffic sent out a Gigabit Ethernet interface. This is useful if the remote side of the connection is a Gigabit Ethernet switch.

You can disable flow control if you want the router to permit unrestricted traffic. To disable flow control, include the no-flow-control statement at the [edit interfaces *interface-name* ggether-options] hierarchy level:

```
[edit interfaces interface-name ggether-options]
no-flow-control;
```

To explicitly reinstate flow control, include the flow-control statement at the [edit interfaces *interface-name* ggether-options] hierarchy level:

```
[edit interfaces interface-name ggether-options]
flow-control;
```

Configure SONET/SDH Physical Interface Properties

To configure SONET/SDH physical interface properties, include the `sonet-options` statement at the `[edit interfaces interface-name]` hierarchy level:

```
[edit interfaces so-fpc/pic/port]
sonet-options {
  aps {
    advertise-interval milliseconds;
    authentication-key key;
    force;
    hold-time milliseconds;
    lockout;
    neighbor address;
    paired-group group-name;
    protect-circuit group-name;
    request;
    revert-time seconds;
    working-circuit group-name;
  }
  bytes {
    e1-quiet value;
    f1 value;
    f2 value;
    s1 value;
    z3 value;
    z4 value;
  }
  fcs (32 | 16);
  loopback (local | remote);
  path-trace trace-string;
  (payload-scrambler | no-payload-scrambler);
  rfc-2615;
  (z0-increment | no-z0-increment);
}
```

Note that on SONET/SDH OC-48 interfaces configured for channelized (multiplexed) mode (by including the `no-concatenate` statement at the `[edit chassis fpc slot-number pic pic-number]` hierarchy level), the bytes `e1-quiet` and bytes `f1` options have no effect. The bytes `f2`, bytes `z3`, bytes `z4`, and `path-trace` options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3.

For DS-3 channels on a Channelized OC-12 interface, the bytes `e1-quiet`, bytes `f1`, bytes `f2`, bytes `z3`, and bytes `z4` options have no effect. The bytes `s1` option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes `s1` value configured on channel 0 applies to all channels on the interface.

You also can include some of the statements in the `sonet-options` statement to set SONET/SDH parameters on ATM interfaces:

```
[edit interfaces at-fpc/pic/port]
sonet-options {
  bytes {
    f2 value;
    s1 value;
    z3 value;
    z4 value;
  }
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
}
```

You can configure the following SONET/SDH physical interface properties:

Configure SONET Header Byte Values on page 54

Configure SONET z0-increment Option on page 55

Configure the SONET Frame Checksum on page 55

Configure SONET Loopback Capability on page 56

Configure the SONET Path Trace Identifier on page 56

Configure SONET HDLC Payload Scrambling on page 57

Configure SONET RFC 2615 Support on page 57

Configure APS on page 57

Configure SONET Header Byte Values

To configure values in SONET header bytes, include the bytes statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
bytes {
  e1-quiet value;
  f1 value;
  f2 value;
  s1 value;
  z3 value;
  z4 value;
}
```

You can configure the following SONET header bytes:

e1-quiet—Default idle byte sent on the orderwire SONET overhead bytes. The router does not support the orderwire channel, and hence sends this byte continuously. For the E1-quiet byte, *value* can be in the range 0 through 255. The default value is 0x7F.

f1, f2, z3, z4—SONET overhead bytes. For these bytes, *value* can be in the range 0 through 255. The default value is 0x00.

s1—Synchronization message SONET overhead byte. This byte is normally controlled as a side effect of the system reference clock configuration and the state of the external clock coming from an interface if the system reference clocks have been configured to use an external reference. For the s1 byte, *value* can be in the range 0 through 255.

On SONET OC-48 interfaces that are configured for channelized (multiplexed) mode (by including the no-concatenate statement at the [edit chassis fpc *slot-number* pic *pic-number*] hierarchy level), the bytes e1-quiet and bytes f1 options have no effect. The bytes f2, bytes z3, bytes z4, and path-trace options work correctly on channel 0 and work in the transmit direction only on channels 1, 2, and 3.

For DS-3 channels on a Channelized OC-12 interface, the bytes e1-quiet, bytes f1, bytes f2, bytes z3, and bytes z4 options have no effect. The bytes s1 option is supported only for channel 0; it is ignored if configured on channels 1 through 11. The bytes s1 value configured on channel 0 applies to all channels on the interface.

Configure SONET z0-increment Option

When configured in SDH framing mode, POS interfaces on a Juniper Networks router might not interoperate with some older versions of ADMs or regenerators that require an incrementing STM ID. To resolve this incompatibility, you can explicitly configure an incrementing STM ID rather than a static one in the SDH overhead by including the z0-increment statement at the [edit interfaces *interface-name* sonet-options] hierarchy level. You should include this statement only for SDH mode; do not use it for SONET mode.

```
[edit interfaces so-fpc/pic/port sonet-options]
z0-increment;
```

To explicitly disable z0-incrementing, include the following statement:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-z0-increment;
```

Current SDH standards specify a set of $3*n$ overhead bytes in an STM- n that includes the J0 section trace byte. The rest are essentially unused (spare Z0) and contain hex values (0x01, 0xCC, 0xCC ... 0xCC). The older version of the standard specified that the same set of bytes should contain an incrementing sequence: 1, 2, 3, ..., $3*n$. Their use was still unspecified, although they might have been used to assist in frame alignment.

The z0-increment option enables Juniper Networks routers to interoperate with older equipment that relies on those bytes for frame alignment.

The STM identifier has a precise definition from the SDH specs. In ITU-T Recommendation G.707, *Network node interface for the synchronous digital hierarchy (SDH)* (03/96), section 9.2.2.2:

NOTE: STM identifier: C1

In earlier versions of the Recommendation, the content of bytes located at S (1, 7, 1) or [1, 6N+ 1] to S (1,7, N) or [1, 7N] was defined as a unique identifier indicating the binary value of the multi-column, interleave depth coordinate, c. It may have been used to assist in frame alignment.

Configure the SONET Frame Checksum

By default, SONET interfaces use a 16-bit frame checksum. You can configure a 32-bit checksum, which provides more reliable packet verification. However, some older equipment may not support 32-bit checksums.

To configure a 32-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 32;
```

To return to the default 16-bit frame checksum, delete the fcs 32 statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options fcs 32
```

To explicitly configure a 16-bit checksum, include the fcs statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
fcs 16;
```

On a Channelized OC-12 interface, the SONET-options fcs statement is not supported. To configure FCS on each DS-3 channel, you must include the t3-options fcs statement in the configuration for each channel.

Configure SONET Loopback Capability

To configure loopback capability on a SONET interface, include the loopback statement at the [edit interfaces *interface-name* sonet-options] hierarchy level. Packets can be looped on either the local or the remote router.

```
[edit interfaces so-fpc/pic/port sonet-options]
loopback (local | remote);
```

To turn off loopback, remove the loopback statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options loopback
```

For DS-3 channels on a Channelized OC-12 interface, the SONET loopback statement is supported only for channel 0; it is ignored if included in the configuration for channels 1 through 11. The SONET loopback configured for channel 0 applies equally to all 12 channels.

You can configure 12 channels per interface, and each interface can have logical interfaces. To configure loopbacks on the DS-3 channels, you must include the t3-options loopback statement in the configuration for each channel. Each DS-3 channel can be put in loopback mode independently.

Configure the SONET Path Trace Identifier

The SONET path trace identifier is a text string that identifies the circuit. If the string contains spaces, enclose it in quotation marks. The common convention is to use the circuit identifier as the path trace identifier. If you do not configure an identifier, the JUNOS software uses the system and interface names. The local system's path trace identifier is displayed when a show interfaces command is issued on the remote system.

For DS-3 channels on a Channelized OC-12 interface, you can configure a unique path trace for each of the 12 channels. Each path trace can be up to 16 bytes.

To configure a path trace identifier, include the path-trace statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
path-trace trace-string;
```

Configure SONET HDLC Payload Scrambling

SONET HDLC payload scrambling, which is enabled by default, provides better link stability. Both sides of a connection must either use or not use scrambling.

On a Channelized OC-12 interface, the SONET payload-scrambler statement is ignored. To configure scrambling on the DS-3 channels on the interface, you can include the t3-options payload-scrambler statement in the configuration for each DS-3 channel.

To disable HDLC payload scrambling, include the no-payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
no-payload-scrambler;
```

To return to the default, that is, to re-enable payload scrambling, delete the no-payload-scrambler statement from the configuration:

```
[edit]
user@host# delete interfaces so-fpc/pic/port sonet-options no-payload-scrambler
```

To explicitly enable payload scrambling, include the payload-scrambler statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
payload-scrambler;
```

Configure SONET RFC 2615 Support

RFC 2615 requires certain C2 byte and FCS settings in addition to the default values configured in accordance with RFC 1619.

To enable support for the RFC 2615 features, include the rfc-2615 statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options]
rfc-2615;
```

Configure APS

Automatic Protection Switching (APS) is used by SONET add/drop multiplexers (ADMs) to protect against circuit failures. The JUNOS implementation of APS allows you to protect against circuit failures between an ADM and one or more routers. When a circuit or router fails, a backup immediately takes over.

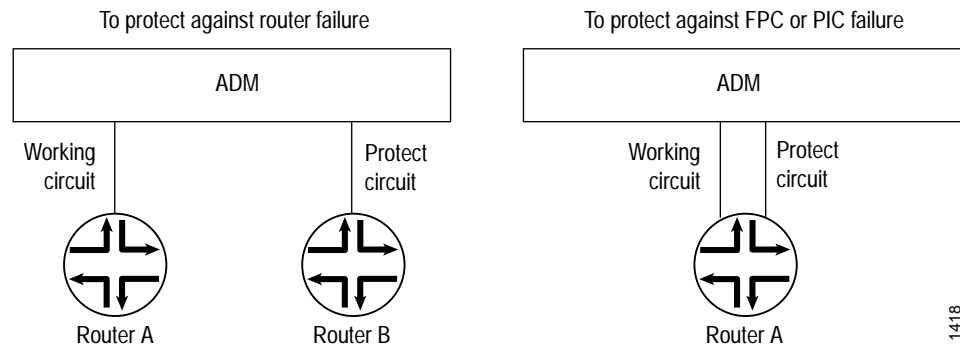
The JUNOS software supports APS 1:1 switching, bidirectional only, and either revertive or nonrevertive mode.

For DS-3 channels on a Channelized OC-12 interface, you can configure APS on channel 0 only. If you configure APS on channels 1 through 11, it is ignored.

With APS, you configure two circuits, a *working circuit* and a *protect circuit*. Normally, traffic is carried on the working circuit (that is, the working circuit is the active circuit), and the protect circuit is disabled. If the working circuit fails or degrades, or if the working router fails, the ADM and the protect router switch the traffic to the protect circuit, and the protect circuit becomes the active circuit.

To configure APS, you configure a working and a protect circuit. To protect against a router failure, you connect two routers to the ADM, configuring one of them as the working router and the second as the protect router. To protect against a PIC or an FPC failure, you connect one router to the ADM through both the working and protect circuits, configuring one of the PICs or FPCs as the working circuit and the second as the protect circuit. See Figure 5.

Figure 5: APS Configuration Topologies



1418

To configure APS, include the `aps` statement at the [edit interfaces *interface-name* sonet-options] hierarchy level:

```
[edit interfaces interface-name sonet-options]
aps {
  advertise-interval milliseconds;
  authentication-key key;
  force;
  hold-time milliseconds;
  lockout;
  neighbor address;
  paired-group group-name;
  protect-circuit group-name;
  request;
  revert-time seconds;
  working-circuit group-name;
}
```

You can configure the following APS properties:

Configure Basic APS Support on page 59

Configure Switching between the Working and Protect Circuits on page 60

Configure Revertive Mode on page 61

Configure APS Timers on page 61

Configure APS Load Sharing between Circuit Pairs on page 62

Configure Basic APS Support

To set up a basic APS configuration, configure one interface to be the working circuit and a second to be the protect circuit. If you are using APS to protect against router failure, configure one interface on each router. If you are using APS to protect against FPC failure, configure two interfaces on the router, one on each FPC.

For each working–protect circuit pair, configure the following:

Group name—Creates the association between the two circuits. Configure the same group name for both the working and protect routers.

Authentication key—You configure this on both interfaces. Configure the same key for both the working and protect routers.

Address of the other interface on the other router—If you are configuring one router to be the working router and a second to be the protect router, you must configure the address of the remote interface. You configure this on one or both of the interfaces.

The address you specify for the neighbor must never be routed through the interface on which APS is configured, or instability will result. We strongly recommend that you directly connect the working and protect routers and that you configure the interface address of this shared network as the neighbor address.

The working and protect configurations on the routers must match the circuits configurations on the ADM; that is, the working router must be connected to the ADM's working circuit and the protect router must be connect to the protect circuit.

To set up a basic APS configuration, include the following statements at the [edit interfaces *interface-name* sonet-options] hierarchy level:

On the working router/circuit:

```
[edit interfaces so-fpc/pic/port sonet-options]
aps {
  working-circuit group-name;
  authentication-key key;
  neighbor address; # Include only if protect circuit is on a different router
}
```

On the protect router/circuit:

```
aps {
  protect-circuit group-name;
  authentication-key key;
  neighbor address; # Include only if working circuit is on a different router
}
```

For example, configure Router A to be the working router and Router B to be the protect router:

On Router A (the working router):

```
[edit interfaces so-6/1/1 sonet-options]
aps {
  working-circuit San-Jose;
  authentication-key "$9$B2612345";
}
```

On Router B (the protect circuit):

```
[edit interfaces so-0/0/0 sonet-options]
aps {
```

```

protect-circuit San-Jose;
authentication-key "$9$B2612345";
neighbor 192.168.1.2; <- address of management interface on Router A
}

```

As a second example, configure one interface on a router to be the working circuit and another interface to be the protect circuit:

```

On Router A:
[edit interfaces so-2/1/1 sonet-options]
aps {
  working-circuit Hayward;
  authentication-key blarney;
}
[edit interfaces so-3/0/2 sonet-options]
aps {
  protect-circuit Hayward;
  authentication-key blarney;
}

```

Configure Switching between the Working and Protect Circuits

When there are multiple reasons to switch between the working and protect circuits, a priority scheme is used to decide which circuit to use. The routers and the ADM might automatically switch traffic between the working and protect circuits because of circuit and router failures. You can also choose to switch traffic manually between the working and protect circuits. There are three priority levels of manual configuration, listed here in order from lowest to highest priority:

Request (also known as manual switch)—Overridden by signal failures, signal degradations, or any higher-priority reasons.

Force (also known as forced switch)—Overrides manual switches, signal failures, and signal degradation.

Lockout (also known as lockout of protection)—Do not switch between the working and protect circuits.

A router failure is considered to be equivalent to a signal failure on a circuit.

To perform a manual switch, include the request statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch.

```

[edit interfaces so-fpc/pic/port sonet-options aps]
request (protect | working);

```

When the working circuit is operating in nonrevertive mode, use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer.

To perform a forced switch, include the force statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level. This statement is honored only if there are no higher-priority reasons to switch. This configuration can be overridden by a signal failure on the protect circuit, thus causing a switch to the working circuit.

```

[edit interfaces so-fpc/pic/port sonet-options aps]
force (protect | working);

```

To configure a lockout of protection, forcing the use of the working circuit and locking out the protect circuit regardless of anything else, include the lockout statement at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
lockout;
```

Configure Revertive Mode

By default, APS is nonrevertive, which means that if the protect circuit becomes active, traffic is not switched back to the working circuit unless the protect circuit fails or you manually configure a switch to the working circuit. In revertive mode, traffic is automatically switched back to the working circuit.

You should configure the ADM and routers consistently with regard to revertive or nonrevertive mode.

To configure revertive mode, include the revert-time statement, specifying the amount of time to wait after the working circuit has again become functional before making the working circuit active again:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
revert-time seconds;
```

If you are using nonrevertive APS, you can use the request working statement to switch the circuit manually to being the working circuit or to override the revert timer (configured with the revert-time statement).

Configure APS Timers

The protect and working routers periodically send packets to their neighbors to advertise that they are operational. By default, these advertisement packets are sent every 1000 milliseconds. A router considers its neighbor to be operational for a period, called the hold time, that is, by default, three times the advertisement interval. If the protect router does not receive an advertisement packet from the working router within the hold time configured on the protect router, the protect router assumes that the working router has failed and becomes active.

APS is symmetric; either side of a circuit can time out the other side (for example, when detecting a crash of the other). Under normal circumstances, the failure of the protect router does not cause any changes because the traffic is already moving on the working router. However, if you had configured request protect and the protect router failed, the working router would enable its interface.

To modify the advertisement interval, include the advertise-interval at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
advertise-interval milliseconds;
```

To modify the hold time, include the hold-time at the [edit interfaces *interface-name* sonet-options aps] hierarchy level:

```
[edit interfaces so-fpc/pic/port sonet-options aps]
hold-time milliseconds;
```

The advertisement intervals and hold times on the protect and working routers can be different.

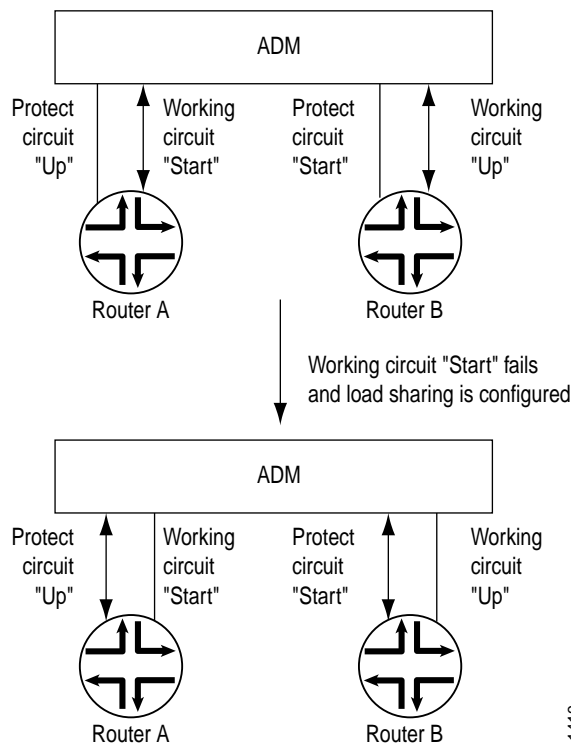
Configure APS Load Sharing between Circuit Pairs

When two routers are connected to a single ADM, you can have them back up each other on two different pairs of circuits. This arrangement provides load balancing between the routers in the event that one of the working circuits fail.

Figure 6 illustrates load sharing between circuits on two routers. Router A has a working circuit "Start" and a protect circuit "Up," and Router B has a working circuit "Up" and a protect circuit "Start." Under normal circumstances, Router A carries the "Start" circuit traffic and Router B carries the "Up" circuit traffic. If the working circuit "Start" were to fail, Router B would end up carrying all the traffic for both the "Start" and "Up" circuits.

To balance the load between the circuits, you pair together the two circuits. In this case, you pair the "Start" and "Up" circuits. Then, if the working circuit "Start" fails, the two routers automatically switch the "Up" traffic from the working to the protect circuit so that each router is still carrying only one circuit's worth of traffic. That is, the working circuit on Router A would be "Up" and the working circuit on Router B would be "Start."

Figure 6: APS Load Sharing between Circuit Pairs



1419

To configure load sharing between two working–protect circuit pairs, include the `paired-circuit` statement when configuring one of the circuits on one of the routers. In this statement, the `group-name` is the name of the group you assigned to one of the circuits with the `working-circuit` and `protect-circuit` statements. The software automatically configures the remainder of the load-sharing setup based on the group name.

```
[edit interfaces so-fpc/pic/port sonet-options aps]
paired-group group-name;
```

Example: Configure APS Load Sharing between Circuit Pairs

Configure APS load sharing to match the configuration shown in Figure 6:

On Router A:

```
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set working-circuit start
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-7/0/0 sonet-options aps]
user@host# set paired-circuit "Router A-Router B"
...
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set protect-circuit up
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-0/0/0 sonet-options aps]
user@host# set paired-circuit "Router A-Router B"
```

On Router B:

```
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set working-circuit up
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set authentication-key woolsey
[edit interfaces so-1/0/0 sonet-options aps]
user@host# set paired-circuit "Router A-Router B"
...
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set protect-circuit start
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set authentication-key linsey
[edit interfaces so-6/0/0 sonet-options aps]
user@host# set paired-circuit "Router A-Router B"
```

Configure 802.1Q VLAN Tagging

For Ethernet, Fast Ethernet, and Gigabit Ethernet interfaces, the software supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or broadcast domain.

The software supports receiving and forwarding routed Ethernet frames with 802.1Q VLAN tags, and running VRRP over 802.1Q-tagged interfaces. To configure the router to receive and forward frames with 802.1Q VLAN tags, include the `vlan-tagging` statement at the `[edit interfaces interface-name]` hierarchy level:

```
[edit interfaces interface-name]
vlan-tagging;
```

