

Chapter 11

Configuring ATM Interfaces

Asynchronous Transfer Mode (ATM) is a network protocol designed to facilitate the simultaneous handling of various types of traffic streams (voice, data, and video) at very high speeds over the same physical connection. By always using 53-byte cells, ATM simplifies the design of hardware, enabling it to quickly determine the destination address of each cell. This allows simple switching of network traffic at much higher speeds than are easily accomplished using protocols with variable sizes of transfer units, such as Frame Relay and Transmission Control Protocol/Internet Protocol (TCP/IP).

Although ATM was designed to operate without the requirement of any other networking protocol, other protocols are frequently segmented and encapsulated across multiple, smaller ATM cells. This makes ATM a transport mechanism for pre-existing technologies such as Frame Relay and the TCP/IP family of protocols.

ATM relies on the concepts of virtual paths and virtual circuits. A virtual path, represented by a specific virtual path identifier (VPI), establishes a route between two devices in a network. Each VPI can contain multiple virtual circuits, each represented by a virtual circuit identifier (VCI).

VPIs and VCIs are local to the routing platform, which means that only the two devices connected by the VCI or VPI need know the details of the connection. In a typical ATM network, user data might traverse multiple connections, using many different VPI and VCI connections. Each end device, just like each device in the network, needs to know only the VCI and VPI information for the path to the next device.



NOTE: The ATM three-bit payload type identifier (PTI) field is not supported.

With ATM2 intelligent queuing (IQ) interfaces, you can configure virtual path (VP) shaping and operation, administration, and maintenance (OAM) F4 cell flows.

This chapter is organized as follows:

ATM1 Physical and Logical Configuration Hierarchies on page 178

ATM2 IQ Physical and Logical Configuration Hierarchies on page 180

Supported Features on ATM1 and ATM2 IQ Interfaces on page 183

Configuring ATM Interfaces on page 186

ATM1 Physical and Logical Configuration Hierarchies

To configure ATM1 physical interface properties, include the `atm-options`, `e3-options`, `t3-options`, and `sonet-options` statements at the [edit interfaces `at-fpc/pic/port`] hierarchy level:

ATM1 Physical Configuration Hierarchy

```
[edit interfaces at-fpc/pic/port]
atm-options {
  ilmi;
  mpls {
    pop-all-labels {
      required-depth number;
    }
  }
  pic-type atm1;
  promiscuous-mode {
    vpi vpi-identifier;
  }
  vpi vpi-identifier {
    maximum-vcs maximum-vcs;
  }
}
e3-options {
  atm-encapsulation (direct | plcp);
  buildout feet;
  framing (g.751 | g.832);
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
}
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;
  }
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
  rfc-2615;
```

```

trigger {
  defect ignore {
    hold-time up milliseconds down milliseconds;
  }
}
(z0-increment | no-z0-increment);
}
t3-options {
  atm-encapsulation (direct | plcp);
  buildout feet;
  (cbit-parity | no-cbit-parity);
  loopback (local | payload | remote);
  (payload-scrambler | no-payload-scrambler);
}

```

ATM1 Logical Configuration Hierarchy

To configure ATM1 logical interface properties, include the following statements:

```

allow-any-vci;
multicast-vci vpi-identifier.vci-identifier;
oam-liveness {
  up-count cells;
  down-count cells;
}
oam-period (disable | seconds);
shaping {
  (cbr rate | vbr peak rate sustained rate burst length);
  queue-length number;
}
vci vpi-identifier.vci-identifier;
vpi vpi-identifier;
family inet {
  address address {
    multipoint-destination address {
      inverse-arp;
      oam-liveness {
        up-count cells;
        down-count cells;
      }
      oam-period (disable | seconds);
      shaping {
        (cbr rate | vbr peak rate sustained rate burst length);
        queue-length number;
      }
      vci vpi-identifier.vci-identifier;
    }
  }
}
}

```

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

ATM2 IQ Physical and Logical Configuration Hierarchies

To configure ATM2 IQ physical interface properties, include the `atm-options` and `sonet-options` statements at the `[edit interfaces at-fpc/pic/port]` hierarchy level:

ATM2 IQ Physical Configuration Hierarchy

```
[edit interfaces at-fpc/pic/port]
atm-options {
  cell-bundle-size cells;
  ilmi;
  linear-red-profiles profile-name {
    high-plp-max-threshold percent;
    low-plp-max-threshold percent;
    queue-depth cells high-plp-threshold percent low-plp-threshold percent;
  }
  mpls {
    pop-all-labels {
      required-depth number;
    }
  }
  pic-type atm2;
  plp-to-clp;
  promiscuous-mode {
    vpi vpi-identifier;
  }
  scheduler-maps map-name {
    forwarding-class class-name {
      epd-threshold cells plp1 cells;
      linear-red-profile profile-name;
      priority (high | low);
      transmit-weight (cells number | percent number);
    }
    vc-cos-mode (alternate | strict);
  }
  vpi vpi-identifier {
    oam-liveness {
      up-count cells;
      down-count cells;
    }
    oam-period (disable | seconds);
    shaping {
      (cbr rate | rtvbr peak rate sustained rate burst length |
       vbr peak rate sustained rate burst length);
    }
  }
}
```

```

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;
  }
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
  rfc-2615;
  trigger {
    defect ignore {
      hold-time up milliseconds down milliseconds;
    }
  }
  (z0-increment | no-z0-increment);
}

```

ATM2 IQ Logical Configuration Hierarchy

To configure ATM2 IQ logical interface properties, include the following statements:

```

allow-any-vci;
atm-scheduler-map (map-name | default);
cell-bundle-size cells;
epd-threshold cells;
multicast-vci vpi-identifier.vci-identifier;
oam-liveness {
  up-count cells;
  down-count cells;
}
oam-period (disable | seconds);
plp-to-clp;
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length |
  vbr peak rate sustained rate burst length);
}
transmit-weight number;
trunk-id number;
vci vpi-identifier.vci-identifier;
vpi vpi-identifier;

```

```

family inet address address {
  multipoint-destination address
  epd-threshold cells;
  inverse-arp;
  oam-liveness {
    up-count cells;
    down-count cells;
  }
  oam-period (disable | seconds);
  shaping {
    (cbr rate | rtvbr peak rate sustained rate burst length |
     vbr peak rate sustained rate burst length);
  }
  transmit-weight number;
  vci vpi-identifier.vci-identifier;
}
}

```

You can include these statements at the following hierarchy levels:

[edit interfaces *interface-name* unit *logical-unit-number*]

[edit logical-routers *logical-router-name* interfaces *interface-name* unit *logical-unit-number*]

Supported Features on ATM1 and ATM2 IQ Interfaces

Table 20 lists the supported features on ATM1 and ATM2 IQ interfaces.

Table 20: ATM1 and ATM2 IQ Supported Features

Item	ATM1	ATM2 IQ	Comments
Encapsulation and Transport Modes			
ATM Adaptation Layer 5 (AAL5) circuit cross-connect (CCC)	Supported	Supported	For ATM1 and ATM2 IQ Physical Interface Cards (PICs), you can configure any combination of AAL5 CCC, nonpromiscuous cell relay, and AAL5 permanent virtual connections (PVCs) on the same PIC at the same time. See "Configuring ATM Interface Encapsulation" on page 230.
Cell-relay accumulation mode: The incoming cells (1 to 8) are packaged into a single packet and forwarded to the label-switched path (LSP).	Supported	Not supported	Cell-relay accumulation mode is per PIC, not per port. If you configure accumulation mode, the entire ATM1 PIC uses the configured mode. See "Configuring ATM Interface Encapsulation" on page 230.
Cell-relay promiscuous port mode: All cells from 0 through 65,535 of all VPIs (0 through 255) are sent to or received from an LSP.	Supported	Supported	For promiscuous mode, you must configure the port with atm-ccc-cell-relay encapsulation. For ATM2 IQ multiport PICs, you can configure one or more ports in port promiscuous mode, and the other ports with any ATM encapsulation.
Cell-relay promiscuous VPI mode: All cells in the VCI range 0 through 65,535 of a single VPI are sent to or received from an LSP.	Supported	Supported	For ATM2 IQ PICs, you can configure one or more logical interfaces in VPI promiscuous mode, and the other logical interfaces with any ATM encapsulation. For ATM1 PICs, if you configure one port in port mode, all ports on the PIC operate in port mode. Likewise if you configure one logical interface in VPI mode, all logical interfaces on the PIC operate in VPI mode. See "Configuring ATM Cell-Relay Promiscuous Mode" on page 193.
Cell-relay VCI mode: All cells in a VCI are sent to or received from an LSP.	Supported	Supported	For ATM1 PICs, nonpromiscuous cell-relay VCI, VPI, and port modes are supported on the same PIC with ATM AAL5 PVCs or ATM AAL5 CCC.
Cell-relay VPI mode: All cells in the VCI range (0 through <i>maximum-vc</i> s) of a single VPI are sent to or received from an LSP.	Supported	Not supported	For ATM2 IQ PICs, nonpromiscuous cell-relay VCI mode is supported on the same PIC with ATM AAL5 PVCs or ATM AAL5 CCC. See "Configuring ATM Interface Encapsulation" on page 230.
Cell-relay port mode: All cells in the VCI range (0 through <i>maximum-vc</i> s) of all VPIs (0 through 255) are sent to or received from an LSP.	Supported	Not supported	
Ethernet over ATM encapsulation: Allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (PDUs).	Supported	Supported	See "Configuring ATM Interface Encapsulation" on page 230.

Item	ATM1	ATM2 IQ	Comments
Layer 2 circuit cell-relay, Layer 2 circuit AAL5, and Layer 2 circuit trunk transport modes: Allow you to send ATM cells or AAL5 PDUs between ATM2 IQ interfaces across a Layer 2 circuit-enabled network. Layer 2 circuits are designed to transport Layer 2 frames between provider edge (PE) routing platforms across a Label Distribution Protocol (LDP)-signaled Multiprotocol Label Switching (MPLS) backbone.	Not supported	Supported	Transport mode is per PIC, not per port. If you configure Layer 2 circuit cell-relay, Layer 2 circuit AAL5, or Layer 2 circuit trunk transport mode, the entire ATM2 IQ PIC uses the configured transport mode. Layer 2 circuit cell-relay mode supports both VP- and port-promiscuous modes. See “Configuring Layer 2 Circuit Transport Mode” on page 198.
Layer 2 VPN cell relay and Layer 2 VPN AAL5: Allow you to carry ATM cells or AAL5 PDUs over an MPLS backbone.	Supported	Supported	See the <i>JUNOS VPNs Configuration Guide</i> .
Point-to-Point Protocol (PPP) over ATM encapsulation: Associates a PPP link with an ATM AAL5 PVC.	Not supported	Supported	For ATM2 IQ interfaces, the JUNOS software supports three PPP over ATM encapsulation types: atm-ppp-llc—PPP over AAL5 logical link control (LLC). atm-ppp-vc-mux—PPP over AAL5 multiplex. atm-mlppp-llc—Multilink PPP over AAL5 LLC. Requires a Link Services or Voice Services PIC. See “Configuring PPP over ATM2 Encapsulation” on page 234.
Other ATM Attributes			
EPD (early packet discard) threshold: Limits the queue size in ATM cells of a particular VC or forwarding class configured over a VC when using VC tunnel class of service (CoS). When the first ATM cell of a new packet is received, the VC’s queue depth is checked against the EPD threshold. If the VC’s queue depth exceeds the EPD threshold, the first and all subsequent ATM cells in the packet are discarded.	Not supported	Supported	If you are using VC tunnel CoS, the EPD threshold configured at the logical unit level has no effect. You should configure each forwarding class for congestion management using either an individual EPD threshold (in other words, tail drop) or weighted random early detection (WRED) profile. See “Configuring the ATM2 IQ EPD Threshold” on page 226 and “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.
OAM F4 cell flows: Identify and report virtual path connection (VPC) defects and failures.	Not supported	Supported	See “Configuring the OAM F4 Cell Flows” on page 214.
OAM F5 loopback cell responses	Supported	Supported	For ATM1 interfaces, when an OAM F5 loopback request is received, the response cell is sent by the PIC. The request and response cells are not counted in the VC, logical interface, or physical interface statistics. For ATM2 IQ interfaces, when an OAM F5 loopback request is received, the response is sent by the routing engine. The OAM, VC, logical interface, and physical interface statistics are incremented. See “Defining the ATM OAM F5 Loopback Cell Period” on page 229 and “Configuring the ATM OAM F5 Loopback Cell Threshold” on page 229.
Passive monitoring mode	Supported	Supported	See “Enabling Passive Monitoring on ATM Interfaces” on page 190.

Item	ATM1	ATM2 IQ	Comments
PIC type	Supported	Supported	For ATM1 interfaces, you can include the pic-type atm1 statement. For ATM2 IQ interfaces, you can include the pic-type atm2 statement. See “Configuring the ATM PIC Type” on page 192.
Ping	Supported	Supported	For ATM1 and ATM2 IQ interfaces, when you issue the ATM ping command, you must include a logical unit number in the interface name, as shown in the following example: ping atm interface at-1/0/0.5 vci 0.123 count 3 The logical unit number is 5 on physical interface at-1/0/0. See the <i>JUNOS Protocols, Class of Service, and System Basics Command Reference</i> .
Queue length: Limits the queue size in packets of a particular VC.	Supported	Not supported	See “Configuring the ATM1 Queue Length” on page 225.
Real-time variable bit rate (VBR): Supports VBR data traffic with average and peak traffic parameters.	Not supported	Supported	Compared to non-real-time VBR, real-time VBR data is serviced at a higher priority. Real-time VBR is suitable for carrying packetized video and audio. See “Configuring ATM2 IQ Real-Time VBR” on page 220.
Shaping rates: Peak and sustained rates of traffic.	Supported	Supported	For ATM1 OC3 interfaces, the rate can be from 33 kilobits per second (Kbps) through 135.6 Megabits per second (Mbps); for ATM1 OC12 interfaces, the rate can be from 33 Kbps through 276 Mbps. For ATM2 IQ OC3 and OC12 interfaces, the rate can be from 33 Kbps through 542,526,792 bits per second (bps). For ATM2 IQ DS3 and E3 interfaces, the rate can be from 33 Kbps to the maximum rate. The maximum rate varies depending on the ATM encapsulation and framing you configure: For DS3 interfaces with direct ATM encapsulation, the maximum rate is 40,038,968 bps. For DS3 interfaces with Physical Layer Convergence Protocol (PLCP) ATM encapsulation, the maximum rate is 36,864,000 bps. For E3 interfaces with g.751 framing and direct ATM encapsulation, the maximum rate is 30,801,509 bps. For E3 interfaces with g.751 framing PLCP ATM encapsulation, the maximum rate is 27,648,000 bps. For E3 interfaces with g.832 framing, the maximum rate is 30,720,000 bps. See “Defining the ATM Traffic-Shaping Profile” on page 218.
VC tunnel CoS: Allows VCs to be opened as VC tunnels.	Not supported	Supported	On M-series platforms (except the M320 router), a VC tunnel can support four CoS queues. On the M320 and T-series platforms, a VC tunnel can support eight CoS queues. Within the VC tunnel, the class-based weighted fair queuing algorithm is used to schedule packet transmission from each queue. You can configure the queue admission policies, such as EPD or WRED, to control the queue size during congestion. See “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.

Item	ATM1	ATM2 IQ	Comments
VCI management	Supported	Supported	<p>For ATM1 interfaces, you must specify the maximum number of VCIs by including the <code>maximum-vc</code> statement in the configuration. This restricts VCIs to the range 0 through <i>maximum-vc</i>. See “Configuring the Maximum Number of ATM1 VCs on a VP” on page 197.</p> <p>For ATM2 interfaces, you must not include the <code>maximum-vc</code> statement in the configuration. All ATM2 IQ interfaces support VCI numbers from 0 through 65,535. The total number of VCIs that you can open on an ATM2 IQ port depends on two factors:</p> <ul style="list-style-type: none"> Number of tunnels Sparseness of VCI numbers (the more sparse, the fewer VCIs supported) <p>For ATM1 and ATM2 IQ interfaces with promiscuous mode, the allowable maximum number of VCIs is 65,535.</p>
VCI statistics	Supported	Supported	<p>For ATM1 interfaces, multipoint VCI statistics are collected from indirect sources.</p> <p>For ATM2 IQ interfaces, multipoint VCI statistics are collected directly from the PIC.</p> <p>For ATM1 and ATM2 IQ interfaces, point-to-point VCI statistics are the same as logical interface statistics.</p>

Configuring ATM Interfaces

You can configure the following ATM properties:

Configuring Communication with Directly Attached ATM Switches and Routers on page 188

Enabling ILMI for Cell Relay on page 188

Enabling Passive Monitoring on ATM Interfaces on page 190

Removing MPLS Labels from Incoming Packets on page 191

Configuring the ATM PIC Type on page 192

Configuring ATM Cell-Relay Promiscuous Mode on page 193

Configuring the Maximum Number of ATM1 VCs on a VP on page 197

Configuring Layer 2 Circuit Transport Mode on page 198

Configuring Layer 2 Circuit Cell-Relay Promiscuous Mode on page 206

Configuring Layer 2 Circuit Trunk Mode Scheduling on page 207

Configuring CoS Queues in Layer 2 Circuit Trunk Mode on page 209

Configuring the Layer 2 Circuit Cell-Relay Cell Maximum on page 212

Configuring the OAM F4 Cell Flows on page 214

- Defining Virtual Path Tunnels on page 215
- Configuring a Point-to-Point ATM1 or ATM2 IQ Connection on page 216
- Configuring a Point-to-Multipoint ATM1 or ATM2 IQ Connection on page 216
- Configuring a Multicast-Capable ATM1 or ATM2 IQ Connection on page 217
- Configuring Inverse ATM1 or ATM2 ARP on page 218
- Defining the ATM Traffic-Shaping Profile on page 218
- Configuring the ATM1 Queue Length on page 225
- Configuring the ATM2 IQ EPD Threshold on page 226
- Configuring Two EPD Thresholds per Queue on page 228
- Configuring the ATM2 IQ Transmission Weight on page 228
- Defining the ATM OAM F5 Loopback Cell Period on page 229
- Configuring the ATM OAM F5 Loopback Cell Threshold on page 229
- Configuring ATM Interface Encapsulation on page 230
- Configuring an ATM1 Cell-Relay Circuit on page 232
- Configuring PPP over ATM2 Encapsulation on page 234
- Configuring E3 and T3 Parameters on ATM Interfaces on page 238
- Configuring SONET/SDH Parameters on ATM Interfaces on page 239
- Configuring ATM2 IQ VC Tunnel CoS Components on page 240
- Examples: Configuring ATM1 Interfaces on page 252
- Examples: Configuring ATM2 IQ Interfaces on page 254

Configuring Communication with Directly Attached ATM Switches and Routers

For ATM1 and ATM2 IQ interfaces, you can configure communication with directly attached ATM switches and routers to enable querying of the IP addresses and switch port numbers. You query the switch or router by entering the following show command:

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

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

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

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

Example: Configuring Communication with Directly Attached ATM Switches and Routers

Enable an interface to communicate directly with an ATM switch or router:

```
[edit interfaces]
at-0/1/0 {
  atm-options {
    vpi 0;
    ilmi;
  }
  unit 0 {
    vci 0.120;
    family inet {
      address 10.33.33.1/30;
    }
  }
}
```

Enabling ILMI for Cell Relay

The JUNOS software supports standard AAL5 and three Layer 2 circuit transport modes: Layer 2 circuit AAL5, Layer 2 circuit cell-relay, and Layer 2 circuit trunk transport mode.

Integrated local management interface (ILMI) is supported on standard AAL5 interfaces, regardless of encapsulation. To enable ILMI on interfaces with cell-relay encapsulation, you must configure an ATM2 IQ PIC to use Layer 2 circuit trunk transport mode. ILMI is not supported with cell-relay encapsulation when the ATM2 IQ PIC is configured with Layer 2 AAL5 or Layer 2 circuit cell-relay transport mode, as shown in as shown in Table 21 on page 189.

Table 21: ILMI Support by Encapsulation Type

Encapsulation Type	ILMI Support
Standard AAL5, with any encapsulation type	Yes
Layer 2 circuit AAL5 mode	No
Layer 2 circuit cell-relay mode	No
Layer 2 circuit trunk mode	Yes

For more information about Layer 2 circuit transport modes, see “Configuring Layer 2 Circuit Transport Mode” on page 198.

To configure ILMI on an interface with cell-relay encapsulation, include the following statements:

```
[edit chassis fpc slot-number pic pic-number]
atm-l2circuit-mode trunk trunk;

[edit interfaces at-fpc/pic/port]
encapsulation atm-ccc-cell-relay;
atm-options {
    ilmi;
    pic-type atm2;
}
unit logical-unit-number {
    trunk-id number;
}
```

For more information about ILMI, see “Configuring Communication with Directly Attached ATM Switches and Routers” on page 188.

Example: Enabling ILMI for Cell Relay

On an ATM2 IQ PIC with Layer 2 circuit trunk transport mode, enable ILMI on an interface with cell-relay encapsulation:

```
[edit chassis]
fpc 0 {
    pic 1 {
        atm-l2circuit-mode trunk uni;
    }
}
[edit interfaces]
at-0/0/0 {
    encapsulation atm-ccc-cell-relay;
    atm-options {
        pic-type atm2;
        ilmi;
    }
}
```

Enabling Passive Monitoring on ATM Interfaces

The Monitoring Services I and Monitoring Services II PICs are designed to enable IP services. If you have a Monitoring Services PIC and an ATM PIC installed in an M160, M40e, or T-series routing platform, you can monitor IP version 4 (IPv4) traffic from another routing platform.

On ATM interfaces, you enable packet flow monitoring by including the `passive-monitor-mode` statement:

```
passive-monitor-mode;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces at-fpc/pic/port]
```

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit logical-unit-number]
```

If you include the `passive-monitor-mode` statement in the configuration, the ATM interface is always up, and the interface does not receive or transmit incoming control packets, such as OAM cell and ILMI.

On monitoring services interfaces, you enable packet flow monitoring by including the `family inet` statement at the `[edit interfaces mo-fpc/pic/port unit logical-unit-number]` hierarchy level, specifying the `inet` option:

```
[edit interfaces mo-fpc/pic/port unit logical-unit-number]  
family inet;
```

For conformity with `cflowd` record structure, you must include the `receive-options-packets` and `receive-ttl-exceeded` statements at the `[edit interfaces mo-fpc/pic/port unit logical-unit-number family inet]` hierarchy level:

```
[edit interfaces mo-fpc/pic/port unit logical-unit-number family inet]  
receive-options-packets;  
receive-ttl-exceeded;
```

For the monitoring services interface, you can configure multiservice physical interface properties. For more information, see “Configuring Multiservice Physical Interface Properties” on page 93 and the *JUNOS Services Interfaces Configuration Guide*.

Removing MPLS Labels from Incoming Packets

The JUNOS software can forward only IPv4 packets to a Monitoring Services PIC. IPv4 packets with MPLS labels cannot be forwarded to a Monitoring Services PIC. By default, if packets with MPLS labels are forwarded to the Monitoring Services PIC, they are discarded. To monitor packets with MPLS labels, you must remove the MPLS labels as the packets arrive on the interface.

You can remove up to two MPLS labels from an incoming packet by including the `pop-all-labels` statement at the `[edit interfaces interface-name atm-options mpls]` hierarchy level:

```
[edit interfaces interface-name atm-options mpls]
pop-all-labels {
    required-depth number;
}
```

By default, the `pop-all-labels` statement takes effect for incoming packets with one or two labels. You can specify the number of MPLS labels an incoming packet must have for the `pop-all-labels` statement to take effect by including the `required-depth` statement at the `[edit interfaces interface-name atm-options mpls pop-all-labels]` hierarchy level:

```
[edit interfaces interface-name atm-options mpls pop-all-labels]
required-depth number;
```

The required depth can be 1, 2, or `[1 2]`. If you include the `required-depth 1` statement, the `pop-all-labels` statement takes effect for incoming packets with one label only. If you include the `required-depth 2` statement, the `pop-all-labels` statement takes effect for incoming packets with two labels only. If you include the `required-depth [1 2]` statement, the `pop-all-labels` statement takes effect for incoming packets with one or two labels. A required depth of `[1 2]` is equivalent to the default behavior of the `pop-all-labels` statement.

When you remove MPLS labels from incoming packets, note the following:

The `pop-all-labels` statement has no effect on IP packets with three or more MPLS labels.

When you enable MPLS label removal, you must configure all ports on a PIC with the same label popping mode and required depth.

You use the `pop-all-labels` statement to enable passive monitoring applications, not active monitoring.

You cannot apply MPLS filters or accounting to the MPLS labels because the labels are removed as soon as the packet arrives on the interface.

The following ATM encapsulation types are not supported on interfaces with MPLS label removal:

```
atm-ccc-cell-relay
atm-ccc-vc-mux
atm-mlppp-llc
atm-tcc-snap
atm-tcc-vc-mux
ether-over-atm-llc
ether-vpls-over-atm-llc
```

Configuring the ATM PIC Type

For ATM1 and ATM2 IQ interfaces, the JUNOS software does not determine from the interface name *at-fpc/pic/port* whether your routing platform has an ATM1 or ATM2 IQ PIC installed. You can configure the PIC type as ATM1 or ATM2 IQ by including the `pic-type` statement at the [edit interfaces *interface-name* atm-options] hierarchy level:

```
[edit interfaces interface-name atm-options]
pic-type (atm1 | atm2);
```

The following guidelines apply to configuring the ATM PIC type:

If you include the `pic-type` statement in the configuration, and you include other statements at the [edit interfaces *interface-name* atm-options] hierarchy level that do not match the configured PIC type, the configuration does not commit. For example, you cannot commit a configuration that includes the `pic-type atm2` statement and the `maximum-vcs` statement.

If you do not include the `pic-type` statement and you do include the `maximum-vcs` statement in the configuration, the JUNOS software assumes you are configuring an ATM1 interface, and sets the PIC type option accordingly. If you do not include the `maximum-vcs` statement in the configuration, the JUNOS software assumes you are configuring an ATM2 IQ interface, and sets the PIC type option accordingly.

If you include the `promiscuous-mode` statement in the configuration of an ATM2 interface, you must also include the `pic-type atm2` statement.

Example: Configuring the ATM PIC Type

Configure the PIC type on an ATM1 and an ATM2 interface.

```

On an ATM1 Interface      [edit interfaces]
                             at-1/0/0 {
                               atm-options {
                                 pic-type atm1;
                                 vpi 0 maximum-vcs 256;
                                 vpi 1 maximum-vcs 512;
                               }
                               ...
                             }

On an ATM2 IQ           [edit interfaces]
Interface               at-1/1/0 {
                               atm-options {
                                 pic-type atm2;
                                 vpi 0;
                                 vpi 2 {
                                   oam-period 6;
                                 }
                               }
                               ...
                             }

```

Configuring ATM Cell-Relay Promiscuous Mode

For ATM1 and ATM2 IQ interfaces with atm-ccc-cell-relay encapsulation, you can map all incoming cells from either an interface port or a virtual path (VP) to a single LSP without restricting the VCI number. Promiscuous mode allows you to map traffic from all 65,535 VCIs to a single LSP, or from all 256 VPIs to a single LSP.

To map incoming traffic from a port or VC to an LSP, include the promiscuous-mode statement at the [edit interfaces *interface-name* atm-options] hierarchy level:

```

[edit interfaces interface-name]
atm-options {
  promiscuous-mode {
    vpi vpi-identifier;
  }
}

```

You can include multiple vpi statements in the configuration.

To enable all VCIs in a VPI to open in ATM CCC cell-relay mode, you must also map the logical interface to a VPI by including the vpi statement in the logical interface configuration:

```
vpi vpi-identifier;
```

You can include this statement at the following hierarchy levels:

```

[edit interfaces interface-name unit logical-unit-number]

[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]

```

Also, note the following:

For promiscuous mode, you must configure the port with `atm-ccc-cell-relay` encapsulation.

For ATM1 and ATM2 IQ PICs, changing modes between promiscuous and nonpromiscuous causes all physical interfaces to be deleted and re-added.

For ATM1 and ATM2 IQ PICs, when you configure promiscuous mode, you cannot configure VCIs.

For ATM1 PICs, if you configure one port in port mode, all ports on the PIC operate in port mode. Likewise if you configure one logical interface in VPI mode, all logical interfaces on the PIC must operate in VPI mode.

For ATM2 IQ PICs, you can configure one or more logical interfaces in VPI promiscuous mode, and the other logical interfaces with any ATM encapsulation.

For ATM2 IQ PICs, when you configure promiscuous mode, you must also include the `pic-type atm2` statement. For more information, see “Configuring the ATM PIC Type” on page 192.

For ATM2 IQ multiport PICs, you can configure one or more ports in port promiscuous mode, and the other ports with any ATM encapsulation.

For interfaces that are configured for cell-relay promiscuous virtual path identifier (VPI) mode, the `show interfaces` command output does not show (OAM) F4 cell statistics.

Examples: Configuring ATM Cell-Relay Promiscuous Mode

This section includes the following examples:

Configuring Port-Promiscuous Mode on page 194

Configuring VP-Promiscuous Mode on page 195

Mapping Incoming Traffic from a Port to an LSP on page 195

Mapping Unit 0 to an LSP on page 196

Mapping a VPI to an LSP on page 196

Configuring Port-Promiscuous Mode

```
[edit interfaces]
at-0/2/1 {
  encapsulation atm-ccc-cell-relay; # at the physical interface level only
  atm-options {
    pic-type atm2;
    promiscuous-mode;
  }
  unit 0 {
    allow-any-vci;
  }
}
```

**Configuring
VP-Promiscuous Mode**

```
[edit interfaces]
at-0/2/0 {
  atm-options {
    pic-type atm2;
    promiscuous-mode {
      vpi 0;
      vpi 1;
    }
    vpi 2;
    vpi 3;
  }
  unit 0 {
    encapsulation atm-ccc-cell-relay; # at the logical interface level only
    vpi 0;
  }
  unit 1 {
    encapsulation atm-ccc-cell-relay;
    vpi 1;
  }
  unit 2 {
    encapsulation atm-snap;
    vci 2.100;
  }
  unit 3 {
    encapsulation atm-vc-mux;
    vci 3.100;
  }
}
```

**Mapping Incoming
Traffic from a Port to an
LSP**

To map incoming traffic from a port to an LSP, include the `allow-any-vci` statement at the `[edit interfaces interface-name unit 0]` hierarchy level. When you include the `allow-any-vci` statement, you cannot configure other logical interfaces in the same physical interface. Next, you must map unit 0 to an LSP using the CCC connection.

```
[edit interfaces at-1/2/0]
encapsulation atm-ccc-cell-relay;
atm-options {
  promiscuous-mode;
}
unit 0 {
  allow-any-vci;
}
```

Mapping Unit 0 to an LSP

```

protocols {
  connections {
    remote-interface-switch router-a-router-c {
      interface at-1/2/0.0;
    }
    lsp-switch router-a-router-c {
      transmit-lsp lsp1
      receive-lsp lsp2;
    }
  }
}

```

Mapping a VPI to an LSP

To map a VPI to an LSP, you must define the allowed VPIs. You can configure one or more logical interfaces, each mapped to a different VPI. You can then route traffic from each of these interfaces to different LSPs.

```

[edit interfaces at-1/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  promiscuous-mode {
    vpi 10;
    vpi 20;
  }
}
unit 0 {
  encapsulation atm-ccc-cell-relay;
  vpi 10;
}
unit 1 {
  encapsulation atm-ccc-cell-relay;
  vpi 20;
}

```

```

[edit interfaces at-3/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm2;
  promiscuous-mode {
    vpi 10;
    vpi 20;
  }
}
unit 0 {
  encapsulation atm-ccc-cell-relay;
  vpi 10;
}
unit 1 {
  encapsulation atm-ccc-cell-relay;
  vpi 20;
}

```

```

[edit protocols]
mpls {
  connections {
    interface-switch router-a-router-c {
      interface at-1/1/0.0;
      interface at-3/1/0.0;
    }
    interface-switch router-a-router-d {
      interface at-1/1/0.1;
      interface at-3/1/0.1;
    }
  }
}

```

Configuring the Maximum Number of ATM1 VCs on a VP

For ATM1 interfaces, you must configure the maximum number of virtual circuits (VCs) allowed on a virtual path (VP) so that sufficient memory on the ATM1 PIC can be allocated for each VC.

To configure the highest-numbered VCs on a VP, include the `maximum-vcs` and `vpi` statements at the `[edit interfaces interface-name atm-options]` hierarchy level:

```

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

```

The VP identifier can be a value from 0 through 255. For most interfaces, you can define a maximum of 4090 VCs per interface, and some interfaces have higher limits. Promiscuous mode removes these limits. For more information, see “Configuring ATM Cell-Relay Promiscuous Mode” on page 193.

All VPIs that you configure in the `atm-options` statement are stored 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 “Configuring SONET/SDH Parameters on ATM Interfaces” on page 239.

Configuring Layer 2 Circuit Transport Mode

On ATM2 IQ interfaces only, you can configure Layer 2 circuit cell-relay, Layer 2 circuit AAL5, or Layer 2 circuit trunk transport mode.

Layer 2 circuit cell-relay and Layer 2 circuit AAL5 are defined in Internet draft draft-martini-l2circuit-encap-mpls-07.txt, *Encapsulation Methods for Transport of Layer 2 Frames Over IP and MPLS Networks* (expires December 2004).

Layer 2 circuit cell-relay and Layer 2 circuit AAL5 transport modes allow you to send ATM cells between ATM2 IQ interfaces across a Layer 2 circuit-enabled network. Layer 2 circuits are designed to transport Layer 2 frames between PE routing platforms across an LDP-signaled MPLS backbone. You use Layer 2 circuit AAL5 transport mode to send AAL5 segmentation and reassembly protocol data units (SAR-PDUs) over the Layer 2 circuit.

A trunk is a collection of ATM VPs. Layer 2 circuit trunk transport mode allows you to send ATM cells over MPLS trunking.

By default, ATM2 IQ PICs are in standard AAL5 transport mode. Standard AAL5 allows multiple applications to tunnel the protocol data units of their Layer 2 protocols over an ATM virtual circuit. Encapsulation of these Layer 2 protocol data units allows a number of these emulated virtual circuits to be carried in a single tunnel. Protocol data units are segmented at one end of the tunnel and reassembled at the other end. The ingress routing platform reassembles the protocol data units received from the incoming VC and transports each PDU as a single packet.

In contrast, Layer 2 circuit cell-relay and Layer 2 circuit AAL5 transport modes accept a stream of ATM cells, convert these to an encapsulated Layer 2 format, then tunnel them over an MPLS or IP backbone, where a similarly configured routing platform segments these packets back into a stream of ATM cells, to be forwarded to the virtual circuit configured for the far-end routing platform.

In Layer 2 circuit cell-relay transport mode, ATM cells are bundled together and transported in packet form to the far-end routing platform, where they are segmented back into individual ATM cells and forwarded to the ATM virtual circuit configured for the far-end routing platform.

The uses for the four transport modes are defined as follows:

To tunnel IP packets over an ATM backbone, use the default standard AAL5 transport mode.

To tunnel a stream of AAL5-encoded ATM SAR-PDUs over an MPLS or IP backbone, use Layer 2 circuit AAL5 transport mode.

To tunnel a stream of ATM cells over an MPLS or IP backbone, use Layer 2 circuit cell-relay transport mode.

To transport ATM cells over an MPLS core network that is implemented between other vendors' switches or routers, use Layer 2 circuit trunk transport mode.



NOTE: You can transport AAL5-encoded traffic with Layer 2 circuit cell-relay transport mode, because Layer 2 circuit cell-relay transport mode ignores the encoding of the cell data presented to the ingress interface.

When you configure AAL5 mode Layer 2 circuits, the control word carries cell loss priority (CLP) information by default.

To configure Layer 2 circuit AAL5, Layer 2 circuit cell-relay, or Layer 2 circuit trunk mode, you must perform the following tasks:

1. Identify the interface as an ATM2 IQ interface by including the `pic-type atm2` statement at the [edit interfaces *at-fpc/pic/port* atm-options] hierarchy level:

```
[edit interfaces at-fpc/pic/port atm-options]
pic-type atm2;
```

2. Include the `atm-l2circuit-mode` statement at the [edit chassis *fpc slot-number* pic *pic-number*] hierarchy level, specifying `aal5`, `cell`, or `trunk`:

```
[edit chassis fpc slot-number pic pic-number]
atm-l2circuit-mode (aal5 | cell | trunk trunk);
```

By default, the trunk mode uses user-to-network interface (UNI) mode. The trunk option can be UNI or network-to-network interface (NNI). For more information about UNI and NNI, see the *JUNOS VPNs Configuration Guide* and the *JUNOS Feature Guide*.

Transport mode is per PIC, not per port. If you do not include the `atm-l2circuit-mode` statement in the configuration, the ATM2 IQ PIC uses standard AAL5 transport mode. If you configure Layer 2 circuit cell-relay, Layer 2 circuit AAL5 transport mode, or Layer 2 circuit trunk mode, the entire ATM2 PIC uses the configured transport mode.

- For Layer 2 circuit trunk mode only, you must also configure a trunk identification number by including the trunk-id statement:

```
trunk-id number;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

The trunk identification number can be from 0 through 31; each trunk on an interface must have a unique trunk ID. When you associate a trunk ID number with a logical interface, you are in effect specifying the interfaces that are allowed to send ATM traffic over an LSP. For UNI mode, the trunk ID range is from 0 through 7. For NNI mode, the trunk ID range is from 0 through 31. Trunk IDs on connecting trunks do not need to be the same.

For information about proportional bandwidth sharing in trunk mode, see “Configuring Layer 2 Circuit Trunk Mode Scheduling” on page 207.

- For Layer 2 circuit AAL5 mode, configure logical interface encapsulation by including the encapsulation statement, specifying the atm-ccc-vc-mux encapsulation type:

```
encapsulation atm-ccc-vc-mux;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

- For Layer 2 circuit cell-relay and Layer 2 circuit trunk modes, configure physical interface encapsulation by including the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level, specifying the atm-ccc-cell-relay encapsulation type:

```
[edit interfaces interface-name]
encapsulation atm-ccc-cell-relay;
```

For more information about Layer 2 circuits, see the *JUNOS VPNs Configuration Guide* and the *JUNOS Routing Protocols Configuration Guide*. For a comprehensive example, see the *JUNOS Feature Guide*.

Examples: Configuring IQ Layer 2 Circuit Transport Mode

This section includes the following configuration examples:

Configuring Layer 2 Circuit AAL5 Transport Mode on page 201

Configuring Layer 2 Circuit Cell-Relay Transport Mode on page 201

Configuring Layer 2 Circuit Trunk Transport Mode on page 202

Configure Layer 2 circuit AAL5 transport mode and cell-relay transport mode.

**Configuring Layer 2
Circuit AAL5 Transport
Mode**

```
[edit chassis]
fpc 0 {
  pic 1 {
    atm-l2circuit-mode aal5;
  }
}

[edit interfaces]
at-0/1/0 {
  atm-options {
    pic-type atm2;
    vpi 0;
  }
  unit 0 {
    encapsulation atm-ccc-vc-mux;
    point-to-point;
    vci 0.32;
  }
}
```

**Configuring Layer 2
Circuit Cell-Relay
Transport Mode**

```
[edit chassis]
fpc 0 {
  pic 1 {
    atm-l2circuit-mode cell;
  }
}

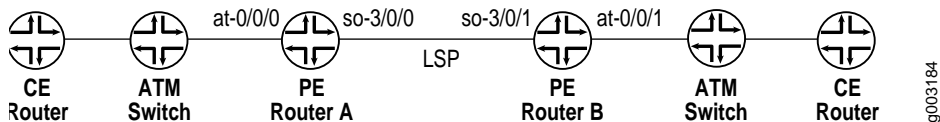
[edit interfaces]
at-0/1/0 {
  encapsulation atm-ccc-cell-relay;
  atm-options {
    pic-type atm2;
    vpi 0;
  }
  unit 0 {
    encapsulation atm-ccc-cell-relay;
    point-to-point;
    vci 0.32;
  }
}
```

**Configuring Layer 2
Circuit Trunk Transport
Mode**

In Figure 11, Router A is a local PE routing platform. Router B is a remote PE router. Both Juniper Networks routing platforms have Layer 2 circuit cell-relay capability. You configure an ATM physical interface on Router A in Layer 2 circuit trunk mode and specify trunks that are allowed to send traffic over the LSP. As a cell is received on this interface, it is classified using the CoS bits in the cell header, and encapsulated as a labeled packet. It is then queued on one of the outgoing queues according to its classification and sent over the LSP to Router B. At Router B, the packet label is removed and the raw cell is put on one of the queues of the ATM interface and forwarded to the second ATM switch. To carry the CoS information and CLP of the cell over the network, the CoS and CLP bits are copied into the EXP bits of the MPLS label. This CoS information is used to select the output queues. Using EPD profiles, the CLP is used to determine whether the cell should be dropped.

For more information about ATM CoS capability, see “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.

Figure 11: Layer 2 Circuit Trunk Topology



On Router A

```
[edit chassis]
fpc 0 {
  pic 1 {
    atm-l2circuit-mode trunk uni;
  }
}
[edit interfaces]
at-0/0/0 {
  encapsulation atm-ccc-cell-relay;
  atm-options {
    pic-type atm2;
    ilmi;
  }
  unit 0 {
    trunk-id 0;
    epd-threshold 10240;
  }
  unit 1 {
    trunk-id 1;
    epd-threshold 10240;
  }
  unit 2 {
    trunk-id 2;
    epd-threshold 10240;
  }
  unit 3 {
    trunk-id 3;
    epd-threshold 10240;
  }
}
```

```

    unit 4 {
        trunk-id 4;
        epd-threshold 10240;
    }
    unit 5 {
        trunk-id 5;
        epd-threshold 10240;
    }
    unit 6 {
        trunk-id 6;
        epd-threshold 10240;
    }
    unit 7 {
        trunk-id 7;
        epd-threshold 10240;
    }
}
so-3/0/0 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.1.1/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.16.0.1/32;
            address 10.255.245.1/32;
        }
    }
}
[edit protocols]
rsvp {
    interface all;
}
mpls {
    interface all;
}
ldp {
    interface all;
}
ospf {
    traffic-engineering;
    reference-bandwidth 4g;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
}

```

```

l2circuit {
  neighbor 10.255.245.2 {
    interface at-0/1/0.0 {
      virtual-circuit-id 100;
    }
    interface at-0/1/0.1 {
      virtual-circuit-id 101;
    }
    interface at-0/1/0.2 {
      virtual-circuit-id 102;
    }
    interface at-0/1/0.3 {
      virtual-circuit-id 103;
    }
    interface at-0/1/0.4 {
      virtual-circuit-id 104;
    }
    interface at-0/1/0.5 {
      virtual-circuit-id 105;
    }
    interface at-0/1/0.6 {
      virtual-circuit-id 106;
    }
    interface at-0/1/0.7 {
      virtual-circuit-id 107;
    }
  }
}

```

On Router B

```

[edit chassis]
fpc 0 {
  pic 1 {
    atm-l2circuit-mode trunk uni;
  }
}
[edit interfaces]
at-0/0/1 {
  encapsulation atm-ccc-cell-relay;
  atm-options {
    pic-type atm2;
  }
  unit 0 {
    trunk-id 0;
    epd-threshold 10240;
  }
  unit 1 {
    trunk-id 1;
    epd-threshold 10240;
  }
  unit 2 {
    trunk-id 2;
    epd-threshold 10240;
  }
  unit 3 {
    trunk-id 3;
    epd-threshold 10240;
  }
}

```

```

    unit 4 {
        trunk-id 4;
        epd-threshold 10240;
    }
    unit 5 {
        trunk-id 5;
        epd-threshold 10240;
    }
    unit 6 {
        trunk-id 6;
        epd-threshold 10240;
    }
    unit 7 {
        trunk-id 7;
        epd-threshold 10240;
    }
}
so-3/0/1 {
    mtu 9192;
    unit 0 {
        family inet {
            address 10.0.1.2/24;
        }
        family mpls;
    }
}
lo0 {
    unit 0 {
        family inet {
            address 172.16.0.1/32;
            address 10.255.245.2/32;
        }
    }
}
[edit protocols]
rsvp {
    interface all;
}
mpls {
    interface all;
}
ldp {
    interface all;
}
ospf {
    traffic-engineering;
    reference-bandwidth 4g;
    area 0.0.0.0 {
        interface all;
        interface fxp0.0 {
            disable;
        }
    }
}
}

```

```

l2circuit {
  neighbor 10.255.245.1 {
    interface at-0/1/0.0 {
      virtual-circuit-id 100;
    }
    interface at-0/1/0.1 {
      virtual-circuit-id 101;
    }
    interface at-0/1/0.2 {
      virtual-circuit-id 102;
    }
    interface at-0/1/0.3 {
      virtual-circuit-id 103;
    }
    interface at-0/1/0.4 {
      virtual-circuit-id 104;
    }
    interface at-0/1/0.5 {
      virtual-circuit-id 105;
    }
    interface at-0/1/0.6 {
      virtual-circuit-id 106;
    }
    interface at-0/1/0.7 {
      virtual-circuit-id 107;
    }
  }
}

```

Configuring Layer 2 Circuit Cell-Relay Promiscuous Mode

By default, all incoming cells are mapped from a single VC to an external LSP. For ATM interfaces with Layer 2 circuit cell-relay transport mode and atm-ccc-cell-relay encapsulation, you can configure promiscuous mode. Promiscuous mode allows you to map all incoming cells from either an interface port or a VP to a single LSP without restricting the VCI number. You can map traffic from all 65,535 VCIs to a single LSP, or from all 256 VPIs to a single LSP. For promiscuous-mode configuration guidelines, see “Configuring ATM Cell-Relay Promiscuous Mode” on page 193.

Example: Configuring Layer 2 Circuit Cell-Relay Promiscuous Mode

Configure Layer 2 circuit cell-relay VP- and port-promiscuous mode:

```

VP-Promiscuous Mode      [edit interfaces]
                           at-0/1/0 {
                             encapsulation atm-ccc-cell-relay;
                             atm-options {
                               pic-type atm2;
                               cell-bundle-size 4;
                               promiscuous-mode {
                                 vpi 0;
                               }
                             }
                           }
                           unit 0 {
                             encapsulation atm-ccc-cell-relay;
                             point-to-point;
                             vci 0.32;
                           }
                           }

```

```

Port-Promiscuous Mode  [edit interfaces]
                           at-0/1/0 {
                             encapsulation atm-ccc-cell-relay;
                             atm-options {
                               pic-type atm2;
                               promiscuous-mode;
                             }
                           }
                           unit 0 {
                             allow-any-vci;
                           }
                           }

```

Configuring Layer 2 Circuit Trunk Mode Scheduling

For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can share a scheduler among 32 trunks on an ATM port. A weighted round robin scheduling algorithm ensures each trunk receives a proportional share of the bandwidth when all trunks are active, and redistributes bandwidth that would have otherwise been reserved by an inactive trunk, thus minimizing the latency on each trunk. For general information about Layer 2 circuit trunk mode, see “Configuring Layer 2 Circuit Transport Mode” on page 198. For general information about ATM CoS scheduling, see “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.

Each trunk is associated with a trunk bandwidth. The trunk bandwidth is the maximum bandwidth used each time a trunk is serviced. We recommend configuring trunk bandwidths so that the ratio between the minimum and maximum bandwidths does not exceed 1:500.

To minimize latency, the JUNOS software does not shape the trunks. As cells are received, they are immediately transmitted.

To configure trunk bandwidth, include the trunk-bandwidth statement:

```
trunk-bandwidth rate;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name
unit logical-unit-number]
```

The trunk bandwidth can be from 1,000,000 through 542,526,792 bps. You can specify the rate in bits per second or cells per second (cps). You can specify a bits-per-second value either as a complete decimal number or as a decimal number followed by the abbreviation k (1000), m (1,000,000), or g (1,000,000,000). You can specify a cells-per-second value by entering a decimal number followed by the abbreviation c; values expressed in cells per second are converted to bits per second by means of the formula 1 cps = 384 bps.

The JUNOS software rounds off the configured value. Therefore, we recommend that you configure a minimum trunk bandwidth of 1m. From 1m, configure values in increments of 500k.

Example: Configuring Layer 2 Circuit Trunk Mode Scheduling

Configure two logical interfaces to use Layer 2 circuit trunk mode, ATM CoS scheduling, and proportional bandwidth sharing:

```
[edit interface]
at-1/1/0 {
  encapsulation atm-ccc-cell-relay;
  atm-options {
    pic-type atm2;
    ilmi;
  }
  scheduler-maps {
    trunk-map {
      vc-cos-mode strict;
      forwarding-class cbr-class {
        priority high;
        transmit-weight percent 40;
        epd-threshold 100;
      }
      forwarding-class rtvbr-class {
        priority low;
        transmit-weight percent 30;
        epd-threshold 100;
      }
      forwarding-class nrtvbr-class {
        priority low;
        transmit-weight percent 20;
        epd-threshold 100;
      }
      forwarding-class ubr-class {
        priority low;
        transmit-weight percent 10;
        epd-threshold 100;
      }
    }
  }
}
```

```

unit 0 {
    encapsulation atm-ccc-cell-relay;
    trunk-id 1;
    trunk-bandwidth 10m;
    atm-scheduler-map trunk-map;
    family ccc {
        filter {
            output atm-trunk-01;
        }
    }
}
unit 1 {
    encapsulation atm-ccc-cell-relay;
    trunk-id 3;
    trunk-bandwidth 30m;
    atm-scheduler-map trunk-map;
}
}

```

Configuring CoS Queues in Layer 2 Circuit Trunk Mode

On ATM2 IQ interfaces, you can configure ATM CoS scheduling for AAL5 mode and Layer 2 circuit trunk mode. For general information about ATM CoS, see “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.

When you configure CoS scheduling in Layer 2 circuit trunk mode, the trunk is defined on the logical interface, and four CoS queues are opened in the trunk. For each CoS queue, you specify a priority and a transmit weight. CoS queues are serviced using a weighted round robin (WRR) algorithm. One queue is serviced with strictly high priority and the remaining queues are serviced with the WRR.

For Layer 2 circuit trunk mode, only strict mode is supported. Alternate mode is not supported.

To configure CoS queues in Layer 2 circuit trunk mode, perform the following tasks:

1. Include the `encapsulation atm-ccc-cell-relay` statement at the [edit interfaces *at-fpc/pic/port*] hierarchy level:

```
[edit interfaces at-fpc/pic/port]
encapsulation atm-ccc-cell-relay;
```

2. Include the `scheduler-maps` statement at the [edit interfaces *at-fpc/pic/port* atm-options] hierarchy level:

```
[edit interfaces at-fpc/pic/port atm-options]
scheduler-maps map-name {
  forwarding-class (class-name | assured-forwarding | best-effort |
    expedited-forwarding | network-control);
  vc-cos-mode strict;
}
```

For information about ATM scheduler maps, see “Configuring an ATM Scheduler Map” on page 242.

3. Include the `atm-scheduler-map`, `trunk-bandwidth`, and `trunk-id` statements at the [edit interfaces *at-fpc/pic/port* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces at-fpc/pic/port unit logical-unit-number]
atm-scheduler-map (map-name | default);
trunk-bandwidth rate;
trunk-id number;
```

For information about trunk identification numbers, see “Configuring Layer 2 Circuit Transport Mode” on page 198. For information about trunk bandwidths, see “Configuring Layer 2 Circuit Trunk Mode Scheduling” on page 207.

Strict mode CoS queue priority works as follows:

Scheduling—One queue has strictly high priority and is always serviced before the remaining queues are serviced by a weighted round robin. This means the packets in a high priority queue are sent first until the queue is empty. Then low priority queues send packets until their weight quota becomes zero or negative.

Latency—Each trunk is associated with a trunk bandwidth. The trunk bandwidth is the maximum bandwidth used each time a trunk is serviced. In the scheduling process, each trunk is serviced in a WRR. The maximum latency for any trunk to begin transmitting is equal to the sum of the weights of all previously queued trunks. Trunks without data do not affect output scheduling. As long as all the trunks have data, the exact weight proportions are maintained. If a trunk runs out of data during its turn, it is no longer included in the WRR. When the trunk gets more data, the trunk is placed at the end of the queue. For more information, see “Configuring Layer 2 Circuit Trunk Mode Scheduling” on page 207.

Within a single trunk, the maximum latency of a high priority queue is the time it takes to transmit one ATM cell. The latency of a low priority queue is the sum of high priority queue burst time and the transmission time of the remaining low priority queues’ weight.

Bandwidth distribution—Trunks are serviced in a WRR based on the trunk bandwidth.

Within a single trunk, the high priority queue consumes the bandwidth first regardless of its weight. The remaining bandwidth is distributed to the low priority queues in proportion to their weights.

Consider the following example:

You configure a trunk with weights 10 percent, 20 percent, 30 percent, and 40 percent for queues 0, 1, 2, and 3, respectively.

You configure queue 0 to be a high priority queue.

Queue 0 does not have cells to transmit.

In this scenario, queues 1, 2 and 3 receive 2/9, 3/9, and 4/9 of the bandwidth, respectively.



NOTE: Constant bit rate (CBR) traffic always enters the strictly high priority queue.

For more information about strict and alternate modes, see “Configuring VC CoS Mode” on page 249.

For general information about Layer 2 circuit trunk mode, see “Configuring Layer 2 Circuit Transport Mode” on page 198.

For interfaces configured in trunk mode, you can also configure dual EPD thresholds depending on packet loss priorities (PLPs). For more information, see “Configuring Two EPD Thresholds per Queue” on page 228.

Example: Configuring CoS Queues in Layer 2 Circuit Trunk Mode

Configure a scheduler map and trunk bandwidth:

```
[edit interfaces]
at-6/1/0 {
  encapsulation atm-ccc-cell-relay;
  atm-options {
    pic-type atm2;
    scheduler-maps {
      cos0 {
        vc-cos-mode strict;
        forwarding-class cbr-class {
          priority high;
          transmit-weight percent 10;
        }
        forwarding-class rtvbr-class {
          priority low;
          transmit-weight percent 20;
        }
        forwarding-class nrtvbr-class {
          priority low;
          transmit-weight percent 30;
        }
        forwarding-class ubr-class {
          priority low;
          transmit-weight percent 40;
        }
      }
    }
  }
  unit 0 {
    trunk-id 0;
    trunk-bandwidth 10m;
    atm-scheduler-map cos0;
  }
}
```

Configuring the Layer 2 Circuit Cell-Relay Cell Maximum

By default, each frame contains one cell. For ATM interfaces with Layer 2 circuit cell-relay transport mode configured, you can configure the maximum number of ATM cells per frame on the physical or logical interface. To set the maximum number of cells per frame, include the `cell-bundle-size` statement:

```
cell-bundle-size cells;
```

You can include this statement at the following hierarchy levels:

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

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

The cell bundle size can be from 1 through 176.

After 125 microseconds, cell bundling times out. This means that after 125 microseconds if the frame does not contain the configured value, the frame is transmitted anyway.

If you include the `cell-bundle-size` statement at the [edit interfaces *interface-name* atm-options] hierarchy level, then the configured value becomes the default for all the logical interface units configured for that physical interface. If you include the `cell-bundle-size` statement for a logical interface, the logical interface configuration overrides the value configured at the physical interface level.

The transmit rates you configure on the routing platforms at each end of the connection must be the same value.

Class-Based Cell Bundling

For Layer 2 circuit trunk mode only, cell bundling is enhanced by a set of CoS and traffic shaping rules, as follows:

CBR and real-time variable bit rate (RTVBR) cells are not bundled. They are always sent as single-cell packets.

Cells with the same CLP bits are bundled together. This means all the cells in a bundle contain the same CLP value.

Cells with the same CoS bits are bundled together. This means all the cells in a bundle belong to the same class of service.

As alluded to in the previous rules, several triggers cause early packet transmission, meaning that the packet is transmitted before the number of cells received is equal to the value configured with the `cell-bundle-size` statement. These triggers are as follows:

The next cell is of type CBR or RTVBR.

The next cell has a different CLP bit.

The next cell has different CoS bits.

The 125-microsecond timer expires.

CoS-based cell bundling optimizes the release of a bundle by sending out the cell that triggers early packet transmission as a single-cell packet. This means that when a cell triggers early packet transmission, that cell is not bundled. Consequently, certain input data patterns might cause primarily single-cell packets to be transmitted. For example, say the output interface receives a steady pattern of two cells from a non-RTVBR queue, followed by two cells from a UBR queue. In this case, all transmitted packets contain a single cell because the first cell triggers a transition and is transmitted by itself. The second cell is also transmitted by itself because the third cell triggers another transition, and so on. This effect might not be dramatic with a mix of traffic; it is most evident with steady traffic patterns, as generated by ATM test equipment programmed to emit regular sequences of CoS queue transitions.

Configuring the OAM F4 Cell Flows

For ATM2 IQ interfaces, the F4 flow cell is used for management of the VP level. If your routing platform is equipped with an ATM2 IQ PIC, you can configure OAM F4 cell flows to identify and report VPC defects and failures. The JUNOS software supports three types of OAM F4 cells in end-to-end F4 flows:

Virtual Path Alarm Indication Signal (VP-AIS)

Virtual Path Remote Defect Indication (VP-RDI)

Virtual Path Loopback

The JUNOS software does not support segment F4 flows, VPC continuity check, or VP performance management functions.

On each VP, you can configure an interval during which to transmit loopback cells by including the `oam-period` statement at the [edit interfaces *interface-name* atm-options vpi *vpi-identifier*] hierarchy level:

```
[edit interfaces interface-name atm-options vpi vpi-identifier]
oam-period (disable | seconds);
```

When you add a VPI at the atm-options hierarchy, an end-to-end F4 VCI is automatically opened to send and receive OAM F4, VP-AIS, and VP-RDI cells. If you enable OAM by including the `oam-period` statement in the configuration, the routing platform sends and receives OAM F4 loopback cells.

To modify OAM liveness values on a VP, include the `oam-liveness` statement at the [edit interfaces *interface-name* atm-options vpi *vpi-identifier*] hierarchy level:

```
[edit interfaces interface-name atm-options vpi vpi-identifier]
oam-liveness {
  up-count cells;
  down-count cells;
}
```

`up-count` is the minimum number of consecutive OAM F4 loopback cells received on a VPI before it is declared up.

`down-count` is the minimum number of consecutive OAM F4 loopback cells lost before a VPI is declared down.

When a VP-AIS or VP-RDI cell is received, the VPI is marked down. When a VP-AIS cell is received on a VPI, a VP-RDI is generated and transmitted on the same VPI. When an OAM F4 loopback request cell is received, the routing platform sends a loopback reply cell, even if the `oam-period` statement is not included in the configuration of the VPI.

When a VPI is marked down because the VPI receives VP-AIS, VP-RDI, VC-AIS, or VC-RDI cells, or because the VPI does not receive down-count consecutive OAM F4 loopback replies, all the VCIs that belong to the VPI are marked down. When a VPI is marked up, all the VCIs that belong to the VPI are marked up. The status of logical interfaces is also changed when the status of the last VCI on that interface is changed.

For a configuration example, see “Examples: Configuring ATM2 IQ Interfaces” on page 254.



NOTE: For interfaces that are configured for cell-relay promiscuous virtual path identifier (VPI) mode, the show interfaces command output does not show (OAM) F4 cell statistics.

Defining Virtual Path Tunnels

For ATM2 IQ interfaces, you can configure shaping on a VPI. When you do this, the VPI is called a VP tunnel. If your routing platform is equipped with an ATM2 IQ PIC, you can configure VP tunnels and a weight for each VC. Each VC is serviced in WRR mode. When VCs have data to send, they send the number of cells equal to their weight before passing control to the next active VC. This allows proportional bandwidth sharing between multiple VCs within a rate-shaped VP tunnel. VP tunnels are not supported on point-to-multipoint interfaces.

If you change or delete VP tunnel traffic shaping, all logical interfaces on a VP are deleted and re-added.

All VPIs you configure on logical interfaces must also be configured on the physical interface, at the [edit interfaces *interface-name* atm-options] hierarchy level.

When you configure a VPI without shaping parameters, the VPI is a regular VPI; no shaping is attached. VCIs that belong to non-shaped VPIs can have VCI shaping.

For point-to-point interfaces, include the shaping statement at the [edit interfaces *interface-name* atm-options vpi *vpi-identifier*] hierarchy level:

```
[edit interfaces interface-name atm-options vpi vpi-identifier]
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length |
   vbr peak rate sustained rate burst length);
  queue-length number;
}
```

For cbr, vbr, and burst statement usage guidelines, see “Defining the ATM Traffic-Shaping Profile” on page 218. For information about ATM2 IQ shaping values, see “Specifying ATM2 IQ Shaping Values” on page 224.

Configuring a Point-to-Point ATM1 or ATM2 IQ Connection

When you use ATM encapsulation on an interface, you must map each logical interface to a VCI. You can optionally map logical interfaces to a VPI.

For ATM1 and ATM2 IQ interfaces, you can configure a VCI and a VPI on a point-to-point ATM interface by including the `vci` statement:

```
vci vpi-identifier.vci-identifier;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

For each VCI, configure the VCI and VPI identifiers. The default VPI identifier is 0. For ATM1 interfaces, the VCI identifier cannot exceed the highest-numbered VC configured for the interface with the `vpi` statement, as described in “Configuring the Maximum Number of ATM1 VCs on a VP” on page 197.

VCIs 0 through 31 are reserved for specific ATM values designated by the ATM Forum.

ATM2 IQ interfaces support only one invalid VC counter for all ports. The invalid VC counter is recorded at port 0 only.

When you are configuring point-to-point connections, the maximum transmission unit (MTU) sizes on both sides of the connections must be the same.

Configuring a Point-to-Multipoint ATM1 or ATM2 IQ Connection

An ATM interface can be a point-to-point interface or a point-to-multipoint (also called a multipoint nonbroadcast multiaccess [NBMA]) connection.

For ATM1 and ATM2 IQ interfaces, you can configure an NBMA ATM connection by including the following statements:

```
multipoint-destination address {
  epd-threshold cells;
  inverse-arp;
  oam-liveness {
    up-count cells;
    down-count cells;
  }
  oam-period (disable | seconds);
  shaping {
    (cbr rate | rtvbr peak rate sustained rate burst length |
     vbr peak rate sustained rate burst length);
    queue-length number;
  }
  vci vpi-identifier.vci-identifier;
}
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number family inet address
address]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number family inet address address]
```

address is the interface's address. The address must include the destination prefix (for example, /24).

For each destination, include one multipoint-destination statement. *address* is the address of the remote side of the connection, and *vci-identifier* and *vpi-identifier* are the VCI and optional VPI identifiers for the connection.

When you configure point-to-multipoint connections, all interfaces in the subnet must use the same MTU size.

Configuring a Multicast-Capable ATM1 or ATM2 IQ Connection

For ATM1 and ATM2 IQ interfaces, you can configure a multicast-capable connection. By default, ATM connections assume unicast traffic. If your ATM switch performs multicast replication, you can configure the connection to support multicast traffic by including the `multicast-vci` statement:

```
multicast-vci vpi-identifier.vci-identifier;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

vci-identifier and *vpi-identifier* are the VCI and VPI identifiers, which define the ATM VCI over which the switch is expecting to receive multicast packets for replication.

You can configure multicast support only on point-to-multipoint ATM connections.

Configuring Inverse ATM1 or ATM2 ARP

For ATM1 and ATM2 IQ interfaces, you can configure inverse ATM Address Resolution Protocol (ARP), as described in RFC 2225, *Classical IP and ARP over ATM*. When inverse ATM ARP is enabled, the routing platform responds to received inverse ATM ARP requests by providing IP address information to the requesting ATM device.

The routing platform does not initiate inverse ATM ARP requests.

By default, inverse ATM ARP is disabled. To configure a VC to respond to inverse ATM ARP requests, include the `inverse-arp` statement:

```
inverse-arp;
```

For a list of hierarchy levels at which you can include this statement, see `inverse-arp` on page 692.

You must configure ATM LLC subnetwork attachment point (SNAP) encapsulation on the logical interface to support inverse ARP. No other ATM encapsulation types are allowed. For more information, see “Configuring ATM Interface Encapsulation” on page 230.

Defining the ATM Traffic-Shaping Profile

When you use an ATM encapsulation on ATM1 and ATM2 IQ interfaces, you can define bandwidth utilization, which consists of either a constant rate or a peak cell rate, with sustained cell rate and burst tolerance.

These values are used in the ATM generic cell-rate algorithm, which is a leaky bucket algorithm that defines the short-term burst rate for ATM cells, the maximum number of cells that can be included in a burst, and the long-term sustained ATM cell traffic rate.

If your routing platform is equipped with an ATM2 IQ PIC, each VC can have independent shaping parameters. For more information, see “Defining Virtual Path Tunnels” on page 215.

By default, the bandwidth utilization is unlimited; that is, unspecified bit rate (UBR) is used. Also, by default, buffer usage by VCs is unregulated.

To define limits to bandwidth utilization, include the `shaping` statement:

```
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length |
   vbr peak rate sustained rate burst length);
  queue-length number;
}
```

For a list of hierarchy levels at which you can include this statement, see `shaping` on page 770.

The `rtvbr` statement is supported on ATM2 IQ PICs only. The `queue-length` statement is supported on ATM1 PICs only.

To configure VP tunnels on ATM2 IQ interfaces, include the shaping statement at the [edit interfaces *interface-name* atm-options vpi *vpi-identifier*] hierarchy level:

```
[edit interfaces interface-name atm-options vpi vpi-identifier]
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length |
   vbr peak rate sustained rate burst length);
}
```

When configuring ATM traffic shaping, you can do the following:

Configuring ATM CBR on page 219

Configuring ATM2 IQ Real-Time VBR on page 220

Configuring ATM VBR on page 220

Specifying ATM1 Shaping Values on page 221

Specifying ATM2 IQ Shaping Values on page 224

Configuring ATM CBR

For traffic that does not require the ability to periodically burst to a higher rate, you can specify a CBR.

To specify a constant bit rate on ATM1 and ATM2 IQ interfaces, include the cbr statement:

```
cbr rate;
```

For a list of hierarchy levels at which you can include this statement, see cbr on page 635.

For ATM1 OC3 interfaces, the rate can be from 33 Kbps through 135.6 Mbps; for ATM1 OC12 interfaces, the rate can be from 33 Kbps through 276 Mbps.

For ATM2 IQ OC3 and OC12 interfaces, the rate can be from 33 Kbps through 542,526,792 bps.

For ATM2 IQ DS3 and E3 interfaces, the rate can be from 33 Kbps to the maximum rate. The maximum rate varies depending on the ATM encapsulation and framing you configure, as shown in Table 22 on page 220.

Table 22: Shaping Rate Range by Interface Type

Interface Type	Maximum Rate
DS3 with direct ATM encapsulation	40,038,968 bps
DS3 with PLCP ATM encapsulation	36,864,000 bps
E3 with g.751 framing and direct ATM encapsulation	30,801,509 bps
E3 with g.751 framing PLCP ATM encapsulation	27,648,000 bps
E3 with g.832 framing	30,720,000 bps

Configuring ATM2 IQ Real-Time VBR

By default, ATM interfaces use UBR; that is, bandwidth utilization is unlimited. For ATM2 IQ interfaces only, you can configure RTVBR, which supports variable bit rate data traffic with average and peak traffic parameters. Compared to non-real-time VBR, RTVBR data is serviced at a higher priority with a relatively small sustainable cell rate (SCR) limit to minimize the delay. Real-time VBR is suitable for carrying packetized video and audio.

To configure RTVBR, include the `rtvbr` statement:

```
rtvbr peak rate sustained rate burst length;
```

For a list of hierarchy levels at which you can include this statement, see `rtvbr` on page 763.

When configuring RTVBR, you can define the following shaping properties:

Peak rate—Top rate at which traffic can burst.

Sustained rate—Normal traffic rate averaged over time.

Burst length—Maximum number of cells that a burst of traffic can contain. It can be a value from 1 through 4000 cells.

The peak and sustained rates can be from 33 Kbps through 542,526,792 bps.

Configuring ATM VBR

By default, ATM interfaces use UBR; that is, bandwidth utilization is unlimited. For ATM1 and ATM2 IQ interfaces, you can configure non-real-time VBR, which supports variable bit rate data traffic with average and peak traffic parameters. Compared to RTVBR, non-real-time VBR is scheduled with a lower priority and with a larger SCR limit, allowing it to recover bandwidth if it falls behind. Non-real-time VBR is suitable for packet data transfers.

To define VBR on ATM1 and ATM2 IQ interfaces, include the `vbr` statement:

```
vbr peak rate sustained rate burst length;
```

For a list of hierarchy levels at which you can include this statement, see `vbr` on page 804.

When configuring VBR, you can define the following shaping properties:

Peak rate—Top rate at which traffic can burst.

Sustained rate—Normal traffic rate averaged over time.

Burst length—Maximum number of cells that a burst of traffic can contain. It can be a value from 1 through 4000 cells.

Specifying ATM1 Shaping Values

For ATM1 interfaces, you can specify the rates in bits per second or cells per second. For OC3c interfaces, the highest rate is 135,631,698 bps (353,207.55 cps), which corresponds to 100 percent of the available line rate. For OC12c interfaces, the highest rate is 271,263,396 bps (706,415.09 cps), which corresponds to 50 percent of the available line rate. Table 23 on page 223 lists some of the other rates you can specify. If you specify a rate that is not listed, it is rounded to the nearest rate.

The exact number of values differs between OC12c and OC3c interfaces. OC12c interfaces have about four times as many value increments as OC3c interfaces.

For OC12c rates between $1/2$ of the line rate and $1/128$ of the line rate, there are 128 steps between each $1/n$ value. This means that there is 128 steps between the $1/2$ and $1/3$ line rate values, and another 128 steps between $1/3$ and $1/4$ and so on. For rates smaller than $1/127$, there are (16,384 minus 127) or 16,257 values. The reason for this is that fractional shaping is ignored at rates below $1/127$. This results in a total of about 32,384 distinct rates for OC12c. When n is larger than or equal to 127, the steps are $1/n$.

For OC3c, the starting point is full line rate, the fraction/integer breakpoint is about $1/31$, and there is a maximum of 4096 scheduler slots for use after $1/31$ of line rate, producing about 8032 total distinct rates. When n is larger than or equal to 31, the steps are $1/n$.

For ATM1 interfaces, the following formula can be used to predict the actual shaping rate:

OC3 shaping settings between 135,631,698 bps (OC3 ATM cell line rate) and 4,375,216 bps (1/31 of OC3 ATM cell line rate).

OC12 shaping settings between 271,263,396 bps (half OC12 ATM cell line rate – the highest rate supported) and 4,271,864 bps (1/127 of OC12 ATM cell line rate).

$$\text{actual-rate} = (128 * \text{line-rate}) / (\text{trunc} ((128 * \text{line-rate}) / \text{desired-rate}))$$

line-rate is the maximum available rate on the interface (in bits per second) after factoring out the overhead for SONET/SDH and ATM (per-cell) overheads. For OC3c interfaces, the line rate is calculated as follows:

$$\text{line-rate} = 155,520,000 \text{ bps} \times (26/27) \times (48/53) = 135,631,698.1 \text{ bps}$$

For OC12c interfaces, the line rate is calculated as follows:

$$\text{line-rate} = 622,080,000 \text{ bps} \times (26/27) \times (48/53) = 542,526,792.45 \text{ bps}$$

desired-rate is the rate you enter in the `vbr` statement, in bits per second.

The `trunc` operator indicates that all digits to the right of the decimal point should be dropped.

For shaping settings smaller than 1/31 of OC3 ATM cell line rate (4,375,216 bps) and 1/127 of OC12 ATM cell line rate (4,271,864 bps), you can predict the actual shaping rate using the following formula:

$$\text{actual-rate} = (1 / (\text{trunc} (\text{line-rate} / \text{desired-rate}) + 1)) * \text{line-rate}$$

For example, for OC12 interfaces, the actual rates for shaping below 4,271,864 bps are calculated as follows:

$$1 / 127 * 542,526,792.45 \text{ bps} = 4,271,864 \text{ bps (11124 cells/second)}$$

$$1 / 128 * 542,526,792.45 \text{ bps} = 4,238,490 \text{ bps (11038 cells/second)}$$

$$1 / 129 * 542,526,792.45 \text{ bps} = 4,205,634 \text{ bps (10952 cells/second)}$$

...

Buffers are shared among all VCs, and by default, there is no limit to the buffer size for a VC. If a VC is particularly slow, it might use all the buffer resources.

Table 23 shows ATM1 traffic-shaping rates.

Table 23: ATM1 Traffic-Shaping Rates

Interface Type	Line Rate (bps)	Line Rate (cps)	Percentage of Total Line Rate
OC3			
	135,600,000	353,125	100.00
	134,542,320	350,370.66	99.22
	133,511,760	347,686.88	98.46
	132,494,760	345,038.44	97.71
	131,491,320	342,425.31	96.97
	130,501,440	339,847.5	96.24
	129,525,120	337,305	95.52
	128,562,360	334,797.81	94.81
	127,626,720	332,361.25	94.12
	126,691,080	329,924.69	93.43
OC12			
	271,263,396	706,415.09	50.00
	270,207,897	703,666.40	49.81
	269,160,579	700,939.01	49.61
	268,121,349	698,232.68	49.42
	267,090,113	695,547.17	49.23
	266,066,779	692,882.24	49.04
	265,051,257	690,237.65	48.85
	264,043,458	687,613.17	48.67
	263,043,293	685,008.58	48.48
	262,050,677	682,423.64	48.30

Example: Specifying ATM1 Shaping Values

Determine the actual rate in ATM1 interfaces when the desired rate is 80 percent of the maximum rate:

OC3c:

$$135,600,000 \text{ bps} * 0.8 = 108,480,000 \text{ bps}$$

Because 108,480,000 bps is greater than 1/31 of OC3 ATM cell line rate:

$$\text{actual-rate} = (128 * 135,600,000.1) / (\text{trunc} ((128 * 135,600,000.1) / 108,480,000))$$

$$\text{actual-rate} = 17,356,800,013 / (\text{trunc} (17,356,800,013 / 108,480,000))$$

$$\text{actual-rate} = 17,356,800,013 / 160$$

$$\text{actual-rate} = 108,480,000 \text{ bps}$$

OC12c:

$$271,263,396 \text{ bps} * 0.8 = 217,010,716.8 \text{ bps}$$

Because 217,010,716.8 bps is greater than 1/127 of OC12 ATM cell line rate:

$$\text{actual-rate} = (128 * 542,526,792.45) / (\text{trunc} ((128 * 542,526,792.45) / 217,010,716.8))$$

$$\text{actual-rate} = 69,443,429,434 / (\text{trunc} (69,443,429,434 / 217,010,716.8))$$

$$\text{actual-rate} = 69,443,429,434 / 320$$

$$\text{actual-rate} = 217,010,717 \text{ bps}$$

Determine the actual rate in ATM1 interfaces when the desired rate is 3,000,000 bps:

OC3c:

Because 3,000,000 bps is smaller than 1/31 of OC3 ATM cell line rate:

$$\text{actual-rate} = (1 / (\text{trunc} (\text{line-rate} / \text{desired-rate}) + 1)) * \text{line-rate}$$

$$\text{actual-rate} = (1 / (\text{trunc} (135,631,698 / 3,000,000) + 1)) * 135,631,698$$

$$\text{actual-rate} = (1 / (45 + 1)) * 135,631,698$$

$$\text{actual-rate} = (1 / 46) * 135,631,698$$

$$\text{actual-rate} = 2,948,515 \text{ bps}$$

OC12c:

Because 3,000,000 bps is smaller than 1/127 of OC12 ATM cell line rate:

$$\text{actual-rate} = (1 / (\text{trunc} (\text{line-rate} / \text{desired-rate}) + 1)) * \text{line-rate}$$

$$\text{actual-rate} = (1 / (\text{trunc} (542,526,792 / 3,000,000) + 1)) * 542,526,792$$

$$\text{actual-rate} = (1 / (180 + 1)) * 542,526,792$$

$$\text{actual-rate} = (1 / 181) * 542,526,792$$

$$\text{actual-rate} = 2,997,386 \text{ bps}$$

Specifying ATM2 IQ Shaping Values

For ATM2 IQ OC3c interfaces, the maximum available rate is 100 percent of line rate, or 135,600,000 bps. For ATM2 IQ OC12c interfaces, the maximum available rate is 100 percent of line rate, or 542,546,792 bps. You can specify the rates in bits per second or cells per second. Fractional shaping is accurate within 0.5 percent of the desired rate.

Configuring the ATM1 Queue Length

ATM1 PICs contain a transmit buffer pool of 16,382 buffers, which are shared by all the PVCs that you configure on the PIC. Even multiple-port ATM PICs have a single buffer pool shared by all the ports.

By default, the ATM1 PIC allows PVCs to consume all the buffers they require. If the sustained traffic rate for a PVC exceeds its shaped rate, buffers are consumed. Eventually, all buffers on the PIC are consumed, and the other PVCs are underserved. This results in head-of-line blocking.

For each PVC, you prevent this situation by configuring the queue length of the PVC. The queue length is a limit on the number of transmit packets that can be queued. Packets that exceed the limit are dropped.

To limit the queue size of a PVC, include the queue-length statement:

```
queue-length number;
```

For a list of hierarchy levels at which you can include this statement, see queue-length on page 754.

The length can be from 1 through 16,383 packets. The default is 16,383 packets. You should include the queue-length statement in the configuration of all the PVCs that you configure on an ATM1 PIC. The queue-length statement performs two functions:

- It prevents head-of-line blocking because it limits the number of packets and therefore buffers that can be consumed by each configured PVC.

- It sets the maximum lifetime that can be sustained by packets over the PVC when traffic has oversubscribed the configured shaping contract.

The total value of all the queue lengths must not exceed the total number of packets that can be held in the buffer space available on the PIC. The total number of packets the buffers can hold depends on the size of the physical interface MTU, including all encapsulation overhead. You can use the following formula to calculate the total number of packets the buffer space can hold:

$$16,382 / (\text{Round Up} (\text{MTU} / 480))$$

For example, assuming default MTU settings for all ATM1 interfaces on a PIC, the total number of packets that can be held is:

$$16,382 / (\text{Round Up} (4482 / 480)) = 1638 \text{ packets}$$

Thus, you can configure up to 1638 for the combined queue length of all the PVCs on an ATM1 PIC that uses default MTU settings for all interfaces.

If you set a queue length to a very low value, small bursts in packets transiting the PVC might not be buffered.

The maximum lifetime that packets can sustain while transiting a PVC depends on the shaping rate you configure for the PVC, the setting for the queue-length statement, and the physical interface MTU. You can use the following formula to calculate the maximum lifetime that packets can sustain while transiting a PVC:

$$(\text{PVC queue-length in packets} \times \text{MTU}) / (\text{PVC shaping in bps} / 8)$$

For example, if you configure a PVC on an ATM1 interface with the default MTU, a CBR shaping rate of 3,840,000 bps (10,000 cps), and a queue length of 25 packets. The maximum lifetime is:

$$(25 \times 4482) / (3,840,000 / 8) = 233 \text{ ms}$$

This is the worst-case lifetime assuming all packets in the queue are MTU sized, and the traffic using the PVC is oversubscribing its configured shaping contract.

In general, we recommend that you use a maximum lifetime under 500 ms.

If you add or change the queue-length setting on the VC, the logical interface associated with the VC is deleted and re-added.

Configuring the ATM2 IQ EPD Threshold

The EPD threshold is a limit on the number of transmit packets that can be queued. Packets that exceed the limit are discarded. This threshold applies to packets that have a PLP of 0. When a beginning of packet (BOP) cell is received, the VC's queue depth is checked against the EPD threshold. If the VC's queue depth exceeds the EPD threshold, the BOP cell and all subsequent cells in the packet are discarded. This prevents a single queue from draining all the buffers on the PIC.

By default, for UBR the EPD threshold is approximately 1 percent of the available cell buffers. If shaping is enabled, the default EPD threshold is proportional to the shaping rate according to the following formula:

$$\text{default epd-threshold} = \text{number of buffers} * \text{shaping rate} / \text{line rate}$$

By default, the software estimates how much buffer space is needed for each PVC. However, you can configure the per-VC buffer space. In general, ATM PVCs need larger buffers for data traffic and smaller buffers for time-sensitive applications. Unnecessarily deep buffers might cause excessive delays on congested PVCs. Overly shallow buffers might cause premature random early detection (RED) or tail packet drops in bursty conditions.

To set the EPD threshold of a PVC, include the epd-threshold statement:

```
epd-threshold cells;
```

For a list of hierarchy levels at which you can include this statement, see epd-threshold on page 663.

The allowable range for EPD threshold varies by interface type, as shown in Table 24.

Table 24: EPD Threshold Range by Interface Type

Interface Type	EPD Range
1-port and 2-port OC12	1 through 425,984 cells
2-port OC3, DS3, and E3	1 through 212,992 cells
4-port DS3 and E3	1 through 106,496 cells

You should include the `epd-threshold` statement in the configuration of all the PVCs that you configure on an ATM2 IQ PIC. The `epd-threshold` statement performs two functions:

- It prevents head-of-line blocking because it limits the number of packets and therefore buffers that can be consumed by each configured PVC.

- It sets the maximum lifetime that can be sustained by packets over the PVC when traffic has oversubscribed the configured shaping contract.

If you add or change the EPD threshold on the VC, the logical interface associated with the VC is deleted and re-added.

On ATM2 IQ DS3 and E3 interfaces, you might be able to enter an EPD threshold or shaping parameter that exceeds the maximum threshold for these interfaces. If the configuration commits, the physical interface might indicate that it is up, but the logical interface fails. As a workaround, configure shaping parameters and EPD thresholds that do not exceed the bandwidth of the interface.

For information about configuring dual EPD thresholds on interfaces configured to use Layer 2 circuit trunk mode, see “Configuring Two EPD Thresholds per Queue” on page 228.

Example: Configuring the ATM2 IQ EPD Threshold

Configure the EPD threshold for a point-to-point ATM2 interface and a point-to-multipoint ATM2 interface.

On a Point-to-Point ATM2 Interface

```
[edit interfaces at-1/0/0]
unit 0 {
  vci 0.123;
  epd-threshold 1300;
  ...
}
```

On a Point-to-Multipoint ATM2 Interface

```
[edit interfaces at-1/0/1]
unit 0 {
  multipoint;
  family inet address 10.0.12.12/24 {
    multipoint-destination 10.0.12.14 vci 0.123 epd-threshold 1300;
    ...
  }
}
```

Configuring Two EPD Thresholds per Queue

For ATM2 IQ interfaces configured to use Layer 2 circuit trunk mode, you can set two EPD thresholds that depend on the PLPs of the packets. When you set a threshold with the `epd-threshold` statement, it applies to packets that have a PLP of 0. When you set a threshold with the `plp1` statement, it applies to packets that have a PLP of 1. If you include the `plp1` statement in the configuration, you must also include the `epd-threshold` statement.

To configure two EPD thresholds, include the `epd-threshold` and `plp1` statements:

```
epd-threshold cells plp1 cells;
```

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name atm-options scheduler-maps map-name
forwarding-class class-name]
```

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

The value you set with the `epd-threshold` statement (for PLP0) should be equal to or greater than the value you set with the `plp1` statement. EPD threshold ranges vary by interface type. See Table 24 on page 227.

For general information about EPD thresholds, see “Configuring the ATM2 IQ EPD Threshold” on page 226.

Configuring the ATM2 IQ Transmission Weight

For ATM2 IQ interfaces configured with VPI shaping, you can control the number of cells a VCI can send each time the VCI has a turn to transmit by including the `transmit-weight` statement:

```
transmit-weight cells;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

VPI traffic shaping is not supported on point-to-multipoint interfaces.

The number of cells can be from 1 through 32,000. For a configuration example, see “Examples: Configuring ATM2 IQ Interfaces” on page 254.

Defining the ATM OAM F5 Loopback Cell Period

For ATM1 and ATM2 IQ interfaces with an ATM encapsulation, you can configure the OAM F5 loopback cell period on virtual circuits. This is the interval at which OAM F5 loopback cells are transmitted.

By default, no OAM F5 loopback cells are sent. To send OAM F5 loopback cells, include the `oam-period` statement:

```
oam-period (disable | seconds);
```

For a list of hierarchy levels at which you can include this statement, see `oam-period` on page 729.

The period can be from 1 through 900 seconds. You can also choose the `disable` option to disable the OAM loopback cell transmit feature.

OAM VC-AIS and VC-RDI defect indication cells are used for identifying and reporting VC defects end-to-end. When a physical link or interface failure occurs, intermediate nodes insert OAM AIS cells into all the downstream VCs affected by the failure. Upon receiving an AIS cell on a VC, the routing platform marks the logical interface down and sends an RDI cell on the same VC to notify the remote end of the error status. When an RDI cell is received on a VC, the routing platform sets the logical interface status to down. When no AIS or RDI cells are received for 3 seconds, the routing platform sets the logical interface status to up. You do not need to configure anything to enable defect indication.

Configuring the ATM OAM F5 Loopback Cell Threshold

For ATM1 and ATM2 IQ interfaces with an ATM encapsulation, you can configure the OAM F5 loopback cell threshold on VCs. This is the minimum number of consecutive OAM F5 loopback cells received before a VC is declared up, or the minimum number of consecutive OAM F5 loopback cells lost before a VC is declared down.

By default, when five consecutive OAM F5 loopback cells are received, the VC is considered to be up, and when five consecutive cells are lost, the VC is considered to be down. To modify these values, include the `oam-liveness` statement:

```
oam-liveness {
    up-count cells;
    down-count cells;
}
```

For a list of hierarchy levels at which you can include this statement, see `oam-liveness` on page 728.

The cell count can be a value from 1 through 255.

Configuring ATM Interface Encapsulation

To configure ATM 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 | ethernet-over-atm);
```

For ATM interfaces, the physical interface encapsulation can be one of the following:

ATM cell-relay—This encapsulation connects two remote virtual circuits or ATM physical interfaces with an LSP. Traffic on the circuit is ATM cells.

ATM PVC—ATM PVC encapsulation is defined in RFC 2684, *Multiprotocol Encapsulation over ATM Adaptation Layer 5*.

Ethernet over ATM—As defined in RFC 1483 (the previous version of RFC 2684), this encapsulation type allows ATM interfaces to connect to devices that support only bridged-mode protocol data units (BPDUs). The JUNOS software does not completely support bridging, but accepts BPDU packets as a default gateway. If you use the router as an edge device, then the router acts as a default gateway. It accepts Ethernet LLC/SNAP frames with IP or ARP in the payload, and drops the rest. For packets destined to the Ethernet LAN, a route lookup is done using the destination IP address. If the route lookup yields a full address match, the packet is encapsulated with an LLC/SNAP and media access control (MAC) header, and the packet is forwarded to the ATM interface.

Generally, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for ATM encapsulations, you can also configure the encapsulation type that is used inside the ATM cell itself. To do this, include the encapsulation statement:

```
encapsulation (atm-ccc-cell-relay | atm-ccc-vc-mux | atm-cisco-nlpid | atm-mlppp-llc |  
atm-nlpid | atm-ppp-llc | atm-ppp-vc-mux | atm-snap | atm-tcc-snap | atm-vc-mux |  
atm-tcc-vc-mux | ether-over-atm-llc | ether-vpls-over-atm-llc);
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit  
logical-unit-number]
```

Table 25 shows the logical interface encapsulation types for ATM interfaces.

Table 25: ATM Logical Interface Encapsulation Types

Encapsulation Types	Comments
ATM CCC cell relay	This encapsulation type connects two remote virtual circuits or ATM physical interfaces with an LSP. This encapsulation type carries traffic in ATM cells. When you use this encapsulation type, you can configure the ccc family only.
ATM CCC VC multiplex	This encapsulation type is for CCC circuits. When you use this encapsulation type, you can configure the ccc family only.
ATM network layer protocol identifier (NLPID)	When you use this encapsulation type, you can configure the inet family only.
ATM SNAP	
ATM SNAP encapsulation on translational cross-connect (TCC) circuits	When you use this encapsulation type, you can configure the tcc family only.
ATM VC multiplex	When you use this encapsulation type, you can configure the inet family only.
ATM VC multiplex on TCC circuits	When you use this encapsulation type, you can configure the tcc family only.
Cell-relay accumulation mode (CAM)	In this mode, the incoming 1 to 8 cells are packaged into a single packet and forwarded to the LSP. To configure CAM, include the atm-cell-relay-accumulation statement at the [edit chassis fpc slot-number pic pic-number] hierarchy level. This encapsulation type is for ATM1 interfaces only. For more information about CAM, see the <i>JUNOS System Basics Configuration Guide</i> .
Cisco ATM NLPID	When you use this encapsulation type, you can configure the inet family only.
Ethernet over ATM	This encapsulation type is for interfaces that carry IPv4 traffic. When you use this encapsulation type, you cannot configure point-to-multipoint interfaces.
Ethernet VPLS over ATM	This encapsulation type enables a VPLS instance to support bridging between Ethernet interfaces and ATM interfaces, as described in RFC 2684. This encapsulation type is for ATM2 IQ interfaces only. When you use this encapsulation type, you cannot configure point-to-multipoint interfaces.
Multilink PPP over AAL5 LLC	This encapsulation type is for ATM2 IQ interfaces only. When you use this encapsulation type, your routing platform must be equipped with a Link Services or Voice Services PIC.
PPP over AAL5 LLC	This encapsulation type is for ATM2 IQ interfaces only. When you use this encapsulation type, you cannot configure point-to-multipoint interfaces.
PPP over AAL5 multiplex	This encapsulation type is for ATM2 IQ interfaces only. When you use this encapsulation type, you cannot configure point-to-multipoint interfaces.

Configuring an ATM1 Cell-Relay Circuit

For ATM1 interfaces, you can create an ATM cell-relay circuit by configuring an entire ATM physical device or an individual VC. When you configure an entire device, only cell-relay encapsulation is allowed on the logical interfaces; for ATM1 PICs, you use the `atm-options` statement to control the number and location of VCs. The configuration of allowed VCs on both ingress and egress ATM interfaces should be the same. For most interfaces, you can define a maximum of 4090 VCs per interface. The highest-numbered VC value you can configure is 4089. Promiscuous mode removes these limits. For more information, see “Configuring ATM Cell-Relay Promiscuous Mode” on page 193.

For ATM1 interfaces, if you are dedicating the entire device to a cell-relay circuit, include the `allow-any-vci` statement in the configuration of unit 0:

```
allow-any-vci;
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit 0]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit 0]
```

Once you include this statement, you cannot configure other logical interfaces in the same physical interface.



NOTE: When you use ATM CCC cell-relay encapsulation, you must configure the logical encapsulation as `atm-ccc-cell-relay`. You cannot mix different logical encapsulation types on an interface that you have configured with ATM CCC cell-relay physical encapsulation.

Examples: Configuring an ATM1 Cell-Relay Circuit

Configure an ATM1 cell-relay circuit:

```
[edit interfaces at-1/2/0]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  vpi 0 maximum-vcs 256;
}
unit 0 {
  point-to-point;
  encapsulation atm-ccc-cell-relay;
  allow-any-vci;
}
```

Configuring an Individual VC on a Logical Interface

```
[edit interfaces at-1/1/0]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  vpi 0 maximum-vcs 256;
}
unit 120 {
  encapsulation atm-ccc-cell-relay;
  vci 0.120;
}
```

Configuring Nonpromiscuous Port Mode

```
[edit interfaces at-0/0/1]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  vpi 0 {
    maximum-vcs 100;
  }
  vpi 1 {
    maximum-vcs 300;
  }
  vpi 4 {
    maximum-vcs 200;
  }
}
unit 0 {
  encapsulation atm-ccc-cell-relay;
  allow-any-vci;
}
}
```

**Configuring
Nonpromiscuous VPI
Mode**

```
[edit interfaces at-0/0/1]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  vpi 0 {
    maximum-vcs 100;
  }
}
unit 0 {
  encapsulation atm-ccc-cell-relay;
  vpi 0;
}
}
```

**Configuring
Nonpromiscuous VCI
Mode**

```
[edit interfaces at-0/0/1]
encapsulation atm-ccc-cell-relay;
atm-options {
  pic-type atm1;
  vpi 0 {
    maximum-vcs 100;
  }
}
unit 0 {
  encapsulation atm-ccc-cell-relay;
  vci 0.50
}
}
```

Configuring PPP over ATM2 Encapsulation

For ATM2 IQ interfaces, you can configure PPP over AAL5 encapsulation, as described in RFC 2364, *PPP over AAL5*. PPP over ATM encapsulation associates a PPP link with an ATM AAL5 PVC.

The JUNOS software supports three PPP over ATM encapsulation types:

atm-ppp-llc—PPP over AAL5 LLC.

atm-ppp-vc-mux—PPP over ATM AAL5 multiplex.

atm-mlppp-llc—Multilink PPP over ATM AAL5 LLC. For this encapsulation type, your routing platform must be equipped with a Link Services or Voice Services PIC.

To enable PPP over ATM encapsulation, include the encapsulation statement, specifying the atm-mlppp-llc, atm-ppp-llc, or atm-ppp-vc-mux encapsulation type:

```
encapsulation (atm-mlppp-llc | atm-ppp-llc | atm-ppp-vc-mux);
```

You can include this statement at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

When you configure PPP over ATM encapsulation, you can enable PPP Challenge Handshake Authentication Protocol (CHAP) and keepalives on the logical interface. For more information about PPP CHAP and keepalives, see “Configuring the PPP Challenge Handshake Authentication Protocol” on page 80 and “Configuring Keepalives” on page 83.



NOTE: When you use PPP over ATM encapsulation, we recommend that you not include the oam-period statement in the configuration. Instead, we recommend that you enable keepalives to detect connection failures.

Example: Configuring PPP over ATM2 IQ Encapsulation

Configure three logical interfaces with PPP over ATM encapsulation:

```
[edit interfaces]
at-0/1/0 {
  atm-options {
    pic-type atm2;
    vpi 0;
    vpi 2;
  }
  unit 0 {
    encapsulation atm-ppp-llc;
    ppp-options {
      chap {
        access-profile pe-B-ppp-clients;
        local-name "pe-A-at-0/1/0";
      }
    }
    keepalives interval 5 up-count 6 down-count 4;
    vci 0.120;
    family inet address 192.168.13.13/30;
  }
  unit 1 {
    encapsulation atm-ppp-vc-mux;
    vci 2.120;
    keepalives interval 6 up-count 6 down-count 4;
    family inet address 192.168.14.13/30;
  }
}
```

**Configuring Multilink
PPP over ATM2 IQ
Encapsulation**

```

    unit 2 {
        encapsulation atm-ppp-vc-mux;
        ppp-options {
            chap {
                passive;
                access-profile pe-A-ppp-clients;
                local-name "pe-A-at-0/1/0";
            }
        }
        keepalives interval 5 up-count 6 down-count 4;
        vci 2.121;
        family inet address 192.168.15.13/30;
    }
}

[edit interfaces]
at-0/0/0 {
    atm-options {
        pic-type atm2;
        vpi 10;
    }
    unit 0 {
        encapsulation atm-mlppp-llc;
        ppp-options {
            chap {
                access-profile pe-B-ppp-clients;
                local-name "pe-A-at-0/0/0";
            }
        }
        keepalive interval 5 up-count 6 down-count 4;
        vci 10.120;
        family mlppp {
            bundle ls-0/3/0.0;
        }
    }
}
at-0/0/1 {
    atm-options {
        pic-type atm2;
        vpi 11;
    }
    unit 1 {
        encapsulation atm-mlppp-llc;
        ppp-options {
            chap {
                access-profile pe-B-ppp-clients;
                local-name "pe-A-at-0/0/0";
            }
        }
        keepalive interval 5 up-count 6 down-count 4;
        vci 11.120;
        family mlppp {
            bundle ls-0/3/0.0;
        }
    }
}
}

```

```

at-1/2/3 {
  atm-options {
    pic-type atm2;
    vpi 12;
  }
  unit 2 {
    encapsulation atm-mlppp-llc;
    ppp-options {
      chap {
        access-profile pe-B-ppp-clients;
        local-name "pe-A-at-0/0/0";
      }
    }
    keepalive interval 5 up-count 6 down-count 4;
    vci 12.120;
    family mlppp {
      bundle ls-0/3/0.0;
    }
  }
}
...
ls-0/3/0 {
  encapsulation multilink-ppp;
  interleave-fragments;
  keepalive;
  unit 0 {
    mrru 4500;
    short-sequence;
    fragment-threshold 16320;
    drop-timeout 2000;
    encapsulation multilink-ppp;
    interleave-fragments;
    minimum-links 8;
    family inet {
      address 10.10.0.1/32 {
        destination 10.10.0.2;
      }
    }
    family iso;
    family inet6 {
      address 8090::0:1/128 {
        destination 8090::0:2;
      }
    }
  }
}
...
}

```

Configuring E3 and T3 Parameters on ATM Interfaces

For ATM1 and ATM2 IQ interfaces, you can configure ATM E3 and T3 interfaces by including the following statements at the [edit interfaces at-*fpc/pic/port*] hierarchy level:

```
[edit interfaces at-fpc/pic/port]
e3-options {
  atm-encapsulation (direct | plcp);
  buildout feet;
  framing (g.751 | g.832);
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
}
t3-options {
  atm-encapsulation (direct | plcp);
  buildout feet;
  (cbit-parity | no-cbit-parity);
  loopback (local | payload | remote);
  (payload-scrambler | no-payload-scrambler);
}
```

The following options and default values differ from those described in the chapters “Configuring E3 Interfaces” on page 341 and “Configuring T3 Interfaces” on page 577:

atm-encapsulation—PLCP is the default value. The E3 line-format option *g.832* supports the direct ATM-encapsulation option only.

buildout—The default value is 10 feet. The number of feet can be any integer value. The range is from 0 through 450 feet (about 137 meters).

cbit-parity—The default option is to enable cbit parity.

framing—There is no default option for E3 interfaces; T3 interfaces use the *cbit-parity* statement in place of the framing statement.

loopback—By default, loopback is disabled.

payload-scrambler—The default option is to enable payload scrambling.

In addition, the ATM E3 and T3 PICs support the clocking statement at the interface level, as do the SONET/SDH PICs. For more information about E3- and T3-specific parameters, see “Configuring E3 Interfaces” on page 341 and “Configuring T3 Interfaces” on page 577.



NOTE: You must configure all the ports on an ATM E3 or T3 PIC with the same framing and encapsulation. Otherwise, the system will set all the ports on the PIC to the slowest framing and encapsulating configuration. For ATM T3, this is PLCP. For ATM E3, this is G.751 PLCP.

Configuring SONET/SDH Parameters on ATM Interfaces

When configuring ATM1 and ATM2 IQ SONET/SDH interfaces, you can also include the following statements in the sonet-options statement to set SONET/SDH parameters on ATM interfaces:

```
[edit interfaces at-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;
  }
  loopback (local | remote);
  (payload-scrambler | no-payload-scrambler);
  rfc-2615;
  trigger {
    defect ignore {
      hold-time up milliseconds down milliseconds;
    }
  }
  (z0-increment | no-z0-increment);
}
```

For information about configuring specific SONET/SDH statements, see “Configuring SONET/SDH Interfaces” on page 531.

Configuring ATM2 IQ VC Tunnel CoS Components

The ATM2 IQ interface allows multiple IP queues into each VC. On M-series platforms (except the M320 router), a VC tunnel can support four CoS queues. On M320 and T-series platforms, a VC tunnel can support up to eight CoS queues. Within a VC tunnel, the WRR algorithm schedules the cell transmission of each queue. You can configure the queue admission policies, such as EPD or WRED, to control the queue size during congestion.

For information about CoS components that apply generally to all interfaces, see “CoS Overview” on page 819 and “CoS Configuration Guidelines” on page 833.

To configure ATM2 IQ VC tunnel CoS components, include the following statements at the [edit interfaces at-*fpc/pic/port*] hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
max-queues-per-interface number;

[edit interfaces at-fpc/pic/port]
atm-options {
  linear-red-profiles profile-name {
    high-plp-max-threshold percent;
    low-plp-max-threshold percent;
    queue-depth cells high-plp-threshold percent low-plp-threshold percent;
  }
  plp-to-clp;
  scheduler-maps map-name {
    forwarding-class class-name {
      epd-threshold cells plp1 cells;
      linear-red-profile profile-name;
      priority (high | low);
      transmit-weight (cells number | percent number);
    }
    vc-cos-mode (alternate | strict);
  }
}
unit 0 {
  atm-scheduler-map (map-name | default);
  family family {
    address address {
      destination address;
    }
  }
  plp-to-clp;
  shaping {
    (cbr rate | rtvbr peak rate sustained rate burst length |
     vbr peak rate sustained rate burst length);
  }
  vci vpi-identifier.vci-identifier;
}
```

This section is organized as follows:

Configuring Linear RED Profiles on page 241

Configuring an ATM Scheduler Map on page 242

Enabling Eight Queues on ATM2 IQ Interfaces on page 244

Configuring VC CoS Mode on page 249

Enabling the PLP Setting to Be Copied to the CLP Bit on page 250

Configuring ATM CoS on the Logical Interface on page 250

Example: Configuring ATM2 IQ VC Tunnel CoS Components on page 251

Configuring Linear RED Profiles

Linear RED profiles define CoS virtual circuit drop profiles. You can configure up to 32 linear RED profiles per port. When a packet arrives, RED checks the queue fill level. If the fill level corresponds to a nonzero drop probability, the RED algorithm determines whether to drop the arriving packet.

To configure linear RED profiles, include the `linear-red-profiles` statement at the [edit interfaces *at-fpc/pic/port* atm-options] hierarchy level:

```
[edit interfaces at-fpc/pic/port atm-options]
linear-red-profiles profile-name {
  high-plp-max-threshold percent;
  low-plp-max-threshold percent;
  queue-depth cells high-plp-threshold percent low-plp-threshold percent;
}
```

The `queue-depth`, `high-plp-threshold`, and `low-plp-threshold` statements are mandatory.

You can define the following options for each RED profile:

`high-plp-max-threshold`—Define the drop profile fill-level for the high PLP CoS VC. When the fill level exceeds the defined percentage, all packets with high PLP are dropped.

`low-plp-max-threshold`—Define the drop profile fill-level for the low PLP CoS VC. When the fill level exceeds the defined percentage, all packets with low PLP are dropped.

`queue-depth`—Define maximum queue depth in the CoS VC drop profile. Packets are always dropped beyond the defined maximum. The range you can configure is from 1 through 64,000 cells.

`high-plp-threshold`—Define CoS VC drop profile fill-level percentage when linear RED is applied to cells with high PLP. When the fill level exceeds the defined percentage, packets with high PLP are randomly dropped by RED.

`low-plp-threshold`—Define CoS VC drop profile fill-level percentage when linear RED is applied to cells with low PLP. When the fill level exceeds the defined percentage, packets with low PLP are randomly dropped by RED.

Configuring an ATM Scheduler Map

To define a scheduler map, you associate it with a forwarding class. Each class is associated with a specific queue, as follows:

```
best-effort—Queue 0
expedited-forwarding—Queue 1
assured-forwarding—Queue 2
network-control—Queue 3
```



NOTE: For M320 and T-series platforms only, you can configure more than four forwarding classes and queues. For more information, see “Enabling Eight Queues on ATM2 IQ Interfaces” on page 244.

When you configure an ATM scheduler map, the JUNOS software creates these CoS queues for a VC. The JUNOS software prefixes each packet delivered to the VC with the next-hop rewrite data associated with each queue.

To configure an ATM scheduler map, include the scheduler-maps statement at the [edit interfaces at-*fpc/pic/port* atm-options] hierarchy level:

```
[edit interfaces at-fpc/pic/port atm-options]
scheduler-maps map-name {
  forwarding-class class-name {
    epd-threshold cells plp1 cells;
    linear-red-profile profile-name;
    priority (high | low);
    transmit-weight (cells number | percent number);
  }
}
```

You can define the following options for each forwarding class:

`epd-threshold` or `linear-red-profile`—An EPD threshold provides a queue of cells that can be stored with tail drop. When a BOP cell is received, the VC's queue depth is checked against the EPD threshold. If the VC's queue depth exceeds the EPD threshold, the BOP cell and all subsequent cells in the packet are discarded.

A linear RED profile defines the number of cells using the `queue-depth` statement within the RED profile. (You configure the `queue-depth` statement at the [edit interfaces *at-fpc/pic/port* atm-options linear-red-profiles *profile-name*] hierarchy level.)

By default, if you include the `scheduler-maps` statement at the [edit interfaces *at-fpc/pic/port* atm-options] hierarchy level, the interface uses an EPD threshold that is determined by the JUNOS software based on the available bandwidth and other parameters. You can override the default EPD threshold by setting an EPD threshold or a linear RED profile.

`priority`—By default, queue 0 is high priority, and the remaining queues are low priority. You can configure high or low queuing priority for each queue.

`transmit-weight`—By default, the transmit weight is 95 percent for queue 0, and 5 percent for queue 3. You can configure the transmission weight in number of cells or percentage. Each CoS queue is serviced in WRR mode. When CoS queues have data to send, they send the number of cells equal to their weight before passing control to the next active CoS queue. This allows proportional bandwidth sharing between multiple CoS queues within a rate-shaped VC tunnel. A CoS queue can send from 1 through 32,000 cells or from 5 through 100 percent of queued traffic before passing control to the next active CoS queue within a VC tunnel.

The AAL5 protocol prohibits cells from being interleaved on a VC; therefore, a complete packet is always sent. If a CoS queue sends more cells than its assigned weight because of the packet boundary, the deficit is carried over to the next time the queue is scheduled to transmit. If the queue is empty after the cells are sent, the deficit is waived, and the queue's assigned weight is reset.



NOTE: If you include the `scheduler-maps` statement at the [edit interfaces *at-fpc/pic/port* atm-options] hierarchy level, the `epd-threshold` statement at the [edit interfaces *interface-name* unit *logical-unit-number*] or [edit interfaces *interface-name* unit *logical-unit-number* address *address* family *family* multipoint-destination *address*] hierarchy level has no effect because either the default EPD threshold, the EPD threshold setting in the forwarding class, or the linear RED profile takes effect instead.

For more information about forwarding classes, see “CoS Configuration Guidelines” on page 833.

Enabling Eight Queues on ATM2 IQ Interfaces

By default, ATM2 IQ PICs on T-series and M320 platforms are restricted to a maximum of four egress queues per interface. You can enable eight egress queues on ATM2 IQ interfaces by including the `max-queues-per-interface` statement at the `[edit chassis fpc slot-number pic pic-number]` hierarchy level:

```
[edit chassis fpc slot-number pic pic-number]
max-queues-per-interface number;
```

The numerical value can be 4 or 8.

If you include the `max-queues-per-interface` statement, all ports on the ATM2 IQ PIC use the configured mode.

When you include the `max-queues-per-interface` statement and commit the configuration, all physical interfaces on the ATM2 IQ PIC are deleted and re-added. Also, the PIC is taken offline and then brought back online immediately. You do not need to manually take the PIC offline and online. You should change modes between four queues and eight queues, or vice versa, only when there is no active traffic going to the ATM2 IQ PIC.

To configure up to eight queues on the ATM2 IQ interface, you must also include the statements described in “Configuring ATM2 IQ VC Tunnel CoS Components” on page 240.

For general information about configuring up to eight forwarding classes and queues on PICs other than ATM2 IQ PICs, see “Configuring up to Eight Forwarding Classes” on page 844.



NOTE: When you are considering enabling eight queues on an ATM2 IQ interface, you should note the following:

ATM2 IQ interfaces using Layer 2 circuit trunk transport mode support only four CoS queues.

ATM2 IQ interfaces with MLPPP encapsulation support only four CoS queues.

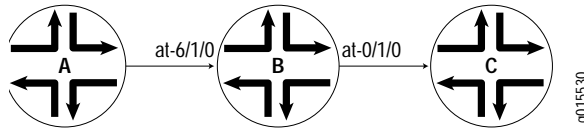
You can configure only four RED profiles for the eight queues. Thus, queue 0 and queue 4 share a single RED profile, as do queue 1 and queue 5, queue 2 and queue 6, and queue 3 and queue 7. There is no restriction on EPD threshold per queue.

The default chassis scheduler allocates resources for queue 0 through queue 3, with 25 percent of the bandwidth allocated to each queue. When you configure the chassis to use more than four queues, you must configure and apply a custom chassis scheduler to override the default. To apply a custom chassis scheduler, include the `scheduler-map-chassis` statement at the `[edit class-of-service interfaces at-fpc/pic/*]` hierarchy level. For more information about configuring and applying a custom chassis scheduler, see “Scheduling Packet Forwarding Component Queues” on page 864.

Example: Enabling Eight Queues on T-series and M320 Platforms

In Figure 12, Router A generates IP packets with different IP precedence settings. Router B is an M320 or T-series platform with two ATM2 IQ interfaces. On Router B, interface at-6/1/0 receives traffic from Router A, while interface at-0/1/0 sends traffic to Router C. This example shows the CoS configuration for Router B.

Figure 12: Example Topology for Router with Eight Queues



On Router B:

```
[edit chassis]
fpc 0 {
  pic 1 {
    max-queues-per-interface 8;
  }
}
fpc 6 {
  pic 1 {
    max-queues-per-interface 8;
  }
}

[edit interfaces]
at-0/1/0 {
  atm-options {
    linear-red-profiles {
      red_1 queue-depth 1k high-plp-threshold 50 low-plp-threshold 80;
      red_2 queue-depth 2k high-plp-threshold 40 low-plp-threshold 70;
      red_3 queue-depth 3k high-plp-threshold 30 low-plp-threshold 60;
      red_4 queue-depth 4k high-plp-threshold 20 low-plp-threshold 50;
    }
  }
  scheduler-maps {
    sch_red {
      vc-cos-mode strict;
      forwarding-class fc_q0 {
        priority high;
        transmit-weight percent 5;
        linear-red-profile red_1;
      }
      forwarding-class fc_q1 {
        priority low;
        transmit-weight percent 10;
        linear-red-profile red_2;
      }
      forwarding-class fc_q2 {
        priority low;
        transmit-weight percent 15;
        linear-red-profile red_3;
      }
    }
  }
}
```

```
forwarding-class fc_q3 {
  priority low;
  transmit-weight percent 20;
  linear-red-profile red_4;
}
forwarding-class fc_q4 {
  priority low;
  transmit-weight percent 5;
  linear-red-profile red_1;
}
forwarding-class fc_q5 {
  priority low;
  transmit-weight percent 10;
  linear-red-profile red_2;
}
forwarding-class fc_q6 {
  priority low;
  transmit-weight percent 15;
  linear-red-profile red_3;
}
forwarding-class fc_q7 {
  priority low;
  transmit-weight percent 20;
  linear-red-profile red_4;
}
}
sch_epd {
  vc-cos-mode alternate;
  forwarding-class fc_q0 {
    priority high;
    transmit-weight percent 5;
    epd-threshold 1024;
  }
  forwarding-class fc_q1 {
    priority low;
    transmit-weight percent 10;
    epd-threshold 2048;
  }
  forwarding-class fc_q2 {
    priority low;
    transmit-weight percent 15;
    epd-threshold 3072;
  }
  forwarding-class fc_q3 {
    priority low;
    transmit-weight percent 20;
    epd-threshold 4096;
  }
  forwarding-class fc_q4 {
    priority low;
    transmit-weight percent 5;
    epd-threshold 2048;
  }
}
```



```

at-6/1/0 {
  atm-options {
    vpi 0;
  }
  unit 0 {
    vci 0.100;
    family inet {
      address 10.10.0.1/24;
    }
  }
  unit 1 {
    vci 0.101;
    family inet {
      address 10.10.1.1/24;
    }
  }
}

[edit class-of-service]
classifiers {
  inet-precedence inet_classifier {
    forwarding-class fc_q0 {
      loss-priority low code-points 000;
    }
    forwarding-class fc_q1 {
      loss-priority low code-points 001;
    }
    forwarding-class fc_q2 {
      loss-priority low code-points 010;
    }
    forwarding-class fc_q3 {
      loss-priority low code-points 011;
    }
    forwarding-class fc_q4 {
      loss-priority low code-points 100;
    }
    forwarding-class fc_q5 {
      loss-priority low code-points 101;
    }
    forwarding-class fc_q6 {
      loss-priority low code-points 110;
    }
    forwarding-class fc_q7 {
      loss-priority low code-points 111;
    }
  }
  forwarding-classes {
    queue 0 fc_q0;
    queue 1 fc_q1;
    queue 2 fc_q2;
    queue 3 fc_q3;
    queue 4 fc_q4;
    queue 5 fc_q5;
    queue 6 fc_q6;
    queue 7 fc_q7;
  }
}

```

```

interfaces {
  at-6/1/0 {
    unit * {
      classifiers {
        inet-precedence inet_classifier;
      }
    }
  }
}
[edit routing-options]
static {
  route 10.10.20.2/32 {
    next-hop at-0/1/0.0;
    retain;
    no-readvertise;
  }
  route 10.10.1.2/32 {
    next-hop at-0/1/0.1;
    retain;
    no-readvertise;
  }
}

```

Check Your Work To see the results of this configuration, you can issue the following operational mode commands:

```

show interfaces at-0/1/0 extensive

show interfaces queue at-0/1/0

show class-of-service forwarding-class

```

Configuring VC CoS Mode

VC CoS mode defines the CoS queue scheduling priority. By default, the VC CoS mode is alternate. When it is a queue's turn to transmit, the queue transmits up to its weight in cells as specified by the transmit-weight statement at the [edit interfaces *at-fpc/pic/port* atm-options scheduler-maps *map-name* forwarding-class *class-name*] hierarchy level. The number of cells transmitted can be slightly over the configured or default transmit weight, because the transmission always ends at a packet boundary.

To configure the VC CoS mode, include the `vc-cos-mode` statement at the [edit interfaces *at-fpc/pic/port* atm-options scheduler-maps] hierarchy level:

```

[edit interfaces at-fpc/pic/port atm-options scheduler-maps]
vc-cos-mode (alternate | strict);

```

Two modes of CoS scheduling priority are supported:

alternate—Assign high priority to one queue. The scheduling of the queues alternates between the high priority queue and the remaining queues. Every other scheduled packet is from the high priority queue.

strict—Assign strictly high priority to one queue. A queue with strictly high priority is always scheduled before the remaining queues. The remaining queues are scheduled in round-robin fashion.

Enabling the PLP Setting to Be Copied to the CLP Bit

For a PE router with customer edge (CE)-facing, egress, ATM2 IQ interfaces configured with standard AAL5 encapsulation, you can enable the PLP setting to be copied into the CLP bit.



NOTE: This configuration setting is not applicable to Layer 2 circuit encapsulations because the control word captures and preserves CLP information. For more information about Layer 2 circuit encapsulations, see “Configuring Layer 2 Circuit Transport Mode” on page 198.

By default, at egress ATM2 IQ interfaces configured with standard AAL5 encapsulation, the PLP information is not copied to the CLP bit. This means the PLP information is not carried beyond the egress interface onto the CE router.

You can enable the PLP information to be copied into the CLP bit by including the `plp-to-clp` statement:

```
plp-to-clp;
```

You can include this statement at the following hierarchy levels:

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

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

Configuring ATM CoS on the Logical Interface

To apply the ATM scheduler map to a logical interface, include the `atm-scheduler-map` statement:

```
atm-scheduler-map (map-name | default);
```

For ATM CoS to take effect, you must configure the VCI and VPI identifiers and traffic shaping on each VC by including the following statements:

```
vci vpi-identifier.vci-identifier;
shaping {
  (cbr rate | rtvbr peak rate sustained rate burst length |
   vbr peak rate sustained rate burst length);
}
```

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name unit logical-unit-number]
```

```
[edit logical-routers logical-router-name interfaces interface-name unit
logical-unit-number]
```

For more information, see “Configuring a Point-to-Point ATM1 or ATM2 IQ Connection” on page 216 and “Defining the ATM Traffic-Shaping Profile” on page 218.

You can also apply a scheduler map to the chassis traffic that feeds the ATM interfaces. For more information, see “Scheduling Packet Forwarding Component Queues” on page 864.

Example: Configuring ATM2 IQ VC Tunnel CoS Components

Configure ATM2 IQ VC tunnel CoS components:

```
[edit interfaces]
at-1/2/0 {
  atm-options {
    vpi 0;
    linear-red-profiles red-profile-1 {
      queue-depth 35000 high-plp-threshold 75 low-plp-threshold 25;
    }
    scheduler-maps map-1 {
      vc-cos-mode strict;
      forwarding-class best-effort {
        priority low;
        transmit-weight percent 25;
        linear-red-profile red-profile-1;
      }
    }
  }
  unit 0 {
    vci 0.128;
    shaping {
      vbr peak 20m sustained 10m burst 20;
    }
    atm-scheduler-map map-1;
    family inet {
      address 192.168.0.100/32 {
        destination 192.168.0.101;
      }
    }
  }
}
```

Examples: Configuring ATM1 Interfaces

The following configuration is sufficient to get an ATM1 OC3 or OC12 interface up and running. By default, ATM interfaces use ATM PVC encapsulation.

```
[edit interfaces]
at-fpc/pic/port {
  atm-options {
    vpi vpi-identifier maximum-vcs maximum-vcs value;
  }
  unit 0 {
    # one unit per VC
    vci vpi-identifier.vci-identifier;
    family inet {
      address local-address {
        destination address;
      }
    }
  }
  unit 1 {
    # second VC
    ...
  }
}
```

Complex Configuration Example

```
[edit interfaces]
at-0/0/0 {
  encapsulation atm-pvc;
  atm-options {
    vpi 0 maximum-vcs 1200;
  }
  unit 2 {
    encapsulation atm-snap;
    inverse-arp;
    vci 0.80;
    family inet {
      mtu 1500;
      address 192.168.0.3/32 {
        destination 192.168.0.1;
      }
    }
  }
  unit 3 {
    encapsulation atm-snap;
    vci 0.32;
    oam-period 60;
    family inet {
      mtu 1500;
      address 192.168.4.3/32 {
        destination 192.168.4.2;
      }
    }
  }
}
```

```

at-0/2/0 {
  encapsulation atm-pvc;
  atm-options {
    vpi 0 maximum-vcs 1200;
  }
  unit 2 {
    encapsulation atm-snap;
    inverse-arp;
    vci 0.82;
    family inet {
      mtu 1500;
      address 192.168.5.3/32 {
        destination 192.168.5.2;
      }
    }
  }
}
at-0/3/0 {
  encapsulation atm-pvc;
  atm-options {
    vpi 0 maximum-vcs 1200;
  }
  unit 140 {
    encapsulation atm-snap;
    multipoint;
    family inet {
      address 192.168.7.4/24 {
        multipoint-destination 192.168.7.5;
        vci 0.100;
        inverse-arp;
      }
    }
  }
}
at-7/3/0 {
  encapsulation atm-pvc;
  atm-options {
    vpi 0 maximum-vcs 1200;
  }
  unit 0 {
    encapsulation atm-snap;
    vci 0.32;
    family inet {
      address 192.168.12.3/32 {
        destination 192.168.12.2;
      }
    }
  }
}

```

Examples: Configuring ATM2 IQ Interfaces

Configure VP tunnel-shaping and OAM F4 on an ATM2 IQ interface:

```

interfaces {
  at-5/2/0 {
    atm-options {
      vpi 0 {
        shaping {
          vbr peak 10m sustained 6m burst 12;
        }
        oam-period 10;
        oam-liveness {
          up-count 6;
          down-count 5;
        }
      }
    }
    vpi 4 {
      shaping {
        vbr peak 7m sustained 4m burst 24;
      }
    }
    vpi 5 {
      oam-period 10;
      oam-liveness {
        up-count 6;
        down-count 5;
      }
    }
    vpi 6;
  }
  unit 0 {
    vci 0.128;
    transmit-weight 20;
    family inet {
      address 192.168.9.225/32 {
        destination 192.168.9.224;
      }
    }
  }
  unit 1 {
    vci 0.129;
    transmit-weight 30;
    family inet {
      address 192.168.9.226/32 {
        destination 192.168.9.227;
      }
    }
  }
}

```

```
unit 2 {  
  vci 5.123;  
  shaping {  
    vbr peak 60m sustained 4m burst 24;  
  }  
  family inet {  
    address 192.168.9.227/32 {  
      destination 192.168.9.230;  
    }  
  }  
}
```

