

Chapter 21

Configure Ethernet Interfaces

Ethernet was developed in the early 1970s at the Xerox Palo Alto Research Center 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 10BaseT. More recently, 100BaseTX (Fast Ethernet, 100 Mbps), Gigabit Ethernet (1 Gbps), and 10-Gigabit Ethernet (10 Gbps) have become available.

Juniper Networks routers support the following types of Ethernet interfaces:

- Fast Ethernet

- Gigabit Ethernet

- Gigabit Ethernet Q Performance Processor (QPP)

- 10-Gigabit Ethernet

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

- Internal Ethernet interface, which connects the Routing Engine to the Packet Forwarding Engine

- 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 router:

- Configure Ethernet Physical Interface Properties on page 264

- Configure 802.1Q VLANs on page 282

- Configure TCC and Layer 2.5 Switching on page 286

- Configure Static ARP Table Entries on page 289

- Configure VRRP on page 290

- Configure the Management Ethernet Interface on page 298

- Configure the Internal Ethernet Interface on page 299

- Configure Aggregated Ethernet Interfaces on page 299

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

Example: Configure Fast Ethernet Interfaces on page 301

Example: Configure Gigabit Ethernet Interfaces on page 301

Example: Configure Aggregated Ethernet Interfaces on page 302

Configure 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 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 `gigether-options` statement at the `[edit interfaces ge-fpc/pic/port]` hierarchy level:

```
[edit interfaces ge-fpc/pic/port]
gigether-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 QPP-specific physical interface properties, include the `gigether-options` statement at the `[edit interfaces ge-fpc/pic/port]` hierarchy level and the `input-vlan-map` and `output-vlan-map` statements at the `[edit interfaces ge-fpc/pic/port unit logical-unit-number]` hierarchy level:

```
[edit interfaces ge-fpc/pic/port]
gigether-options {
  802.3ad aex;
  (flow-control | no-flow-control);
  (loopback | no-loopback);
  ethernet-switch-profile {
    ethernet-policer-profile {
      ieee802.1-priority-map premium [ bits ];
    }
    policer cos-policer-name {
      aggregate {
        bandwidth-limit rate;
        bandwidth-percent percent;
        burst-size-limit length;
      }
      premium {
        bandwidth-limit rate;
        bandwidth-percent percent;
        burst-size-limit length;
      }
    }
  }
  (mac-learn-enable | no-mac-learn-enable);
  tag-protocol-id [ tpid ];
}

[edit interfaces ge-fpc/pic/port unit logical-unit-number]
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;
}
```

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);
  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, Gigabit Ethernet QPP, and 10-Gigabit Ethernet interfaces:

Configure Ethernet Link Aggregation on page 266

Configure Gratuitous ARP on page 281

Configure Aggregated Ethernet Link Speed on page 267

Configure Aggregated Ethernet Minimum Links on page 267

Configure Gigabit Ethernet QPP Interfaces on page 267

Configure Ethernet MAC Address Filtering on page 279

Configure Ethernet Loopback Capability on page 280

Configure Flow Control on page 280

Configure the Link Characteristics on page 281

Configure the Interface Speed on page 282

Configure the Ingress Rate Limit on page 282

Configure 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 range from 0 through 15, for a total of 16 aggregated interfaces. You must also include a statement defining `aex` at the `[edit interfaces]` hierarchy level. For more information, see “Configure Aggregated Ethernet Interfaces” on page 299. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configure Ethernet Physical Interface Properties” on page 264, and for a sample configuration, see “Example: Configure Aggregated Ethernet Interfaces” on page 302.

Configure 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).

Configure 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, the minimum number of links is one. The minimum number of links can be from one through eight.

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;
```

Configure Gigabit Ethernet QPP Interfaces

The one-port and two-port Gigabit Ethernet PICs with QPP are primarily used for virtual metropolitan-area network (VMAN) aggregation and ISP peering applications. The Gigabit Ethernet PIC with QPP requires an enhanced FPC.

Table 22 lists the capabilities of the Gigabit Ethernet PIC with QPP.

Table 22: Capabilities of the Gigabit Ethernet PIC with QPP

Layer 2	
802.3ad link aggregation	Not applicable
Maximum VLANs per port	384
MTU size	9192
MAC filters:	
Destinations per port	960
Sources per port	64
Layer 2 VPNs	
VLAN CCC	Yes
Port-based CCC	Yes
Extended VLAN CCC VMANs Tag Protocol	Yes
CoS	
PIC-based egress queues	Yes

On Gigabit Ethernet QPP interfaces, you can perform the following tasks:

Configure a Gigabit Ethernet QPP Policer Profile on page 269

Configure Gigabit Ethernet QPP MAC Address Filtering on page 270

Configure Gigabit Ethernet QPP MAC Address Accounting on page 272

Stack and Rewrite Gigabit Ethernet QPP VLAN Tags on page 273

For example configurations, see the following sections:

Example: Configure Gigabit Ethernet QPP MAC Address Filtering on page 271

Examples: Stack and Rewrite Gigabit Ethernet QPP VLAN Tags on page 274

You can also configure class of service (CoS) on logical QPP interfaces. For more information, see “Associate a Scheduler with a DLCI or VLAN on a Channelized QPP Interface” on page 597.

Configure a Gigabit Ethernet QPP Policer Profile

On Gigabit Ethernet QPP interfaces, you can define rate limits for premium and aggregate traffic received on the interface. These policers allow you to perform simple traffic policing on Gigabit Ethernet QPP interfaces without configuring a firewall filter. First you configure the Ethernet policer profile, then you can apply the policer to an interface. For information about applying the policer to an interface, see “Configure Gigabit Ethernet QPP MAC Address Filtering” on page 270.

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

```
[edit interfaces interface-name gigether-options ethernet-switch-profile]
ethernet-policer-profile {
  ieee802.1-priority-map premium [ bits ];
  policer cos-policer-name {
    aggregate {
      bandwidth-limit rate;
      bandwidth-percent percent;
      burst-size-limit length;
    }
    premium {
      bandwidth-limit rate;
      bandwidth-percent percent;
      burst-size-limit length;
    }
  }
}
```



Note

When configuring the bandwidth, use the bandwidth-limit statement and not the bandwidth-percent statement. If you include the bandwidth-percent statement at the [edit interfaces *interface-name* gigether-options ethernet-policer-profile policer (aggregate | premium)] hierarchy level, it currently has no effect.

In the Ethernet policer profile, the aggregate-priority policer is mandatory; the premium-priority policer is optional. If you include a premium-priority policer, you can specify premium IEEE 802.1p bits by including the ieee802.1-priority-map statement at the [edit interfaces *interface-name* gigether-options ethernet-policer-profile] hierarchy level:

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

Specify values of the code-point bits, in binary code. The remaining bits are classified as nonpremium (or aggregate).

To configure rate limiting for premium and aggregate 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.

You can rate-limit based on port speed. You specify this port speed as a bandwidth percentage in a policer. The percentage must be a whole decimal number between 1 and 100.

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.

Configure Gigabit Ethernet QPP MAC Address Filtering

On Gigabit Ethernet QPP interfaces, you can enable MAC address filtering to accept packets from specific MAC addresses, and then apply a policer to the accepted packets. To configure MAC address filtering, include the `accept-source-mac` statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
accept-source-mac {
  mac-address mac-address {
    policer {
      input policer-name;
      output policer-name;
    }
  }
}
```

You can specify the MAC address as *nn:nn:nn:nn:nn:nn* or *nnnn.nnnn.nnnn*, where *n* is a hexadecimal number. To specify more than one address, include multiple `mac-address` statements at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

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 router's MAC database. To view the router's MAC database, enter the `show interfaces mac-database interface-name` command:

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

Optionally, you can apply input and output policers that define rate limits for premium and aggregate traffic received on the interface. These policers allow you to perform simple traffic policing on Gigabit Ethernet QPP interfaces without configuring a firewall filter. For information about defining these policers, see "Configure a Gigabit Ethernet QPP Policer Profile" on page 269.

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.

You cannot define traffic with specific MAC addresses to be rejected; however, you can block all incoming packets that do not have a source address specified at the [edit interfaces *interface-name* unit *logical-unit-number* accept-source-mac] hierarchy level. To enable this blocking, include the source-filtering statement at the [edit interfaces *interface-name* gigeother-options] hierarchy level:

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

For more information, see “Configure Ethernet MAC Address Filtering” on page 279.

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

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

Example: Configure Gigabit Ethernet QPP MAC Address Filtering

Configure interface ge-6/0/0 to treat priority levels 2 and 3 as premium. On ingress, this means that IEEE 802.1p priority values 2 and 3 are premium. On egress, it means traffic classified into Queue 1 is 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 router 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 {
  gigeother-options {
    ether-switch-profile {
      ether-policer-profile {
        ieee802.1-priority-map {
          premium [ 2 3 ];
        }
      }
      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;
      }
    }
  }
}

```

Configure Gigabit Ethernet QPP MAC Address Accounting

For Gigabit Ethernet QPP interfaces only, 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* `gigether-options` `ethernet-switch-profile`] hierarchy level:

```

[edit interfaces interface-name gigether-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* `gigether-options` `ethernet-switch-profile`] hierarchy level:

```

[edit interfaces interface-name gigether-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.

Stack and Rewrite Gigabit Ethernet QPP VLAN Tags

On Gigabit Ethernet QPP 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) routers that share one VLAN ID.

To stack and rewrite VLAN tags, include the `input-vlan-map` and `output-vlan-map` statements at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level:

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

To stack a VLAN tag on top of all tagged frames entering or exiting the interface, include the `push`, `vlan-id`, and `tag-protocol-id` statements at the `[edit interfaces interface-name unit logical-unit-number input-vlan-map]` or `[edit interfaces interface-name unit logical-unit-number output-vlan-map]` hierarchy level.

To remove a VLAN tag from all tagged frames entering or exiting the interface, include the `pop` statement at the `[edit interfaces interface-name unit logical-unit-number input-vlan-map]` or `[edit interfaces interface-name unit logical-unit-number output-vlan-map]` hierarchy level.

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 and output VLAN maps are stacked on top of the VLAN ID you define at the `[edit interfaces interface-name unit logical-unit-number vlan-id number]` hierarchy level.

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 together-options ethernet-switch-profile ethernet-policer-profile]` hierarchy level:

```
[edit interfaces interface-name together-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 together-options ethernet-switch-profile tag-protocol-id [tpids]]` hierarchy level.

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 at the `[edit interfaces interface-name unit logical-unit-number input-vlan-map]` hierarchy level.

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 at the [edit interfaces *interface-name* unit *logical-unit-number* output-vlan-map] hierarchy level. The `swap` operation works on the outer tag only, independent of whether you include the `stacked-vlan-tagging` statement in the configuration. If you include the `swap` statement in the configuration, the VLAN ID in outgoing frames is rewritten to the VLAN ID you configure at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level.

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. To configure dual VLAN tags on a logical interface, include the `vlan-tag` statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]  
vlan-tag [ tpid.vlan-id ];
```

Configure the outer tag first, then the inner tag. The outer tag VLAN ID range is 1 through 511 for normal interfaces, and 512 and above for VLAN CCC interfaces. The inner tag does not have this restriction. If you configure stacked VLAN tagging, you can configure a maximum of two TPID VLAN ID pairs.

Examples: Stack and Rewrite Gigabit Ethernet QPP 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.

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;
  gigeather-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;
  }
}

```

Swap a VLAN ID on ingress

```

[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  gigeather-options {
    ethernet-switch-profile {
      tag-protocol-id Ox9100;
    }
  }
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 1000;
    input-vlan-map {
      swap;
      tag-protocol-id Ox9100;
      vlan-id 2000;
    }
  }
}
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;
  }
}

```

Swap a VLAN ID on egress

```

[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;
  gigheter-options {
    ethernet-switch-profile {
      tag-protocol-id 0x8800;
    }
  }
  ...
  unit 1 {
    encapsulation vlan-ccc;
    vlan-id 2000;
    output-vlan-map {
      swap;
      tag-protocol-id 0x8800; # No egress VLAN ID rewrite support; only TPID rewrite
    }
  }
}

[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;
  }
}

```

Swap a VLAN ID on both ingress and egress

```

[edit interfaces]
ge-4/0/0 {
  vlan-tagging;
  encapsulation vlan-ccc;
  gigether-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;
  gigether-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;
  }
}

```

Configure Ethernet MAC Address Filtering

By default, source address filtering is disabled. On aggregated Ethernet, Fast Ethernet, Gigabit Ethernet, 10-Gigabit Ethernet, and Gigabit Ethernet QPP interfaces, you can enable source address filtering, which blocks all incoming packets to that interface. To enable the filtering, include the source-filtering statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* ggether-options] hierarchy level:

```
source-filtering;
```

This method of MAC address filtering is not supported on Gigabit Ethernet QPP interfaces. For more information, see “Configure Gigabit Ethernet QPP MAC Address Filtering” on page 270.

To explicitly disable filtering, include the no-source-filtering statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* ggether-options] hierarchy level:

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

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

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

For Gigabit Ethernet QPP interfaces, you do this by including the accept-source-mac statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
accept-source-mac {
  mac-address mac-address {
    policer {
      input policer-name;
      output policer-name;
    }
  }
}
```

For more information, see “Configure Gigabit Ethernet QPP MAC Address Filtering” on page 270.

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

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



Note

Support for source address filters is limited on the Fast Ethernet 12-port and 48-port PIC interfaces.

Configure 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 at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
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 at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
no-loopback;
```

Configure Flow Control

By default, the router 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 router to permit unrestricted traffic. To disable flow control, include the no-flow-control statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

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

To explicitly reinstate flow control, include the flow-control statement at the [edit interfaces *interface-name* aggregated-ether-options], [edit interfaces *interface-name* fastether-options], or [edit interfaces *interface-name* gigether-options] hierarchy level:

```
flow-control;
```

Configure the Link Characteristics

By default, the router's management Ethernet interface, `fxp0`, autonegotiates whether to operate in full-duplex or half-duplex mode. Fast Ethernet interfaces can operate in either full-duplex or half-duplex mode, and all other interfaces can operate only in full-duplex mode. For Gigabit Ethernet 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);
```

Configure Gratuitous ARP

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

By default, the router 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 router receives a gratuitous ARP reply, the router 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 router. 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;
```

Configure 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);
```

Configure 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.

Configure 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:

Enable VLAN Tagging on page 283

Configure VLAN CCC or VPLS Encapsulation on page 284

Configure Extended VLAN CCC or VLAN VPLS Encapsulation on page 285

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

Example: Configure VLAN CCC or VPLS Encapsulation on page 284

Example: Configure Extended VLAN CCC or VLAN VPLS Encapsulation on page 285

Enable VLAN Tagging

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

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

Gigabit Ethernet interfaces can be partitioned; you can assign up to 4095 different logical interfaces, one for each VLAN, but you are limited to a maximum of 1024 VLANs on any single Gigabit Ethernet or 10-Gigabit Ethernet port; and you are limited to a maximum of 384 (383 for tagged and 1 for untagged VLANs) on any single Gigabit Ethernet QPP port. Fast Ethernet interfaces can also be partitioned, with a maximum of 1024 logical interfaces for the 4-port Fast Ethernet PIC, and 16 logical interfaces for the 8-port, 12-port, and 48-port Fast Ethernet PICs. Table 23 lists VLAN ID range by interface type.

Table 23: 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 QPP	1 through 4094
10-Gigabit Ethernet	1 through 4094
Management and internal Ethernet interfaces	1 through 1023

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 at the `[edit interfaces interface-name unit logical-unit-number]` hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]  
vlan-id number;
```



Note

Because 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 about IS-IS capabilities, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*.

Configure VLAN CCC or VPLS Encapsulation

Ethernet interfaces with VLAN tagging enabled can use VLAN circuit cross-connect (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 `vlan-ccc` or `vlan-vpls`:

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

Ethernet interfaces in VLAN mode can have multiple logical interfaces, but in CCC and VPLS modes VLAN IDs from 1 through 511 are reserved for normal VLANs, and VLAN IDs 512 and up are reserved for CCC VLANs.

In general, you configure an interface's encapsulation at the [edit interfaces *interface-name*] hierarchy level. However, for some encapsulation types, including 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 at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

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

You cannot configure a logical interface with VLAN CCC or VLAN VPLS encapsulation unless you also configure the physical device with the same encapsulation. The logical interface must have a VLAN ID in the range from 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.

Example: Configure VLAN CCC or VPLS 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;
  }
}
```

Configure Extended VLAN CCC or VLAN VPLS Encapsulation

Gigabit Ethernet and four-port Fast Ethernet interfaces with VLAN tagging enabled can use extended VLAN circuit cross-connect (CCC) or VLAN Virtual Private LAN Service (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: Configure Extended VLAN CCC or VLAN VPLS 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;
  }
}
```

Configure 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, circuit cross-connect (CCC). However, while CCC requires the same Layer 2 encapsulations on both sides of a router (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:

Configure Extended VLAN TCC Encapsulation on page 286

Configure an Ethernet TCC or Extended VLAN TCC on page 287

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

Example: Configure an Ethernet TCC or Extended VLAN TCC on page 287

Example: Configure Extended VLAN CCC or VLAN VPLS Encapsulation on page 285

Configure Extended VLAN TCC Encapsulation

One-port Gigabit Ethernet, two-port Gigabit Ethernet, and four-port Fast Ethernet PICs with VLAN tagging enabled can use extended VLAN TCC encapsulation, which allows circuits to have different media on either side of the connection. 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;
```

For extended 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 four-port Gigabit Ethernet PICs.

Configure an Ethernet TCC or Extended VLAN TCC

For Layer 2.5 VPNs employing an Ethernet interface as the TCC router, you can configure an Ethernet TCC or an extended VLAN TCC.

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

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

To configure an Ethernet TCC or an extended VLAN TCC, include the proxy and remote statements at the [edit interfaces *interface-name* unit *logical-unit-number* family tcc] hierarchy level:

```
[edit interfaces interfaces interface-name unit logical-unit-number family tcc]
  proxy {
    inet-address address;
  }
  remote {
    (inet-address address | mac-address address);
  }
```

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

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

Ethernet TCC is supported on interfaces that carry IPv4 traffic only. Ethernet TCC encapsulation is supported on one-port Gigabit Ethernet, two-port Gigabit Ethernet, four-port Gigabit Ethernet, and four-port Fast Ethernet PICs only.

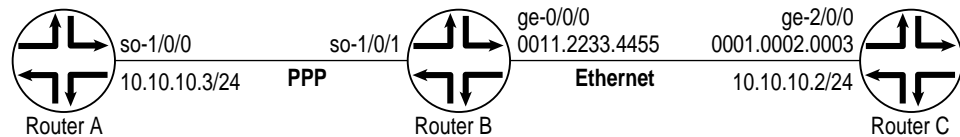
Example: Configure an Ethernet TCC or Extended VLAN TCC

Configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks router, Router B, as the TCC interface. Ethernet TCC encapsulation provides an Ethernet wide area circuit for interconnecting IP traffic. (See the topology in Figure 21.)

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 21: 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 10.10.10.3/24;
        }
        remote {
          inet-address 10.10.10.2/24;
        }
      }
    }
  }

```

Configure an Extended VLAN TCC Configure a full-duplex Layer 2.5 translational cross-connect between Router A and Router C, using a Juniper Networks router, 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 21).

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;
        }
      }
    }
  }

```

Configure 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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
  arp ip-address (mac | multicast-mac) mac-address <publish>;
```

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 router replies to proxy ARP requests.



Caution

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 “Apply Policers” on page 87.

Example: Configure Static ARP Table Entries

Configure two static ARP table entries on the router’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;
      }
    }
  }
}
```

Configure 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 routers on that LAN without requiring more than the static configuration of a single default route on the hosts. The VRRP routers share the IP address corresponding to the default route configured on the hosts. At any time, one of the VRRP routers 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 router and allowing traffic on the LAN to be routed without relying on a single router.

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

To configure VRRP, include the `vrp-group` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
vrp-group group-number {
  (accept-data | no-accept-data);
  advertise-interval seconds;
  authentication-key key;
  authentication-type authentication;
  (preempt | no-preempt);
  priority number;
  track {
    interface interface-name priority-cost cost;
    virtual-address [ addresses ];
  }
}
```

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 “Trace VRRP Operations” on page 295.

You can configure the following VRRP properties:

- Configure Basic VRRP Support on page 291
- Configure VRRP Authentication on page 292
- Configure the Advertisement Interval for the VRRP Master Router on page 293
- Configure a Backup Router to Preempt the Master Router on page 293
- Accept Packets Destined for the Virtual IP Address on page 294
- Configure a Logical Interface to Be Tracked on page 295
- Trace VRRP Operations on page 295

For a VRRP configuration example, see “Example: Configure VRRP” on page 296.

Configure Basic VRRP Support

To configure basic VRRP support, configure VRRP groups on interfaces by including the following statements at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address]
  vrrp-group group-number {
    priority number;
    virtual-address [ addresses ];
  }
```

An interface can be a member of one or more VRRP groups. On a single router, you cannot configure the same VRRP group on multiple interfaces. 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 “Configure Ethernet MAC Address Filtering” on page 279, 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 routers 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 routers 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 router'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 router 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 router with the highest priority within the group becomes the master router.

Within a single VRRP group, the master and backup routers cannot be the same router.

Configure VRRP Authentication

All VRRP protocol exchanges can be authenticated to guarantee that only trusted routers 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 router 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 router 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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

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

authentication can be none, simple, or md5. The authentication type must be the same for all routers 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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

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

The key (password) is an ASCII string. For simple authentication, it can be 1 through 8 characters long. For MD-5 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 routers in the VRRP group.

Configure 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.

To modify the time between the sending of VRRP advertisement packets, include the advertise-interval statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

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

The interval can range from 1 through 255 seconds. The interval must be the same for all routers in the VRRP group.

Configure 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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

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

To prohibit a higher priority backup router from preempting a lower priority master router, include the no-preempt statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
no-preempt;
```



The router that owns the IP address(es) associated with the virtual router always preempts, independent of the setting of this flag.

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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

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

To prohibit the interface from accepting packets destined for the virtual IP address, include the `no-accept-data` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
no-accept-data;
```

The `accept-data` statement has the following consequences:

You do not need to include the `accept-data` statement to activate this feature if the master router owns the virtual IP address.

If you do not include the `accept-data` statement, and if the master router owns the virtual IP address, the master router responds to ICMP message requests only.

You cannot include the `accept-data` statement when the priority of the master router is set to 255.

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 router 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.

Configure 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 at the [edit interfaces *interface-name* unit *logical-unit-number* family inet address *address* vrrp-group *group-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet address address
vrrp-group group-number]
track {
    interface interface-name priority-cost cost;
}
```

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 range 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.

Trace 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 1MB, 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: Configure VRRP

Configure one master (Router A) and one backup (Router B) router. Note that the address configured in the virtual-address statements differs from the addresses configured in the address statements.

```
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;
          }
        }
      }
    }
  }
}
```

When configuring multiple VRRP groups on an interface, configure one to be the master virtual router for that 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;
          }
        }
      }
    }
  }
}
```

Configure 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 {
  together-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;
        }
      }
    }
  }
}
```

Configure the Management Ethernet Interface

The router'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;
      }
    }
  }
}
```



Caution

The management Ethernet interface must be configured for the router to function.

Configure the MAC Address on the Management Ethernet Interface

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

Configure the Internal Ethernet Interface

The internal Ethernet interface, fxp1, connects the Routing Engine with the System Control Board (SCB), System and Switch Board (SSB), Forwarding Engine Board (FEB), or Switching and Forwarding Module (SFM), depending on router model, in the Packet Forwarding Engine. The router software automatically configures this interface.



Caution

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

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

Configure 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 Internet Software Guide: Routing and Routing Protocols*.



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.

The JUNOS software does not support the Link Aggregation Control Protocol (LACP).

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 either Fast Ethernet, Gigabit Ethernet, or 10-Gigabit Ethernet devices; however, do not use a combination of these within the same aggregated link.

To specify aggregated Ethernet interfaces, include the `vlan-tagging` statement at the `[edit interfaces aex]` hierarchy level and include the `vlan-id` statement at the `[edit interfaces aex unit logical-unit-number]` hierarchy level, as in the following example:

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

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 interfaces is 16, and the assigned number can range from 0 through 15. For information about configuring aggregated devices, see the *JUNOS Internet Software Configuration Guide: Getting Started*.

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 “Configure Ethernet Link Aggregation” on page 266. You can optionally specify other physical properties that apply specifically to the aggregated Ethernet interfaces; for details, see “Configure Ethernet Physical Interface Properties” on page 264. For a sample configuration, see “Example: Configure Aggregated Ethernet Interfaces” on page 302.

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: Configure 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: Configure 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, T320, and T640 two-port Gigabit Ethernet PIC supports two independent Gigabit Ethernet links. This PIC is supported on the M160, T320, and T640 platforms only and it requires a Type 2 M160, Type 2 T320, or Type 2 T640 FPC.

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: Configure 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;
}
```