



Mark Packets to Change Priority Settings

- [Packet Marking Overview, on page 1](#)
- [Class-based Unconditional Packet Marking Feature and Benefits, on page 7](#)
- [Configure Class-based Unconditional Packet Marking, on page 8](#)
- [Class-based Unconditional Packet Marking: Examples, on page 9](#)
- [IP Precedence Compared to IP DSCP Marking, on page 14](#)
- [In-Place Policy Modification, on page 14](#)

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.

QoS Behavior for Generic Routing Encapsulation (GRE) Tunnels

Table 1: Feature History Table

Feature Name	Release Information	Feature Description
QoS Behavior for Generic Routing Encapsulation (GRE) Tunnels: Default Marking	Release 7.3.1	With the support for GRE encapsulation and decapsulation tunnel interfaces, there are some important updates to QoS behavior for GRE tunnels. These updates are applicable for default packet marking and involve Type of Service (ToS) and MPLS experimental bits.

GRE Encapsulation

If you do not configure Type of Service (ToS), the outer IP precedence value or the differentiated services code point (DSCP) value is copied from the inner IP header. If you configure ToS, the outer IP precedence value or DSCP value is as per the ToS configuration.

GRE Decapsulation

During decapsulation, the MPLS experimental bits (EXP) are derived from the outer IP packet.

For more information on GRE tunnels, see the *Interfaces Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 7.3.x*.

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.

Table 2: Supported Packet Marking Operations for Egress and Ingress

Supported Mark Types	Range	Support for Unconditional Marking	Support for Conditional Marking
set discard-class	0-1	ingress	No
set dscp	0-63	ingress, egress	No
set mpls experimental topmost	0-7	ingress, egress	No
set precedence	0-7	ingress, egress	No
set qos-group	0-31	ingress	No
set cos	0-7	ingress, egress	No
set dei	0-1	ingress, egress	No

QoS Behavior for Generic Routing Encapsulation (GRE) Tunnels

Table 3: Feature History Table

Feature Name	Release Information	Feature Description
QoS Behavior for Generic Routing Encapsulation (GRE) Tunnels: Explicit Marking	Release 7.3.1	With the support for GRE encapsulation and decapsulation tunnel interfaces, there are some important updates to QoS behavior for GRE tunnels. These updates are applicable for explicit packet marking and involve QoS behavior during ingress and egress.

GRE Encapsulation

During encapsulation of IPv4/IPv6 payload inside the GRE header, QoS behavior is as follows:

- Ingress: QoS supports classification on the payload Layer 3 fields or EXP and remarking payload IP header DSCP.
- Egress: QoS supports setting outer GRE IP header DSCP. It doesn't overwrite the Tunnel Type of Service (ToS) configuration and doesn't remark GRE IP header DCSP.

GRE Decapsulation

During decapsulation of the outer GRE header (during which the inner IPv4/IPv6/MPLS payload is forwarded to the next-hop router), QoS behavior is as follows:

- Ingress: QoS supports classification on Layer 3 fields of outer GRE using the **set qos-group** command. Setting DSCP on the ingress interface sets DSCP for the inner headers.
- Egress: QoS supports classification using **qos-group** to set DSCP or EXP for egress packets.

For more information on GRE tunnels, see the *Interfaces Configuration Guide for Cisco 8000 Series Routers, IOS XR Release 7.3.x*.

Classification and Marking for L2 Traffic

Table 4: Feature History Table

Feature Name	Release Information	Feature Description
Classification and Marking for L2 Traffic	Release 7.7.1	You can now perform packet remarking on Layer 2 subinterfaces. Traffic marking allows you to fine tune the traffic attributes on your network. The increased granularity helps single out traffic that requires special handling, and achieves optimal application performance.

Traffic marking is a method used to identify certain traffic types for unique handling, effectively partitioning network traffic into different categories. Cisco 8000 Series routers support Layer 2 marking of Ethernet packets in the egress and ingress directions.

Configuration Example for Classification and Marking for L2 Traffic

Follow these steps to enable this feature:

For guidelines on creating a traffic class and egress traffic classification, refer the *Create a Traffic Class* section in the *Classify Packets to Identify Specific Traffic* chapter.

- Enable Layer 2 configuration. Refer the [L2VPN Configuration Guide](#) for details.
- Enable egress remarking with the **qos-group** match type.
- Verify the egress remark QoS policy configuration.
- Enable egress traffic remarking.

Egress Remarking With *qos-group* Match Type

/ Associate a specific qos-group with a traffic class using the set qos-group command */*

```
Router# configure terminal
Router(config)# policy-map INGRESS_L2_AC
Router(config-pmap)# class CONTROL_PLANE
Router(config-pmap-c)# set traffic-class 7
Router(config-pmap-c)# set qos-group 27
Router(config-pmap-c)# exit

Router(config-pmap)# class VOIP
Router(config-pmap-c)# set traffic-class 6
Router(config-pmap-c)# set qos-group 26
Router(config-pmap-c)# exit

Router(config-pmap)# class VIDEO_STREAM
Router(config-pmap-c)# set traffic-class 5
Router(config-pmap-c)# set qos-group 25
Router(config-pmap-c)# exit

Router(config-pmap)# class TRANSACTIONAL_DATA
Router(config-pmap-c)# set traffic-class 4
Router(config-pmap-c)# set qos-group 24
Router(config-pmap-c)# exit

Router(config-pmap)# class DB_SYNC
Router(config-pmap-c)# set traffic-class 3
Router(config-pmap-c)# set qos-group 23
Router(config-pmap-c)# exit

Router(config-pmap)# class BULK_DATA
Router(config-pmap-c)# set traffic-class 2
Router(config-pmap-c)# set qos-group 22
Router(config-pmap-c)# exit

Router(config-pmap)# class SCAVENGER
Router(config-pmap-c)# set traffic-class 1
Router(config-pmap-c)# set qos-group 21
Router(config-pmap-c)# exit

Router(config-pmap)# class class-default
```

```

Router(config-pmap-c) # end-policy-map
Router(config) # commit

/* Associate egress policy map with the qos-groups */

Router# configure terminal
Router(config) # policy-map REMARK_EGRESS_QG_L2_AC
Router(config-pmap) # class QoS_GROUP_27
Router(config-pmap-c) # set cos 7
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_26
Router(config-pmap-c) # set cos 7
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_25
Router(config-pmap-c) # set cos 5
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_24
Router(config-pmap-c) # set cos 5
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_23
Router(config-pmap-c) # set cos 3
Router(config-pmap-c) # set dei 0
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_22
Router(config-pmap-c) # set cos 3
Router(config-pmap-c) # set dei 1
Router(config-pmap-c) # exit

Router(config-pmap) # class QoS_GROUP_21
Router(config-pmap-c) # set cos 1
Router(config-pmap-c) # set dei 1
Router(config-pmap-c) # exit

Router(config-pmap) # class class-default
Router(config-pmap-c) # set dei 1
Router(config-pmap-c) # exit
Router(config) # commit

/* Associate Policy-Map REMARK_EGRESS_QG_L2_AC With the Designated Subinterface */

Router(config) # interface bundle-Ether 1.104
Router(config-subif) # service-policy output REMARK_EGRESS_QG_L2_AC
Router(config-subif) # commit

```

Verification

Verify the egress remark QoS policy configuration:

```
Router# show policy-map interface Bundle-Ether 1.104 output
```

```
Bundle-Ether1.104 output: REMARK_EGRESS_QG_L2_AC
```

```

Class QoS_GROUP_7
  Classification statistics          (packets/bytes)      (rate - kbps)
  Matched                          :          21319554/29847375600      985424
  Transmitted                       :          21319554/29847375600      985424
  Total Dropped                     :                   0/0                          0
Class QoS_GROUP_6

```

```

Classification statistics          (packets/bytes)    (rate - kbps)
  Matched                        :      21581741/30214437400      997540
  Transmitted                     :      21581741/30214437400      997540
  Total Dropped                   :                   0/0              0
Class QoS_GROUP_5
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :      106597824/149236953600      4927114
    Transmitted                     :      106597824/149236953600      4927114
    Total Dropped                   :                   0/0              0
Class QoS_GROUP_4
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :      490350115/686490161000      22664721
    Transmitted                     :      490350115/686490161000      22664721
    Total Dropped                   :                   0/0              0
Class QoS_GROUP_3
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :                   0/0              0
    Transmitted                     :                   0/0              0
    Total Dropped                   :                   0/0              0
Class QoS_GROUP_2
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :                   0/0              0
    Transmitted                     :                   0/0              0
    Total Dropped                   :                   0/0              0
Class QoS_GROUP_1
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :                   0/0              0
    Transmitted                     :                   0/0              0
    Total Dropped                   :                   0/0              0
Class class-default
  Classification statistics          (packets/bytes)    (rate - kbps)
    Matched                        :      1087299229/1522218920600      50256530
    Transmitted                     :      1087299229/1522218920600      50256530
    Total Dropped                   :                   0/0              0
Policy Bag Stats time: 1657531125148 [Local Time: 07/11/22 09:18:45.148]

```

Remarking of Egress Traffic

```

Router# configure terminal
Router(config)# policy-map REMARK_EGRESS_L2_AC
Router(config-pmap)# class CONTROL_PLANE
Router(config-pmap-c)# set traffic-class 7
Router(config-pmap-c)# exit

Router(config-pmap)# class VOIP
Router(config-pmap-c)# set traffic-class 6
Router(config-pmap-c)# set cos 7
Router(config-pmap-c)# exit

Router(config-pmap)# class VIDEO_STREAM
Router(config-pmap-c)# set traffic-class 5
Router(config-pmap-c)# set cos 5
Router(config-pmap-c)# exit

Router(config-pmap)# class TRANSACTIONAL_DATA
Router(config-pmap-c)# set traffic-class 4
Router(config-pmap-c)# set cos 5
Router(config-pmap-c)# exit

Router(config-pmap)# class DB_SYNC
Router(config-pmap-c)# set traffic-class 3
Router(config-pmap-c)# set dei 0
Router(config-pmap-c)# exit

```

```
Router(config-pmap)# class BULK_DATA
Router(config-pmap-c)# set traffic-class 2
Router(config-pmap-c)# set cos 3
Router(config-pmap-c)# set dei 0
Router(config-pmap-c)# exit

Router(config-pmap)# class SCAVENGER
Router(config-pmap-c)# set traffic-class 1
Router(config-pmap-c)# exit

Router(config-pmap)# class class-default
Router(config-pmap-c)# set dei 1
Router(config-pmap-c)# end-policy-map
Router(config)# commit
```

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 8.

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
- 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 output 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

```

Queuing and Marking Policies: Examples

These examples explain how queuing and marking can be applied to an interface.

Ingress Traffic Classification and QoS Mapping Configuration for Input Interfaces

In this example, a policy map named "INGRESS_L3" is configured, specifically for classifying and marking incoming traffic.

```

RP/0/RP0/CPU0:Router#show policy-map pmap-name INGRESS_L3 detail

class-map match-all CONTROL_PLANE
match dscp cs7
end-class-map
!
class-map match-all VOIP
match dscp cs6
end-class-map
!
class-map match-all VIDEO_STREAM
match dscp cs5
end-class-map
!
class-map match-all TRANSACTIONAL_DATA
match dscp cs4
end-class-map
!
class-map match-all DB_SYNC
match dscp cs3
end-class-map
!

```

```

class-map match-all BULK_DATA
match dscp cs2
end-class-map
!
class-map match-all SCAVENGER
match dscp cs1
end-class-map
!
policy-map INGRESS_L3
class CONTROL_PLANE
set traffic-class 7
set qos-group 27
!
class VOIP
set traffic-class 6
set qos-group 26
!
class VIDEO_STREAM
set traffic-class 5
set qos-group 25
!
class TRANSACTIONAL_DATA
set traffic-class 4
set qos-group 24
!
class DB_SYNC
set traffic-class 3
set qos-group 23
!
class BULK_DATA
set traffic-class 2
set qos-group 22
!
class SCAVENGER
set traffic-class 1
set qos-group 21
!
class class-default
!
end-policy-map
!

```

Egress Queueing Policy Configuration

In this example, a policy map named "EGRESS_QOS_QUEUEING" is configured for the egress (outgoing) direction.

```
RP/0/RP0/CPU0:Router#show policy-map pmap-name EGRESS_QOS_QUEUEING detail
```

```

class-map match-any TC7_CONTROL_PLANE
match traffic-class 7
end-class-map
!
class-map match-any TC6_VOIP
match traffic-class 6
end-class-map
!
class-map match-any TC5_VIDEO_STREAM
match traffic-class 5
end-class-map
!
class-map match-any TC4_TRANSACTIONAL_DATA
match traffic-class 4
end-class-map

```

```

!
class-map match-any TC3_DB_SYNC
match traffic-class 3
end-class-map
!
class-map match-any TC2_BULK_DATA
match traffic-class 2
end-class-map
!
class-map match-any TC1_SCAVENGER
match traffic-class 1
end-class-map
!
policy-map EGRESS_QOS_QUEUEING
class TC7_CONTROL_PLANE
priority level 1
!
class TC6_VOIP
priority level 2
!
class TC5_VIDEO_STREAM
bandwidth remaining ratio 20
random-detect 50 mbytes 100 mbytes
!
class TC4_TRANSACTIONAL_DATA
bandwidth remaining ratio 20
queue-limit 75 mbytes
!
class TC3_DB_SYNC
bandwidth remaining ratio 10
!
class TC2_BULK_DATA
bandwidth remaining ratio 10
!
class TC1_SCAVENGER
bandwidth remaining ratio 5
!
class class-default
!
end-policy-map
!

```

Egress Marking Policy Configuration

In this example, a policy map named "EGRESS_REMARK" is configured for the egress (outgoing) direction.

```
RP/0/RP0/CPU0:Router#show policy-map pmap-name EGRESS_REMARK detail
```

```

class-map match-any QG4
match qos-group 4
end-class-map
!
class-map match-any QG3
match qos-group 3
end-class-map
!
class-map match-any QG2
match qos-group 2
end-class-map
!
policy-map EGRESS_REMARK
class QG4
set dscp af42
!

```

```

class QG3
  set dscp af33
!
class QG2
  set dscp af21
!
class class-default
!
end-policy-map
!

```

Output Interface Configuration with Queuing Policy and Remark Policy Applied

```

interface FourHundredGigE0/2/0/29
mtu 10000
service-policy output EGRESS_REMARK
service-policy output EGRESS_QOS_QUEUEING
ipv4 address 18.29.0.1 255.255.255.0
ipv6 address 18:1d::1/96
!

```

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.

