Mark Packets to Change Priority Settings

Packet Marking Overview

You can use packet marking in input policy maps to set or modify the attributes for traffic belonging to a specific class. For example, you can change the CoS value in a class or set IP DSCP or IP precedence values for a specific type of traffic. These new values are then used to determine how the traffic should be treated.

Default Marking

When an ingress or egress interface adds VLAN tags or MPLS labels, it requires a default value for the class of service and EXP values that go into those tags and labels.

On the router, one ingress default QoS mapping profile and one egress default QoS mapping profile are created and configured per device during initialization.

Packet Marking

The packet marking feature, also called explicit marking, provides users with a means to differentiate packets based on the designated markings. The router supports ingress and egress packet marking.

Supported Packet Marking Operations

This table shows the supported packet marking operations.

Supported Mark Types	Range	Support for Unconditional Marking	Support for Conditional Marking
set discard-class	0-1	ingress	No
set dscp	0-63	ingress	No

Supported Mark Types	Range	Support for Unconditional Marking	Support for Conditional Marking
set mpls experimental topmost	0-7	ingress	No
set precedence	0-7	ingress	No
set qos-group	0-7	ingress	No

Class-based Unconditional Packet Marking Feature and Benefits

The packet marking feature allows you to partition your network into multiple priority levels or classes of service, as follows:

- Use QoS unconditional packet marking to set the IP precedence or IP DSCP values for packets entering the network. Routers within your network can then use the newly marked IP precedence values to determine how the traffic should be treated.

  On ingress direction, after matching the traffic based on either the IP Precedence or DSCP value, you can set it to a particular discard-class. Weighted random early detection (WRED), a congestion avoidance technique, thereby uses discard-class values to determine the probability that a packet is dropped.

- Use QoS unconditional packet marking to assign MPLS packets to a QoS group. The router uses the QoS group to determine how to prioritize packets for transmission. To set the QoS group identifier on MPLS packets, use the **set qos-group** command in policy map class configuration mode.

Note   Setting the QoS group identifier does not automatically prioritize the packets for transmission. You must first configure an egress policy that uses the QoS group.

- Mark Multiprotocol Label Switching (MPLS) packets by setting the EXP bits within the imposed or topmost label.
- Mark packets by setting the value of the *qos-group* argument.
- Mark packets by setting the value of the *discard-class* argument.

Note   *qos-group* and *discard-class* are variables internal to the router, and are not transmitted.

The configuration task is described in the Configure Class-based Unconditional Packet Marking, on page 2.

Configure Class-based Unconditional Packet Marking

This configuration task explains how to configure the following class-based, unconditional packet marking features on your router:

- IP precedence value

- IP DSCP value

- QoS group value (ingress only)

- CoS value (egress only on Layer 3 subinterfaces)

- MPLS experimental value

- Discard class

Note IPv4 and IPv6 QoS actions applied to MPLS tagged packets are not supported. The configuration is accepted, but no action is taken.

Configuration Example

Follow these steps to configure unconditional packet marking features on your router.

1. Create or modify a policy map that can be attached to one or more interfaces to specify a service policy and enter the policy map configuration mode.

2. Configure an interface and enter the interface configuration mode.

3. Attach a policy map to an input or output interface to be used as the service policy for that interface.

Configuration Example

```
router# configure
router(config)# interface hundredGigE 0/0/0/24
router(config-pmap)# policy-map policy1
Router(config-int)# commit
```

Running Configuration

```
router(config)# policy-map policy1

class-map match-any class1
 match protocol ipv4
 end-class-map
!
!
policy-map policy1
 class class1
  set precedence 1
 !
 class class-default
 !
 end-policy-map
!
interface HundredGigE0/0/0/24
 service-policy input policy1

!
```

Verification

Run this command to display policy configuration information for all classes configured for all service policies on the specified interface.

```
router# show run interface hundredGigE 0/0/0/24
```

Class-based Unconditional Packet Marking: Examples

These are typical examples for class-based unconditional packet marking.

IP Precedence Marking Configuration: Example

In this example, a service policy called *policy1* is created. This service policy is associated to a previously defined class map called *class1* through the use of the **class** command, and then the service policy is attached to the output HundredGigE interface 0/7/0/1. The IP precedence bit in the ToS byte is set to 1:

```
policy-map policy1
  class class1
    set precedence 1
!
interface HundredGigE 0/7/0/1
  service-policy output policy1
```

IP DSCP Marking Configuration: Example

In this example, a service policy called policy1 is created. This service policy is associated to a previously defined class map through the use of the **class** command. In this example, it is assumed that a class map called class1 was previously configured and new class map called class2 is created.

In this example, the IP DSCP value in the ToS byte is set to 5:

```
policy-map policy1
  class class1
    set dscp 5

  class class2
    set dscp ef
```

After you configure the settings shown for voice packets at the edge, all intermediate routers are configured to provide low-latency treatment to the voice packets, as follows:

```
class-map voice
  match dscp ef
policy-map qos-policy
  class voice
    priority level 1
    police rate percent 10
```

QoS Group Marking Configuration: Example

In this example, a service policy called *policy1* is created. This service policy is associated to a class map called *class1* through the use of the **class** command, and then the service policy is attached in the input direction on a HundredGigE 0/7/0/1. The qos-group value is set to 1.

```
class-map match-any class1
  match protocol ipv4
  match access-group ipv4 101

policy-map policy1
  class class1
    set qos-group 1
  !
interface HundredGigE 0/7/0/1
  service-policy input policy1
```

Note   The **set qos-group** command is supported only on an ingress policy.

CoS Marking Configuration: Example

In this example, a service policy called *policy1* is created. This service policy is associated to a class map called *class1* through the use of the **class** command, and then the service policy is attached in the output direction on a HundredGigE 0/7/0/1.100. The IEEE 802.1p (CoS) bits in the Layer 2 header are set to 1.

```
class-map match-any class1
  match protocol ipv4
  match access-group ipv4 101

policy-map policy1
  class class1
    set cos 1
  !

interface HundredGigE 0/7/0/1.100
  service-policy input policy1
```

MPLS Experimental Bit Imposition Marking Configuration: Example

In this example, a service policy called *policy1* is created. This service policy is associated to a class map called *class1* through the use of the **class** command, and then the service policy is attached in the input direction on a HundredGigE 0/7/0/1. The MPLS EXP bits of all imposed labels are set to 1.

```
class-map match-any class1
  match protocol ipv4
  match access-group ipv4 101

policy-map policy1
  class class1
    set mpls exp imposition 1
 !
interface HundredGigE 0/7/0/1
  service-policy input policy1
```

✎

Note   The **set mpls exp imposition** command is supported only on an ingress policy.

MPLS Experimental Topmost Marking Configuration: Example

In this example, a service policy called *policy1* is created. This service policy is associated to a class map called *class1* through the use of the **class** command, and then the service policy is attached in the output direction on a HundredGigE 0/7/0/1. The MPLS EXP bits on the TOPMOST label are set to 1:

```
class-map match-any class1
  match mpls exp topmost 2

policy-map policy1
  class class1
    set mpls exp topmost 1
  !
interface HundredGigE 0/7/0/1
  service-policy output policy1
```

IP Precedence Compared to IP DSCP Marking

If you need to mark packets in your network and all your devices support IP DSCP marking, use the IP DSCP marking to mark your packets because the IP DSCP markings provide more unconditional packet marking options. If marking by IP DSCP is undesirable, however, or if you are unsure if the devices in your network support IP DSCP values, use the IP precedence value to mark your packets. The IP precedence value is likely to be supported by all devices in the network.

You can set up to 8 different IP precedence markings and 64 different IP DSCP markings.

Configure DSCP CS7 (Precedence 7)

See the following example to configure options in DSCP for a particular source address in IPv4 packets.

Configuration Example

```
policy-map policy1
 class class1
  set dscp cs7
 !
```

In-Place Policy Modification

The In-Place policy modification feature allows you to modify a QoS policy even when the QoS policy is attached to one or more interfaces. A modified policy is subjected to the same checks that a new policy is subject to when it is bound to an interface. If the policy-modification is successful, the modified policy takes effect on all the interfaces to which the policy is attached. However, if the policy modification fails on any

one of the interfaces, an automatic rollback is initiated to ensure that the pre-modification policy is in effect on all the interfaces.

You can also modify any class map used in the policy map. The changes made to the class map take effect on all the interfaces to which the policy is attached.

Note	• The QoS statistics for the policy that is attached to an interface are lost (reset to 0) when the policy is modified.
	• When a QoS policy attached to an interface is modified, there might not be any policy in effect on the interfaces in which the modified policy is used for a short period of time.
	• An in-place modification of an ACL does not reset the policy-map statistics counter.

Verification

If unrecoverable errors occur during in-place policy modification, the policy is put into an inconsistent state on target interfaces. No new configuration is possible until the configuration session is unblocked. It is recommended to remove the policy from the interface, check the modified policy and then re-apply accordingly.

Recommendations for Using In-Place Policy Modification

For a short period of time while a QoS policy is being modified, there might not be any policy in effect on the interfaces in which the modified policy is used. For this reason, modify QoS policies that affect the fewest number of interfaces at a time. Use the **show policy-map targets** command to identify the number of interfaces that will be affected during policy map modification.