

# Chapter 7

## Configure Protocol Family and Address Interface Properties

For each logical interface, you must configure one or more protocol families. You can also configure interface address properties. To do this, you can include the following statements at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number]
family family {
  accounting {
    destination-class-usage;
    source-class-usage {
      (input | output | [input output]);
    }
  }
}
bundle (ml-fpc/pic/port | ls-fpc/pic/port);
filter {
  input filter-name;
  output filter-name;
  group filter-group-number;
}
ipsec-sa sa-name;
keep-address-and-control;
mtu bytes;
multicasts-only;
no-asynchronous-notification;
no-redirects;
policer {
  arp policer-template-name;
  input policer-template-name;
  output policer-template-name;
}
primary;
proxy inet-address address;
remote (inet-address address | mac-address address);
rpf-check <fail-filter filter-name> {
  <mode loose>;
}
sampling {
  [ input output ];
}
(translate-discard-eligible | no-translate-discard-eligible);
(translate-fecn-and-becn | no-translate-fecn-and-becn);
```

```

address address {
  arp ip-address (mac | multicast-mac) mac-address <publish>;
  destination destination-address;
  eui-64;
  broadcast address;
  multipoint-destination destination-address (dlci dlcI-identifier | vci vci-identifier);
  multipoint-destination destination-address {
    epd-threshold cells;
    inverse-arp;
    oam-liveness {
      up-count cells;
      down-count cells;
    }
    oam-period seconds;
    shaping {
      (cbr rate | rtvbr peak rate sustained rate burst length |
        vbr peak rate sustained rate burst length);
      queue-length number;
    }
    vci vpi-identifier.vci-identifier;
  }
  preferred;
  primary;
  vrrp-group group-number {
    virtual-address [addresses];
    priority number;
    (accept-data | no-accept-data);
    advertise-interval seconds;
    authentication-type authentication;
    authentication-key key;
    (preempt | no-preempt);
    track {
      interface interface-name priority-cost cost;
    }
  }
}
}

```

This chapter describes the configuration of the interface protocol and address properties:

Configure the Protocol Family on page 79

Configure the Interface Address on page 81

Configure an Unnumbered Interface on page 82

Set the Protocol MTU on page 83

Configure Default, Primary, and Preferred Addresses and Interfaces on page 84

Disable the Sending of Redirect Messages on an Interface on page 84

Configure Default, Primary, and Preferred Addresses and Interfaces on page 84

Apply Policers on page 87

Apply a Filter to an Interface on page 88

Configure Unicast RPF on page 89

Enable Source Class and Destination Class Usage on page 94

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

## Configure the Protocol Family

For each logical interface, you can configure one or more of the following protocols that run on the interface:

`ccc`—Circuit Cross-Connect (CCC). You can configure this protocol family for the logical interface of CCC physical interfaces. When you use this encapsulation type, you can configure the family `ccc` only.

`inet`—IP (Internet Protocol). You must configure this protocol family for the logical interface to support IP protocol traffic, including OSPF, BGP, and ICMP.

`inet6`—IP (Internet Protocol) version 6. You must configure this protocol family for the logical interface to support IPv6 protocol traffic, including RIPng, IS-IS, and BGP. For more information about IPv6, see “IPv6 Introduction” on page 80.

`iso`—ISO. You must configure this protocol family for the logical interface to support IS-IS traffic.

`mlfr-uni-nni`—Multilink Frame Relay (MLFR) FRF.16 user-to-network network-to-network (UNI NNI) . You must configure this protocol or `mlfr-end-to-end` for the logical interface to support link services bundling.

`mlfr-end-to-end`—Multilink Frame Relay End-to-End. You must configure this protocol or `MLPPP` for the logical interface to support multilink bundling.

`mlppp`—Multilink Point-to-Point Protocol (MLPPP). You must configure this protocol (or `mlfr-end-to-end`) for the logical interface to support multilink bundling.

`mpls`—Multiprotocol Label Switching (MPLS). You must configure this protocol family for the logical interface to participate in an MPLS path.

`tcc`—Translational Cross-Connect (TCC). You can configure this protocol family for the logical interface of TCC physical interfaces.

`tnp`—Trivial Network Protocol. This protocol is used to communicate between the Routing Engine and the System Control Board (SCB), System and Switch Board (SSB), Forwarding Engine Board (FEB), or System and Forwarding Module (SFM), depending on router model, in the router’s Packet Forwarding Engine. The JUNOS software automatically configures this protocol family on the router’s internal interfaces only, as discussed in “Configure the Internal Ethernet Interface” on page 299.

`vpls`—Virtual Private LAN Service (VPLS). You can optionally configure this protocol family for the logical interface on which you configure VPLS. VPLS provides an Ethernet-based point-to-multipoint Layer 2 VPN to connect customer edge (CE) routers across an MPLS backbone. When you configure a VPLS encapsulation type, the family `vpls` statement is assumed by default. For more information about VPLS, see the *JUNOS Internet Software Configuration Guide: VPNs* and the *JUNOS Internet Software Feature Guide*.

To configure the logical interface's protocol family, include the family statement at the [edit interfaces *interface-name* unit *logical-unit-number*] hierarchy level, specifying the selected family. To configure more than one protocol family on a logical interface, include multiple family statements. Following is the minimum configuration:

```
[edit interfaces interface-name unit logical-unit-number]
family family {
  mtu size;
  multicasts-only;
  no-redirects;
  primary;
  address address {
    destination address;
    broadcast address;
    preferred;
    primary;
  }
}
```

## IPv6 Introduction

IPv4 has been widely deployed and used to network the Internet today. With the rapid growth of the Internet, enhancements to IPv4 are needed to support the influx of new subscribers, Internet-enabled devices, and applications. IPv6 is designed to enable the global expansion of the Internet.

IPv6 builds upon the functionality of IPv4, providing improvements to addressing, configuration and maintenance, and security.

IPv6 is defined in the following documents:

RFC 2460, *Internet Protocol, Version 6 (IPv6)*

RFC 2373, *IP Version 6 Addressing Architecture*

## IPv4-to-IPv6 Transition

Implementing IPv6 requires a transition mechanism to allow interoperability between IPv6 nodes (both routers and hosts) and IPv4 nodes. The transition mechanism is the key factor in the successful deployment of IPv6. Because millions of IPv4 nodes already exist, upgrading every node to IPv6 at the same time is not feasible.

As a result, transition from IPv4 to IPv6 happens gradually, allowing nodes to be upgraded independently and without disruption to other nodes. While a gradual upgrade occurs, compatibility between IPv6 and IPv4 nodes becomes a requirement. Otherwise, an IPv6 node would not be able to communicate with an IPv4 node.

Transition mechanisms allow IPv6 and IPv4 nodes to coexist together in the same network, and make gradual upgrading possible. The transition mechanism supported by the JUNOS Internet software is tunneling. Tunnels allow IPv6 packets to be encapsulated into IPv4 headers and sent across an IPv4 infrastructure. For more information about configuring tunnels to support IPv4-to-IPv6 transition, see "Configure an IPv6-over-IPv4 Tunnel" on page 412.

## Configure the Interface Address

You assign an address to an interface by specifying the address when configuring the protocol family. For the inet family, you configure the interface's IP address. For the iso family, you configure one or more addresses for the loopback interface. For the ccc, tcc, mpls, tnp, and vpls families, you never configure an address.

To assign an address to an interface, include the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
address address {
  destination address;
  eui-64;
  broadcast address;
  preferred;
  primary;
}
```

In the address statement, specify the network address of the interface.

For each address, you can optionally configure one or more of the following:

Address of the remote side of the connection (for point-to-point interfaces only)—Specify this in the destination statement.

Whether the router automatically generates the host number portion of interface addresses—The eui-64 statement applies only to interfaces that carry IPv6 traffic, where the prefix length of the address is 64 bits or less, and the low-order 64 bits of the address are zero. This option does not apply to the loopback interface (lo0) because IPv6 addresses configured on the loopback interface must have a 128-bit prefix length.

Broadcast address for the interface's subnet—Specify this in the broadcast statement; this applies only to Ethernet interfaces, such as the management interface fxp0, the Fast Ethernet interface, and the Gigabit Ethernet interface.

Whether this address is the preferred address—Each subnet on an interface has a preferred local address. If you configure more than one address on the same subnet, the preferred local address is chosen by default as the source address when you originate packets to destinations on the subnet. For more information about preferred addresses, see “Configure Default, Primary, and Preferred Addresses and Interfaces” on page 84.

By default, the preferred address is the lowest numbered address on the subnet. To override the default and explicitly configure the preferred address, include the preferred statement when configuring the address.

Whether this address is the primary address—Each interface has a primary local address. If an interface has more than one address, the primary local address is used by default as the source address when you originate packets out the interface where the destination gives no hint about the subnet (for example, some ping commands). For more information about primary addresses, see “Configure Default, Primary, and Preferred Addresses and Interfaces” on page 84.

By default, the primary address on an interface is the lowest numbered non-127 preferred address on the interface. To override the default and explicitly configure the preferred address, include the primary statement when configuring the address.

## Configure the IPv6 Address on an Interface

You represent IPv6 addresses in hexadecimal notation using a colon-separated list of 16-bit values.

You assign a 128-bit IPv6 address to an interface by including the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet6] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family inet6]
address aaaa:bbbb:...:zzzz/nn;
```

The double colon (::) represents all bits set to 0, as shown in the following example:

```
interfaces fe-0/0/1 {
  unit 0 {
    family inet6 {
      address fec0:1:1:1::2/64;
    }
  }
}
```

## Configure an Unnumbered Interface

When you need to conserve IP addresses, you can configure unnumbered interfaces. To do this, configure the protocol family, but do not include the address statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

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

For example:

```
[edit]
interfaces {
  so-6/1/0 {
    unit 0 {
      family inet;
      family iso;
    }
  }
}
```

When configuring unnumbered interfaces, you must ensure that a source address is configured on some interface in the router. This address is the default address. We recommend that you do this by assigning an address to the loopback interface (lo0), as described in “Configure the Loopback Interface” on page 313. If you configure an address (other than a martian) on the lo0 interface, that address is always the default address, which is preferable because the loopback interface is independent of any physical interfaces and therefore is always accessible.

## Set the Protocol MTU

For each interface, you can configure an interface-specific MTU by including the `mtu` statement at the [edit interfaces *interface-name*] hierarchy level. If you need to modify this MTU for a particular protocol family, include the `mtu` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
mtu bytes;
```

The default protocol MTU is 4470 bytes for ATM PVC, Cisco HDLC, Frame Relay, and PPP encapsulations. For Ethernet encapsulation on IPv4, the default protocol MTU is 1500 bytes. For Ethernet encapsulation on ISO, the default protocol MTU is 1497 bytes.



**Note**

When you initially configure an interface, the protocol MTU is calculated automatically. However, if you subsequently change the media MTU, the protocol MTU on existing address families does not automatically adjust.

If you increase the size of the protocol MTU, you must ensure that the size of the media MTU is equal to or greater than the sum of the protocol MTU and the encapsulation overhead. If you reduce the media MTU size, but there are already one or more address families configured and active on the interface, you must also reduce the protocol MTU size. (You configure the media MTU by including the `mtu` statement at the [edit interfaces *interface-name*] hierarchy level, as discussed in “Configure the Media MTU” on page 47.)

For Ethernet encapsulation when the family is `mpls`, the default protocol MTU is 1500 bytes, including 4 to 12 bytes of overhead. The maximum number of DLCIs is determined by the MTU on the interface. If you have keepalives enabled, the maximum number of DLCIs is 1000, with the MTU set to 5012.



**Note**

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

## Disable the Removal of Address and Control Bytes

For PPP CCC-encapsulated interfaces, the address and control bytes are removed by default before the packet is encapsulated into a tunnel.

You can disable the removal of address and control bytes. To do this, include the `keep-address-and-control` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *ccc*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family ccc]
  keep-address-and-control;
```

When you use PPP CCC encapsulation, and one PE router is running JUNOS Software Release 5.7 or earlier and the other PE router is running JUNOS Software Release 6.0 or later, you must include the `keep-address-and-control` statement in the configuration of the PE router running JUNOS Software Release 6.0 or later.

## Disable the Sending of Asynchronous Notification Upon Link Failure

For PPP- and Cisco HDLC-encapsulated serial interfaces on T-series platforms only, you can disallow sending of asynchronous notification upon link failure. To do this, include the `no-asynchronous-notification` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (*ccc* | *tcc*)] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family (ccc | tcc)]
  no-asynchronous-notification;
```

## Disable the Sending of Redirect Messages on an Interface

By default, the interface sends protocol redirect messages. To disable the sending of these messages on an interface, include the `no-redirects` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family family]
  no-redirects;
```

To disable the sending of protocol redirect messages for the entire router, include the `no-redirects` statement at the [edit system] hierarchy level.

## Configure Default, Primary, and Preferred Addresses and Interfaces

The router has a default address and a primary interface, and interfaces have primary and preferred addresses.

The *default address* of the router is used as the source address on unnumbered interfaces. The routing protocol process tries to pick the default address as the router ID, which is used by protocols, including OSPF and IBGP.

The *primary interface* for the router is the interface that packets go out when no interface name is specified and when the destination address does not imply a particular outgoing interface.

An interface's *primary address* is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. An interface's *preferred address* is the default local address used for packets sourced by the local router to destinations on the subnet.

The default address of the router is chosen using the following sequence:

1. The primary address on the loopback interface lo0 that is not 127.0.0.1 is used.
2. The primary address on the primary interface is used.

To configure these addresses and interfaces, you can do the following:

Configure the Primary Interface for the Router on page 85

Configure the Primary Address for an Interface on page 86

Configure the Preferred Address for an Interface on page 86

## **Configure the Primary Interface for the Router**

The *primary interface* for the router has the following characteristics:

It is the interface that packets go out when you type a command such as ping 255.255.255.255—that is, a command that does not include an interface name (there is no interface *type-0/0/0.0* qualifier) and where the destination address does not imply any particular outgoing interface.

It is the interface on which multicast applications running locally on the router, such as SAP, do group joins by default.

It is the interface from which the default local address is derived for packets sourced out an unnumbered interface if there are no non-127 addresses configured on the loopback interface, lo0.

By default, the multicast-capable interface with the lowest-index address is chosen as the primary interface. If there is no such interface, the point-to-point interface with the lowest index address is chosen. Otherwise, any interface with an address could be picked. In practice, this means that, on the router, the fxp0 interface is picked by default.

To configure a different interface to be the primary interface, include the primary statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family*] hierarchy level:

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

## Configure the Primary Address for an Interface

The *primary address* on an interface is the address that is used by default as the local address for broadcast and multicast packets sourced locally and sent out the interface. For example, the local address in the packets sent by a ping interface `so-0/0/0.0 255.255.255.255` command is the primary address on interface `so-0/0/0.0`. The primary address flag also can be useful for selecting the local address used for packets sent out unnumbered interfaces when multiple non-127 addresses are configured on the loopback interface, `lo0`. By default, the primary address on an interface is selected as the numerically lowest local address configured on the interface.

To set a different primary address, include the primary statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

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

## Configure the Preferred Address for an Interface

The *preferred address* on an interface is the default local address used for packets sourced by the local router to destinations on the subnet. By default, the numerically lowest local address is chosen. For example, if the addresses `128.100.1.1/24`, `128.100.1.2/24`, and `128.100.1.3/24` are configured on the same interface, the preferred address on the subnet (by default, `128.100.1.1`) would be used as a local address when you issue a ping `128.100.1.5` command.

To set a different preferred address for the subnet, include the preferred statement at the [edit interfaces *interface-name* unit *logical-unit-number* family *family* address *address*] hierarchy level:

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

## Apply Policers

Policies allow you to perform simple traffic policing on specific interfaces or Layer 2 VPNs without configuring a firewall filter. To apply policies, include the policy statement when configuring the logical interface at the [edit interfaces *interface-name* unit *logical-unit-number* family (ccc | inet | tcc)] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
  unit logical-unit-number {
    family (ccc | inet | tcc) {
      policer {
        arp policer-template-name;
        input policer-template-name;
        output policer-template-name;
      }
    }
  }
}
```



**Note**

To use policing on a CCC or TCC interface, you must include the family (ccc | tcc) statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level.

In the arp statement, list the name of one policer template to be evaluated when Address Resolution Protocol (ARP) packets are received on the interface. By default, an ARP policer is installed that is shared among all the Ethernet interfaces on which you have configured the family inet statement. If you want more stringent or lenient policing of ARP packets, you can configure an interface-specific policer and apply it to the interface. You configure an ARP policer just as you would configure any other policer, at the [edit firewall policer] hierarchy level. If you apply this policer to an interface, the default ARP packet policer is overridden. If you delete this policer, the default policer takes effect again.

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 configure a different policer on each protocol family under an interface. You can configure one input policer only and one output policer only for each protocol family. You can use the same policer one or more times. On M-series routers, you can apply to multiple interfaces a policer that polices the total traffic arriving on those interfaces. This does not work the same way on T-series platforms because interfaces can reside on different Packet Forwarding Engines (PFEs).

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

If you apply the policer to the interface lo0, it is applied to packets received or transmitted by the Routing Engine.

For more information about policies, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

## Apply a Filter to an Interface

To apply firewall filters to an interface, include the filter statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6 | mpls)] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
  unit logical-unit-number {
    family (inet | inet6 | mpls) {
      filter {
        group filter-group-number;
        input filter-name;
        output filter-name;
      }
    }
  }
}
```

In the group statement, specify the interface group number to associate with the filter.

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

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

You can use the same filter one or more times.

For filter-based forwarding (FBF), you can configure input packet filters only; FBF is not supported for output filters.

If you apply the filter to the interface lo0, it is applied to packets received or transmitted by the Routing Engine. You cannot apply MPLS filters to the management interface (fxp0) or the loopback interface (lo0).

For more information about firewall filters, see the *JUNOS Internet Software Configuration Guide: Policy Framework*. For more information about MPLS filters, see the *JUNOS Internet Software Configuration Guide: MPLS Applications*. For more information about FBF, see the *JUNOS Internet Software Configuration Guide: Routing Protocols*.

## Define Interface Groups in Firewall Filters

When applying a firewall filter, you can define an interface to be part of an *interface group*. Packets received on that interface are tagged as being part of the group. You can then match these packets using the `interface-group` match statement, as described in the *JUNOS Internet Software Configuration Guide: Policy Framework*.

To define the interface to be part of an interface group, include the `group` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6 | mpls) filter] hierarchy level:

```
[edit interfaces]
interfaces interface-name {
  unit logical-unit-number {
    family (inet | inet6 | mpls) {
      filter {
        group filter-group-number;
      }
    }
  }
}
```

## Configure Unicast RPF

For interfaces that carry IPv4 or IPv6 traffic, you can reduce the impact of denial of service (DoS) attacks by configuring unicast reverse path forwarding (RPF). Unicast RPF helps determine the source of attacks and rejects packets from unexpected source addresses on interfaces where unicast RPF is enabled.



**Note**

If you want to configure unicast RPF, your router must be equipped with the Internet Processor II ASIC.

If you enable unicast RPF on live traffic, some packets are dropped while the Packet Forwarding Engine is updating.

The following sections describe unicast RPF in detail:

Configure Unicast RPF Strict Mode on page 90

Configure Unicast RPF Loose Mode on page 91

Unicast RPF and Default Routes on page 91

Unicast RPF with Routing Asymmetry on page 92

Example: Configure Unicast RPF on page 93

## Configure Unicast RPF Strict Mode

In strict mode, unicast RPF checks whether the incoming packet has a source address that matches a prefix in the routing table, and whether the interface expects to receive a packet with this source address prefix.

If the incoming packet fails the unicast RPF check, the packet is not accepted on the interface. When a packet is not accepted on an interface, unicast RPF counts the packet and sends it to an optional fail filter.

The optional fail filter allows you to apply a filter to packets that fail the unicast RPF check. You can define the fail filter to perform any filter operation, including accepting, rejecting, logging, sampling, or policing.

When unicast RPF is enabled on an interface, Bootstrap Protocol (BOOTP) packets and Dynamic Host Configuration Protocol (DHCP) packets are not accepted on the interface. To allow the interface to accept BOOTP packets and DHCP packets, you must apply a fail filter that accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255. For a configuration example, see “Example: Configure Unicast RPF” on page 93.

For more information about defining fail filters, see the *JUNOS Internet Software Configuration Guide: Policy Framework*.

To configure unicast RPF, include the `rpf-check` statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6)] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family (inet | inet6)]
rpf-check <fail-filter filter-name>;
```

Using unicast RPF can have several consequences when implemented with traffic filters:

RPF fail filters are evaluated after input filters and before output filters.

If you configure a filter counter for packets dropped by an input filter, and you want to know the total number of packets dropped, you must also configure a filter counter for packets dropped by the RPF check.

To count packets that fail the RPF check and are accepted by the RPF fail filter, you must configure a filter counter.

If an input filter forwards packets anywhere other than the inet.0 or inet6.0 routing tables, the unicast RPF check is not performed.

If an input filter forwards packets anywhere other than the routing instance the input interface is configured for, the unicast RPF check is not performed.

## Configure Unicast RPF Loose Mode

By default, unicast RPF uses strict mode. Unicast RPF loose mode is similar to unicast RPF strict mode and has the same configuration restrictions. The only check in loose mode is whether the packet has a source address with a corresponding prefix in the routing table; loose mode does not check whether the interface expects to receive a packet with a specific source address prefix. If a corresponding prefix is not found, unicast RPF loose mode does not accept the packet. As in strict mode, loose mode counts the failed packet and optionally forwards it to a fail filter, which either accepts, rejects, logs, samples, or polices the packet.

To configure unicast RPF loose mode, include the mode statement at the [edit interfaces *interface-name* unit *logical-unit-number* family (inet | inet6) rpf-check] hierarchy level:

```
[edit interfaces interface-name unit logical-unit-number family (inet | inet6)]
rpf-check <fail-filter filter-name> {
  mode loose;
}
```

## Unicast RPF and Default Routes

When the active route cannot be chosen from the routes in a routing table, the router chooses a default route. A default route is equivalent to an IP address of 0.0.0.0/0. If you configure a default route, and you configure unicast RPF on an interface that the default route uses, unicast RPF behaves differently than it does otherwise. For information about configuring default routes, see the *JUNOS Internet Software Configuration Guide: Routing and Routing Protocols*. The following sections describe how unicast RPF behaves when a default route uses an interface and when a default route does not use an interface:

Unicast RPF Behavior with a Default Route on page 91

Unicast RPF Behavior without a Default Route on page 92

To determine whether the default route uses an interface, enter the show route command:

```
user@host> show route destination-address
```

*destination-address* is the next-hop address of the configured default route. The default route uses the interfaces shown in the output of the show route command.

### Unicast RPF Behavior with a Default Route

If you configure a default route that uses an interface configured with unicast RPF, unicast RPF behaves as follows:

**Strict mode**—If no corresponding route is found in the routing table, the packet is accepted. A packet is not accepted when:

The packet has a source address that matches a prefix in the routing table.

The interface does not expect to receive a packet with this source address prefix.

**Loose mode**—All packets are automatically accepted. For this reason, we recommend that you not configure unicast RPF loose mode on interfaces that the default route uses.

### Unicast RPF Behavior without a Default Route

If you do not configure a default route, or if the default route does not use an interface configured with unicast RPF, unicast RPF behaves as described in “Configure Unicast RPF Strict Mode” on page 90 and “Configure Unicast RPF Loose Mode” on page 91. To summarize, unicast RPF without a default route behaves as follows:

Strict mode—The packet is not accepted when either of the following is true:

The packet has a source address that does not match a prefix in the routing table.

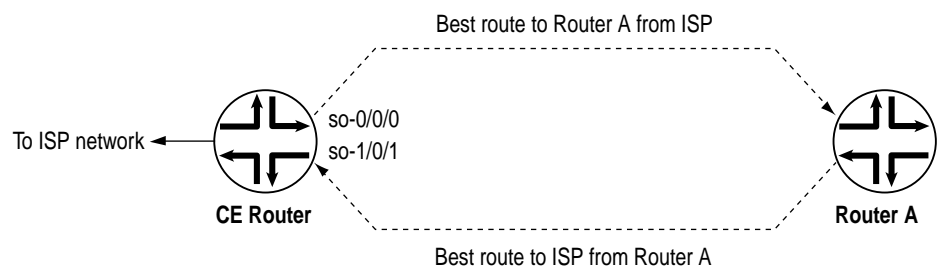
The interface does not expect to receive a packet with this source address prefix.

Loose mode—The packet is not accepted when the packet has a source address that does not match a prefix in the routing table.

### Unicast RPF with Routing Asymmetry

In general, we recommend that you not enable unicast RPF on interfaces that are internal to the network because internal interfaces are likely to have *routing asymmetry*. Routing asymmetry means that a packet’s outgoing and return paths are different. Routers in the core of the network are more likely to have asymmetric reverse paths than routers at the customer or provider edge. Figure 3 shows unicast RPF in an environment with routing asymmetry.

Figure 3: Unicast RPF with Routing Asymmetry



In Figure 3, if you enable unicast RPF on interface so-0/0/0, traffic destined for Router A is not rejected. If you enable unicast RPF on interface so-1/0/1, traffic from Router A is rejected.

If you need to enable unicast RPF in an asymmetric routing environment, you can use fail filters to allow the router to accept incoming packets that are known to be arriving by specific paths. For an example of a fail filter that accepts packets with a specific source and destination address, see “Example: Configure Unicast RPF” on page 93.

**Example: Configure Unicast RPF**

Configure unicast RPF strict mode, and apply a fail filter that allows the interface to accept BOOTP packets and DHCP packets. The filter accepts all packets with a source address of 0.0.0.0 and a destination address of 255.255.255.255.

```
[edit firewall]
filter rpf-special-case-dhcp-bootp {
  term allow-dhcp-bootp {
    from {
      source-address {
        0.0.0.0/32;
      }
      destination-address {
        255.255.255.255/32;
      }
    }
    then {
      count rpf-dhcp-bootp-traffic;
      accept;
    }
  }
  term default {
    then {
      log;
      reject;
    }
  }
}

[edit]
interfaces {
  so-0/0/0 {
    unit 0 {
      family inet {
        rpf-check fail-filter rpf-special-case-dhcp-bootp;
      }
    }
  }
}
```

## Enable Source Class and Destination Class Usage

For interfaces that carry IPv4 traffic, you can maintain packet counts based on the entry and exit points for traffic passing through your network. Entry and exit points are identified by source and destination prefixes grouped into disjoint sets defined as *source classes* and *destination classes*. You can define classes based on a variety of parameters, such as routing neighbors, autonomous systems, and route filters.

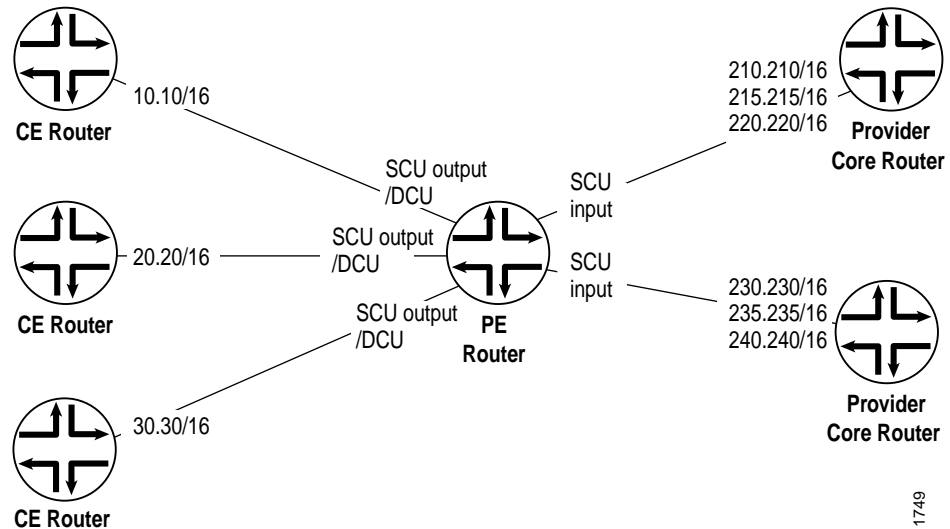
Source class usage (SCU) counts packets sent to customers by performing lookup on the IP source address and the IP destination address. SCU makes it possible to track traffic originating from specific prefixes on the provider core and destined for specific prefixes on the customer edge. You must enable SCU accounting on both the inbound and outbound physical interfaces.

Destination class usage (DCU) counts packets from customers by performing lookup of the IP destination address. DCU makes it possible to track traffic originating from the customer edge and destined for specific prefixes on the provider core router.

Figure 4 illustrates an ISP network. In this topology, you can use DCU to count packets customers send to specific prefixes. For example, you can have three counters, one per customer, that count the packets destined for prefix 210.210/16 and 220.220/16.

You can use SCU to count packets the provider sends from specific prefixes. For example, you can count the packets sent from prefix 210.210/16 and 215.215/16 and transmitted on a specific output interface.

Figure 4: Prefix Accounting with Source and Destination Classes



You can configure up to 126 source classes and 126 destination classes. For each interface on which you enable destination class usage and source class usage, the JUNOS software maintains an interface-specific counter for each corresponding class up to the 126 class limit.



**Note**

To configure source class and destination class usage, your router must be equipped with the Internet Processor II ASIC.

To enable packet counting on an interface, include the accounting statement at the [edit interfaces *interface-name* unit *logical-unit-number* family inet] hierarchy level:

```
accounting {
  destination-class-usage;
  source-class-usage {
    (input | output | [input output]);
  }
}
```

For SCU to work, you must configure at least one input interface and at least one output interface. An incoming packet is counted only once, and SCU takes priority over DCU. This means that when a packet arrives on an interface on which you include the source-class-usage input and destination-class-usage statements in the configuration, and when the source and destination both match accounting prefixes, the JUNOS software associates the packet with the source class only. To ensure the outgoing packet is counted, include the source-class-usage output statements in the configuration of the outgoing interface.

Once you enable accounting on an interface, the JUNOS software maintains packet counters for that interface. You must then configure the source class and destination class attributes in policy action statements, which must be included in forwarding-table export policies. For a complete discussion about source and destination class accounting profiles, see the *JUNOS Internet Software Configuration Guide: Network Management*.

## **Examples: Enable Source Class and Destination Class Usage**

Configure DCU and SCU output on one interface:

```
[edit]
interfaces {
  so-6/1/0 {
    unit 0 {
      family inet {
        accounting {
          destination-class-usage;
          source-class-usage {
            output;
          }
        }
      }
    }
  }
}
```

**Configure SCU input on another interface**

```
[edit]
interfaces {
  ge-0/1/0 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            input;
          }
        }
      }
    }
  }
}
```

**Optionally, you can include the input and output statements on a single interface**

```
[edit]
interfaces {
  ge-0/1/2 {
    unit 0 {
      family inet {
        accounting {
          source-class-usage {
            input;
            output;
          }
        }
      }
    }
  }
}
```

**Enable Packet Counting for Layer 3 VPNs**

You can use SCU and DCU to count packets on Layer 3 VPNs. To enable packet counting for Layer 3 VPN implementations at the egress point of the MPLS tunnel, you must configure a virtual loopback tunnel interface (vt) on the PE router, map the VRF instance type to the virtual loopback tunnel interface, and send the traffic received from the VPN out the source class output interface, as shown in the following example:

1. Configure a virtual loopback tunnel interface on a provider edge router equipped with a tunnel PIC:

```
[edit interfaces]
vt-0/3/0 {
  unit 0 {
    family inet {
      accounting {
        source-class-usage {
          input;
        }
      }
    }
  }
}
```

- Map the VRF instance type to the virtual loopback tunnel interface:

```
[edit]
routing-instances {
  VPN-A {
    instance-type vrf;
    interface at-2/1/1.0;
    interface vt-0/3/0.0;
    route-distinguisher 10.255.14.225:100;
    vrf-import import-policy-A;
    vrf-export export-policy-A;
    protocols {
      bgp {
        group to-r4 {
          local-address 10.27.253.1;
          peer-as 400;
          neighbor 10.27.253.2;
        }
      }
    }
  }
}
```



**Caution**

For SCU and DCU to work, you must not include the `vrf-table-label` statement at the `[edit routing-instances instance-name]` hierarchy level.

- Send traffic received from the VPN out the source class output interface:

```
[edit interfaces]
at-1/1/0 {
  unit 0 {
    family inet {
      accounting {
        source-class-usage {
          output;
        }
      }
    }
  }
}
```

For more information about VPNs, see the *JUNOS Internet Software Configuration Guide: VPNs*. For more information about virtual loopback tunnel interfaces, see “Configure Tunnel Interfaces” on page 407.

