

## Chapter 22

# Additional CoS Configuration Examples

This chapter includes the following examples:

- Example: Configuring Classifiers, Rewrite Markers, and Schedulers on page 273
- Example: Configuring a CoS Policy for IPv6 Packets on page 278

### Example: Configuring Classifiers, Rewrite Markers, and Schedulers

---

1. Define a classifier that matches IP traffic arriving on the interface. The affected IP traffic has IP precedence bits with patterns matching those defined by aliases A or B. The loss priority of the matching packets is set to low, and the forwarding class is mapped to best effort (queue 0):

```
[edit]
class-of-service {
  classifiers {
    inet-precedence normal-traffic {
      forwarding-class best-effort {
        loss-priority low code-points [my1 my2];
      }
    }
  }
}
```

Following are the code-point alias and forwarding-class mappings referenced in the `normal-traffic` classifier:

```
[edit]
class-of-service {
  code-point-aliases {
    inet-precedence {
      my1 000;
      my2 001;
      ...
    }
  }
}
```

```
[edit]
class-of-service {
  forwarding-classes {
    queue 0 best-effort;
    queue 1 expedited-forwarding;
  }
}
```

2. Use rewrite markers to redefine the bit pattern of outgoing packets. Assign the new bit pattern based on specified forwarding classes, regardless of the loss priority of the packets:

```
[edit]
class-of-service {
  rewrite-rules {
    inet-precedence clear-prec {
      forwarding-class best-effort {
        loss-priority low code-point 000;
        loss-priority high code-point 000;
      }
      forwarding-class expedited-forwarding {
        loss-priority low code-point 100;
        loss-priority high code-point 100;
      }
    }
  }
}
```

3. Configure a scheduler map associating forwarding classes with schedulers and drop-profiles:

```
[edit]
class-of-service {
  scheduler-maps {
    one {
      forwarding-class expedited-forwarding scheduler special;
      forwarding-class best-effort scheduler normal;
    }
  }
}
```

Schedulers establish how to handle the traffic within the output queue for transmission onto the wire. Following is the scheduler referenced in scheduler map one:

```
[edit]
class-of-service {
  schedulers {
    special {
      transmit-rate percent 30;
      priority high;
    }
    normal {
      transmit-rate percent 70;
      priority low;
    }
  }
}
```

4. Apply the **normal-traffic** classifier to all SONET/SDH interfaces and all logical interfaces of SONET/SDH interfaces; apply the **clear-prec** rewrite marker to all Gigabit Ethernet interfaces and all logical interfaces of Gigabit Ethernet interfaces; and apply the one scheduler map to all interfaces:

```
[edit]
class-of-service {
  interfaces {
    so-0/0/0 {
      scheduler-map one;
      unit 0 {
        classifiers {
          inet-precedence normal-traffic;
        }
      }
    }
    so-0/0/1 {
      scheduler-map one;
      unit 1 {
        classifiers {
          inet-precedence normal-traffic;
        }
      }
    }
  }
  ge-1/0/0 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
}
```

```

ge-1/0/1 {
  scheduler-map one;
  unit 0 {
    rewrite-rules {
      inet-precedence clear-prec;
    }
  }
  unit 1 {
    rewrite-rules {
      inet-precedence clear-prec;
    }
  }
}
}

```

Following is the complete configuration:

```

[edit class-of-service]
classifiers {
  inet-precedence normal-traffic {
    forwarding-class best-effort {
      loss-priority low code-points [my1 my2];
    }
  }
}
code-point-aliases {
  inet-precedence {
    my1 000;
    my2 001;
    cs1 010;
    cs2 011;
    cs3 100;
    cs4 101;
    cs5 110;
    cs6 111;
  }
}
drop-profiles {
  high-priority {
    fill-level 20 drop-probability 100;
  }
  low-priority {
    fill-level 90 drop-probability 95;
  }
  big-queue {
    fill-level 100 drop-probability 100;
  }
}
forwarding-classes {
  queue 0 best-effort;
  queue 1 expedited-forwarding;
}

```

```

interfaces {
  so-0/0/0 {
    scheduler-map one;
    unit 0 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  so-0/0/1 {
    scheduler-map one;
    unit 1 {
      classifiers {
        inet-precedence normal-traffic;
      }
    }
  }
  ge-1/0/0 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
  ge-1/0/1 {
    scheduler-map one;
    unit 0 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
    unit 1 {
      rewrite-rules {
        inet-precedence clear-prec;
      }
    }
  }
}
rewrite-rules {
  inet-precedence clear-prec {
    forwarding-class best-effort {
      loss-priority low code-point 000;
      loss-priority high code-point 000;
    }
    forwarding-class expedited-forwarding {
      loss-priority low code-point 100;
      loss-priority high code-point 100;
    }
  }
}
}

```

```

scheduler-maps {
  one {
    forwarding-class expedited-forwarding scheduler special;
    forwarding-class best-effort scheduler normal;
  }
}
schedulers {
  special {
    transmit-rate percent 30;
    priority high;
  }
  normal {
    transmit-rate percent 70;
    priority low;
  }
}

```

## Example: Configuring a CoS Policy for IPv6 Packets

---

1. Define a new classifier of type DSCP IPv6.

```

[edit class-of-service]
classifiers {
  dscp-ipv6 core-dscp-map {
    forwarding-class best-effort {
      loss-priority low code-points 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-points 001010;
    }
    forwarding-class network-control {
      loss-priority low code-points 110000;
    }
  }
}

```

2. Define a new rewrite rule of type DSCP IPv6.

```

[edit class-of-service]
rewrite-rules {
  dscp-ipv6 core-dscp-rewrite {
    forwarding-class best-effort {
      loss-priority low code-point 000000;
    }
    forwarding-class assured-forwarding {
      loss-priority low code-point 001010;
    }
    forwarding-class network-control {
      loss-priority low code-point 110000;
    }
  }
}

```

3. Assign the classifier and rewrite rule to a logical interface.

```
[edit class-of-service]
interfaces so-2/0/0 {
  unit 0 {
    classifiers { # Both dscp and dscp-ipv6 classifiers on this interface.
      dscp default;
      dscp-ipv6 core-dscp-map;
    }
    rewrite-rules { # Both dscp and dscp-ipv6 rewrite rules on this interface.
      dscp default;
      dscp-ipv6 core-dscp-rewrite;
    }
  }
}
```

