Chapter 28
CoS Configuration Guidelines

To configure CoS properties, you can include the following statements at the [edit class-of-service] hierarchy level of the configuration:

```
class-of-service {
    input {
        fpc fpc-number {
            precedence-map map-name;
        }
        interfaces {
            interface-name {
                inet-precedence-map;
                mpls-cos-map;
                unit unit-number {
                    output-queue queue-number;
                }
            }
        }
        precedence-map map-name {
            bits precedence-bit output-queue queue-number;
        }
    }
    output {
        drop-profile profile-name {
            stream-profile {
                fill-level fill-percentage drop-probability probability-percentage;
            }
            plp-set-queue-profile {
                fill-level fill-percentage drop-probability probability-percentage;
            }
            plp-clear-queue-profile {
                fill-level fill-percentage drop-probability probability-percentage;
            }
        }
        fpc fpc-number {
            drop-profile profile-name;
        }
        interfaces {
            interface-name {
                transmit-queues {
                    output-queue queue-number buffer-percentage percentage;
                }
                weighted-round-robin {
                    output-queue queue-number weight percentage;
                }
            }
```

```
            unit unit-number {
                precedence-rewrite {
                    output-queue queue-number {
                        plp-clear rewrite-bits precedence-bit;
                        plp-set rewrite-bits precedence-bit;
                    }
                }
            }
        }
    }
    policy {
        class class-name {
            classification-override {
                output-queue queue-number;
            }
        }
    }
}
```

This chapter discusses the tasks to configure CoS:

Classify and Map Incoming IPv4 Packets

The JUNOS software classifies all incoming IPv4 traffic based on the value of the IPv4 ToS precedence bits. To use the JUNOS CoS features to differentiate service types, do one or more of the following:

For a configuration example, see “Examples: Classify and Map Input IPv4 Packets” on page 331.

When classifying incoming IPv4 traffic, you can map the precedence bits to different output transmission queues (item 1 in Figure 17 on page 326) or map all the incoming traffic on an interface to a specific output transmission queue (item 2 in Figure 17). These two classification schemes are mutually exclusive.

For each IPv4 traffic stream on a link, you can configure up to four output transmission queues. Each FPC can have a maximum of 16 traffic links, so each FPC can have a total of 64 output transmission queues.

For MPLS traffic, the mapping of CoS bits to output transmission queues is static and cannot be configured.

The application that generates the IPv4 packet controls the value in the ToS field.

The precedence bits in the ToS field of the IP packet header comprise the first three bits in the eight-bit ToS field, so there can be eight different precedence values (see Table 15). To configure output transmission queues for CoS, map the eight precedence bit values to the four output queues available on the link. Creating the mapping between the ToS values and the output transmission queues effectively creates the output transmission queues on the FPC.

Assign Precedence Bits to Output Transmission Queues

When classifying incoming IPv4 traffic, you can assign the IP precedence bits to different output transmission queues (item 1 in Figure 17 on page 326). By default, if you do not classify and map incoming packets, packets whose precedence bits range from 000 through 101 are placed in queue 0, and network control packets are placed in queue 3. These packets are generated management traffic, such as OSPF and RSVP control traffic, whose precedence bits are 110 and 111.

After you define the mapping, you must associate it with an FPC in the router and with interfaces on the FPC.

> **Caution**
> If you assign the IP precedence bits to output transmission queues, you cannot also map an interface's packet to an output transmission queue (described in "Map an Input Interface to an Output Transmission Queue" on page 330). That is, you cannot include both the inet-precedence-map and the output-queue statements. These two input packet classification schemes are mutually exclusive.

To assign precedence bits to output transmission queues, do the following:

1. Assign one or more of the precedence bit values to specific output transmission queue by including the precedence-map statement at the [edit class-of-service input] hierarchy level:

   ```
   [edit class-of-service input]
   precedence-map map-name {
       bits bits queue queue-number;
   }
   ```

   Include one bits statement for each precedence bit that you want to map to a nondefault queue. For clarity, you can explicitly specify mappings for all precedence bits, even those being directed to the default queue.

   The *queue-number* can be a value from 0 through 3. To differentiate service types, you must configure at least two queues.

   > **Caution**
   > If you change the queue assignments for precedence bits 110 and 111, which are used by network control packets, the routing protocols running on the router will no longer function correctly.

   In a configuration, you can include an unlimited number of precedence-map statements to define an unlimited number of precedence bit mappings.

2. Associate the precedence bit mapping with a particular FPC in the router and bind the mapping to the FPC by including the fpc statement at the [edit class-of-service input] hierarchy level:

```
[edit class-of-service input]
fpc fpc-number {
    precedence-map map-name;
}
```

map-name is a mapping you defined with the precedence-map statement.

You can bind one precedence map with each FPC.

3. Associate the precedence bit mapping with the physical interface by including the inet-precedence-map statement at the [edit class-of-service input interfaces *interface-name*] hierarchy level:

```
[edit class-of-service input]
interfaces {
    interface-name {
        inet-precedence-map;
    }
}
```

Include the inet-precedence-map statement for each physical interface on the FPC.

Map an Input Interface to an Output Transmission Queue

When classifying incoming IPv4 traffic, you can assign all the incoming traffic on a logical interface to a specific output transmission queue (item 2 in Figure 17 on page 326). By default, if you do not classify and map incoming packets, packets whose precedence bits range from 000 through 101 are placed in queue 0, and network control packets are placed in queue 3. These packets are generated management traffic, such as OSPF and RSVP control traffic, whose precedence bits are 110 and 111.

When mapping traffic to output transmission queues on a per-interface basis, you actually map the traffic on each logical interface.

> **Caution**
>
> If you map an interface's packet to an output transmission queue, you cannot also map the IP precedence bits to output transmission queues (described in "Assign Precedence Bits to Output Transmission Queues" on page 329). That is, you cannot include both the inet-precedence-map and the output-queue statements. These two input packet classification schemes are mutually exclusive.

To map incoming traffic on a logical interface to a specific output transmission queue, include the unit statement at the [edit class-of-service input interfaces *interface-name*] hierarchy level:

```
[edit class-of-service input]
interfaces {
    interface-name {
        unit unit-number {
            output-queue queue-number;
        }
    }
}
```

unit-number is the logical unit number.

queue-number is the output transmission queue. It can be a value from 0 through 3. To differentiate service types, you must configure at least two queues.

For any logical interfaces that you do not configure, the default output transmission queue is queue 0.

Examples: Classify and Map Input IPv4 Packets

Map the precedence bits to output transmission queues:

```
[edit]
class-of-service {
    input {
        precedence-map import-remapping {
            bits 000 output-queue 2;
            bits 001 output-queue 1;
            bits 010 output-queue 1;
            bits 011 output-queue 2;
            bits 100 output-queue 2;
            bits 101 output-queue 0;
        }
        fpc 1 {
            precedence-map import-remapping;
        }
        interfaces {
            so-1/0/0 {
                inet-precedence-map;
            }
            so-1/2/0 {
                inet-precedence-map;
            }
            so-1/3/0 {
                inet-precedence-map;
            }
        }
    }
}
```

Map the incoming traffic on logical interface 0 on interface so-0/0/0 to an output transmission queue 2. If this interface has other logical interfaces, their incoming traffic is placed in the default queue, which is queue 0.

```
[edit]
class-of-service {
    input {
        interfaces {
            so-0/0/0 {
                unit 0 {
                    output-queue 2;
                }
            }
        }
    }
}
```

Classify and Map Incoming MPLS Packets

By default, all MPLS packets are placed into output transmission queue 0 (see item 3 in Figure 17 on page 326). To map MPLS packets to multiple output transmission queues based on the CoS field in the MPLS header, you configure the software to use the MPLS CoS mapping by including the mpls-cos-map statement at the [edit class-of-service input interfaces *interface-name*] hierarchy level:

```
[edit class-of-service input]
interfaces {
    interface-name {
        mpls-cos-map;
    }
}
```

You can also include the class-of-service statement in the MPLS configuration (in the [edit protocols mpls] hierarchy), as described in "Configure the MPLS CoS Value" in *JUNOS Internet Software Configuration Guide: MPLS Applications*. (This statement is optional; if you do not include it, the default is to queue the packet based on the MPLS CoS bits in the packet's header.)

Configure Packet Transmission

After incoming traffic has been classified, you can do one or more of the following:

Override the Input Classification on page 333

Rewrite the IP Precedence Bits on page 334

Configure Weighted Round-Robin to Schedule Packet Transmission on page 336

Configure Congestion Avoidance Using RED on page 337

Override the Input Classification

For IPv4 packets, you can override the incoming classification, assigning them to the same output transmission queue based on their input interface, input precedence bits, or destination address (item 3 in Figure 17). You do so by defining a policy class when configuring CoS properties and referencing this class when configuring a routing policy.

When you override the classification of incoming packets, any mappings you configured for associated precedence bits or incoming interfaces to output transmission queues are ignored. Also, if the packet loss priority (PLP) bit was set in the packet by the incoming interlace, the PLP bit is cleared.

To override the input packet classification, do the following:

1. Define the policy class by including the class statement at the [edit class-of-service policy] hierarchy level:

    ```
    [edit class-of-service]
    policy {
        class class-name {
            classification-override {
                output-queue queue-number;
            }
        }
    }
    ```

    class-name is a name that identifies the class.

    queue-number is the output transmission queue to which to assign packets. It can be a value from 0 through 3.

2. Associate the policy class with a routing policy by including it in a policy-statement statement at the [edit policy-options] hierarchy level. Specify the destination prefixes in the route-filter statement and the CoS policy class name in the then statement.

    ```
    [edit policy-options]
    policy-statement policy-name {
        term term-name {
            from {
                route-filter destination-prefix match-type <class class-name>;
            }
            then class class-name;
        }
    }
    ```

3. Apply the policy by including the export statement at the [edit routing-option] hierarchy level:

    ```
    [edit routing-options]
    forwarding-table {
        export policy-name;
    }
    ```

Rewrite the IP Precedence Bits

After IPv4 packets have been received and classified, you can map traffic to an output queue based on the packet's input interface, input precedence bits, or destination address (item 4 in Figure 17 on page 326). You can also specify the rewrite value for each output queue associated with a particular logical interface (item 5 in Figure 17 on page 326).

You rewrite the IP headers of all packets in an output transmission queue based on whether the packet's PLP bit is set. For packets whose PLP bit is set, you can rewrite the IP precedence bits to one value, and for packets whose PLP bit is not set, you can rewrite the IP precedence bits to a second value.

To rewrite the IP precedence bits in a packet's IP header for packets whose PLP bit is set, include the precedence-rewrite statement at the [edit class-of-service output interfaces *interface-name* unit *unit-number*] hierarchy level:

```
[edit class-of-service]
output {
    interfaces {
        interface-name {
            unit unit-number {
                precedence-rewrite {
                    output-queue queue-number {
                        plp-clear rewrite-bits precedence-bit;
                        plp-set rewrite-bits precedence-bit;
                    }
                }
            }
        }
    }
}
```

In the output-queue statement, specify the number of the output transmission queue whose IP precedence bits you are rewriting. The number can be a value from 0 through 3.

In the plp-clear statement, define the new IP precedence bits to write in packets whose PLP bit is not set.

In the plp-set statement, define the new IP precedence bits to write in packets whose PLP bit is set.

For both MPLS and IP traffic, if the least significant bit (LSB) of the precedence field is set, then the PLP bit is set by the incoming interface on downstream routers. This increases the probability of drop by RED on downstream routers. Therefore, whenever you explicitly set the PLP on a packet, you should also rewrite its precedence field to a value that has the LSB of the precedence field set as follows:

```
001
011
101
111
```

Conversely, with packets that should receive priority treatment, and any traffic that does not have its PLP bit set, you should rewrite the precedence field to values that do not contain a 1 in the LSB:

```
000
010
100
110
```

For MPLS packets, in addition to the LSB of the precedence field affecting downstream treatment, the first two bits identify the outbound transmission queue used on all downstream outbound interfaces. You can manipulate the MPLS CoS field through MPLS CoS configuration.

Example: Rewrite the IP Precedence Bits

Map the incoming traffic on logical interface 0 on interface ge-0/1/0 to output transmission queue 1. On the output queue, rewrite the precedence bits as the packet leaves the router. The ToS bits will be rewritten to 110 upon transmission if the packet transits interface ge-4/0/0.0 without the PLP bit set.

```
[edit]
class-of-service {
    input {
        interfaces {
            ge-0/1/0 {
                unit 0 {
                    output-queue 1;
                }
            }
        }
    }
    output {
        interfaces {
            ge-4/2/0 {
                unit 0 {
                    precedence-rewrite {
                        output-queue 0 {
                            plp-clear rewrite-bits 111;
                        }
                        output-queue 1 {
                            plp-clear rewrite-bits 110;
                        }
                        output-queue 2 {
                            plp-clear rewrite-bits 000;
                        }
                        output-queue 3 {
                            plp-clear rewrite-bits 101;
                        }
                    }
                }
            }
        }
    }
}
```

Configure Weighted Round-Robin to Schedule Packet Transmission

A weighted round-robin scheme allows you to control the bandwidth allocated to the output transmission queues on each link (see item 5 in Figure 17 on page 326). The weighting is based on the amount of bandwidth allocated for each queue. Packets from queues with more bandwidth are transmitted more often than those from queues with less bandwidth. On Juniper Networks routers, up to four transmission queues can be configured for each output link. You weight the queues to influence the amount of bandwidth each queue consumes to transmit packets. You assign the weights as a ratio (percentage) of the total link transmission bandwidth.

By default, 95 percent of the bandwidth is allocated to queue 0 and 5 percent is allocated to queue 3. To modify this allocation, include the weighted-round-robin statement when you configure each of the FPC's interfaces. You specify the weight by assigning each queue a percentage of the total transmission bandwidth.

To configure weighted round-robin, include the weighted-round-robin statement at the [edit class-of-service output interfaces *interface-name*] hierarchy level:

```
[edit class-of-service]
output {
    interfaces {
        interface-name {
            weighted-round-robin {
                output-queue queue-number weight percentage;
            }
        }
    }
}
```

In the output-queue statement, specify the number of the output transmission queue. The number can be a value from 0 through 3.

In the weight option, specify the percentage of weight allocated to the output transmission queue. The sum of the bandwidth assigned to the four queues must be 100 percent.

Example: Configure Weighted Round-Robin to Schedule Packet Transmission

Configure weighted round-robin to transmit packets from the output transmission queues:

```
[edit class-of-service output interfaces interface-name]
weighted-round-robin {
    output-queue 0 weight 50;
    output-queue 1 weight 30;
    output-queue 2 weight 10;
    output-queue 3 weight 10;
}
```

Configure Congestion Avoidance Using RED

RED provides a congestion-avoidance mechanism (see item in Figure 17 on page 326). If you do not configure RED (or other active queue management techniques), the router waits for a queue to become 100 percent full and then drops all subsequent packets arriving at that queue until the queue has transmitted enough packets that there is space again in the queue. The RED algorithm anticipates incipient congestion and reacts by dropping a small percentage of packets from the head of the queue to prevent the queue from becoming congested. To configure RED, you define drop profiles, which are tables that map the fullness of a queue to a drop probability, and you configure the packet buffer space associated with each of the queues.

To provide information to RED about congestion elsewhere in the router, the input interfaces monitor the level of traffic congestion on their interfaces to determine whether the amount of traffic exceeds a threshold level. This mechanism allows you to police the amount of traffic that a subscriber is sending and to increase the drop probability of traffic above the threshold level. You configure traffic threshold levels by defining the receive leaky bucket threshold, as described in "Configure Receive and Transmit Leaky Bucket Properties" on page 36. If the traffic level exceeds the receive bucket's threshold, the PIC sets the PLP bit in the packet.

The incoming PIC also provides information to RED about incoming congestion. It checks bit 5 of the IPv4 type-of-service (ToS) field, and if it is set, the PIC sets the packet's PLP bit, thereby informing RED that the packet previously experienced congestion. This mechanism can be useful for across-the network CoS applications, in which the ingress router can rewrite ToS bits to reflect the service level that is being provided to individual customers.

To configure RED, you define drop profiles for each FPC. A drop profile is a table of pairs of values that describe the fullness, or depth, of a queue and a probability of dropping a packet.

You can define up to 64 pairs of values in the drop profile for a single FPC.

In the RED drop profiles, you map the fullness of a queue to a drop probability, expressing the fullness of a queue as a percentage of the total output bandwidth allocated to that queue. For each FPC, you define three drop profiles: one for the entire output traffic stream, one for packets whose PLP bit is not set, and one for packets whose PLP bit is set. RED periodically examines the queue and the packet at the head of the queue. If the congestion level on the queue corresponds to a nonzero drop probability, RED decides whether to drop the packet at the head of the queue.

You can configure three different RED drop profiles for each FPC:

Stream drop profile—Defines the drop profile for the entire packet stream passing through a physical output interface. A stream represents the entire traffic on the interface. When all traffic on the interface is traffic on an LSP the stream is equivalent to the LSP. If more than one LSP is defined on an output interface, the stream represents all the LSPs.

Drop profile for packets whose non-PLP bit is not set—Defines the drop profile for queues in which the packet at the head of the queue is one in which the PLP bit is not set.

Drop profile for packets whose PLP bit is set—Defines the drop profile for queues in which the packet at the head of the queue is one in which the PLP bit is set.

For a packet to be dropped, it must pass the drop profile tests specified either in the stream drop and PLP queue drop profiles or in the stream drop and non-PLP queue drop profiles.

To configure congestion avoidance, do the following:

1. Configure RED drop profiles by including the drop-profile statement at the [edit class-of-service output] hierarchy level:

```
[edit class-of-service output]
drop-profile profile-name {
    stream-profile {
        fill-level fill-percentage drop-probability probability-percentage;
        ...
    }
    plp-clear-queue-profile {
        fill-level fill-percentage drop-probability probability-percentage;
        ...
    }
    plp-set-queue-profile {
        fill-level fill-percentage drop-probability probability-percentage;
        ...
    }
}
```

   profile-name is a text name that identifies the profile.

   For each drop profile, specify one or more fill levels and the drop probability that corresponds to that fill level. List the fill levels incrementally, in increasing order. Both the fill level and the drop probability can be numbers in the range 0 through 100.

2. Associate a drop profile with an FPC by including the fpc statement at the [edit class-of-service output] hierarchy level:

```
[edit class-of-service]
output {
    fpc fpc-number {
        drop-profile profile-name;
    }
}
```

   You can associate one drop profile with each FPC.

3. Allow packet buffering in each of the output transmission queues by configuring the percentage of the output buffer memory allocated to each queue by including the transmit-queue statement at the [edit class-of-service output interfaces *interface-name*] hierarchy level. You specify the memory as a percentage of the total transmission buffer storage.

```
[edit class-of-service]
output {
    interfaces {
        interface-name {
            transmit-queues {
                output-queue queue-number buffer-percentage percentage;
            }
        }
    }
}
```

   In the output-queue statement, specify the number of the output transmission queue. The number can be a value from 0 through 3. Include one output-queue statement for each output queue you are configuring.

In the buffer-percentage option, specify the percentage of buffer memory to allocate to the transmit queue. The sum of the allocations for all the queues must be 100 percent. By default, the following percentages are allocated to the four output queues:

Queue 0—95 percent

Queue 1—0 percent

Queue 2—0 percent

Queue 3—5 percent

Example: Configure Congestion Avoidance Using RED

Configure RED, using the drop profiles shown in Figure 16 on page 325 and associate the drop profiles with interfaces on the FPC in slot 3:

```
[edit class-of-service output]
drop-profile sample-profile {
    stream-profile {
        fill-level 25 drop-probability 25;
        fill-level 50 drop-probability 50;
        fill-level 75 drop-probability 75;
        fill-level 90 drop-probability 100;
    }
    plp-clear-queue-profile {
        fill-level 60 drop-probability 25;
        fill-level 80 drop-probability 60;
        fill-level 90 drop-probability 100;
    }
    plp-set-queue-profile {
        fill-level 0 drop-probability 10;
        fill-level 10 drop-probability 20;
        fill-level 25 drop-probability 50;
        fill-level 50 drop-probability 100;
    }
}
fpc 3 {
    drop-profile sample-profile;
}
interfaces {
    so-3/0/0 {
        transmit-queues {
            output-queue 0 buffer-percentage 40;
            output-queue 1 buffer-percentage 20;
            output-queue 2 buffer-percentage 20;
            output-queue 3 buffer-percentage 20;
        }
    }
}
```