Chapter 6

Classifying Packets by Behavior Aggregate

By default, packet classification is performed by IP precedence and MPLS EXP classifiers. The IP precedence classifier handles all incoming IP packets, and the MPLS EXP classifier handles all incoming MPLS packets.

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

```
class-of-service {
    classifiers {
        (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
            import (classifier-name | default);
            forwarding-class class-name {
                loss-priority level {
                    code-points [ aliases ] [ 6-bit-patterns ];
                }
            }
        }
    }
    interfaces {
        interface-name {
            unit logical-unit-number {
                classifiers {
                    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) (classifier-name
                        | default);
                }
            }
        }
    }
    routing-instances routing-instance-name {
        classifiers {
            exp (classifier-name | default);
        }
    }
}
```

This chapter discusses the following topics:

Classifier Types

You can configure the following classifier types:

- DSCP, DSCP IPv6, or IP precedence—IP packet classification (Layer 3 headers)
- MPLS EXP—MPLS packet classification (Layer 2 headers)
- IEEE 802.1p—Packet classification (Layer 2 headers)

If you apply an IEEE 802.1 classifier to a logical interface, this classifier takes precedence and is not compatible with any other classifier type. Classifiers for IP (DSCP or IP precedence) and MPLS (EXP) can coexist on a logical interface if the hardware platform requirements are met. (See Table 11 on page 49.)

The simplest way to classify a packet is to use behavior aggregate classification. The DSCP, DSCP IPv6, or IP precedence bits of the IP header convey the behavior aggregate class information. The information might also be found in the MPLS EXP bits or IEEE 802.1p CoS bits.

Default Behavior Aggregate Classification

Table 10 shows the default system classification scheme for the well-known DSCPs.

Table 10: Default Behavior Aggregate Classification

DSCP and DSCP IPv6	Forwarding Class	PLP
ef	expedited-forwarding	low
af11	assured-forwarding	low
af12	assured forwarding	high
af13	assured forwarding	high
af21	best-effort	low
af22	best-effort	low
af23	best-effort	low
af31	best-effort	low
af32	best-effort	low
af33	best-effort	low
af41	best-effort	low
af42	best-effort	low
af43	best-effort	low
be	best-effort	low
cs1	best-effort	low
cs2	best-effort	low
cs3	best-effort	low
cs4	best-effort	low
cs5	best-effort	low
nc1/cs6	network-control	low
nc2/cs7	network control	low
other	best-effort	low

All af classes other than af1X are mapped to best-effort, because RFC 2597 prohibits a node from aggregating classes. In effect, mapping to best-effort implies that the node does not support that class.

Defining Classifiers

To define new classifiers for all code-point types, include the classifiers statement at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
classifiers {
    (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) classifier-name {
      import [classifier-name | default];
      forwarding-class class-name {
        loss-priority level {
          code-points [ aliases ] [ 6-bit-patterns ];
        }
      }
    }
}
```

The map sets the forwarding class and PLP for a specific set of code-point aliases and bit patterns. The inputs of the map are code-point aliases and bit patterns. The outputs of the map are the forwarding class and the PLP. For more information about how CoS maps work, see Table 5 on page 14.

The classifiers work as follows:

- dscp—Handles incoming IPv4 packets.
- dscp-ipv6—Handles incoming IPv6 packets. For more information, see "Applying DSCP IPv6 Classifiers" on page 50.
- exp—Handles MPLS packets using Layer 2 headers.
- ieee-802.1—Handles Layer 2 CoS.
- inet-precedence—Handles incoming IPv4 packets. IP precedence mapping requires only the upper three bits of the DSCP field.

A classifier takes a specified bit pattern as either the literal pattern or as a defined alias and attempts to match it to the type of packet arriving on the interface. If the information in the packet's header matches the specified pattern, the packet is sent to the appropriate queue, defined by the forwarding class associated with the classifier.

The code-point aliases and bit patterns are the input for the map. The loss priority and forwarding class are outputs of the map. In other words, the map sets the PLP and forwarding class for a given set of code-point aliases and bit patterns.

You can use any table, including the default, in the definition of a new classifier by including the import statement. The imported classifier is used as a template and is not modified. Whenever you commit a configuration that assigns a new *class-name* and loss-priority value to a code-point alias or set of bits, it replaces that entry in the imported classifier template. As a result, you must explicitly specify every code point in every designation that requires modification.

NOTE: If an interface is mounted on an M-series FPC, you can apply to the interface the default exp classifier only. If an interface is mounted on an enhanced FPC, you can create a new exp classifier and apply it to an interface.

On M320 and T-series platforms, the behavior aggregate (BA) classifier loss priority setting has no effect. For more information, see "Overriding the Default PLP on M320 and T-series Platforms" on page 55.

Applying a Classifier to a Logical Interface

You can apply the classification map to a logical interface by including the classifiers statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
classifiers (dscp | dscp-ipv6 | exp | ieee-802.1 | inet-precedence) (classifier-name
        | default);
```

You can use interface wildcards for *interface-name* and *logical-unit-number*.

NOTE: If you apply an IEEE 802.1p classifier to a logical interface, you cannot apply non-IEEE classifiers on other logical interfaces on the same physical interface.

Table 11 shows the classifiers you can combine on a single logical interface, by platform.

Table 11: Supported Classifier Combinations by Platform

Classifier Combinations	T-series and M320	Other M-series with Regular FPCs	Other M-series with Enhanced FPCs
dscp and inet-precedence	No	No	No
dscp-ipv6 and (dscp | inet-precedence)	Yes	No	No
exp and ieee 802.1	No	No	No
ieee 802.1 and (dscp | dscp-ipv6 | exp | inet-precedence)	No	No	Yes
exp and (dscp | dscp-ipv6 | inet-precedence)	Yes	No	Yes

Applying DSCP IPv6 Classifiers

For T-series platforms, you can apply separate classifiers for IPv4 and IPv6 packets per logical interface by including the classifiers statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number*] hierarchy level and specifying the dscp and dscp-ipv6 classifier types:

```
[edit class-of-service interfaces interface-name unit logical-unit-number]
classifiers dscp (classifier-name | default);
classifiers dscp-ipv6 (classifier-name | default);
```

For M-series enhanced FPCs, you cannot apply separate classifiers for IPv4 and IPv6 packets on a single logical interface. Instead, classifier assignment works as follows:

- If you assign a DSCP classifier only, IPv4 and IPv6 packets are classified using the DSCP classifier.
- If you assign an IP precedence classifier only, IPv4 and IPv6 packets are classified using the IP precedence classifier. In this case, the lower three bits of the DSCP field are ignored because IP precedence mapping requires the upper three bits only.
- If you assign either the DSCP or the IP precedence classifier in conjunction with the DSCP IPv6 classifier, the commit fails.
- If you assign a DSCP IPv6 classifier only, IPv4 and IPv6 packets are classified using the DSCP IPv6 classifier, but the commit displays a warning message.

For more information, see Table 11 on page 49.

Applying MPLS EXP Classifiers to Routing Instances

When you enable VRF table labels and you do not explicitly apply a classifier configuration to the routing instance, the default MPLS EXP classifier is applied to the routing instance. For detailed information about VRF table labels, see the *JUNOS VPNs Configuration Guide*.

The default MPLS EXP rewrite table contents are shown in Table 12.

Table 12: Default MPLS EXP Classification Table

Forwarding Class	Loss Priority	Code Point
best-effort	low	000
best-effort	high	001
expedited-forwarding	low	010
expedited-forwarding	high	011
assured-forwarding	low	100
assured-forwarding	high	101
network-control	low	110
network-control	high	111

For PICs that are installed on enhanced FPCs, you can override the default MPLS EXP classifier and apply a custom classifier to the routing instance. To do this, perform the following configuration tasks:

1. Filter traffic based on the IP header by including the vrf-table-label statement at the [edit routing-instances *routing-instance-name*] hierarchy level:

```
[edit routing-instances routing-instance-name]
vrf-table-label;
```

2. Configure a custom MPLS EXP classifier by including the following statements at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
classifiers {
    exp classifier-name {
        import (classifier-name | default);
        forwarding-class class-name {
            loss-priority level {
                code-points [ aliases ] [ 6-bit-patterns ];
            }
        }
    }
}
forwarding-classes {
    queue queue-number class-name priority (high | low);
}
```

3. Configure the routing instance to use the custom MPLS EXP classifier by including the exp statement at the [edit class-of-service routing-instances *routing-instance-name* classifiers] hierarchy level:

```
[edit class-of-service routing-instances routing-instance-name classifiers]
exp classifier-name;
```

To display the MPLS EXP classifiers associated with all routing instances, issue the show class-of-service routing-instances command.

NOTE: The following caveats apply to custom MPLS EXP classifiers for routing instances:

- An enhanced FPC is required.
- Logical routers are not supported.

For more details, see the following sections:

- Configuring Global Classifiers and Wildcard Routing Instances on page 52
- Examples: Applying MPLS EXP Classifiers to Routing Instances on page 53

Configuring Global Classifiers and Wildcard Routing Instances

To configure a global routing instance classifier, include the all statement at the [edit class-of-service routing-instances] hierarchy level:

```
[edit class-of-service routing-instances]
all {
    classifiers {
        exp classifier-name;
    }
}
```

For routing instances associated with specific classifiers, the global configuration is ignored.

To use a wildcard in the routing instance classifier configuration, include an asterisk (*) in the name of the routing instance:

```
[edit class-of-service routing-instances]
instance-name* {
    classifiers {
        exp classifier-name;
    }
}
```

The wildcard configuration follows the longest match. If there is a specific configuration, it is given precedence over the wildcard configuration.

NOTE: Wildcards and the all keyword are supported at the [edit class-of-service routing-instances] hierarchy level but not at the [edit routing-instances] hierarchy level.

If you configure a routing instance at the [edit routing-instances] hierarchy level with, for example, the name vpn*, the JUNOS software treats vpn* as a valid and distinct routing instance name. If you then try to apply a classifier to the vpn* routing instance at the [edit class-of-service routing-instances] hierarchy level, the JUNOS software treats the vpn* routing instance name as a wildcard, and all the routing instances that start with vpn and do not have a specific classifier applied receive the classifier associated with vpn*. This same behavior applies with the all keyword.

Examples: Applying MPLS EXP Classifiers to Routing Instances

Configuring a Global Classifier

Configure a global classifier for all routing instances and override the global classifier for a specific routing instance. In this example, there are three routing instances: vpn1, vpn2, and vpn3, each with VRF table label enabled. The classifier exp-classifier-global is applied to vpn1 and vpn2. The classifier exp-classifier-3 is applied to vpn3.

```
[edit routing-instances]
vpn1 {
    vrf-table-label;
}
vpn2 {
    vrf-table-label;
}
vpn3 {
    vrf-table-label;
}

[edit class-of-service routing-instances]
all {
    classifiers {
        exp exp-classifier-global;
    }
}
vpn3 {
    classifiers {
        exp exp-classifier-3;
    }
}
```

Configuring a Wildcard Routing Instance

Configure a wildcard routing instance and override the wildcard with a specific routing instance. In this example, there are three routing instances: vpn-red, vpn-yellow, and vpn-green, each with VRF table label enabled. The classifier exp-class-wildcard is applied to vpn-yellow and vpn-green. The classifier exp-class-red is applied to vpn-red.

```
[edit routing-instances]
vpn-red {
      vrf-table-label;
}
vpn-yellow {
      vrf-table-label;
}
vpn-green {
      vrf-table-label;
}

[edit class-of-service routing-instances]
vpn* {
      classifiers {
         exp exp-class-wildcard;
      }
}
vpn-red {
      classifiers {
         exp exp-class-red;
      }
}
```

Monitoring a Configuration

Display the MPLS EXP classifiers associated with two routing instances:

```
[edit class-of-service routing-instances]
vpn1 {
      classifiers {
         exp default;
      }
}
vpn2 {
      classifiers {
         exp class2;
      }
}

user@router>show class-of-service routing-instances
Routing Instance : vpn1
    Object          Name              Type              Index
    Classifier      exp-default       exp                  8

  Routing Instance : vpn2
    Object          Name              Type              Index
    Classifier      class2            exp               57507
```

Overriding the Default PLP on M320 and T-series Platforms

By default, the least significant bit of the code point sets the packet loss priority (PLP) value. For example, code point 000 is associated with PLP low, and code point 001 is associated with PLP high. In general, you can override this default by configuring a BA classifier, as discussed in “Classifier Types” on page 46.

NOTE: On M320 and T-series platforms, you configure the rewrite rule for loss-priority high by setting the least-significant bit or by setting loss-priority high within a multifield classifier. If the least-significant bit is set, the loss-priority is high, regardless of how you configure the multifield classifier. However, if you do not configure the least-significant bit, you can overwrite the loss priority using a multifield classifier.

For more information about MF classifiers, see the *JUNOS Policy Framework Configuration Guide*.

Example: Overriding the Default PLP on M320 and T-series Platforms

Override the default PLP:

1. The least-significant bit of the code point is 1; therefore, the loss priority is high on M320 and T-series platforms. The loss priority is set to low for the code point 111; however, on M320 and T-series platforms, this loss-priority setting has no effect.

```
[edit class-of-service]
classifiers {
    dscp ba-classifier {
      forwarding-class expedited-forwarding {
        loss-priority low code-points 111;
      }
    }
}
```

2. For M320 and T-series platforms, this MF classifier overrides the default.

```
[edit firewall filter ef-filter term ef-multifield]
then {
    forwarding-class expedited-forwarding;
    loss-priority low;
}
```

Classifying Frame Relay Traffic

For J-series Services Router interfaces with Frame Relay encapsulation, you can set the loss priority of Frame Relay traffic, based on the discard eligibility (DE) bit. For each incoming frame with the DE bit containing the code point value 0 or 1, you can configure a Frame Relay loss priority value of low, medium-low, medium-high, or high.

You can apply a classifier to the same interface on which you configure a Frame Relay loss priority value. The Frame Relay loss priority map is applied first, followed by the classifier. The classifier can change the loss priority to a higher value only (for example, from low to high). If the classifier specifies a loss priority with a lower value than the current loss priority of a particular packet, the classifier does not change the loss priority of that packet.

This section is organized as follows:

Assigning the Default Frame Relay Loss Priority Map to an Interface

The default Frame Relay loss priority map contains the following settings:

```
loss-priority low code-point 0;
loss-priority high code-point 1;
```

This default map sets the loss priority to low for each incoming frame with the DE bit containing the 0 code point value. The map sets the loss priority to high for each incoming frame with the DE bit containing the 1 code point value.

To assign the default map to an interface, include the frame-relay-de default statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* loss-priority-maps] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
loss-priority-maps]
frame-relay-de default;
```

Defining a Custom Frame Relay Loss Priority Map

To define a custom Frame Relay loss priority map, include the following statements at the [edit class-of-service] hierarchy level:

```
[edit class-of-service]
loss-priority-maps {
    frame-relay-de map-name {
        loss-priority (low | medium-low | medium-high | high) code-point (0 | 1);
    }
}
```

A custom loss priority map sets the loss priority to low, medium-low, medium-high or high for each incoming frame with the DE bit containing the specified 0 or 1 code point value.

Applying the Map to a Logical Interface

The map does not take effect until you apply it to a logical interface. To apply a map to a logical interface, include the frame-relay-de *map-name* statement at the [edit class-of-service interfaces *interface-name* unit *logical-unit-number* loss-priority-maps] hierarchy level:

```
[edit class-of-service interfaces interface-name unit logical-unit-number
loss-priority-maps]
frame-relay-de map-name;
```

Verifying Your Configuration

To verify your configuration, you can issue the following operational-mode commands:

- show class-of-service forwarding-table loss-priority-map
- show class-of-service forwarding-table loss-priority-map mapping
- show chassis forwarding
- show pfe fwdd