

Chapter 5

Configure Physical Interface Properties

The software driver for each network media type sets reasonable default values for general interface properties, such as the interface's maximum transmission unit (MTU) size, receive and transmit leaky bucket properties, link operational mode, and clock source. To modify any of the default general interface properties, include the appropriate statements at the [edit interfaces *interface-name*] hierarchy level:

```
interfaces {
  interface-name {
    accounting-profile name;
    aggregated-ether-options {
      aggregated-ether-interface-options;
    }
    aggregated-sonet-options {
      aggregated-sonet-interface-options;
    }
    atm-options {
      atm-interface-options;
    }
    clocking clock-source;
    dce;
    description text;
    disable;
    ds0-options {
      ds0-interface-options;
    }
    e1-options {
      e1-interface-options;
    }
    e3-options {
      e3-interface-options;
    }
    es-options {
      es-interface-options;
    }
    encapsulation type;
    fastether-options {
      fastether-interface-options;
    }
    gigether-options {
      gigether-interface-options;
    }
    (gratuitous-arp-reply | no-gratuitous-arp-reply);
    hold-time up milliseconds down milliseconds;
    keepalives <down-count number> <interval seconds> <up-count number>;
    link-mode mode;
```

```

lmi {
  lmi-type (ansi | itu);
  n391dte number;
  n392dce number;
  n392dte number;
  n393dce number;
  n393dte number;
  t391dte seconds;
  t392dce seconds;
}
mac mac-address;
mtu bytes;
multiservice-options {
  boot-command filename;
  (core-dump | no-core-dump);
  (syslog | no-syslog);
}
no-gratuitous-arp-request;
no-keepalives;
no-partition {
  interface-type type;
}
partition partition-number oc-slice oc-slice-range interface-type type {
  timeslots time-slot-range;
}
passive-monitor-mode;
per-unit-scheduler;
ppp-options {
  chap {
    access-profile name;
    local-name name;
    passive;
  }
}
receive-bucket {
  overflow (tag | discard);
  rate percentage;
  threshold bytes;
}
serial-options {
  serial-interface-options;
}
service-options {
  service-interface-options;
}
sonet-options {
  sonet-interface-options;
}
speed (10m | 100m);
stacked-vlan-tagging;
t1-options {
  t1-interface-options;
}
t3-options {
  t3-interface-options;
}
traceoptions {
  flag flag <flag-modifier> <disable>;
}

```

```

    transmit-bucket {
        overflow discard;
        rate percentage;
        threshold bytes;
    }
    (traps | no-traps);
    unit {
        logical-interface-statements;
    }
    vlan-tagging;
}
}

```

This chapter discusses configuration of the following physical interface properties:

- Configure Aggregated Interfaces on page 45
- Add an Interface Description to the Configuration on page 46
- Configure the Link Characteristics on page 46
- Configure the Media MTU on page 47
- Configure Interface Encapsulation on page 51
- Configure PPP Challenge Handshake Authentication Protocol on page 55
- Configure the Interface Speed on page 57
- Configure Keepalives on page 58
- Configure the Clock Source on page 59
- Configure the Router as a DCE on page 59
- Configure Receive and Transmit Leaky Bucket Properties on page 60
- Configure Accounting for the Physical Interface on page 61
- Configure BERT Properties on page 62
- Trace Operations of an Individual Router Interface on page 64
- Damp Interface Transitions on page 64
- Configure Multiservice Physical Interface Properties on page 65
- Enable or Disable SNMP Notifications on Physical Interfaces on page 65
- Disable a Physical Interface on page 65

For information about interface-specific physical properties, see “Interface Types” on page 109.

Table 2 lists statements that you can use to configure physical interfaces.

Table 2: Statements for Physical Interface Properties

Statement	Interface Types	Usage Guidelines
access-profile <i>name</i>	Interfaces with PPP encapsulation	“Configure PPP Challenge Handshake Authentication Protocol” on page 55
accounting-profile <i>name</i>	All	“Configure Accounting for the Physical Interface” on page 61
acknowledge-retries <i>number</i>	Link services interface	“Configure Link Services Acknowledgment Timers” on page 329
acknowledge-timer <i>milliseconds</i>	Link services interface	“Configure Link Services Acknowledgment Timers” on page 329
aggregated-ether-options	Aggregated Ethernet interfaces	“Configure Aggregated Ethernet Interfaces” on page 299
aggregated-sonet-options	Aggregated SONET/SDH interfaces	“Configure Aggregated SONET/SDH Interfaces” on page 380
atm-options	ATM 1 and ATM 2 interfaces	“Configure ATM 1 and ATM 2 Physical Interface Properties” on page 125
boot-command <i>filename</i>	Monitoring services interfaces	“Configure Multiservice Physical Interface Properties” on page 65
chap	Interfaces with PPP encapsulation types	“Configure PPP Challenge Handshake Authentication Protocol” on page 55
clock-rate <i>rate</i>	Serial interfaces (EIA-530 and V.35)	“Configure the DTE Clock Rate” on page 351
clocking-mode (dce dte loop)	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Clocking Mode” on page 350
clocking <i>clock-source</i>	ATM, DS-0, E1, E3, SONET/SDH, T1, and T3 interfaces	“Configure the Clock Source” on page 59
control-leads	Serial interfaces (EIA-530, V.35, and X.21)	“Configure the Serial Signal Handling” on page 352
control-polarity (positive negative)	Serial interfaces (X.21)	“Configure Serial Signal Polarities” on page 355
control-signal (assert de-assert normal)	Serial interfaces (X.21)	“Configure the Serial Signal Handling” on page 352
(core-dump no-core-dump)	Monitoring services interfaces	“Configure Multiservice Physical Interface Properties” on page 65
cts (ignore normal require)	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Signal Handling” on page 352
cts-polarity (positive negative)	Serial interfaces (EIA-530 and V.35)	“Configure Serial Signal Polarities” on page 355
dcd (ignore normal require)	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Signal Handling” on page 352
dcd-polarity (positive negative)	Serial interfaces (EIA-530 and V.35)	“Configure Serial Signal Polarities” on page 355
dce	Interfaces with Frame Relay encapsulation	“Configure the Router as a DCE” on page 59
description <i>text</i>	All	“Add an Interface Description to the Configuration” on page 46
disable	All	“Disable a Physical Interface” on page 65
ds0-options	DS-0 interfaces	“Channelized Interfaces Overview” on page 163
dsr (ignore normal require)	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Signal Handling” on page 352

Statement	Interface Types	Usage Guidelines
dsr-polarity (positive negative)	Serial interfaces (EIA-530 and V.35)	“Configure Serial Signal Polarities” on page 355
dtr <i>signal-handling-option</i>	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Signal Handling” on page 352
dtr-circuit (balanced unbalanced)	Serial interfaces (EIA-530 and V.35)	“Configure the Serial DTR Circuit” on page 355
dtr-polarity (positive negative)	Serial interfaces (EIA-530 and V.35)	“Configure Serial Signal Polarities” on page 355
e1-options	E1 interfaces	“Configure E1 Interfaces” on page 243
e3-options	E3 interfaces	“Configure E3 Interfaces” on page 251
es-options	ES interfaces	“Configure ES PIC Redundancy” on page 259
encapsulation <i>type</i>	All interface types except aggregated Ethernet, loopback, and multicast tunnel	“Configure Interface Encapsulation” on page 51
encoding (nrz nrzi)	Serial interfaces (EIA-530, V.35, and X.21)	“Configure Serial Line Encoding” on page 357
fastether-options	Fast Ethernet interfaces	“Configure Ethernet Physical Interface Properties” on page 264
gigether-options	Gigabit Ethernet interfaces	“Configure Ethernet Physical Interface Properties” on page 264
(gratuitous-arp-reply no-gratuitous-arp-reply)	Ethernet interfaces	“Configure Gratuitous ARP” on page 281
hold-time up <i>milliseconds</i> down <i>milliseconds</i>	All interface types except aggregated SONET/SDH, GRE tunnel, and IP tunnel	“Damp Interface Transitions” on page 64
ignore-all	Serial interfaces (EIA-530, V.35 and X.21)	“Configure the Serial Signal Handling” on page 352
indication (ignore normal require)	Serial interfaces (X.21)	“Configure the Serial Signal Handling” on page 352
indication-polarity (positive negative)	Serial interfaces (X.21)	“Configure Serial Signal Polarities” on page 355
interface-type <i>type</i>	Channelized QPP interfaces	“Channelized Interfaces Overview” on page 163
keepalives <down-count <i>number</i> > <interval <i>seconds</i> > <up-count <i>number</i> >	Aggregated SONET/SDH, DS-0, E1, E3, SONET, T1, and T3 interfaces	“Configure Keepalives” on page 58
line-protocol <i>protocol</i>	Serial interfaces (EIA-530, V.35, and X.21)	“Configure the Serial Line Protocol” on page 347
link-mode <i>mode</i>	Management Ethernet (fxp0) and Fast Ethernet interfaces	“Configure the Link Characteristics” on page 46
lmi	Interfaces with Frame Relay encapsulation	“Configure Frame Relay Keepalives” on page 307
lmi-type (ansi itu)	Interfaces with Frame Relay encapsulation	“Configure Frame Relay Keepalives” on page 307
local-name <i>name</i>	Interfaces with PPP encapsulation	“Configure PPP Challenge Handshake Authentication Protocol” on page 55

Statement	Interface Types	Usage Guidelines
<code>loopback mode</code>	E1/E3/T1/T3, Ethernet, SONET/SDH, and Serial interfaces (EIA-530, V.35, and X.21)	“Configure Ethernet Loopback Capability” on page 280, “Configure E1 Loopback Capability” on page 247, “Configure E3 Loopback Capability” on page 254, “Configure SONET Loopback Capability” on page 364, “Configure T1 Loopback Capability” on page 392, “Configure T3 Loopback Capability” on page 401, and “Configure Serial Loopback Capability” on page 356
<code>mac mac-address</code>	Management Ethernet interface (fxp0)	“Configure the MAC Address on the Management Ethernet Interface” on page 298
<code>minimum-links</code>	Multilink and link services interfaces	“Configure Multilink and Link Services Interfaces” on page 319
<code>mtu bytes</code>	All interface types except Management Ethernet (fxp0), loopback, multilink, and multicast tunnel	“Configure the Media MTU” on page 47
<code>multiservice-options</code>	Monitoring services interfaces	“Configure Multiservice Physical Interface Properties” on page 65
<code>no-gratuitous-arp-request</code>	Ethernet interfaces	“Configure Gratuitous ARP” on page 281
<code>no-keepalives</code>	Interfaces with PPP, Frame Relay, or Cisco HDLC encapsulation	“Configure Keepalives” on page 58
<code>no-partition partition-number</code>	Channelized QPP interfaces	“Channelized Interfaces Overview” on page 163
<code>oc-slice oc-slice-range</code>	Channelized OC-12 QPP interfaces	“Configure Channelized OC-12 Interfaces” on page 183
<code>partition partition-number</code>	Channelized QPP interfaces	“Channelized Interfaces Overview” on page 163
<code>passive</code>	Interfaces with PPP encapsulation	“Configure PPP Challenge Handshake Authentication Protocol” on page 55
<code>per-unit-scheduler</code>	QPP interfaces	“Associate a Scheduler with a DLCI or VLAN on a Channelized QPP Interface” on page 597
<code>passive-monitor-mode</code>	SONET/SDH interfaces	“Enable Passive Monitoring” on page 374
<code>ppp-options</code>	Interfaces with PPP encapsulation	“Configure PPP Challenge Handshake Authentication Protocol” on page 55
<code>receive-bucket</code>	All interface types except ATM, Fast Ethernet, and Gigabit Ethernet	“Configure Receive and Transmit Leaky Bucket Properties” on page 60
<code>rts (assert de-assert normal)</code>	Serial interfaces (EIA-530 and V.35)	“Configure the Serial Signal Handling” on page 352
<code>rts-polarity (positive negative)</code>	Serial interfaces (EIA-530 and V.35)	“Configure Serial Signal Polarities” on page 355
<code>serial-options</code>	Serial interfaces (EIA-530, V.35, and X.21)	“Configure Serial Interfaces” on page 345
<code>sonet-options</code>	SONET interfaces	“Configure SONET/SDH Physical Interface Properties” on page 360
<code>speed (10m 100m)</code>	Management Ethernet interface (fxp0) and the Fast Ethernet 12-port and 48-port PIC	“Configure the Interface Speed” on page 57
<code>stacked-vlan-tagging</code>	Gigabit Ethernet QPP interfaces	“Stack and Rewrite Gigabit Ethernet QPP VLAN Tags” on page 273

Statement	Interface Types	Usage Guidelines
(syslog no-syslog)	Monitoring services interfaces	“Configure Multiservice Physical Interface Properties” on page 65
t1-options	T1 interfaces	“Configure T1 Interfaces” on page 387
t3-options	T3 interfaces	“Configure T3 Interfaces” on page 395
timeslots <i>time-slot-range</i>	Channelized T1 QPP and Channelized E1 QPP interfaces	“Channelized Interfaces Overview” on page 163
tm (ignore normal require)	Serial interfaces (EIA-530)	“Configure the Serial Signal Handling” on page 352
tm-polarity (positive negative)	Serial interfaces (EIA-530)	“Configure Serial Signal Polarities” on page 355
traceoptions	All	“Trace Operations of an Individual Router Interface” on page 64
transmit-bucket	All interface types except ATM, Fast Ethernet, and Gigabit Ethernet	“Configure Receive and Transmit Leaky Bucket Properties” on page 60
transmit-clock invert	Serial interfaces (EIA-530, V.35, and X.21)	“Configure the Serial Clocking Mode” on page 350
(traps no-traps)	All	“Enable or Disable SNMP Notifications on Physical Interfaces” on page 65
vlan-tagging	Fast Ethernet and Gigabit Ethernet interfaces	“Configure 802.1Q VLANs” on page 282

Configure Aggregated Interfaces

You specify aggregated interfaces by assigning a number for the aggregated interface. For aggregated Ethernet interfaces, configure `aex` as in the following example:

```
[edit interfaces]
ae0 {
...
}
```

For aggregated SONET/SDH interfaces, configure `asx` as in the following example:

```
[edit interfaces]
as0 {
...
}
```

The maximum number of aggregated interfaces is 16, and the assigned number can range from 0 through 15. You should not mix SONET and SDH mode on the same aggregated interface.



Note

SONET aggregation is proprietary to the JUNOS software and might not work with other software.

For aggregated Ethernet interfaces, you must include the `vlan-tagging` statement at the `[edit interfaces aex]` hierarchy level to complete the association.

For more information, see “Configure Aggregated Ethernet Interfaces” on page 299 and “Configure Aggregated SONET/SDH Interfaces” on page 380.

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, and is also exposed in the `ifAlias` MIB object. 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.

For information about describing logical units, see “Add a Logical Unit Description to the Configuration” on page 71.

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: phI01 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: phI01 P12/0/0 - local wire";
  encapsulation cisco-hdlc;
  ...
}
```

Configure the Link Characteristics

By default, the router’s management Ethernet interface, `fxp0`, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and 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 an 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 used on that interface. Table 3, Table 4, Table 5, Table 6, and Table 7 list the media MTU sizes by interface type, and Table 8 lists the encapsulation overhead by encapsulation type.

Table 3: Media MTU Sizes by Interface Type for M5, M10, 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	1500
E3/T3	4474	9192	4470
Fast Ethernet	1514		1500 (IPv4) 1497 (ISO)
4-port		9192	
8-port		1532	
12-port		1532	
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
Serial	1504	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	9192	4470

Table 4: Media MTU Sizes by Interface Type for M40e Routers

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
E1/T1	1504	4500	1500
E3/T3	4474	4500	4470
Fast Ethernet	1514	4500	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514		1500 (IPv4) 1497 (ISO)
1- or 2-port		9192	
4-port		4500	

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
Serial	1504	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474		4470
4-port OC-3		4500	
4-port OC-3c		4500	
1-port OC-12		4500	
4-port OC-12		4500	
4-port OC-12c		4500	
1-port OC-48		4500	
2-port OC-3		9192	
2-port OC-3c		9192	
1-port OC-12c		9192	
1-port OC-48c		9192	
1-port OC-192		9192	
1-port OC-192c		9192	

Table 5: 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	1500
E3/T3	4474	4500	4470
Fast Ethernet	1514	4500	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514		1500 (IPv4) 1497 (ISO)
1- or 2-port		9192	
4-port		4500	
Serial	1504	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474		4470
1- or 2-port		9192	
4-port		4500	

Table 6: Media MTU Sizes by Interface Type for T320 Platforms

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
ATM	4482	9192	4470
Fast Ethernet	1514		1500 (IPv4) 1497 (ISO)
4-port		4500	
48-port		1532	
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	9192	4470

Table 7: Media MTU Sizes by Interface Type for T640 Platforms

Interface Type	Default Media MTU (Bytes)	Maximum MTU (Bytes)	Default IP Protocol MTU (Bytes)
48-port Fast Ethernet	1514	1532	1500 (IPv4) 1497 (ISO)
Gigabit Ethernet	1514	9192	1500 (IPv4) 1497 (ISO)
SONET/SDH	4474	9192	4470

Table 8: Encapsulation Overhead by Encapsulation Type

Interface Encapsulation	Encapsulation Overhead (Bytes)
ATM Cell Relay	4
ATM PVC	12
Cisco HDLC	4
Frame Relay	4
Point-to-Point Protocol	4
Ethernet over ATM	28
Ethernet version 2	14
Ethernet 802.3	17
802.1Q/Ethernet version 2	18
802.1Q/Ethernet 802.3	21
Ethernet CCC and VPLS	4
Ethernet TCC	18
Ethernet SNAP	22
802.1Q/Ethernet SNAP	26
VLAN CCC and VPLS	4
Extended VLAN CCC and VPLS	4
Extended VLAN TCC	22

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.



Note

The actual frames transmitted also contain cyclic redundancy check (CRC) bits, which are not part of the media MTU. For example, the media MTU for a gigabit Ethernet interface is specified as 1500 bytes, but the largest possible frame size is actually 1504 bytes; you need to consider the extra bits in calculations of MTUs for interoperability.

The physical MTU for Ethernet interfaces does not include the 4-byte FCS field of the Ethernet frame.

A SONET interface operating in concatenated mode has a “c” added to the rate descriptor. For example, a concatenated OC-48 interface is referred to as OC-48c.

For information about configuring the encapsulation on an interface, see “Configure Interface Encapsulation” on page 51.

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 83.

Configure Interface Encapsulation

Point-to-Point Protocol (PPP) encapsulation is the default encapsulation type for physical interfaces. You need not configure encapsulation for any physical interfaces that support PPP encapsulation. If you do not configure encapsulation, PPP is used by default. For physical interfaces that do not support PPP encapsulation, 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. For more information about logical interface encapsulation, see “Configure the Encapsulation on a Logical Interface” on page 75.

This section is organized as follows:

Configure the Encapsulation on a Physical Interface on page 51

Encapsulation Capabilities on page 54

Example: Configure the Encapsulation on a Physical Interface on page 55

Configure the Encapsulation on a Physical Interface

By default, Point-to-Point Protocol (PPP) is the encapsulation type for physical interfaces. 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-ccc-cell-relay | atm-pvc | cisco-hdlc | cisco-hdlc-ccc | cisco-hdlc-tcc |
ethernet-ccc | ethernet-over-atm | ethernet-tcc | ethernet-vpls | extended-vlan-ccc |
extended-vlan-tcc | extended-vlan-vpls | frame-relay | frame-relay-ccc | frame-relay-tcc |
multilink-frame-relay-uni-nni | ppp | ppp-ccc | ppp-tcc | vlan-ccc | vlan-vpls);
```

The physical interface encapsulation can be one of the following:

ATM Cell Relay—Connects two remote virtual circuits or ATM physical interfaces with a label-switched path (LSP). Traffic on the circuit is ATM cells.

You can configure an ATM 1 PIC to use cell-relay accumulation mode (CAM). In this mode, the incoming cells (1 to 8 cells) are packaged into a single packet and forwarded to the label-switched path (LSP). Cell-relay accumulation mode is not supported on ATM 2 PICs. You configure CAM as shown in the following example:

```
[edit chassis]
fpc 1 {
  pic 0 {
    atm-cell-relay-accumulation;
  }
}
```

For more information, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

ATM PVC—Defined in RFC 1483, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*. When you configure physical ATM interfaces with ATM PVC encapsulation, an RFC 2684-compliant ATM Adaptation Layer 5 (AAL5) tunnel is set up to route the ATM cells over an MPLS path that is typically established between two MPLS-capable routers using the Label Distribution Protocol (LDP).

Cisco HDLC—E1, E3, SONET, T1, and T3 interfaces can use Cisco HDLC encapsulation. Two related versions are supported:

CCC version (cisco-hdlc-ccc)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation type, you can configure the family `ccc` only.

TCC version (cisco-hdlc-tcc)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

Ethernet over ATM—Allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (PDUs).

Ethernet Cross-Connect—Ethernet interfaces without VLAN tagging can use Ethernet CCC encapsulation. Two related versions are supported:

CCC version (ethernet-ccc)—Ethernet interfaces with standard Tag Protocol ID (TPID) tagging can use Ethernet CCC encapsulation. When you use this encapsulation type, you can configure the family `ccc` only.

TCC version (ethernet-tcc)—Similar to CCC, but used for circuits with different media on either side of the connection. One-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Ethernet TCC encapsulation.

VLAN Circuit Cross-Connect (CCC) (vlan-ccc)—Ethernet interfaces with virtual local area network (VLAN) tagging enabled can use VLAN CCC encapsulation. VLAN CCC encapsulation supports TPID 0x8100 only. When you use this encapsulation type, you can configure the family `ccc` only.

Extended VLAN Cross-Connect—Gigabit Ethernet interfaces with virtual local area network (VLAN) 802.1Q tagging enabled can use extended VLAN cross-connect encapsulation. (Ethernet interfaces with standard TPID tagging can use VLAN CCC encapsulation.) Two related versions of extended VLAN cross-connect are supported:

CCC version (extended-vlan-ccc)—Extended VLAN CCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. Extended VLAN CCC is not supported on four-port Gigabit Ethernet PICs. When you use this encapsulation type, you can configure the family `ccc` only.

TCC version (extended-vlan-tcc)—Similar to CCC, but used for circuits with different media on either side of the connection. One-port Gigabit Ethernet, two-port Gigabit Ethernet, and four-port Fast Ethernet PICs can use Extended Ethernet TCC encapsulation.

Ethernet VPLS (ethernet-vpls)—Ethernet interfaces with Virtual Private LAN Service (VPLS) enabled can use Ethernet VPLS encapsulation. For more information about VPLS, see the *JUNOS Internet Software Configuration Guide: VPNs* and the *JUNOS Internet Software Feature Guide*.

Ethernet VLAN VPLS (`vlan-vpls`)—Ethernet interfaces with virtual local area network (VLAN) tagging and VPLS enabled can use Ethernet VLAN VPLS encapsulation. For more information about VPLS, see the *JUNOS Internet Software Configuration Guide: VPNs* and the *JUNOS Internet Software Feature Guide*.

Extended VLAN VPLS (`extended-vlan-vpls`)—Ethernet interfaces with virtual local area network (VLAN) 802.1Q tagging and VPLS enabled can use Ethernet Extended VLAN VPLS encapsulation. (Ethernet interfaces with standard TPID tagging can use Ethernet VLAN VPLS encapsulation.) Extended Ethernet VLAN VPLS encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. For more information about VPLS, see the *JUNOS Internet Software Configuration Guide: VPNs* and the *JUNOS Internet Software Feature Guide*.

Frame Relay—Defined in RFC 1490, *Multiprotocol Interconnect over Frame Relay*. E1, E3, link services, SONET, T1, and T3 interfaces can use Frame Relay encapsulation. Two related versions are supported:

CCC version (`frame-relay-ccc`)—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. When you use this encapsulation type, you can configure the family `ccc` only.

TCC version (`frame-relay-tcc`)—Similar to Frame Relay CCC and has the same configuration restrictions, but is used for circuits with different media on either side of the connection.

Multilink Frame Relay (MLFR) User-to-Network Interface (UNI) and Network-to-Network Interface (NNI) (`multilink-frame-relay-uni-nni`)—Link services interfaces functioning as FRF.16 bundles can use multilink Frame Relay UNI NNI encapsulation. This encapsulation is also used on link services interfaces' constituent T1, E1, or NxDS-0 interfaces.

Point-to-Point Protocol (PPP)—Defined in RFC 1661, *The Point-to-Point Protocol (PPP) for the Transmission of Multiprotocol Datagrams over Point-to-Point Links*. PPP is the default encapsulation type for physical interfaces. E1, E3, SONET, T1, and T3 interfaces can use PPP encapsulation. Two related versions are supported:

Circuit cross-connect (CCC) version (`ppp-ccc`)—The logical interfaces do not require an encapsulation statement. When you use this encapsulation type, you can configure the family `ccc` only.

Translational cross-connect (TCC) version (`ppp-tcc`)—Similar to CCC and has the same configuration restrictions, but used for circuits with different media on either side of the connection.

Encapsulation Capabilities

When you configure a point-to-point encapsulation (such as PPP or 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 CCC encapsulation for Ethernet interfaces with standard TPID tagging requires that the physical interface have only a single logical interface. Ethernet interfaces in VLAN mode can have multiple logical interfaces.

For Ethernet interfaces in VLAN mode, VLAN IDs are applicable as follows:

VLAN ID 0 is reserved for tagging the priority of frames.

For encapsulation type `vlan-ccc`, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 and above are reserved for CCC VLANs.

For encapsulation type `vlan-vpls`, VLAN IDs 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 and above are reserved for VPLS VLANs.

For encapsulation types `extended-vlan-ccc` and `extended-vlan-vpls`, all VLAN IDs are valid.

The upper limits for configurable VLAN IDs vary by interface type. For more information, see “Configure 802.1Q VLANs” on page 282 and Table 23 on page 283.

When you configure a TCC encapsulation, some modifications are needed to handle VPN connections over unlike Layer 2 and Layer 2.5 links and terminate the Layer 2 and Layer 2.5 protocol locally. The router performs the following media-specific changes:

PPP TCC—Both Link Control Protocol (LCP) and Network Control Protocol (NCP) are terminated on the router. Internet Protocol Control Protocol (IPCP) IP address negotiation is not supported. The JUNOS software strips all PPP encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to PPP encapsulation.

Cisco HDLC TCC—Keepalive processing is terminated on the router. The JUNOS software strips all Cisco HDLC encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Cisco HDLC encapsulation.

Frame Relay TCC—All Local Management Interface (LMI) processing is terminated on the router. The JUNOS software strips all Frame Relay encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to Frame Relay encapsulation.

ATM—Operation, Administration, and Maintenance (OAM) and Interim Local Management Interface (ILMI) processing is terminated at the router. Cell relay is not supported. The JUNOS software strips all ATM encapsulation data from incoming frames before forwarding them. For output, the next hop is changed to ATM encapsulation.

Example: Configure the Encapsulation on a Physical Interface

Configure PPP encapsulation on a SONET interface. The second and third 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 PPP Challenge Handshake Authentication Protocol

For interfaces with PPP encapsulation, you can configure interfaces to support PPP Challenge Handshake Authentication Protocol (CHAP), as defined in RFC 1994. When you enable CHAP on an interface, the interface can authenticate its peer and can be authenticated by its peer.

By default, PPP CHAP is disabled. If CHAP is not explicitly enabled, the interface makes no CHAP challenges and denies all incoming CHAP challenges. To enable CHAP, you must create an access profile, and you must configure the interfaces to use CHAP.

To configure a CHAP access profile, include the profile statement and specify a profile name at the [edit access] hierarchy level:

```
[edit access]
profile profile-name {
  client name chap-secret data;
}
```

For more information about configuring access profiles, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

When you configure an interface to use CHAP, you must assign an access profile to the interface. When an interface receives CHAP challenges and responses, the access profile in the packet is used to look up the shared secret, as defined in RFC 1994.

To configure PPP CHAP on an interface with PPP encapsulation, include the chap statement at the [edit interfaces *interface-name* ppp-options] hierarchy level:

```
[edit interfaces interface-name ppp-options]
chap {
  access-profile name;
  local-name name;
  passive;
}
```

On each interface with PPP encapsulation, you can configure the following PPP CHAP properties:

Assign an Access Profile to an Interface on page 56

Configure the Local Name on page 56

Configure Passive Mode on page 56

When you configure PPP over ATM or Multilink PPP over ATM encapsulation, you can enable CHAP on the logical interface. For more information, see “Configure PPP over ATM 2 Encapsulation” on page 150.

Assign an Access Profile to an Interface

To assign an access profile to an interface, include the access-profile statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
access-profile name;
```

You must include the access-profile statement when you configure the CHAP authentication method. If an interface receives a CHAP challenge or response from a peer that is not in the applied access profile, the link is immediately dropped.

Configure the Local Name

By default, when CHAP is enabled on an interface, the interface uses the router's system hostname as the name sent in CHAP challenge and response packets.

To configure the name the interface uses in CHAP challenge and response packets, include the local-name statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
local-name name;
```

Configure Passive Mode

By default, when CHAP is enabled on an interface, the interface always challenges its peer and responds to challenges from its peer.

You can configure the interface not to challenge its peer, and only respond when challenged. To configure the interface not to challenge its peer, include the passive statement at the [edit interfaces *interface-name* ppp-options chap] hierarchy level:

```
[edit interfaces interface-name ppp-options chap]
passive;
```

Example: Configure PPP Challenge Handshake Authentication Protocol

Configure CHAP:

```
[edit access]
  profile pe-A-ppp-clients;
    client cpe-1 chap-secret "$1$dQYsZ$B5ojUeUjDsUo.yKwcCZ0"; # SECRET-DATA
    client cpe-2 chap-secret "$1$kdAsfaDAfkdjDsASxfadKdFKJ"; # SECRET-DATA
  }
}

[edit interfaces so-1/2/0]
  encapsulation ppp;
  ppp-options {
    chap {
      access-profile pe-A-ppp-clients;
      local-name "pe-A-so-1/1/1";
    }
  }
}

[edit interfaces so-1/1/2]
  encapsulation ppp;
  ppp-options {
    chap {
      access-profile pe-A-ppp-clients;
      local-name "pe-A-so-1/1/2";
    }
  }
}
```

Configure the Interface 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 the *JUNOS Internet Software Guide: Getting Started*).

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 Keepalives

By default, physical interfaces configured with Cisco HDLC or PPP encapsulation send keepalive packets at 10-second intervals. The Frame Relay term for keepalives is Local Management Interface (LMI) packets; the JUNOS software supports both ANSI T1.617 Annex D LMIs and ITU Q933 Annex A LMIs. On ATM networks, Operation, Administration, and Maintenance (OAM) cells perform the same function. You configure OAM cells at the logical interface level; for more information, see “Define the ATM 1 and ATM 2 OAM F5 Loopback Cell Period” on page 144.

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;
```

When you configure PPP over ATM or Multilink PPP over ATM encapsulation, you can enable or disable keepalives on the logical interface. For more information, see “Configure PPP over ATM 2 Encapsulation” on page 150.

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;
```

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>;
```

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

`interval seconds`—The time in seconds between successive keepalive requests. The range is 1 second 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.

For information about Frame Relay keepalive settings, see “Configure Frame Relay Keepalives” on page 307.

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 timing) 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 T3 channels on an Channelized OC-12 interface, T1 channels on an Channelized T3 interface, and DS-0 channels on an Channelized E1 interface, the clocking statement is supported only for channel 0; it is ignored if included in the configuration of other channels. The clock source configured for channel 0 applies to all channels on the Channelized OC-12, Channelized DS-3, and Channelized E1 interfaces. The individual DS-3, DS-1, and DS-0 channels use a gapped 45-MHz clock as the transmit clock. For more information, see "Clock Sources on Channelized Interfaces" on page 166.



On Channelized STM-1 interfaces, you should configure the clock source at one side of the connection to be internal (the default JUNOS configuration) and configure the other side of the connection to be external.

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

```
[edit 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

Congestion control is particularly difficult in high-speed networks with high volumes of traffic. When congestion occurs in such a network, it is usually too late to react. You can avoid congestion by regulating the flow of packets into your network. Smoother flows prevent bursts of packets from arriving at (or being transmitted from) the same interface and causing congestion.

For all interface types except ATM, Channelized E1, E1, Fast Ethernet, Gigabit Ethernet, and Channelized QPP, 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.



Instead of configuring leaky bucket properties, you can limit traffic flow by configuring policers. Policers work on all interfaces. For more information, see “Apply Policers” on page 87 and the *JUNOS Internet Software Configuration Guide: Policy Framework*.

The leaky bucket is used at the host-network interface to allow packets into the network at a constant rate. Packets might be generated in a bursty manner, but after they pass through the leaky bucket, they enter the network evenly spaced. In some cases, you might want to allow short bursts of packets to enter the network without smoothing them out. By controlling the number of packets that can accumulate in the bucket, the threshold property controls burstiness. The maximum number of packets entering the network in t time units is $\text{threshold} + \text{rate} * t$.

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

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]
receive-bucket {
  overflow (tag | discard);
  rate percentage;
  threshold bytes;
}
transmit-bucket {
  overflow discard;
  rate percentage;
  threshold bytes;
}
```

In the rate statement, 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 statement, 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 statement, specify how to handle packets that exceed the threshold:

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

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

Configure Accounting for the Physical Interface

Juniper Networks routers can collect various kinds of data about traffic passing through the router. You can set up one or more *accounting profiles* that specify some common characteristics of this data, including the following:

The fields used in the accounting records

The number of files that the router retains before discarding, and the number of bytes per file

The polling period that the system uses to record the data

You configure the profiles and define a unique name for each profile using statements at the [edit accounting-options] hierarchy level. There are two types of accounting profiles: interface profiles and filter profiles. You configure interface profiles by including the interface-profile statement at the [edit accounting-options] hierarchy level. You configure filter profiles by including the filter-profile statement at the [edit accounting-options] hierarchy level. For more information, see the *JUNOS Internet Software Configuration Guide: Network Management*.

You apply interface profiles by including the accounting-profile statement at the [edit interfaces *interface-name*] and [edit interfaces *interface-name* unit *number*] hierarchy levels. You apply filter profiles by including the accounting-profile statement at the [edit firewall filter *filter-name*] and [edit firewall family *family* filter *filter-name*] hierarchy levels.

Apply an Accounting Profile to the Physical Interface

To enable accounting on an interface, include the accounting-profile statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
accounting-profile name;
```

You can also reference profiles by logical unit; for more information, see “Configure Accounting for the Logical Interface” on page 72. For information about configuring a firewall filter accounting profile, see the *JUNOS Internet Software Configuration Guide: Network Management*.

Example: Apply an Accounting Profile to the Physical Interface

Configure an accounting profile for an interface and apply it to a physical interface:

```
[edit]
accounting-options {
  file if_stats {
    size 4m files 10 transfer-interval 15;
    archive-sites {
      "ftp://login:password@host/path";
    }
  }
}
interface-profile if_profile {
  interval 15;
  file if_stats {
    fields {
      input-bytes;
      output-bytes;
      input-packets;
      output-packets;
      input-errors;
      output-errors;
    }
  }
}
}

[edit interfaces ge-1/0/1]
accounting-profile if_profile;
```

Configure BERT Properties

You can configure any of the following interfaces to execute a bit error rate test (BERT) when the interface receives a request to run this test: E1, E3, T1, T3, the Channelized DS-3, OC-3, OC-12, and STM-1 interfaces, and the Channelized DS-3, E1, and OC-12 QPP interfaces. On all of the specified interface types, you set the duration of the test and the error rate to include in the bit stream by including the `bert-period` and `bert-error-rate` statements at the [edit interfaces *interface-name* *interface-options*] hierarchy level:

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

seconds is the duration of the BERT procedure. The test can last from 1 to 240 seconds; the default is 10 seconds.

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, 1 error per bit) to 10^{-7} (that is, 1 error per 10 million bits).

algorithm is the pattern to send in the bit stream. The algorithm for the E1 BERT procedure is pseudo-2e15-o151 (pattern is $2^{15}-1$, as defined in the CCITT/ITU O.151 standard). On T1, E3, T3, NxDS-0, and Channelized E1 and T3 QPP interfaces, you can also select the pattern to send in the bit stream by including the bert-algorithm statement at the [edit interfaces *interface-name interface-options*] hierarchy level:

```
[edit interfaces interface-name interface-options]
bert-algorithm algorithm;
```

For a list of supported algorithms, see the CLI possible completions; for example:

```
[edit interfaces t1-0/0/0 t1-options]
user@host# set bert-algorithm ?
Possible completions:
pseudo-2e11-o152  Pattern is 2^11 -1 (per O.152 standard)
pseudo-2e15-o151  Pattern is 2^15 - 1 (per O.152 standard)
pseudo-2e20-o151  Pattern is 2^20 - 1 (per O.151 standard)
pseudo-2e20-o153  Pattern is 2^20 - 1 (per O.153 standard)
...
```

See individual interface types for specific hierarchy information. For information about running the BERT procedure, see the *JUNOS Internet Software Operational Mode Command Reference*.

Table 9 shows the BERT capabilities for various interface types.

Table 9: BERT Capabilities by Interface Type

Interface	T1 BERT	T3 BERT	Comments
E1 or T1	Yes (port 0–3)	Yes (port 0–3)	Single port at a time Limited algorithms
E3 or T3	Yes (port 0–3)	Yes (port 0–3)	Single port at a time
Channelized OC-12	N/A	Yes (channel 0–11)	Single channel at a time Limited algorithms No bit count
Channelized STM-1	Yes (channel 0–62)	N/A	Multiple channels Only one algorithm No error insert No bit count
Channelized T3 and Multichannel T3	Yes (channel 0–27)	Yes (port 0–3 on channel 0)	Multiple ports and channels Limited algorithms for T1 No error insert for T1 No bit count for T1

For information about BERT capabilities on channelized QPP interfaces, see “Channelized QPP Interface Properties” on page 169.

Trace Operations of an Individual Router Interface

To trace the operations of individual router interfaces, include the `traceoptions` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]
traceoptions {
  flag flag <disable>;
}
```

You can specify the following interface tracing flags:

`all`—Trace all interface operations.

`event`—Trace all interface events.

`ipc`—Trace all interface IPC messages.

`media`—Trace all interface media changes.

The interfaces `traceoptions` statement does not support a trace file. The logging is done by the kernel, so the tracing information is placed in the system syslog files.

For more information about trace operations, see “Trace Interface Operations” on page 107.

Damp Interface Transitions

By default, when an interface changes 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 passed, called the *hold-time*. 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 65,534 milliseconds. Upon execution, the time value that you specify is rounded up to the nearest whole second; therefore, we recommend that you configure the up and down options to multiples of 1000. The default value is 0, which means that interface transitions are not damped.

Configure Multiservice Physical Interface Properties

The monitoring services PIC is one of a group of multiservice PICs specifically designed to enable IP services. To configure multiservice physical interface properties on the monitoring services interface, include the multiservice-options statement at the [edit interfaces *mo-fpc/pic/port*] hierarchy level:

```
[edit interfaces mo-fpc/pic/port]
multiservice-options {
  boot-command filename
  (core-dump | no-core-dump);
  (syslog | no-syslog);
}
```

For more information about the monitoring services interface, see “Enable Passive Monitoring” on page 374.

Enable or Disable SNMP Notifications on Physical Interfaces

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

```
[edit interfaces interface-name]
(traps | 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]
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;
      }
    }
  }
}
```