

Chapter 22

Configuring Ethernet Interfaces

Ethernet was developed in the early 1970s at the Xerox Palo Alto Research Center (PARC) as a data-link control layer protocol for interconnecting computers. It was first widely used at 10 Mbps over coaxial cables and later over unshielded twisted pairs using 10Base-T. More recently, 100Base-TX (Fast Ethernet, 100 Mbps), Gigabit Ethernet (1 Gbps), and 10-Gigabit Ethernet (10 Gbps) have become available.

Juniper Networks routing platforms support the following types of Ethernet interfaces:

- Fast Ethernet

- Gigabit Ethernet

- Gigabit Ethernet intelligent queuing (IQ)

- 10-Gigabit Ethernet

- Management Ethernet interface, which is an out-of-band management interface within the routing platform

- Internal Ethernet interface, which connects the Routing Engine to the packet forwarding components

- Aggregated Ethernet interface, a logical linkage of Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet physical connections

This chapter discusses the following topics specific to configuring the different types of Ethernet interfaces in the routing platform:

Configuring Ethernet Physical Interface Properties on page 344

Configuring 802.1Q VLANs on page 355

Configuring TCC and Layer 2.5 Switching on page 359

Configuring Static ARP Table Entries on page 362

Configuring VRRP on page 363

Configuring Gigabit Ethernet Accounting and Policing on page 374

Configuring the Management Ethernet Interface on page 390

Displaying the Internal Ethernet Interface on page 391

Configuring Aggregated Ethernet Interfaces on page 392

For examples of Ethernet interface configuration, see the following sections:

Example: Configuring Fast Ethernet Interfaces on page 394

Example: Configuring Gigabit Ethernet Interfaces on page 395

Example: Configuring Aggregated Ethernet Interfaces on page 395

Configuring Ethernet Physical Interface Properties

To configure Fast Ethernet-specific physical interface properties, include the `fastether-options` statement at the `[edit interfaces fe-fpc/pic/port]` hierarchy level:

```
[edit interfaces fe-fpc/pic/port]
link-mode (full-duplex | half-duplex);
speed (10m | 100m);
vlan-tagging;
fastether-options {
    802.3ad aex;
    (flow-control | no-flow-control);
    ingress-rate-limit rate;
    (loopback | no-loopback);
    source-address-filter {
        mac-address;
    }
    (source-filtering | no-source-filtering);
}
```



NOTE: The statement `speed (10m | 100m)` applies only to the management Ethernet interface (fxp0) and to the Fast Ethernet 12-port and 48-port Physical Interface Cards (PICs). The 4-port and 8-port Fast Ethernet PICs support a speed of 100 Mbps only.

To configure Gigabit Ethernet- and 10-Gigabit Ethernet-specific physical interface properties, include the `gether-options` statement at the `[edit interfaces ge-fpc/pic/port]` hierarchy level:

```
[edit interfaces ge-fpc/pic/port]
gether-options {
  802.3ad aex;
  (flow-control | no-flow-control);
  (loopback | no-loopback);
  source-address-filter {
    mac-address;
  }
  (source-filtering | no-source-filtering);
}
```

To configure Gigabit Ethernet IQ-specific physical interface properties, include the `gether-options` statement at the `[edit interfaces ge-fpc/pic/port]` hierarchy level. Some of these statements are also supported on Gigabit Ethernet PICs with small form-factor pluggable transceivers (SFPs) (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform). For more information, see “Example: Configuring Gigabit Ethernet Interfaces” on page 395.

```
[edit interfaces ge-fpc/pic/port]
gether-options {
  802.3ad aex;
  (flow-control | no-flow-control);
  (loopback | no-loopback);
  (source-filtering | no-source-filtering);
  ethernet-switch-profile {
    (mac-learn-enable | no-mac-learn-enable);
    tag-protocol-id [ tpids ];
    ethernet-policer-profile {
      input-priority-map {
        ieee802.1p premium [ values ];
      }
      output-priority-map {
        classifier {
          premium {
            forwarding-class class-name {
              loss-priority (high | low);
            }
          }
        }
      }
    }
  }
  policer cos-policer-name {
    aggregate {
      bandwidth-limit bps;
      burst-size-limit bytes;
    }
    premium {
      bandwidth-limit bps;
      burst-size-limit bytes;
    }
  }
}
```

To configure Gigabit Ethernet IQ-specific logical interface properties, include the `input-vlan-map` and `output-vlan-map` statements:

```

input-vlan-map {
  pop;
  push;
  swap;
  vlan-id number;
  tag-protocol-id tpid;
}
output-vlan-map {
  pop;
  push;
  swap;
  vlan-id number;
  tag-protocol-id tpid;
}
vlan-tags inner tpid.vlan-id outer tpid.vlan-id;

```

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]
```

To configure aggregated Ethernet-specific physical interface properties, include the `aggregated-ether-options` statement at the `[edit interfaces aex]` hierarchy level:

```

[edit interfaces aex]
aggregated-ether-options {
  (flow-control | no-flow-control);
  lACP mode {
    periodic interval;
  }
  link-speed speed;
  (loopback | no-loopback);
  minimum-links number;
  source-address-filter {
    mac-address;
  }
  (source-filtering | no-source-filtering);
}

```

You can configure the following properties specific to aggregated Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces:

- Configuring Ethernet Link Aggregation on page 347
- Configuring Aggregated Ethernet LACP on page 348
- Configuring Aggregated Ethernet Link Speed on page 350
- Configuring Aggregated Ethernet Minimum Links on page 351
- Enabling Ethernet MAC Address Filtering on page 351
- Configuring Ethernet Loopback Capability on page 352
- Configuring Flow Control on page 353
- Configuring the Link Characteristics on page 353
- Configuring Gratuitous ARP on page 354
- Configuring the Interface Speed on page 354
- Configuring the Ingress Rate Limit on page 355

Configuring Ethernet Link Aggregation

On Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces, you can associate a physical interface with an aggregated Ethernet interface. To enable the aggregated link, include the 802.3ad statement at the [edit interfaces *interface-name* fastether-options] or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
802.3ad aex;
```

You specify the interface instance number *x* to complete the link association; *x* can be from 0 through 127, for a total of 128 aggregated interfaces. You must also include a statement defining *aex* at the [edit interfaces] hierarchy level. For more information, see “Configuring Aggregated Ethernet Interfaces” on page 392. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configuring Ethernet Physical Interface Properties” on page 344, and for a sample configuration, see “Example: Configuring Aggregated Ethernet Interfaces” on page 395.

Configuring Aggregated Ethernet LACP

For aggregated Ethernet interfaces, you can configure the Link Aggregation Control Protocol (LACP). LACP allows you to bundle several physical interfaces to form one logical interface. LACP exchanges are made between actors and partners. An actor is the local interface in an LACP exchange. A partner is the remote interface in an LACP exchange.

LACP is defined in IEEE 802.3ad, *Aggregation of Multiple Link Segments* .

LACP was designed to achieve the following:

- Automatic addition and deletion of individual links to the aggregate bundle without user intervention

- Link monitoring to check whether both ends of the bundle are connected to the correct group

The JUNOS implementation of LACP provides link monitoring but not automatic addition and deletion of links.

The LACP mode can be active or passive. If the actor and partner are both in passive mode, they do not exchange LACP packets, which results in the aggregated Ethernet links not coming up. If either the actor or partner is active, they do exchange LACP packets. By default, LACP is in passive mode on aggregated Ethernet interfaces. To initiate transmission of LACP packets and response to LACP packets, you must enable LACP active mode.

To enable LACP active mode, include the `lacp` statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level, and specify the active option:

```
[edit interfaces interface-name aggregated-ether-options]
lacp active;
```

To restore the default behavior, include the `lacp` statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level, and specify the passive option:

```
[edit interfaces interface-name aggregated-ether-options]
lacp passive;
```

Configuring the LACP Interval

By default, the actor and partner send LACP packets every second. You can configure the interval at which the interfaces send LACP packets by including the periodic statement at the [edit interfaces *interface-name* aggregated-ether-options lacp] hierarchy level:

```
[edit interfaces interface-name aggregated-ether-options lacp]
  periodic interval;
```

The interval can be fast (every second) or slow (every 30 seconds). You can configure different periodic rates on active and passive interfaces. When you configure the active and passive interfaces at different rates, the transmitter honors the receiver's rate.



NOTE: Source address filtering does not work when LACP is enabled. For more information about source address filtering, see “Enabling Ethernet MAC Address Filtering” on page 351.

Percentage policers are not supported on aggregated Ethernet interfaces with the protocol family circuit cross-connect (CCC) configured. For more information about percentage policers, see the *JUNOS Policy Framework Configuration Guide*.

Generally, LACP is supported on all untagged aggregated Ethernet interfaces.

For M-series routers with enhanced Flexible PIC Concentrators (FPCs) and T-series routing platforms, LACP over virtual local area networks (VLAN)-tagged aggregated Ethernet interfaces is supported. For 8-port, 12-port, and 48-port Fast Ethernet PICs, LACP over VLAN-tagged interfaces is not supported.

Example: Configuring Aggregated Ethernet LACP

Configure aggregated Ethernet LACP over a VLAN-tagged interface:

```
[edit interfaces]
fe-5/0/1 {
  fastether-options {
    802.3ad ae0;
  }
}
ae0 {
  aggregated-ether-options {
    lacp {
      active;
    }
  }
  vlan-tagging;
  unit 0 {
    vlan-id 100;
    family inet {
      address 10.1.1.2/24 {
        vrrp-group 0 {
          virtual-address 10.1.1.4;
          priority 200;
        }
      }
    }
  }
}
```

Configuring Aggregated Ethernet Link Speed

On aggregated Ethernet interfaces, you can set the required link speed for all interfaces included in the bundle. All interfaces that make up a bundle must be the same speed. If you include in the aggregated Ethernet interface an individual link that has a speed different from the speed you specify in the link-speed parameter, an error message will be logged. To set the required link speed, include the link-speed statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level:

```
[edit interfaces interface-name aggregated-ether-options]
link-speed speed;
```

speed can be in bits per second 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).

Configuring Aggregated Ethernet Minimum Links

On aggregated Ethernet interfaces, you can configure the minimum number of links that must be up for the bundle as a whole to be labeled up. By default, only one link must be up for the bundle to be labeled up.

To configure the minimum number of links, include the `minimum-links` statement at the [edit interfaces *interface-name* aggregated-ether-options] hierarchy level:

```
[edit interfaces interface-name aggregated-ether-options]
  minimum-links number;
```

The number can be from one through eight. (The maximum number of links supported in an aggregate is eight.)

Enabling Ethernet MAC Address Filtering

By default, source address filtering is disabled. On aggregated Ethernet interfaces, Fast Ethernet, Gigabit Ethernet, Gigabit Ethernet IQ, and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can enable source address filtering, which blocks all incoming packets to an interface.

To enable the filtering, include the `source-filtering` statement:

```
source-filtering;
```

To explicitly disable filtering, include the `no-source-filtering` statement:

```
no-source-filtering;
```

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name aggregated-ether-options]
```

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

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



NOTE: When you integrate a standalone T640 routing node into a routing matrix, the PIC MAC addresses for the integrated T640 routing node are derived from a pool of MAC addresses maintained by the TX Matrix platform. For each MAC address you specify in the configuration of a formerly standalone T640 routing node, you must specify the same MAC address in the configuration of the TX Matrix platform.

Filter Specific MAC Addresses

When source address filtering is enabled, you can configure the interface to receive packets from specific media access control (MAC) addresses. To do this, specify the MAC addresses in the source-address-filter statement:

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

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name aggregated-ether-options]
```

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

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

You can specify the MAC address as *nn:nn:nn:nn:nn:nn* or *nnnn.nnnn.nnnn*, where *n* is a hexadecimal number. You can configure up to 64 source addresses. To specify more than one address, include the source-address-filter statement multiple times.



NOTE: The source-address-filter statement is not supported on Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform); instead, include the accept-source-mac statement. For more information, see “Configuring MAC Address Filtering” on page 379.

If the remote Ethernet card is changed, the interface cannot receive packets from the new card because it has a different MAC address.

Source address filtering does not work when LACP is enabled. For more information about LACP, see “Configuring Aggregated Ethernet LACP” on page 348.

Configuring Ethernet Loopback Capability

By default, local aggregated Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces connect to a remote system. To place an interface in loopback mode, include the loopback statement:

```
loopback;
```

To return to the default—that is, to disable loopback mode—delete the loopback statement from the configuration:

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

To explicitly disable loopback mode, include the no-loopback statement:

```
no-loopback;
```

You can include the loopback and no-loopback statements at the following hierarchy levels:

```
[edit interfaces interface-name aggregated-ether-options]
```

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

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

Configuring Flow Control

By default, the routing platform imposes flow control to regulate the amount of traffic sent out a Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interface. This is useful if the remote side of the connection is a Fast Ethernet or Gigabit Ethernet switch.

You can disable flow control if you want the routing platform to permit unrestricted traffic. To disable flow control, include the `no-flow-control` statement:

```
no-flow-control;
```

To explicitly reinstate flow control, include the `flow-control` statement:

```
flow-control;
```

You can include these statements at the following hierarchy levels:

```
[edit interfaces interface-name aggregated-ether-options]
```

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

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

Configuring the Link Characteristics

Full-duplex communication means that both ends of the communication can send and receive signals at the same time. *Half-duplex* is also bidirectional communication, but signals can flow in only one direction at a time.

By default, the routing platform's management Ethernet interface, `fxp0`, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and all other interfaces can operate only in full-duplex mode. For Gigabit Ethernet and 10-Gigabit Ethernet, the link partner must also be set to full duplex.

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

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

Configuring Gratuitous ARP

Gratuitous Address Resolution Protocol (ARP) requests provide duplicate IP address detection. A gratuitous ARP request is a broadcast request for a routing platform's own IP address. If a routing platform sends an ARP request for its own IP address and no ARP replies are received, the routing platform's assigned IP address is not being used by other nodes. If a routing platform sends an ARP request for its own IP address and an ARP reply is received, the routing platform's assigned IP address is already being used by another node.

By default, the routing platform responds to gratuitous ARP requests. On Ethernet interfaces, you can disable responses to gratuitous ARP requests. To disable responses to gratuitous ARP requests, include the `no-gratuitous-arp-request` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
no-gratuitous-arp-request;
```

To return to the default—that is, to respond to gratuitous ARP requests—delete the `no-gratuitous-arp-request` statement from the configuration:

```
[edit]  
user@host# delete interfaces interface-name no-gratuitous-arp-request
```

Gratuitous ARP replies are reply packets sent to the broadcast MAC address with the target IP address set to be the same as the sender's IP address. When the routing platform receives a gratuitous ARP reply, the routing platform can insert an entry for that reply in the ARP cache.

By default, updating the ARP cache on gratuitous ARP replies is disabled on the routing platform. On Ethernet interfaces, you can enable handling of gratuitous ARP replies on a specific interface by including the `gratuitous-arp-reply` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
gratuitous-arp-reply;
```

To restore the default behavior, include the `no-gratuitous-arp-reply` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
no-gratuitous-arp-reply;
```

Configuring the Interface Speed

On Fast Ethernet 12-port and 48-port PIC interfaces and the management Ethernet interface (fxp0) only, you can explicitly set the interface speed to either 10 Mbps or 100 Mbps.

To explicitly configure the speed on an interface, include the `speed` statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
speed (10m | 100m);
```

Configuring the Ingress Rate Limit

On Fast Ethernet 8-port, 12-port, and 48-port PIC interfaces only, you can apply port-based rate limiting to the ingress traffic that arrives at the PIC.

To configure an ingress rate limit on a Fast Ethernet 8-port, 12-port, or 48-port PIC interface, include the `ingress-rate-limit` statement at the `[edit interfaces interface-name fastether-options]` hierarchy level:

```
[edit interfaces interface-name fastether-options]
ingress-rate-limit rate;
```

rate can range in value from 1 through 100 Mbps.

Configuring 802.1Q VLANs

For Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces, the JUNOS software supports a subset of the IEEE 802.1Q standard for channelizing an Ethernet interface into multiple logical interfaces, allowing many hosts to be connected to the same Gigabit Ethernet switch, but preventing them from being in the same routing or broadcast domain.

You can configure the following 802.1Q VLAN properties:

Enabling VLAN Tagging on page 355

Binding a VLAN ID to a Logical Interface on page 356

Configuring VLAN Encapsulation on page 357

Configuring Extended VLAN Encapsulation on page 358

For examples of 802.1Q VLAN configuration, see the following sections:

Example: Configuring VLAN Encapsulation on page 357

Example: Configuring Extended VLAN Encapsulation on page 358

Enabling VLAN Tagging

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

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

Binding a VLAN ID to a Logical Interface

Ethernet interfaces can be partitioned; you can assign up to 4095 different logical interfaces, one for each VLAN.

VLAN ID 0 is reserved for tagging the priority of frames. To bind a VLAN ID to a logical interface, include the `vlan-id` statement:

```
vlan-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]
```



NOTE: For Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), VLAN IDs on a single interface can differ from each other.

Because Intermediate System-to-Intermediate System (IS-IS) has an 8-bit limit for broadcast multiaccess media, you cannot set up more than 255 adjacencies over Gigabit Ethernet using VLAN tagging. For more information, see the *JUNOS Routing Protocols Configuration Guide*.

VLAN IDs 1 through 511 are reserved for normal VLANs. VLAN IDs 512 and above are reserved for VLAN CCCs.

For Gigabit Ethernet IQ interfaces and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can configure flexible Ethernet services encapsulation on the physical interface. With flexible Ethernet services encapsulation, VLAN IDs from 1 through 511 are no longer reserved for normal VLANs.

Table 30 lists VLAN ID range by interface type.

Table 30: VLAN ID Range by Interface Type

Interface Type	VLAN ID Range
Aggregated Ethernet	1 through 1023
4-port, 8-port, and 12-port Fast Ethernet	1 through 1023
48-port Fast Ethernet	1 through 4094
Gigabit Ethernet	1 through 4094
Gigabit Ethernet IQ	1 through 4094
10-Gigabit Ethernet	1 through 4094
Management and internal Ethernet interfaces	1 through 1023

Configuring VLAN Encapsulation

Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform) with VLAN tagging enabled can use flexible Ethernet services, VLAN CCC, or VLAN virtual private LAN service (VPLS) encapsulation. To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level, specifying flexible-ethernet-services, vlan-ccc, or vlan-vpls:

```
[edit interfaces interface-name]
encapsulation (flexible-ethernet-services | vlan-ccc | vlan-vpls);
```

Ethernet interfaces in VLAN mode can have multiple logical interfaces. In CCC and VPLS modes, VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 through 4094 are reserved for CCC or VPLS VLANs. For 4-port Fast Ethernet interfaces, you can use VLAN IDs 512 through 1024 for CCC or VPLS VLANs.

For encapsulation type flexible-ethernet-services, all VLAN IDs are valid.

In general, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for some encapsulation types, including flexible Ethernet services, Ethernet VLAN CCC and VLAN VPLS, you can also configure the encapsulation type that is used inside the VLAN circuit itself. To do this, include the encapsulation statement:

```
encapsulation (vlan-ccc | vlan-tcc | vlan-vpls);
```

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]
```

You cannot configure a logical interface with VLAN CCC or VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation or with flexible Ethernet services encapsulation. In general, the logical interface must have a VLAN ID of 512 or higher; if the VLAN ID is 511 or lower, it will be subject to the normal destination filter lookups in addition to source address filtering. However if you configure flexible Ethernet services encapsulation, this VLAN ID restriction is removed.

Example: Configuring VLAN Encapsulation

Configure VLAN CCC encapsulation on a Gigabit Ethernet interface:

```
interfaces ge-2/1/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  unit 0 {
    encapsulation vlan-ccc;
    vlan-id 600;
  }
}
```

Configuring Extended VLAN Encapsulation

Gigabit Ethernet and 4-port Fast Ethernet interfaces with VLAN tagging enabled can use extended VLAN CCC or VLAN VPLS, which allow 802.1Q tagging. To configure the encapsulation on a physical interface, include the encapsulation statement at the [edit interfaces *interface-name*] hierarchy level, specifying extended-vlan-ccc or extended-vlan-vpls:

```
[edit interfaces interface-name]
encapsulation (extended-vlan-ccc | extended-vlan-vpls);
```

For extended VLAN CCC and extended VLAN VPLS encapsulation, all VLAN IDs 1 and higher are valid. VLAN ID 0 is reserved for tagging the priority of frames.



NOTE: For extended VLAN CCC, the VLAN IDs on ingress and egress interfaces must be the same. For back-to-back connections, all VLAN IDs must be the same.

Example: Configuring Extended VLAN Encapsulation

Configure extended VLAN CCC encapsulation on Gigabit Ethernet ingress and egress interfaces:

```
interfaces ge-0/0/0 {
  vlan-tagging;
  encapsulation extended-vlan-ccc;
  unit 0 {
    vlan-id 2;
    family ccc;
  }
}

interfaces ge-1/0/0 {
  vlan-tagging;
  encapsulation extended-vlan-ccc;
  unit 0 {
    vlan-id 2;
    family ccc;
  }
}
```

Configuring TCC and Layer 2.5 Switching

Translational cross-connect (TCC) is a switching concept that allows you to establish interconnections between a variety of Layer 2 protocols or circuits. It is similar to its predecessor, CCC. However, while CCC requires the same Layer 2 encapsulations on both sides of a routing platform (such as PPP-to-PPP or Frame Relay-to-Frame Relay), TCC lets you connect different types of Layer 2 protocols interchangeably. With TCC, combinations such as PPP-to-ATM and Ethernet-to-Frame Relay cross-connections are possible.

You can configure the following Layer 2.5 switching properties:

Configuring VLAN TCC Encapsulation on page 359

Configuring an Ethernet TCC or VLAN TCC on page 360

For examples of Layer 2.5 switching configuration, see the following sections:

Example: Configuring an Ethernet TCC or Extended VLAN TCC on page 361

Example: Configuring Extended VLAN Encapsulation on page 358

Configuring VLAN TCC Encapsulation

VLAN TCC encapsulation allows circuits to have different media on either side of the connection. Extended VLAN TCC encapsulation supports TPIDs 0x8100, 0x9100, and 0x9901. VLAN TCC encapsulation supports TPID 0x8100 only.

One-port Gigabit Ethernet, 2-port Gigabit Ethernet, and 4-port Fast Ethernet PICs with VLAN tagging enabled can use VLAN TCC encapsulation. To configure the encapsulation on a physical interface, include the encapsulation extended-vlan-tcc statement at the [edit interfaces *interface-name*] hierarchy level:

```
[edit interfaces interface-name]  
encapsulation (extended-vlan-tcc | vlan-tcc);
```

For VLAN TCC encapsulation, all VLAN IDs from 1 through 1024 are valid. VLAN ID 0 is reserved for tagging the priority of frames.

Extended VLAN TCC is not supported on 4-port Gigabit Ethernet PICs.

Configuring an Ethernet TCC or VLAN TCC

For Layer 2.5 VPNs using an Ethernet interface as the TCC routing platform, you can configure an Ethernet TCC or a VLAN TCC.

To configure an Ethernet TCC, include the encapsulation `ethernet-tcc` statement at the `[edit interfaces interface-name]` hierarchy level. To configure a VLAN TCC, include the encapsulation `extended-vlan-tcc` or `encapsulation vlan-tcc` statement at the `[edit interfaces interface-name]` hierarchy level:

```
[edit interfaces interface-name
 encapsulation (ethernet-tcc | extended-vlan-tcc | flexible-ethernet-services |
 vlan-tcc);
```

For Ethernet TCC and VLAN TCC, you must also configure the logical interface by including the `proxy` and `remote` statements:

```
proxy {
    inet-address address;
}
remote {
    (inet-address address | mac-address address);
}
```

You can include these statements at the following hierarchy levels:

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

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

The proxy address is the IP address of the non-Ethernet TCC neighbor for which the TCC routing platform is acting as a proxy.

The remote address is the IP or MAC address of the remote routing platform. The remote statement provides ARP capability from the TCC switching routing platform to the Ethernet neighbor. The MAC address is the physical Layer 2 address of the Ethernet neighbor, also known as the remote routing platform.

Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on 1-port Gigabit Ethernet, 2-port Gigabit Ethernet, 4-port Gigabit Ethernet, and 4-port Fast Ethernet PICs only.

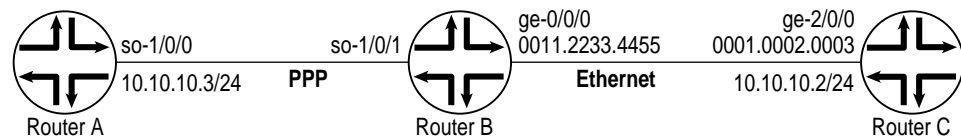
Example: Configuring an Ethernet TCC or Extended VLAN TCC

Configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks routing platform, Router B, as the TCC interface. Ethernet TCC encapsulation provides an Ethernet wide area circuit for interconnecting IP traffic. (See the topology in Figure 27.)

The Router A-to-Router B circuit is PPP, and the Router B-to-Router C circuit accepts packets carrying standard TPID values.

If traffic flows from Router A to Router C, the JUNOS software strips all PPP encapsulation data from incoming packets and adds Ethernet encapsulation data before forwarding the packets. If traffic flows from Router C to Router A, the JUNOS software strips all Ethernet encapsulation data from incoming packets and adds PPP encapsulation data before forwarding the packets.

Figure 27: Example Topology of Layer 2.5 Translational Cross-Connect



1748

```

On Router B
interfaces ge-0/0/0 {
  encapsulation ethernet-tcc;
  unit 0 {
    family tcc {
      proxy {
        inet-address 10.10.10.3;
      }
      remote {
        inet-address 10.10.10.2;
      }
    }
  }
}

```

Configuring an Extended VLAN TCC Configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks routing platform, Router B, as the TCC interface. Extended VLAN TCC encapsulation provides an Ethernet wide area circuit for interconnecting IP traffic. (See the topology in Figure 27 on page 361).

The Router A-to-Router B circuit is PPP, and the Router B-to-Router C circuit is Ethernet with VLAN tagging enabled.

```

On Router B      interfaces ge-0/0/0 {
                    vlan-tagging;
                    encapsulation extended-vlan-tcc;
                    unit 0 {
                      vlan-id 1;
                      family tcc {
                        proxy {
                          inet-address 10.10.10.3/24;
                        }
                        remote {
                          inet-address 10.10.10.2/24;
                        }
                      }
                    }
                  }

```

Configuring Static ARP Table Entries

For Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces, you can configure static ARP table entries, defining mappings between IP and MAC addresses. To configure static ARP table entries, include the `arp` statement:

```
arp ip-address (mac | multicast-mac) mac-address <publish>;
```

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]
```

The IP address that you specify must be part of the subnet defined in the enclosing address statement.

To associate a multicast MAC address with a unicast IP address, include the `multicast-mac` statement.

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

For unicast MAC addresses only, if you include the `publish` option, the routing platform replies to proxy ARP requests.



NOTE: By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the `family inet` statement. By including the `arp` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet policer] hierarchy level, you can apply a specific ARP-packet policer to an interface. For more information, see “Applying Policers” on page 121.

Example: Configuring Static ARP Table Entries

Configure two static ARP table entries on the routing platform’s management interface:

```
[edit interfaces]
fxp0 {
  unit 0 {
    family inet {
      address 10.10.0.11/24 {
        arp 10.10.0.99 mac 0001.0002.0003;
        arp 10.10.0.101 mac 00:11:22:33:44:55 publish;
      }
    }
  }
}
```

Configuring VRRP

For Ethernet, Fast Ethernet, Gigabit Ethernet, and 10-Gigabit Ethernet interfaces, you can configure the Virtual Router Redundancy Protocol (VRRP). VRRP allows hosts on a LAN to make use of redundant routing platforms on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routing platforms share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routing platforms is the master (active) and the others are backups. If the master fails, one of the backup routers becomes the new master, thus always providing a virtual default routing platform and allowing traffic on the LAN to be routed without relying on a single routing platform.

VRRP is defined in RFC 2338, *Virtual Router Redundancy Protocol*.

To configure VRRP, include the `vrrp-group` statement:

```
vrrp-group group-number {
  (accept-data | no-accept-data);
  advertise-interval seconds;
  authentication-key key;
  authentication-type authentication;
  fast-interval milliseconds;
  (preempt | no-preempt) {
    hold-time seconds;
  }
  priority number;
  track {
    interface interface-name priority-cost cost;
    virtual-address [ addresses ];
  }
}
```

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]
```

To trace VRRP operations, include the `traceoptions` statement at the [edit protocols vrrp] hierarchy level:

```
[edit protocols vrrp traceoptions]
file {
  filename filename;
  files number;
  size size;
  (world-readable | no-world-readable);
}
flag flag;
```

For more information, see “Tracing VRRP Operations” on page 371.

You can configure the following VRRP properties:

Configuring Basic VRRP Support on page 365

Configuring VRRP Authentication on page 366

Configuring the Advertisement Interval for the VRRP Master Router on page 367

Configuring a Backup Router to Preempt the Master Router on page 368

Accept Packets Destined for the Virtual IP Address on page 369

Configuring a Logical Interface to Be Tracked on page 370

Tracing VRRP Operations on page 371

For a VRRP configuration example, see “Example: Configuring VRRP” on page 372.

Configuring Basic VRRP Support

An interface can be a member of one or more VRRP groups. To configure basic VRRP support, configure VRRP groups on interfaces by including the following statements:

```

vrp-group group-number {
    priority number;
    virtual-address [ addresses ];
}

```

You can include these statements 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]
```

On a single routing platform, you cannot configure the same VRRP group on the same virtual IP address. Within a single VRRP group, the master and backup routers cannot be the same routing platform.

For each group, you must configure the following:

Group number—Identifies the VRRP group. It can be a value from 0 through 255. If you also enable MAC source address filtering on the interface, as described in “Enabling Ethernet MAC Address Filtering” on page 351, you must include the virtual MAC address in the list of source MAC addresses that you specify in the source-address-filter statement. MAC addresses ranging from 00:00:5e:00:01:00 through 00:00:5e:00:01:ff are reserved for VRRP, as defined in RFC 2338. The VRRP group number must be the decimal equivalent of the last hexadecimal byte of the virtual MAC address.

Addresses of one or more virtual router that are members of the VRRP group—Virtual IP addresses associated with the virtual router in the VRRP group. Normally, you configure only one virtual IP address per group. The virtual IP addresses must be the same for all routing platforms in the VRRP group. You can configure up to eight addresses. In the addresses, specify the address only. Do not include a prefix length.

If you configure a virtual IP address to be the same as the interface’s address (the address configured with the address statement), the interface becomes the master virtual router for the group. In this case, you must configure the priority to be 255 and you must configure preemption by including the preempt statement. If you have multiple VRRP groups on an interface, the interface can be the master virtual router for only one of the groups. If the virtual IP address you choose is not the same as the interface’s address, you must ensure that this address does not appear anywhere else in the routing platform’s configuration. Check that you do not use this address for other interfaces, for the IP address of a tunnel, or for the IP address of static ARP entries.

Priority for this routing platform to become the master virtual router—Value used to elect the master virtual router in the VRRP group. It can be a number from 1 through 255. The default value for backup routers is 100. A larger value indicates a higher priority. The routing platform with the highest priority within the group becomes the master router.

Configuring VRRP Authentication

All VRRP protocol exchanges can be authenticated to guarantee that only trusted routing platforms participate in the AS’s routing. By default, VRRP authentication is disabled. You can configure one of the following authentication methods; each VRRP group must use the same method:

Simple authentication—Uses a text password included in the transmitted packet. The receiving routing platform uses an authentication key (password) to verify the packet.

MD5 algorithm—Creates the authentication data field in the IP authentication header. This header is used to encapsulate the VRRP protocol data unit (PDU). The receiving routing platform uses an authentication key (password) to verify the authenticity of the IP authentication header and VRRP PDU.

To enable authentication and specify an authentication method, include the authentication-type statement:

```
authentication-type authentication;
```

You can include this statement at the following hierarchy levels:

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

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

authentication can be none, simple, or md5. The authentication type must be the same for all routing platforms in the VRRP group.

If you included the *authentication-type* statement to select an authentication method, you can configure a key (password) on each interface by including the *authentication-key* statement:

```
authentication-key key;
```

You can include this statement at the following hierarchy levels:

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

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

The key (password) is an ASCII string. For simple authentication, it can be 1 through 8 characters long. For MD5 authentication, it can be 1 through 16 characters long. If you include spaces, enclose all characters in quotation marks (" "). The key must be the same for all routing platforms in the VRRP group.

Configuring the Advertisement Interval for the VRRP Master Router

By default, the master router sends VRRP advertisement packets every second to all members of the VRRP group. These packets indicate that the master router is still operational. If the master router fails or becomes unreachable, the backup router with the highest priority value becomes the new master router.

You can modify the advertisement interval in seconds or in milliseconds; the interval must be the same for all routing platforms in the VRRP group.

Modifying the Advertisement Interval in Seconds

To modify the time, in seconds, between the sending of VRRP advertisement packets, include the *advertise-interval* statement:

```
advertise-interval seconds;
```

You can include this statement at the following hierarchy levels:

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

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

The interval can be from 1 through 255 seconds.

Modifying the Advertisement Interval in Milliseconds

To modify the time, in milliseconds, between the sending of VRRP advertisement packets, include the `fast-interval` statement:

```
fast-interval milliseconds;
```

You can include this statement at the following hierarchy levels:

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

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

The interval can be from 100 through 999 milliseconds.



NOTE: In the VRRP PDU, the JUNOS software sets the advertisement interval to 0. When you configure VRRP with other vendors' routers, the `fast-interval` statement works correctly only when the other routers also have an advertisement interval set to 0 in the VRRP PDUs. Otherwise, the JUNOS software interprets other routers' settings as advertisement timer errors.

Configuring a Backup Router to Preempt the Master Router

By default, a higher-priority backup router preempts a lower priority master router. To explicitly allow the master router to be preempted, include the `preempt` statement:

```
preempt;
```

To prohibit a higher-priority backup router from preempting a lower priority master router, include the `no-preempt` statement:

```
no-preempt;
```

You can include these statements at the following hierarchy levels:

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

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



NOTE: The routing platform that owns the IP address or addresses associated with the virtual router always preempts, independent of the setting of this flag.

Modifying the Preemption Hold-Time Value

The hold time is the maximum number of seconds allowed to elapse before a higher-priority backup router preempts the master router. You might want to configure a hold time to allow all the JUNOS software components to converge before preemption.

By default, the hold-time value is 0 seconds. A value of 0 means that preemption can occur immediately after the backup router comes online. Note that the hold time is counted from the time the backup router comes online. The hold time is only valid when the VRRP router is just coming online.

To modify the preemption hold-time value, include the hold-time statement:

```
hold-time seconds;
```

You can include this statement at the following hierarchy levels:

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

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

The hold time can be a value from 0 through 3600 seconds.

Accept Packets Destined for the Virtual IP Address

To configure an interface to accept packets destined for the virtual IP address, include the accept-data statement:

```
accept-data;
```

To prohibit the interface from accepting packets destined for the virtual IP address, include the no-accept-data statement:

```
no-accept-data;
```

You can include these statements at the following hierarchy levels:

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

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

The accept-data statement has the following consequences:

If the master router owns the virtual IP address, the accept-data statement is not valid.

If the master router owns the virtual IP address, the master router responds to ICMP message requests only.

when the priority of the master router is set to 255, the accept-data statement is not valid.

To restrict incoming IP packets to ICMP only, you must configure firewall filters to accept only ICMP packets.

If you include the accept-data statement, your routing platform configuration will not comply with RFC 2338.

If you include the accept-data statement, VRRP clients should be able to process Gratuitous ARP.

If you include the accept-data statement, VRRP clients should not use packets other than ARP replies to update their ARP cache.

Configuring a Logical Interface to Be Tracked

VRRP can track whether a logical interface is up, down, or not present and dynamically change the priority of the VRRP group based on the state of the tracked logical interface, which might trigger a new master router election.

When interface tracking is enabled, you cannot configure a priority of 255, thereby designating the master router. For each VRRP group, you can track up to 10 logical interfaces.

To configure a logical interface to be tracked, include the track statement:

```
track {
  interface interface-name priority-cost cost;
}
```

You can include this statement at the following hierarchy levels:

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

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

The priority cost is the value to be subtracted from the configured VRRP priority when the tracked logical interface is down, forcing a new master router election. The value can be from 1 through 254. The sum of the costs for all tracked logical interfaces or routes must be less than or equal to the configured priority of the VRRP group.

Tracing VRRP Operations

To trace VRRP operations, include the `traceoptions` statement at the `[edit protocols vrrp]` hierarchy level.

By default, VRRP logs the error, DCD configuration, and routing socket events in a file in the `/var/log` directory. By default, this file is named `/var/log/vrrpd`. The default file size is 1 MB, and three files are created before the first one gets overwritten.

To change the configuration of the logging file, include the `file` statement at the `[edit protocols vrrp traceoptions]` hierarchy level:

```
[edit protocols vrrp traceoptions]
file {
  filename filename;
  files number;
  size size;
  (world-readable | no-world-readable);
}
flag flag;
```

You can specify the following VRRP tracing flags:

- `all`—Trace all VRRP operations.
- `database`—Trace all database changes.
- `general`—Trace all general events.
- `interfaces`—Trace all interface changes.
- `normal`—Trace all normal events.
- `packets`—Trace all packets sent and received.
- `state`—Trace all state transitions.
- `timer`—Trace all timer events.

Example: Configuring VRRP

Configure one master (Router A) and one backup (Router B) routing platform. Note that the address configured in the virtual-address statements differs from the addresses configured in the address statements. Note that when you configure multiple VRRP groups on an interface, you configure one to be the master virtual router for that group.

```

On Router A    [edit]
                  interfaces {
                    ge-0/0/0 {
                      unit 0 {
                        family inet {
                          address 192.168.1.20/24 {
                            vrrp-group 27 {
                              virtual-address 192.168.1.15;
                              priority 254;
                              authentication-type simple;
                              authentication-key booJUM;
                            }
                          }
                        }
                      }
                    }
                  }

On Router B    [edit]
                  interfaces {
                    ge-4/2/0 {
                      unit 0 {
                        family inet {
                          address 192.168.1.24/24 {
                            vrrp-group 27 {
                              virtual-address 192.168.1.15;
                              priority 200;
                              authentication-type simple;
                              authentication-key booJUM;
                            }
                          }
                        }
                      }
                    }
                  }

```

**Configuring One Router
to be the Master Virtual
Router for the Group**

```
[edit]
interfaces {
  ge-0/0/0 {
    unit 0 {
      family inet {
        address 192.168.1.20/24 {
          vrrp-group 2 {
            virtual-address 192.168.1.20;
            priority 255;
            advertise-interval 3;
            preempt;
          }
          vrrp-group 10 {
            virtual-address 192.168.1.55;
            priority 201;
            advertise-interval 3;
          }
          vrrp-group 1 {
            virtual-address 192.168.1.54;
            priority 22;
            advertise-interval 4;
          }
        }
      }
    }
  }
}
```

**Configuring VRRP and
MAC Source Address
Filtering**

The VRRP group number is the decimal equivalent of the last byte of the virtual MAC address.

```
[edit interfaces]
ge-5/2/0 {
  gigether-options {
    source-filtering;
    source-address-filter {
      00:00:5e:00:01:0a;<— Virtual MAC address
    }
  }
  unit 0 {
    family inet {
      address 192.168.1.10/24 {
        vrrp-group 10 {<— VRRP group number
          virtual-address 192.168.1.10;
          priority 255;
          preempt;
        }
      }
    }
  }
}
```

Configuring Gigabit Ethernet Accounting and Policing

For Gigabit Ethernet IQ PICs and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can configure granular per-VLAN class-of-service (CoS) capabilities and extensive instrumentation and diagnostics on a per-VLAN and per-MAC address basis.

VLAN rewrite, tagging, and deleting enables you to use VLAN address space to support more customers and services.

Virtual private LAN service (VPLS) allows you to provide a point-to-multipoint LAN between a set of sites in a virtual private network (VPN). Ethernet IQ PICs and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform) are combined with VPLS to deliver metro Ethernet service.

The Gigabit Ethernet PICs require an enhanced FPC.

Table 31 on page 374 lists the capabilities of Gigabit Ethernet IQ PICs and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform).

Table 31: Capabilities of Gigabit Ethernet IQ and Gigabit Ethernet with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform)

Capability	Gigabit Ethernet IQ (SFP)	Gigabit Ethernet (SFP)
Layer 2		
802.3ad link aggregation	Yes	Yes
Maximum VLANs per port	384	1023
MTU size	9192	9192
MAC learning	Yes	Yes
MAC accounting	Yes	Yes
MAC filtering	Yes	Yes
Destinations per port	960	960
Sources per port	64	64
Hierarchical MAC policers	Yes, premium and aggregate	No, aggregate only
Multiple TPID support and IP service for nonstandard TPIDs	Yes	Yes
Multiple Ethernet encapsulations	Yes	Yes
Dual VLAN tags	Yes	No
VLAN rewrite	Yes	No
Layer 2 VPNs		
VLAN CCC	Yes	Yes
Port-based CCC	Yes	Yes
Extended VLAN CCC VMANs Tag Protocol	Yes	Yes

Capability	Gigabit Ethernet IQ (SFP)	Gigabit Ethernet (SFP)
CoS		
PIC-based egress queues	Yes	Yes
Queued VLANs	Yes	No
VLAN virtual private LAN service (VPLS)	Yes	Yes

For more information about configuring VPLS, see the *JUNOS VPNs Configuration Guide* and the *JUNOS Feature Guide*.

This section is organized as follows:

Configuring Gigabit Ethernet Policers on page 375

Configuring MAC Address Accounting on page 381

Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags on page 382

For example configurations, see the following sections:

Example: Configuring Gigabit Ethernet Policers on page 380

Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags on page 386

You can also configure CoS on logical IQ interfaces. For more information, see “Associating a Scheduler Map with a DLCI or VLAN” on page 846.

Configuring Gigabit Ethernet Policers

On Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can define rate limits for premium and aggregate traffic received on the interface. These policers allow you to perform simple traffic policing without configuring a firewall filter. First you configure the Ethernet policer profile, next you classify ingress and egress traffic, then you can apply the policer to a logical interface.

For Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), the policer rates you configure can be different than the rates on the Packet Forward Engine. The difference results from Layer 2 overhead. The PIC accounts for this difference.

This section is organized as follows:

Configuring a Policer on page 376

Specifying an Input Priority Map on page 377

Specifying an Output Priority Map on page 377

Applying a Policer on page 378

Configuring MAC Address Filtering on page 379

Example: Configuring Gigabit Ethernet Policers on page 380

Configuring a Policer

To configure an Ethernet policer profile, include the `ethernet-policer-profile` statement at the [edit interfaces *interface-name* together-options ethernet-switch-profile] hierarchy level:

```
[edit interfaces interface-name together-options ethernet-switch-profile]
ethernet-policer-profile {
  policer cos-policer-name {
    aggregate {
      bandwidth-limit bps;
      burst-size-limit bytes;
    }
    premium {
      bandwidth-limit bps;
      burst-size-limit bytes;
    }
  }
}
```

In the Ethernet policer profile, the aggregate-priority policer is mandatory; the premium-priority policer is optional.

For aggregate and premium policers, you specify the bandwidth limit in bits per second. You can specify the value 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). There is no absolute minimum value for bandwidth limit, but any value below 61,040 bps will result in an effective rate of 30,520 bps. The maximum bandwidth limit is 4.29 Gbps.

The maximum burst size controls the amount of traffic bursting allowed. To determine the burst-size limit, you can multiply the bandwidth of the interface on which you are applying the filter by the amount of time you allow a burst of traffic at that bandwidth to occur:

$$\text{burst size} = \text{bandwidth} \times \text{allowable time for burst traffic}$$

If you do not know the interface bandwidth, you can multiply the maximum transmission unit (MTU) of the traffic on the interface by 10 to obtain a value. For example, the burst size for an MTU of 4700 would be 47,000 bytes. The burst size should be at least 10 interface MTUs. The maximum value for the burst-size limit is 100 MB.

Specifying an Input Priority Map

An input priority map identifies ingress traffic with specified IEEE 802.1p priority values, and classifies that traffic as premium.

If you include a premium-priority policer, you can specify an input priority map by including the `ieee802.1p premium` statement at the [edit interfaces *interface-name* *gigether-options* ethernet-policer-profile input-priority-map] hierarchy level:

```
[edit interfaces interface-name gigether-options ethernet-policer-profile
input-priority-map]
ieee802.1p premium [ values ];
```

The priority values can be from 0 through 7. The remaining traffic is classified as nonpremium (or aggregate). For a configuration example, see “Example: Configuring Gigabit Ethernet Policers” on page 380.

Specifying an Output Priority Map

An output priority map identifies egress traffic with specified queue classification and PLP, and classifies that traffic as premium.

If you include a premium-priority policer, you can specify an output priority map by including the `classifier` statement at the [edit interfaces *interface-name* *gigether-options* ethernet-policer-profile output-priority-map] hierarchy level:

```
[edit interfaces interface-name gigether-options ethernet-policer-profile
output-priority-map]
classifier {
  premium {
    forwarding-class class-name {
      loss-priority (high | low);
    }
  }
}
```

You can define a forwarding class, or you can use a predefined forwarding class. Table 32 shows the predefined forwarding classes and their associated queue assignments.

Table 32: Default Forwarding Classes

Forwarding Class Name	Queue
best-effort	queue 0
expedited-forwarding	queue 1
assured-forwarding	queue 2
network-control	queue 3

For more information about CoS forwarding classes, see “Configuring Forwarding Classes” on page 821 and “Classifying Packets by Behavior Aggregate” on page 830. For a configuration example, see “Example: Configuring Gigabit Ethernet Policers” on page 380.

Applying a Policer

On Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can apply input and output policers that define rate limits for premium and aggregate traffic received on the logical interface. Aggregate policers are supported on Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform).

These policers allow you to perform simple traffic policing without configuring a firewall filter. For information about defining these policers, see “Configuring Gigabit Ethernet Policers” on page 375.

To apply policers to specific source MAC addresses, include the `accept-source-mac` statement:

```
accept-source-mac {
  mac-address mac-address {
    policer {
      input cos-policer-name;
      output cos-policer-name;
    }
  }
}
```

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]
```

You can specify the MAC address as `nn:nn:nn:nn:nn:nn` or `nnnn.nnnn.nnnn`, where *n* is a hexadecimal number. You can configure up to 64 source addresses. To specify more than one address, include multiple `mac-address` statements in the logical interface configuration.



NOTE: If the remote Ethernet card is changed, the interface does not accept traffic from the new card because the new card has a different MAC address.

The MAC addresses you include in the configuration are entered into the routing platform’s MAC database. To view the routing platform’s MAC database, enter the `show interfaces mac-database interface-name` command:

```
user@host> show interfaces mac-database interface-name
```

In the input statement, list the name of one policer template to be evaluated when packets are received on the interface.

In the output statement, list the name of one policer template to be evaluated when packets are transmitted on the interface.

You can use the same policer one or more times.

If you apply both policers and firewall filters to an interface, input policers are evaluated before input firewall filters, and output policers are evaluated after output firewall filters.

Configuring MAC Address Filtering

You cannot explicitly define traffic with specific source MAC addresses to be rejected; however, for Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can block all incoming packets that do not have a source address specified in the `accept-source-mac` statement. For more information about the `accept-source-mac` statement, see “Applying a Policer” on page 378.

To enable this blocking, include the `source-filtering` statement at the `[edit interfaces interface-name gigether-options]` hierarchy level:

```
[edit interfaces interface-name gigether-options]
source-filtering;
```

For more information about the `source-filtering` statement, see “Enabling Ethernet MAC Address Filtering” on page 351.

To accept traffic even though it does not have a source address specified in the `accept-source-mac` statement, include the `no-source-filtering` statement at the `[edit interfaces interface-name gigether-options]` hierarchy level:

```
[edit interfaces interface-name gigether-options]
no-source-filtering;
```

For more information about the `accept-source-mac` statement, see “Applying a Policer” on page 378.

Example: Configuring Gigabit Ethernet Policers

Configure interface ge-6/0/0 to treat priority values 2 and 3 as premium. On ingress, this means that IEEE 802.1p priority values 2 and 3 are treated as premium. On egress, it means traffic that is classified into queue 0 or 1 with PLP of low and queue 2 or 3 with PLP of high, is treated as premium.

Define a policer that limits the premium bandwidth to 100 Mbps and burst size to 3 k, and the aggregate bandwidth to 200 Mbps and burst size to 3 k.

Specify that frames received from the MAC address 00:01:02:03:04:05 and the VLAN ID 600 are subject to the policer on input and output. On input, this means frames received with the source MAC address 00:01:02:03:04:05 and the VLAN ID 600 are subject to the policer. On output, this means frames transmitted from the routing platform with the destination MAC address 00:01:02:03:04:05 and the VLAN ID 600 are subject to the policer.

```
[edit interfaces]
ge-6/0/0 {
  gigether-options {
    ether-switch-profile {
      ether-policer-profile {
        input-priority-map {
          ieee-802.1p {
            premium [ 2 3 ];
          }
        }
      }
    }
    output-priority-map {
      classifier {
        premium {
          forwarding-class best-effort {
            loss-priority low;
          }
          forwarding-class expedited-forwarding {
            loss-priority low;
          }
          forwarding-class assured-forwarding {
            loss-priority high;
          }
          forwarding-class network-control {
            loss-priority high;
          }
        }
      }
    }
  }
}
```

```

    policer policer-1 {
      premium {
        bandwidth-limit 100m;
        burst-size-limit 3k;
      }
      aggregate {
        bandwidth-limit 200m;
        burst-size-limit 3k;
      }
    }
  }
}
unit 0 {
  accept-source-mac {
    mac-address 00:01:02:03:04:05 {
      policer {
        input policer-1;
        output policer-1;
      }
    }
  }
}
}
}

```

Configuring MAC Address Accounting

For Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can configure whether source and destination MAC addresses are dynamically learned. To configure MAC address accounting, include the `mac-learn-enable` statement at the [edit interfaces *interface-name* gigeother-options ethernet-switch-profile] hierarchy level:

```
[edit interfaces interface-name gigeother-options ethernet-switch-profile]
mac-learn-enable;
```

To prohibit the interface from dynamically learning source and destination MAC addresses, include the `no-mac-learn-enable` statement at the [edit interfaces *interface-name* gigeother-options ethernet-switch-profile] hierarchy level:

```
[edit interfaces interface-name gigeother-options ethernet-switch-profile]
no-mac-learn-enable;
```

MAC address learning is based on source addresses. You can start accounting for traffic after it has been sent from the MAC address. Once the MAC address is learned, the frames and bytes transmitted to or received from the MAC address can be tracked.

Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags

On Gigabit Ethernet IQ interfaces with encapsulation type extended-vlan-ccc or vlan-ccc, you can stack and rewrite VLAN tags. Stacking and rewriting VLAN tags allows you to use an additional (outer) VLAN tag to differentiate between customer edge (CE) routing platforms that share one VLAN ID.

This section is organized as follows:

Stacking a VLAN Tag on All Tagged Frames Entering the Interface on page 382

Stacking a VLAN Tag on All Tagged Frames Exiting the Interface on page 383

Removing a VLAN Tag from All Tagged Frames Entering the Interface on page 383

Removing a VLAN Tag from All Tagged Frames Exiting the Interface on page 384

Configuring Frames with Particular TPIDs to be Processed as Tagged Frames on page 384

Rewriting the VLAN Tag on All Tagged Frames Entering the Interface on page 384

Rewriting the VLAN Tag on All Tagged Frames Exiting the Interface on page 385

Configuring Stacked VLAN Tagging on page 385

Configuring Dual VLAN Tags on page 386

Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags on page 386

Stacking a VLAN Tag on All Tagged Frames Entering the Interface

To stack a VLAN tag on all tagged frames entering the interface, include the push, vlan-id, and tag-protocol-id statements in the input VLAN map:

```
input-vlan-map {
    push;
    vlan-id number;
    tag-protocol-id tpid;
}
```

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]
```

If you include the push statement in an interface's input VLAN map, you must include the pop statement in the interface's output VLAN map.

The VLAN IDs you define in the input VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see “Configuring 802.1Q VLANs” on page 355.

All TPIDs you include in input and output VLAN maps must be among those you specify at the [edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]] hierarchy level. For more information, see “Configuring Frames with Particular TPIDs to be Processed as Tagged Frames” on page 384.

Stacking a VLAN Tag on All Tagged Frames Exiting the Interface

To stack a VLAN tag on all tagged frames exiting the interface, include the push, vlan-id, and tag-protocol-id statements in the output VLAN map:

```
output-vlan-map {
  push;
  vlan-id number;
  tag-protocol-id tpid;
}
```

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]
```

The VLAN IDs you define in the output VLAN maps are stacked on top of the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see “Configuring 802.1Q VLANs” on page 355.

All TPIDs you include in input and output VLAN maps must be among those you specify at the [edit interfaces *interface-name* gigether-options ethernet-switch-profile tag-protocol-id [*tpids*]] hierarchy level. For more information, see “Configuring Frames with Particular TPIDs to be Processed as Tagged Frames” on page 384.

Removing a VLAN Tag from All Tagged Frames Entering the Interface

To remove a VLAN tag from all tagged frames entering the interface, include the pop statement in the input VLAN map:

```
input-vlan-map {
  pop;
}
```

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]
```

Removing a VLAN Tag from All Tagged Frames Exiting the Interface

To remove a VLAN tag from all tagged frames entering or exiting the interface, include the `pop` statement in the output VLAN map:

```
output-vlan-map {
    pop;
}
```

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]
```

Configuring Frames with Particular TPIDs to be Processed as Tagged Frames

For Gigabit Ethernet IQ and Gigabit Ethernet PICs with SFPs (except the 10-port Gigabit Ethernet PIC and the built-in Gigabit Ethernet port on the M7i platform), you can configure frames with particular TPIDs to be processed as tagged frames. To do this, you specify up to eight IEEE 802.1Q TPID values per port; a frame with any of the specified TPIDs is processed as a tagged frame. To configure the TPID values, include the `tag-protocol-id` statement at the `[edit interfaces interface-name ggether-options ethernet-switch-profile]` hierarchy level:

```
[edit interfaces interface-name ggether-options ethernet-switch-profile]
tag-protocol-id [ tpids ];
```

All TPIDs you include in input and output VLAN maps must be among those you specify at the `[edit interfaces interface-name ggether-options ethernet-switch-profile tag-protocol-id [tpids]]` hierarchy level.

Rewriting the VLAN Tag on All Tagged Frames Entering the Interface

To rewrite the VLAN tag on all tagged frames entering the interface to a specified VLAN ID and TPID, include the `swap`, `tag-protocol-id`, and `vlan-id` statements in the input VLAN map:

```
input-vlan-map {
    swap;
    vlan-id number;
    tag-protocol-id tpid;
}
```

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]
```

The `swap` operation works on the outer tag only, whether or not you include the `stacked-vlan-tagging` statement in the configuration. For more information, see “Configuring Stacked VLAN Tagging” on page 385.

Rewriting the VLAN Tag on All Tagged Frames Exiting the Interface

To rewrite the VLAN tag on all tagged frames exiting the interface to a specified VLAN ID and TPID, include the `swap` and `tag-protocol-id` statements in the output VLAN map:

```
output-vlan-map {
    swap;
    tag-protocol-id tpid;
}
```

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]
```

You cannot include both the `swap` statement and the `vlan-id` statement in the output VLAN map configuration. If you include the `swap` statement in the configuration, the VLAN ID in outgoing frames is rewritten to the VLAN ID bound to the logical interface. For more information about binding a VLAN ID to the logical interface, see “Configuring 802.1Q VLANs” on page 355.

The `swap` operation works on the outer tag only, whether or not you include the `stacked-vlan-tagging` statement in the configuration. For more information, see “Configuring Stacked VLAN Tagging” on page 385.

Configuring Stacked VLAN Tagging

To configure stacked VLAN tagging for all logical interfaces on a physical interface, include the `stacked-vlan-tagging` statement at the `[edit interfaces interface-name]` hierarchy level:

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

If you include the `stacked-vlan-tagging` statement in the configuration, you must configure dual VLAN tags for all logical interfaces on the physical interface. For more information, see “Configuring Dual VLAN Tags” on page 386.

Configuring Dual VLAN Tags

To configure dual VLAN tags on a logical interface, include the `vlan-tags` statement:

```
vlan-tags inner tpid.vlan-id outer tpid.vlan-id;
```

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 outer tag VLAN ID range is from 1 through 511 for normal interfaces, and from 512 through 4094 for VLAN CCC or VLAN VPLS interfaces. The inner tag does not have this restriction.

You must also include the `stacked-vlan-tagging` statement in the configuration. See “Configuring Stacked VLAN Tagging” on page 385.

Examples: Stacking and Rewriting Gigabit Ethernet IQ VLAN Tags

Configure a VLAN CCC tunnel, in which Ethernet frames enter the tunnel at interface `ge-4/0/0` and exit the tunnel at interface `ge-4/2/0`. The following examples show how to perform the following tasks:

Push a TPID and VLAN ID pair on ingress.

Stack inner and outer VLAN tags.

Swap a VLAN ID on ingress.

Swap a VLAN ID on egress.

Swap a VLAN ID on both ingress and egress.

Push a TPID and VLAN ID Pair on Ingress

```
[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  ggether-options {
    ethernet-switch-profile {
      tag-protocol-id 0x9909;
    }
  }
  unit 0 {
    encapsulation vlan-ccc;
    vlan-id 512;
    input-vlan-map {
      push;
      tag-protocol-id 0x9909;
      vlan-id 520;
    }
    output-vlan-map pop;
  }
}
```

```

ge-4/2/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  unit 0 {
    encapsulation vlan-ccc;
    vlan-id 520;
  }
}

[edit protocols]
mpls {
  interface ge-4/0/0.0;
  interface ge-4/2/0.0;
}
connections {
  interface-switch vlan-tag-push {
    interface ge-4/0/0.0;
    interface ge-4/2/0.0;
  }
}

Stack Inner and Outer
VLAN Tags
[edit interfaces]
ge-0/2/0 {
  stacked-vlan-tagging;
  mac 00.01.02.03.04.05;
  gigheter-options {
    loopback;
  }
  unit 0 {
    vlan-tags outer 0x8100.200 inner 0x8100.200;
  }
}

Swap a VLAN ID on
Ingress
[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  gigheter-options {
    ethernet-switch-profile {
      tag-protocol-id 0x9100;
    }
  }
}
...
unit 1 {
  encapsulation vlan-ccc;
  vlan-id 1000;
  input-vlan-map {
    swap;
    tag-protocol-id 0x9100;
    vlan-id 2000;
  }
}
}

```

Swap a VLAN ID on Egress

```

ge-4/2/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 2000;
  }
}

[edit protocols]
mpls {
  ...
  interface ge-4/0/0.1;
  interface ge-4/2/0.1;
}
connections {
  ...
  interface-switch vlan-tag-swap {
    interface ge-4/2/0.1;
    interface ge-4/0/0.1;
  }
}

[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 1000;
  }
}
ge-4/2/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  gigether-options {
    ethernet-switch-profile {
      tag-protocol-id 0x8800;
    }
  }
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 2000;
    output-vlan-map {
      swap;
      tag-protocol-id 0x8800;
    }
  }
}

```

**Swap a VLAN ID on
Both Ingress and
Egress**

```
[edit protocols]
mpls {
  ...
  interface ge-4/0/0.1;
  interface ge-4/2/0.1;
}
connections {
  ...
  interface-switch vlan-tag-swap {
    interface ge-4/2/0.1;
    interface ge-4/0/0.1;
  }
}

[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  together-options {
    ethernet-switch-profile {
      tag-protocol-id [ 0x8800 0x9100 ];
    }
  }
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 1000;
    input-vlan-map {
      swap;
      tag-protocol-id 0x9100;
      vlan-id 2000;
    }
  }
}
ge-4/2/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  together-options {
    ethernet-switch-profile {
      tag-protocol-id [ 0x8800 0x9100 ];
    }
  }
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 2000;
    output-vlan-map {
      swap;
      tag-protocol-id 0x8800;
    }
  }
}
}
```

```
[edit protocols]
mpls {
  ...
  interface ge-4/0/0.1;
  interface ge-4/2/0.1;
}
connections {
  ...
  interface-switch vlan-tag-swap {
    interface ge-4/2/0.1;
    interface ge-4/0/0.1;
  }
}
```

Configuring the Management Ethernet Interface

The routing platform's management Ethernet interface, fxp0, is an out-of-band management interface. You must configure an IP address and prefix length for this interface, which you commonly do when you first install the JUNOS software:

```
[edit]
user@host# set interfaces fxp0 unit 0 family inet address/prefix-length
[edit]
user@host# show
interfaces {
  fxp0 {
    unit 0 {
      family inet {
        address/prefix-length;
      }
    }
  }
}
```



NOTE: The management Ethernet interface must be configured for the routing platform to function.

Configuring the MAC Address on the Management Ethernet Interface

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

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

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

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



NOTE: When you integrate a standalone T640 routing node into a routing matrix, the PIC MAC addresses for the integrated T640 routing node are derived from a pool of MAC addresses maintained by the TX Matrix platform. For each MAC address you specify in the configuration of a formerly standalone T640 routing node, you must specify the same MAC address in the configuration of the TX Matrix platform.

Displaying the Internal Ethernet Interface

The internal Ethernet interface, fxp1, connects the Routing Engine with the routing platform's packet forwarding components. The JUNOS software automatically configures this interface.



NOTE: Do not modify or remove the configuration for the internal Ethernet interface that the JUNOS software automatically configures. If you do, the routing platform will stop functioning.

```
user@host> show configuration
...
interfaces {
...
    fxp1 {
        unit 0 {
            family tnp {
                address 1;
            }
        }
    }
}
```

Configuring Aggregated Ethernet Interfaces

Link aggregation of Ethernet interfaces is defined in the IEEE 802.3ad standard. The JUNOS implementation of 802.3AD balances traffic across the member links within an aggregated Ethernet bundle based on the Layer 3 information carried in the packet. This implementation uses the same load-balancing algorithm used for per-packet load balancing. For information about per-packet load balancing, see the *JUNOS Routing Protocols Configuration Guide*.



NOTE: The JUNOS software does not provide load balancing for multicast traffic on aggregated interfaces. If a link carrying multicast data goes down, another link carries the traffic. This provides redundancy, not more bandwidth.

For information about configuring circuit cross-connects over aggregated Ethernet, see “Examples: Configuring Switching Cross-Connects” on page 146.

You configure an aggregated Ethernet virtual link by specifying the link number as a physical device and then associating a set of ports that have the same speed and are in full-duplex mode. The physical interfaces can be Fast Ethernet, Gigabit Ethernet, Gigabit Ethernet IQ, or 10-Gigabit Ethernet devices. Generally, you cannot use a combination of these interfaces within the same aggregated link; however, you can combine Gigabit Ethernet and Gigabit Ethernet IQ interfaces in a single aggregated Ethernet bundle.

On the aggregated bundle, no IQ-specific capabilities such as MAC accounting, VLAN rewrites, and VLAN queuing are available. For more information about IQ-specific capabilities, see “Configuring Gigabit Ethernet Accounting and Policing” on page 374.

Aggregated Ethernet interfaces can be either tagged or untagged. In either case, you must set the number of aggregated Ethernet interfaces on the chassis, as described in the following sections:

Configuring Tagged Aggregated Ethernet Interfaces on page 392

Configuring Untagged Aggregated Ethernet Interfaces on page 393

Set the Number of Aggregated Ethernet Interfaces on the Chassis on page 394

Configuring Tagged Aggregated Ethernet Interfaces

To specify aggregated Ethernet interfaces, include the `vlan-tagging` statement at the `[edit interfaces aex]` hierarchy level:

```
[edit interfaces aex]
vlan-tagging;
```

You must also include the `vlan-id` statement:

```
vlan-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]
```

For more information about the `vlan-tagging` and `vlan-id` statements, see “Configuring 802.1Q VLANs” on page 355.

Configuring Untagged Aggregated Ethernet Interfaces

When you configure an untagged Aggregated Ethernet interface, the existing rules for untagged interfaces apply. These rules are as follows:

You can configure only one logical interface (unit 0) on the port. The logical unit 0 is used to send and receive LACP or marker PDUs to and from the individual links.

You cannot include the `vlan-id` statement in the configuration of the logical interface.

Table 33 lists untagged aggregated Ethernet and LACP support by PIC and platform.

Table 33: Untagged Aggregated Ethernet and LACP Support by PIC and Platform

PIC Type	M-series	LACP	T-Series	LACP
4-port Fast Ethernet PIC Type 1	Yes	Yes	Yes	Yes
1-port Gigabit Ethernet PIC Type 1	Yes	Yes	Yes	Yes
2-port Gigabit Ethernet PIC Type 2	Yes	Yes	Yes	Yes
4-port Gigabit Ethernet PIC Type 2	Yes	Yes	Yes	Yes
1-port 10-Gigabit Ethernet M160	Yes	Yes	NA	NA
10-port Gigabit Ethernet PIC Type 3	NA	NA	Yes	Yes
1-port 10-Gigabit Ethernet PIC Type 3	NA	NA	Yes	Yes

Syslog messages are logged if you try to configure an untagged aggregated Ethernet interface using an unsupported PIC type.

To configure an untagged aggregated Ethernet interface, omit the `vlan-tagging` and `vlan-id` statements from the configuration, as shown in “Example: Configuring Untagged Aggregated Ethernet Interfaces interfaces” on page 393. For more information about configuring LACP, see “Configuring Aggregated Ethernet LACP” on page 348

Example: Configuring Untagged Aggregated Ethernet Interfaces interfaces

```
[edit interfaces]
fe-5/0/1 {
  fastether-options {
    802.3ad ae0;
  }
}
ae0 {
  unit 0 {
    family inet {
      address 13.1.1.2/24 {
```

```

        vrrp-group 0 {
            virtual-address 13.1.1.4;
            priority 200;
        }
    }
}

```

Set the Number of Aggregated Ethernet Interfaces on the Chassis

By default, no aggregated Ethernet interfaces are created. You must define the number of aggregated Ethernet interfaces by including the `device-count` statement at the `[edit chassis aggregated-devices ethernet]` hierarchy level:

```

[edit chassis]
aggregated-devices {
    ethernet {
        device-count number;
    }
}

```

The maximum number of aggregated devices you can configure is 128. The aggregated interfaces are numbered from `ae0` through `ae127`. For information about configuring aggregated devices, see the *JUNOS System Basics Configuration Guide*.

You must also specify the constituent physical links by including the `802.3ad` statement at the `[edit interfaces interface-name fastether-options]` or `[edit interfaces interface-name gigether-options]` hierarchy level; for more information, see “Configuring Ethernet Link Aggregation” on page 347. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configuring Ethernet Physical Interface Properties” on page 344. For a sample configuration, see “Example: Configuring Aggregated Ethernet Interfaces” on page 395.

To delete an aggregated Ethernet interface from the configuration, issue the `delete interfaces aex` command at the `[edit]` hierarchy level in configuration mode:

```

[edit]
user@host# delete interfaces aex

```

If you delete an aggregated Ethernet interface from the configuration, the JUNOS software removes the configuration statements related to `aex` and sets this interface to down state. However, the aggregated Ethernet interface is not deleted until you delete the `chassis aggregated-devices ethernet device-count` configuration statement.

Example: Configuring Fast Ethernet Interfaces

The following configuration is sufficient to get a Fast Ethernet interface up and running. By default, IPv4 Fast Ethernet interfaces use 802.3 encapsulation.

```

[edit]
user@host# set interfaces fe-fpc/pic/port unit 0 family inet address local-address
user@host# show

```

```

interfaces {
  fe-fpc/pic/port {
    unit 0 {
      family inet {
        address local-address;
      }
    }
  }
}

```

Example: Configuring Gigabit Ethernet Interfaces

The following configuration is sufficient to get a Gigabit Ethernet or 10-Gigabit Ethernet interface up and running. By default, IPv4 Gigabit Ethernet interfaces use 802.3 encapsulation.

```

[edit]
user@host# set interfaces ge-fpc/pic/port unit 0 family inet address
local-address
user@host# show
interfaces {
  ge-fpc/pic/port {
    unit 0 {
      family inet {
        address local-address;
      }
    }
  }
}

```

The M160, M320, T320, and T640 2-port Gigabit Ethernet PIC supports two independent Gigabit Ethernet links.

Each of the two interfaces on the PIC is named:

```
ge-fpc/pic/[0.1]
```

Each of these interfaces has functionality identical to the Gigabit Ethernet interface supported on the single-port PIC.

Example: Configuring Aggregated Ethernet Interfaces

The following set of configurations is sufficient to get an aggregated Gigabit Ethernet interface up and running.

```

[edit interfaces]
ae0 {
  vlan-tagging;
  unit 0 {
    vlan-id 100;
    family inet {
      address 10.1.1.1/24;
    }
  }
}

```

```
}  
  
[edit chassis]  
aggregated-devices {  
  ethernet {  
    device-count 15;  
  }  
}  
  
[edit interfaces]  
ge-1/3/0 {  
  gigheter-options {  
    802.3ad ae0;  
  }  
}  
  
[edit interfaces ae0]  
aggregated-ether-options {  
  link-speed 1g;  
  minimum-links 5;  
}
```