Packet Classification Overview

Packet classification involves categorizing a packet within a specific group (or class) and assigning it a traffic descriptor to make it accessible for QoS handling on the network. The traffic descriptor contains information about the forwarding treatment (quality of service) that the packet should receive. Using packet classification, you can partition network traffic into multiple priority levels or classes of service. The source agrees to adhere to the contracted terms and the network promises a quality of service. Traffic policers and traffic shapers use the traffic descriptor of a packet to ensure adherence to the contract.

Traffic policers and traffic shapers rely on packet classification features, such as IP precedence, to select packets (or traffic flows) traversing a router or interface for different types of QoS service. After you classify packets, you can use other QoS features to assign the appropriate traffic handling policies including congestion management, bandwidth allocation, and delay bounds for each traffic class.

The Modular Quality of Service (QoS) CLI (MQC) defines the traffic flows that must be classified, where each traffic flow is called a class of service, or class. Later, a traffic policy is created and applied to a class. All traffic not identified by defined classes fall into the category of a default class.

You can classify packets at the ingress on L3 subinterfaces for (CoS, DEI) for IPv4, IPv6, and MPLS flows. IPv6 packets are forwarded by paths that are different from those for IPv4. To enable classification of IPv6 packets based on (CoS, DEI) on L3 subinterfaces, run the hw-module profile qos ipv6 short-l2qos-enable command and reboot the line card for the command to take effect.

Restrictions for NCS 560 Series Routers

The **hw-module profile qos ingress-model peering** command is not supported.

Traffic Class Elements

The purpose of a traffic class is to classify traffic on your router. Use the **class-map** command to define a traffic class.

A traffic class contains three major elements:

- A name
- A series of **match** commands - to specify various criteria for classifying packets.
- An instruction on how to evaluate these **match** commands (if more than one **match** command exists in the traffic class)

Packets are checked to determine whether they match the criteria that are specified in the **match** commands. If a packet matches the specified criteria, that packet is considered a member of the class and is forwarded according to the QoS specifications set in the traffic policy. Packets that fail to meet any of the matching criteria are classified as members of the default traffic class.

This table shows the details of match types that are supported on the router.

Match Type Supported	Min, Max	Max Entries	Support for Match NOT	Support for Ranges	Direction Supported on Interfaces
IPv4 DSCP IPv6 DSCP DSCP	(0,63)	64	Yes	Yes	Ingress
IPv4 Precedence IPv6 Precedence Precedence	(0,7)	8	Yes	No	Ingress
MPLS Experimental Topmost	(0,7)	8	Yes	No	Ingress
Access-group	Not applicable	8	No	Not applicable	Ingress
QoS-group	(1,7) (1,511) for peering profile	7	No	No	• Egress • Ingress for QoS Policy Propagation Using Border Gateway Protocol (QPPB) • Ingress for peering profile
Traffic-class	(1,7)	7	No	No	• Egress
CoS	(0,7)	8	No	Yes	Ingress

Match Type Supported	Min, Max	Max Entries	Support for Match NOT	Support for Ranges	Direction Supported on Interfaces
DEI	(0,1)	1	No	No	Ingress
Protocol	(0,255)	1	Yes	Not applicable	Ingress

Note Egress queue statistics are displayed only for those classes which have a corresponding match criteria in the egress. Therefore, if you have a **set traffic-class** *x* configured in the ingress, you must have a corresponding **match traffic-class** *x* in the egress, in order to see the statistics in the egress side.

Note A maximum value of up to 64 unique queues is supported. Each unique queue-limit consumes one rate profile in the Traffic manager. Out of 64 unique queues, 4 are reserved and 60 are usable. So, you can program 60 unique queue-limit in the hardware.

Consider the following points while planning for the policy maps:

You need to be cautious of the unique queue-limit while configuring the egress policy-map. Following are some scenarios that can exhaust the 60 configurable rate-profiles.

- Having different shape rate, without configuring queue limits, could exhaust the rate-profiles as 10ms of GSR converts to different value in bytes based on the shape rate.
- Configuring queue limit in time could exhaust the rate-profiles. For example, 20 ms of 50 Mbps and 20 ms of 100 Mbps are two different values in bytes.

Tip You can avoid exhausting the rate-profile using multiple egress policy-maps. By configuring queue limits in absolute unit (bytes/kbytes/mbytes) and sharing between different policy-maps.

Default Traffic Class

Unclassified traffic (traffic that does not meet the match criteria specified in the traffic classes) is treated as belonging to the default traffic class.

If the user does not configure a default class, packets are still treated as members of the default class. However, by default, the default class has no enabled features. Therefore, packets belonging to a default class with no configured features have no QoS functionality. These packets are then placed into a first in, first out (FIFO) queue and forwarded at a rate determined by the available underlying link bandwidth.

For egress classification, match on **traffic-class** (1-7) is supported. Match **traffic-class 0** cannot be configured. The class-default in the egress policy maps to **traffic-class 0**.

This example shows how to configure a traffic policy for the default class:

```
configure
```

```
policy-map ingress_policy1
class class-default
 police rate percent 30
!
```

Create a Traffic Class

To create a traffic class containing match criteria, use the **class-map** command to specify the traffic class name, and then use the **match** commands in class-map configuration mode, as needed.

Guidelines

- Users can provide multiple values for a match type in a single line of configuration; that is, if the first value does not meet the match criteria, then the next value indicated in the match statement is considered for classification.
- Use the **not** keyword with the **match** command to perform a match based on the values of a field that are not specified.
- All **match** commands specified in this configuration task are considered optional, but you must configure at least one match criterion for a class.
- If you specify **match-any**, one of the match criteria must be met for traffic entering the traffic class to be classified as part of the traffic class. This is the default. If you specify **match-all**, the traffic must match all the match criteria.
- From Release 7.7.1 onwards, for the **match access-group** command, QoS classification based on the packet length field in the IPv4 and IPv6 headers is supported. Prior to this, support was not available for packet length and TTL (time to live) fields.
- For the **match access-group** command, when an ACL list is used within a class-map, the deny action of the ACL is ignored and the traffic is classified based on the specified ACL match parameters.

  An empty ACL (contains no rules, only remarks), when used within a class-map permits all traffic by default, and the implicit deny condition doesn't work with an empty ACL. The corresponding **class-map** matches all traffic not yet matched by the preceding traffic classes.
- The **traffic-class** and **discard-class** are supported only in egress direction, and these are the only match criteria supported in egress direction.
- The egress default class implicitly matches **qos-group** 0 for marking policy and **traffic-class** 0 for queuing policy.
- Multicast takes a system path that is different than unicast on router, and they meet later on the egress in a multicast-to-unicast ratio of 20:80 on a per interface basis. This ratio is maintained on the same priority level as that of the traffic.
- When conditional marking policy map is applied, the MPLS EXP value is set to 0 for multicast traffic.
- When an ingress policy-map is applied to mark an MPLS EXP topmost label, the MPLS EXP topmost and inner MPLS labels are marked for multicast traffic.
- Egress QoS for multicast traffic treats traffic classes 0-5 as low-priority and traffic classes 6-7 as high priority. Currently, this is not user-configurable.
- Egress shaping does not take effect for multicast traffic in the high priority (HP) traffic classes. It only applies to unicast traffic.

- If you set a traffic class at the ingress policy and do not have a matching class at egress for the corresponding traffic class value, then the traffic at ingress with this class will not be accounted for in the default class at the egress policy map.
- Only traffic class 0 falls in the default class. A non-zero traffic class assigned on ingress but with no assigned egress queue, falls neither in the default class nor any other class.

Configuration Example

You have to accomplish the following to complete the traffic class configuration:

1. Creating a class map
2. Specifying the match criteria for classifying the packet as a member of that particular class

   (For a list of supported match types, see Traffic Class Elements, on page 2.)

```
Router# configure
Router(config)# class-map match-any qos-1
Router(config-cmap)# match qos-group 1
Router(config-cmap)# end-class-map
Router(config-cmap)# commit
```

Use this command to verify the class-map configuration:

```
Router#show class-map qos-1
1) ClassMap: qos-1    Type: qos
    Referenced by 2 Policymaps
```

Also see, Running Configuration, on page 8.

Also see, Verification, on page 9.

Related Topics

- Traffic Class Elements, on page 2
- Traffic Policy Elements, on page 6

Associated Commands

- class-map
- match access-group
- match dscp
- match mpls experimental topmost
- match precedence
- match qos-group

Traffic Policy Elements

A traffic policy contains three elements:

- Name
- Traffic class
- QoS policies

After choosing the traffic class that is used to classify traffic to the traffic policy, the user can enter the QoS features to be applied to the classified traffic.

The MQC does not necessarily require that the users associate only one traffic class to one traffic policy.

The order in which classes are configured in a policy map is important. The match rules of the classes are programmed into the TCAM in the order in which the classes are specified in a policy map. Therefore, if a packet can possibly match multiple classes, only the first matching class is returned and the corresponding policy is applied.

The router supports 32 classes per policy-map in the ingress direction and 8 classes per policy-map in the egress direction.

Note The policer maximum scale value is enhanced to 32K per system from Cisco IOS XR Release 7.1.1 and the scale value supported before this release is 4K per system.

This table shows the supported class-actions on the router.

Supported Action Types	Direction supported on Interfaces
minimum-bandwidth	egress
bandwidth-remaining	egress
mark	(See Packet Marking, on page 10)
police	ingress
priority	egress (level 1 to level 7)
queue-limit	egress
shape	egress
wred	egress

WRED supports **default** and **discard-class** options; the only values to be passed to the discard-class being 0 and 1.

Create a Traffic Policy

The purpose of a traffic policy is to configure the QoS features that should be associated with the traffic that has been classified in a user-specified traffic class or classes.

To configure a traffic class, see Create a Traffic Class, on page 4.

After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces to specify the traffic policy for those interfaces by using the **service-policy** command in interface configuration mode. With dual policy support, you can have two traffic policies, one marking and one queuing attached at the output. See, Attach a Traffic Policy to an Interface, on page 8.

Configuration Example

You have to accomplish the following to complete the traffic policy configuration:

1. Creating a policy map that can be attached to one or more interfaces to specify a service policy
2. Associating the traffic class with the traffic policy
3. Specifying the class-action(s) (see Traffic Policy Elements, on page 6)

```
Router# configure
Router(config)# policy-map  test-shape-1
Router(config-pmap)# class qos-1

/* Configure class-action ('shape' in this example).
Repeat as required, to specify other class-actions */
Router(config-pmap-c)# shape average percent 40
Router(config-pmap-c)# exit

/* Repeat class configuration as required, to specify other classes */

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

See, Running Configuration, on page 8.

See, Verification, on page 9.

Related Topics

- Traffic Policy Elements, on page 6
- Traffic Class Elements, on page 2

Associated Commands

- bandwidth
- bandwidth remaining
- class
- police
- policy-map

- priority
- queue-limit
- service-policy
- set discard-class
- set dscp
- set mpls experimental
- set precedence
- set qos-group
- shape

Attach a Traffic Policy to an Interface

After the traffic class and the traffic policy are created, you must attach the traffic policy to interface, and specify the direction in which the policy should be applied.

Configuration Example

You have to accomplish the following to attach a traffic policy to an interface:

1. Creating a traffic class and the associated rules that match packets to the class (see #unique_11)
2. Creating a traffic policy that can be attached to one or more interfaces to specify a service policy (see Create a Traffic Policy, on page 7)
3. Associating the traffic class with the traffic policy
4. Attaching the traffic policy to an interface, in the ingress or egress direction

Running Configuration

```
RP/0/RP0/CPU0:R1# show run interface TwentyFiveGigE0/0/0/26.1

interface TwentyFiveGigE0/0/0/26.1 l2transport
encapsulation dot1q 25
service-policy input cos
!

RP/0/RP0/CPU0:R1# show run policy-map cos

policy-map cos
class cos1
police rate 3 mbps
!
!
class cos2
police rate 2 mbps
!
!
class cos3
police rate 3 mbps
!
```

```
!
class class-default
police rate 4 mbps
!
!
end-policy-map
!

RP/0/RP0/CPU0:R1#
```

```
Router# configure
Router(config)# interface HundredGigE 0/6/0/18
Router(config-int)# service-policy output test-shape-1
Router(config-int)# commit

Running Configuration

/* Class-map configuration */

class-map match-any traffic-class-1
match traffic-class 1
end-class-map
!
- - -
- - -

/* Traffic policy configuration */
policy-map test-shape-1
class traffic-class-1
shape average percent 40
!
class class-default
!
end-policy-map
!
- - -
- - -

/* Attaching traffic policy to an interface in egress direction */
interface HundredGigE0/6/0/18
service-policy output test-shape-1
!
```

Verification

```
Router# show qos interface hundredGigE 0/6/0/18 output

NOTE:- Configured values are displayed within parentheses Interface HundredGigE0/6/0/18 ifh
 0x30001f8  -- output policy
NPU Id:                        3
Total number of classes:       2
Interface Bandwidth:           100000000 kbps
VOQ Base:                      11112
VOQ Stats Handle:              0x88430698
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                             =   qos-1
Egressq Queue ID                         =   11113 (LP queue)
Queue Max. BW.                           =   40329846 kbps (40 %)
Queue Min. BW.                           =   0 kbps (default)
Inverse Weight / Weight                  =   1 / (BWR not configured)
Guaranteed service rate                  =   40000000 kbps
TailDrop Threshold                       =   50069504 bytes / 10 ms (default)
```

```
WRED not configured for this class

Level1 Class                              =   class-default
Egressq Queue ID                          =   11112 (Default LP queue)
Queue Max. BW.                            =   101803495 kbps (default)
Queue Min. BW.                            =   0 kbps (default)
Inverse Weight / Weight                   =   1 / (BWR not configured)
Guaranteed service rate                   =   50000000 kbps
TailDrop Threshold                        =   62652416 bytes / 10 ms (default)
WRED not configured for this class
```

Related Topics

- Traffic Policy Elements, on page 6
- Traffic Class Elements, on page 2

Associated Commands

- service-policy

Packet Marking

The packet marking feature provides users with a means to differentiate packets based on the designated markings. The router supports egress packet marking. match on **discard-class** on egress, if configured, can be used for a marking policy only.

The router also supports L2 ingress marking.

For ingress marking:

Ingress traffic— For the ingress pop operation, re-marking the customer VLAN tag (CoS, DEI) is not supported.

Egress traffic— The ingress ‘pop VLAN’ is translated to a ‘push VLAN’ for the egress traffic, and (CoS, DEI) marking is supported for newly pushed VLAN tags. If two VLAN tags are pushed to the packet header at the egress side, both inner and outer VLAN tags are marked. For example:

1. rewrite ingress tag pop 1 symmetric
2. rewrite ingress tag pop 2 symmetric
3. rewrite ingress tag translate 2-to-1 dot1q <> symmetric

Packet Marking Guidelines and Limitations

- While marking a packet, ensure you don’t set the IP DSCP (using the **set dscp** command) and the MPLS experimental imposition values (using the **set mpls experimental imposition** command) for the same class map. Else, neither the DSCP remarking nor the MPLS EXP values may take effect at the ingress, causing unintended traffic to flow across your network.
- The statistics and counters for the egress marking policy cannot be viewed on the router.
- For QOS EXP-Egress marking applied on a Layer 3 interface on Cisco NCS550x and NCS55Ax routers, there is a limit of two unique policy maps per NPU. This limit is three unique policy maps per NPU for routers that have the Cisco NC57 line cards installed.

You can apply these policies to as many interfaces as your system resources allow. However, if you apply more than the permitted limit of unique policies, you may encounter unexpected failure.

- For QOS egress marking (CoS, DEI) applied on a Layer 2 interface, there is a limit of 13 unique policy-maps per NPU. If you exceed this number, you may encounter unexpected failure.
- Cisco NCS series routers do not support push or translate operations for dot1ad.

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 cos	0-7	ingress	No
set dei	0-1	ingress	No
set discard-class	0-3	ingress	No
set dscp	0-63	ingress	No
set mpls experimental topmost	0-7	ingress	No
set precedence	0-7	ingress	No
set QoS-group	0-7	ingress	No
set traffic-class	0-7	ingress	No

Class-based Unconditional Packet Marking

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.

  If however, you set a discard-class of 3, the packet is dropped at ingress itself.

- 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 traffic class identifier on MPLS packets, use the **set traffic-class** command in policy map class configuration mode.

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

- Use QoS unconditional packet marking to assign packets to set the priority value of IEEE 802.1p/ Inter-Switch Link (ISL) packets. The router uses the CoS value to determine how to prioritize packets for transmission and can use this marking to perform Layer 2-to-Layer 3 mapping. To set the Layer 2 CoS value of an outgoing packet, use the **set cos** command in policy map configuration mode.
- Use QoS unconditional packet marking to mark a packet based on the drop eligible indicator value (DEI) bit on 802.1ad frames. To set the DEI value, use the **set dei** command to set the drop eligible indicator value (DEI) in policy map class configuration mode.

Note

- Unless otherwise indicated, the class-based unconditional packet marking for Layer 3 physical interfaces applies to bundle interfaces.

Setting QoS-group and DSCP at Ingress

With the introduction of this feature, you can set both qos-group and DSCP values within the same QoS policy that is applied in the ingress direction. You can use any permitted value to set the qos-group value.

Restrictions and Guidelines

The following restrictions and guidelines apply when you are setting qos-group and DSCP values within the same QoS policy that is applied in the ingress direction.

- You can set only up to eight DSCP values (determined by the **hw-module** config) in any QoS policy that contains both **set qos-group** and **set dscp** configurations.
- You cannot configure both set qos-group and set dscp settings in the *same* class. Each configuration must be in a *different* class, but within the same policy.

Configuring QoS-group and DSCP at Ingress

To configure a QoS policy that has both qos-group and set dscp configurations, at ingress you must:

1. Enable the QoS profile to set both qos-group and set dscp configurations.
2. Reload the chassis.
3. Specify class names and define class maps for these class names.
4. Match packets to these classes.
5. Create a policy map and include the class maps you created earlier.
6. Set the qos-group and dscp configurations within this policy map.
7. Attach the policy map to the interface.

Note Starting from Release 7.1.2, you can configure **set qos-group** and **set dscp** packet marking operations within same ingress QoS policy.

To configure static mapping between DSCP and qos-group, use the **hw-module profile qos qosg-dscp-mark-enable** command with only, set dscp values as 0, 8, 16, 24, 32, 40, 48, or 56 in the hardware profile.

```
RP/0/RP0/CPU0:router(config)#hw-module profile qos qosg-dscp-mark-enable 13 16
In order to activate this new qos profile, you must manually reload the chassis/all line
cards
RP/0/RP0/CPU0:router(config)#commit
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)#class-map cm-dscp-cs1
RP/0/RP0/CPU0:router(config-cmap)#match dscp cs1
RP/0/RP0/CPU0:router(config-cmap)#commit
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)# class-map cm-dscp-cs4
RP/0/RP0/CPU0:router(config-cmap)# match dscp cs4
RP/0/RP0/CPU0:router(config-cmap)#commit
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)#policy-map pm-in
RP/0/RP0/CPU0:router(config-pmap)#class cm-dscp-cs1
RP/0/RP0/CPU0:router(config-pmap-c)#set dscp cs2
RP/0/RP0/CPU0:router(config-pmap-c)#exit
RP/0/RP0/CPU0:router(config-pmap)# class cm-dscp-cs4
RP/0/RP0/CPU0:router(config-pmap-c)#set qos-group 2
RP/0/RP0/CPU0:router(config-pmap-c)#commit
RP/0/RP0/CPU0:router(config)#
RP/0/RP0/CPU0:router(config)#interface TenGigE 0/0/0/1
RP/0/RP0/CPU0:router(config-if)#service-policy input pm-in
RP/0/RP0/CPU0:router(config-if)#commit
RP/0/RP0/CPU0:router(config)# exit
```

Running Configuration

```
hw-module profile qos qosg-dscp-mark-enable 13 16

class-map match-any cm-dscp-cs1
match dscp cs1
 end-class-map
!
class-map match-any cm-dscp-cs4
match dscp cs4
 end-class-map
!

policy-map pm-in
class cm-dscp-cs1
  set dscp cs2
!
class cm-dscp-cs4
  set qos-group 2
!
 class class-default
!
 end-policy-map
!

interface TenGigE0/0/0/1
```

```
service-policy input pm-in
shutdown
!
```

Verification

```
RP/0/RP0/CPU0:ios#show policy-map pmap-name pm-in detail
class-map match-any cm-dscp-cs1
match dscp cs1
 end-class-map
!
class-map match-any cm-dscp-cs4
match dscp cs4
 end-class-map
!

policy-map pm-in
class cm-dscp-cs1
  set dscp cs2
!
class cm-dscp-cs4
  set qos-group 2
!
 class class-default
!
 end-policy-map
!
```

Associated Commands

- hw-module profile qos qosg-dscp-mark-enable

QoS Re-marking of IP Packets in Egress Direction

The router support the marking of IP DSCP bits of all IP packets to zero, in the egress direction. This feature helps to re-mark the priority of IP packets, which is mostly used in scenarios like IP over Ethernet over MPLS over GRE. This functionality is achieved using the ingress policy-map with **set dscp 0** option configured in class-default.

Configuration Example

```
Router# configure
Router(config)# policy-map ingress-set-dscp-zero-policy
Router(config-pmap)# class class-default
Router(config-pmap-c)# set dscp 0
Router(config-pmap-c)# end-policy-map
Router(config-pmap)# commit
```

Running Configuration

```
policy-map ingress-set-dscp-zero-policy
class class-default
  set dscp 0
!
end-policy-map
!
```

QoS Re-marking of Ethernet Packets in Egress Direction

The router supports Layer 2 marking of Ethernet packets in the egress direction.

QoS L2 Re-marking of Ethernet Packets in Egress Direction

The router supports Layer 2 marking of Ethernet packets in the egress direction.

To enable this feature, you must:

- Configure the policy maps for queuing and marking at the egress interface.
- Set traffic-class in the ingress and use **match traffic-class** in the egress for queuing.
- Ensure that the **set qos-group** command is configured in ingress policy and the corresponding **match qos-group** command is configured in the egress marking policy. If there is no corresponding QoS group, you will experience traffic failure.

  The ingress 'push VLAN' is translated to 'pop VLAN' for the egress traffic. In this case, (CoS, DEI) re-marking is not supported for the VLAN tag. For example:

  1. rewrite ingress tag push dot1q/dot1ad <> symmetric
  2. rewrite ingress tag push dot1q/dot1ad <> second-dot1q <> symmetric
  3. rewrite ingress tag translate 1-to-2 dot1q/dot1ad <> second-dot1q <> symmetric

Running Configuration

```
policy-map egress-marking
class qos1
set cos 1
!
class qos2
set cos 2
set dei 1
!
class qos3
set cos 3
!
class class-default
set cos 7
!
end-policy-map
!
```

QoS L2 Re-Marking of Ethernet Packets on L3 Flows in Egress Direction

The router supports Layer 2 marking of Ethernet packets on Layer 3 flows in the egress direction. To enable this feature, you must:

- Configure the policy maps for marking at the egress interface.
- Ensure that the **set qos-group** command is configured in ingress policy and the corresponding **match qos-group** command is configured in the egress marking policy. If there is no corresponding QoS group, you will experience traffic failure.

Restrictions

The following restrictions apply while configuring the Layer 2 marking of Ethernet packets on Layer 3 flows in the egress direction.

- Egress marking statistics are not available.
- Layer 2 (802.1p) Egress marking is supported on Layer 3 flows only for MPLS-to-IP traffic.

Running Configuration

Ingress Policy:

You must first set up the qos-group at ingress.

```
class-map match-any Class0
 match mpls experimental topmost 0

 end-class-map
class-map match-any Class1
 match mpls experimental topmost 1

 end-class-map
class-map match-any Class2
 match mpls experimental topmost 2

 end-class-map
class-map match-any Class3
 match mpls experimental topmost 3

 end-class-map
class-map match-any Class4
 match mpls experimental topmost 4

 end-class-map
class-map match-any Class5
 match mpls experimental topmost 5

 end-class-map
class-map match-any Class6
 match mpls experimental topmost 6
end-class-map
class-map match-any Class7
 match mpls experimental topmost 7

 end-class-map
!

policy-map ncs_input
 class Class7
  set traffic-class 7
    set qos-group 7
!
 class Class6
  set traffic-class 6
  set qos-group 6
!
 class Class5
  set traffic-class 5
    set qos-group 5
!
 class Class4
  set traffic-class 4
```

```
  set qos-group 4
 !
 class Class3
  set traffic-class 4
    set qos-group 3
 !
 class Class2
  set traffic-class 2
    set qos-group 2
 !
 class Class1
  set traffic-class 2
    set qos-group 1
 !
 class Class0
  set traffic-class 0
    set qos-group 0
 !
 end-policy-map
!
```

Egress Policy:

At the egress, run these commands to mark the packets.

```
class-map match-any qos7
match gos-group 7
 end-class-map
!
class-map match-any qos6
match gos-group 6
 end-class-map
!
class-map match-any qos5
match qos-group 5
 end-class-map
!
class-map match-any qos4
match gos-group 4
 end-class-map
!
class-map match-any qos3
match gos-group 3
 end-class-map
!
class-map match-any qos2
 match gos-group 2
 end-class-map
!
class-map match-any qos1
match gos-group 1
 end-class-map
!

policy-map ncs_output
 class qos7
  set cos 7
  !
 class qos6
  set cos 6
!
 class qos5
  set cos 5
  !
 class qos4
```

```
  set cos 4
!
 class qos3
  set cos 3
!
 class qos2
  set cos 2

!
 class qos1
  set cos 1
  !
 end-policy-map
!
```

QoS L2 Re-Marking of Ethernet Packets on L3 Flows in Egress Direction on L3 sub-interfaces

The router supports Layer 2 marking of Ethernet packets on Layer 3 flows in the egress direction on L3 subinterfaces.

To enable this feature, you must:

- Configure the policy maps for marking at the egress interface.
- Ensure that the **set qos-group** command is configured in ingress policy and the corresponding **match qos-group** command is configured in the egress marking policy. If there is no corresponding QoS group, you experience traffic failure.

Restrictions

The following restrictions apply while configuring the Layer 2 marking of Ethernet packets on Layer 3 flows in the egress direction.

- **set discard-class** is not supported in ingress policy with peering mode.
- Egress marking statistics are not available.
- Layer 2 (CoS, DEI) Egress marking is supported on Layer 3 flows on L3 subinterfaces for these types of traffic: IP-to-IP, IP-to-MPLS, and MPLS-to-IP traffic.

Running Configuration

Ingress Policy:

You must first set up the qos-group at ingress. This is applicable only when you want to mark packets at the egress.

```
class-map match-all COS0_DEI0
 match cos 0
 match dei 0
 end-class-map
class-map match-all COS0_DEI1
 match cos 0
 match dei 1
 end-class-map
class-map match-all COS1_DEI0
 match cos 1
 match dei 0
 end-class-map
class-map match-all COS1_DEI1
```

```
 match cos 1
 match dei 1
 end-class-map
class-map match-all COS2_DEI0
 match cos 2
 match dei 0
 end-class-map
class-map match-all COS2_DEI1
 match cos 2
 match dei 1
 end-class-map
class-map match-all COS3_DEI0
 match cos 3
 match dei 0
 end-class-map
class-map match-all COS3_DEI1
 match cos 3
 match dei 1
 end-class-map
class-map match-all COS4_DEI0
 match cos 4
 match dei 0
 end-class-map
class-map match-all COS4_DEI1
 match cos 4
 match dei 1
 end-class-map
class-map match-all COS5_DEI0
 match cos 5
 match dei 0
 end-class-map
class-map match-all COS5_DEI1
 match cos 5
 match dei 1
 end-class-map
class-map match-all COS6_DEI0
 match cos 6
 match dei 0
 end-class-map
class-map match-all COS6_DEI1
 match cos 6
 match dei 1
 end-class-map
class-map match-all COS7_DEI0
 match cos 7
 match dei 0
 end-class-map
class-map match-all COS7_DEI1
 match cos 7
 match dei 1
 end-class-map

policy-map ncs_input
 class COS7_DEI0
  set qos-group 7
  set discard-class 0
!
 class COS7_DEI1
  set qos-group 7
  set discard-class 1
!
 class COS6_DEI0
  set qos-group 6
  set discard-class 0
```

```
!
 class COS6_DEI1
  set qos-group 6
  set discard-class 1
!
 class COS5_DEI0
  set qos-group 5
  set discard-class 0
!
 class COS5_DEI1
  set qos-group 5
  set discard-class 1
!
 class COS4_DEI0
  set qos-group 4
  set discard-class 0
!
 class COS4_DEI1
  set qos-group 4
  set discard-class 1
!
 class COS3_DEI0
  set qos-group 3
  set discard-class 0
!
 class COS3_DEI1
  set qos-group 3
  set discard-class 1
!
 class COS2_DEI0
  set qos-group 2
  set discard-class 0
!
 class COS2_DEI1
  set qos-group 2
  set discard-class 1
!
 class COS1_DEI0
  set qos-group 1
  set discard-class 0
!
 class COS1_DEI1
  set qos-group 1
  set discard-class 1
!
 class COS0_DEI0
  set qos-group 0
  set discard-class 0
!
 class COS0_DEI1
  set qos-group 0
  set discard-class 1
!
```

Egress Policy:

At the egress, run these commands to mark the packets.

```
class-map match-all qos7_dc0
match qos-group 7
match discard-class 0
 end-class-map
!
class-map match-all qos7_dc1
match qos-group 7
```

```
match discard-class 1
 end-class-map
!

class-map match-all qos6_dc0
match qos-group 6
match discard-class 0
 end-class-map
!
class-map match-all qos6_dc1
match qos-group 6
match discard-class 1
 end-class-map
!
class-map match-all qos5_dc0
match qos-group 5
match discard-class 0
 end-class-map
!
class-map match-all qos5_dc1
match qos-group 5
match discard-class 1
 end-class-map
!
class-map match-all qos4_dc0
match qos-group 4
match discard-class 0
 end-class-map
!
class-map match-all qos4_dc1
match qos-group 4
match discard-class 1
 end-class-map
!
class-map match-all qos3_dc0
match qos-group 3
match discard-class 0
 end-class-map
!
class-map match-all qos3_dc1
match qos-group 3
match discard-class 1
 end-class-map
!
class-map match-all qos2_dc0
match qos-group 2
match discard-class 0
 end-class-map
!
class-map match-all qos2_dc1
match qos-group 2
match discard-class 1
 end-class-map
!
class-map match-all qos1_dc0
match qos-group 1
match discard-class 0
 end-class-map
!
class-map match-all qos1_dc1
match qos-group 1
match discard-class 1
 end-class-map
!
```

```
class-map match-all qos0_dc0
match qos-group 0
match discard-class 0
 end-class-map
!
class-map match-all qos0_dc1
match qos-group 0
match discard-class 1
 end-class-map
!


policy-map ncs_output
 class qos7_dc0
  set cos 7
  set dei 0
  set mpls experimental imposition 7
!
 class qos7_dc1
  set cos 7
  set dei 1
  set mpls experimental imposition 7
!
 class qos6_dc0
  set cos 6
  set dei 0
  set mpls experimental imposition 6
!
 class qos6_dc1
  set cos 6
  set dei 1
  set mpls experimental imposition 6
!
 class qos5_dc0
  set cos 5
  set dei 0
  set mpls experimental imposition 5
!
 class qos5_dc1
  set cos 5
  set dei 1
  set mpls experimental imposition 5
!
 class qos4_dc0
  set cos 4
  set dei 0
  set mpls experimental imposition 4
!
 class qos4_dc1
  set cos 4
  set dei 1
  set mpls experimental imposition 4
!
 class qos3_dc0
  set cos 3
  set dei 0
  set mpls experimental imposition 3
!
 class qos3_dc1
  set cos 3
  set dei 1
  set mpls experimental imposition 3
!
 class qos2_dc0
```

```
  set cos 2
  set dei 0
  set mpls experimental imposition 2
!
 class qos2_dc1
  set cos 2
  set dei 1
  set mpls experimental imposition 2
!
 class qos1_dc0
  set cos 1
  set dei 0
  set mpls experimental imposition 1
!
 class qos1_dc1
  set cos 1
  set dei 1
  set mpls experimental imposition 1
!
 class qos0_dc0
  set cos 0
  set dei 0
  set mpls experimental imposition 0
!
 class qos0_dc1
  set cos 0
  set dei 1
  set mpls experimental imposition 0
!
 end-policy-map
!
```

Bundle Traffic Policies

A policy can be bound to bundles. When a policy is bound to a bundle, the same policy is programmed on every bundle member (port). For example, if there is a policer or shaper rate, the same rate is configured on every port. Traffic is scheduled to bundle members based on the load balancing algorithm.

Both ingress and egress traffic is supported. Percentage-based policies , absolute rate-based policies, and time-based policies are supported.

Note Egress marking is not supported on BVI interfaces.

For details, see Configure QoS on Link Bundles.

Ingress Short-Pipe

When QoS traffic leaves an MPLS network, the MPLS label stack is removed on the penultimate ingress Label Switch Router (LSR), leaving an IPv4 or IPv6 packet to be forwarded. MPLS experimental bits (or EXP or pipe mode) carries out this disposition process and the packet is marked with a Differentiated Services Code Point (DSCP) or precedence value (also called DSCP or Precedence-based classification).

Usually, QoS traffic supports DSCP and precedence-based classifications only when there is no MPLS label in the packet. Using the ingress short-pipe feature, however, you can classify a packet that contains one MPLS

label using the type-of-service (ToS) field of the IPv4 or IPv6 header. This classification method is called ingress short-pipe. To classify an IP packet this way, you must:

1. Create a child class map.
2. Specify a ToS value in the child class map.
3. Attach the child class map to a parent class map.
4. Create a policy map containing the parent class map.
5. Set any ingress action such as traffic class or QoS group. From Release 7.1.1 onwards, you can also set ingress action DSCP (or precedence value).

With the ingress short-pipe feature, you get an increased visibility into traffic packets. Plus, the feature also removes the limitation of classifying MPLS packets that come into IPv4 or IPv6 networks.

Restrictions and Other Important Points

Ensure that you read these points before you configure the ingress short-pipe feature.

- This feature works only when there is one MPLS header in the traffic packet. If there are two or more MPLS headers, the ingress-short pipe feature fails. For example, in case of Explicit Null where there are two labels at the disposition, this feature will not work.
- You can carry out ingress classification using either the MPLS experimental bits (or EXP or pipe mode) classification OR the DSCP/precedence (or short-pipe) classification. Ensure that you do not mix the classification methods, else it may result in an unknown behavior, and the classification may not work at all.
- This feature is supported only on L3VPN, and not supported on L2VPN.
- This feature works for regular IPv4/IPv6 traffic, but will not work for IPv6 VPN Provider Edge over MPLS (6VPE).
- You can add only one child class map to a parent class map.
- This feature supports the invocation of short-pipe and legacy DSCP classification for the same parent class map.
- The child class map can contain only match precedence and match dscp commands.
- This feature is not supported in peering mode.

Configure Ingress Short-Pipe

This section details a sample configuration for the ingress short-pipe feature and another sample to configure classification for labeled and non-labeled packets under the same parent class.

Sample configuration to classify a packet that contains one MPLS label using the type-of-service (ToS) field of the IPv4 or IPv6 header (or the ingress short-pipe method):

```
class-map match-any in_pipe
 match mpls disposition class-map child_pipe
end-class-map
!
class-map match-any child_pipe
```

```
 match precedence 1
match dscp ipv4 af11
 end-class-map
!
class-map match-any ingress-business-high
match dscp af21 af22
end-class-map

class-map match-any ingress-business-low
match dscp af11 af12
end-class-map

policy-map ingress-classifier
class in_pipe
set traffic-class 5
set dscp af31
class ingress-business-high
set traffic-class 4
class ingress-business-low
set traffic-class 2
class class-default
set traffic-class 0
!
```

Note The **set dscp** option is available from Release 7.1.1 onwards.

You can configure classification for both labeled and non-labeled packets under the same parent class as in the following sample configuration. In this example, for MPLS labeled packets, DSCP configured under the child class is classified, while for non-labeled packets, DSCP/ToS configured in the **match dscp <value>** statement is classified.

DSCP value range is from 0 through 63. The range option is not supported. Up to 8 items per class are supported. Up to 64 **match dscp** values in total.

```
class-map match-any in_pipe
match mpls disposition class-map child_pipe  (labeled case)
match dscp af11 (non-labeled case)
end-class-map
!
class-map match-any child_pipe
match precedence 1
match dscp ipv4 af11
end-class-map
!
class-map match-any ingress-business-high
match dscp af21 af22
end-class-map

class-map match-any ingress-business-low
match dscp af11 af12
end-class-map

policy-map ingress-classifier
class in_pipe
set traffic-class 5
set dscp af31
class ingress-business-high
set traffic-class 4
class ingress-business-low
set traffic-class 2
```

```
class class-default
set traffic-class 0
!
```

Note The **set dscp** option is available from Release 7.1.1 onwards. A maximum of one set dscp command is supported per class-map.

Associated Commands

- match mpls disposition class-map

Selective Egress Policy-Based Queue Mapping

With selective egress policy-based queue mapping, you can combine traffic class (TC) maps in various permutations at the egress.

The primary aim of introducing the egress TC (traffic class) mapping is to classify the traffic in the ingress using a single policy and place the classified traffic into queues, by assigning the traffic classes. At the egress, you can support different grouping of TCs.

Based on different Service Level Agreements (SLAs) that each customer has signed up for, you can group some TCs into priority queues for real time (RT) traffic, other TCs into guaranteed bandwidth (BW) traffic, and the rest into best effort (BE) traffic delivery.

Let us consider an example where three customers have purchased these services, based on their requirements:

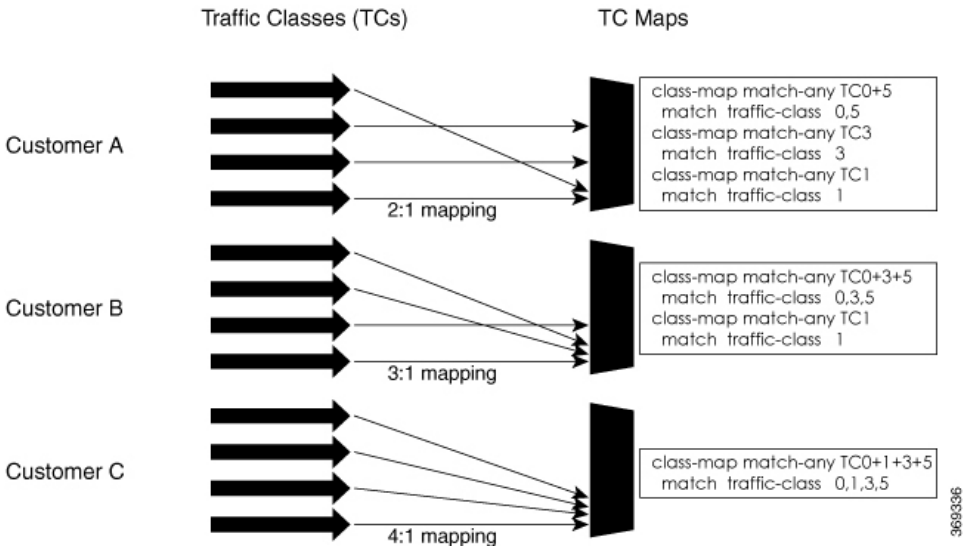
- Customer A - Requires RT traffic, reserved BW traffic and BE traffic delivery.
- Customer B – Requires reserved BW traffic and BE traffic delivery.
- Customer C – Needs only BE traffic delivery.

Using the selective egress policy-based queue mapping, you can create three profiles this way:

- Customer A – Priority queue RT traffic (TC1), Guaranteed BW traffic (TC3), Best effort traffic (TC0, TC5)
- Customer B – Guaranteed BW traffic (TC1), Best effort traffic (TC0, TC3, TC5)
- Customer C - Best effort traffic (TC0, TC1, TC3, TC5)

Using the egress TC-mapping, you can create three different profiles that you can use for each customer based on their SLAs with the provider.

Figure 1: Selective Egress Policy-Based Queue Mapping Helps Create Customer Profiles Based on Their SLAs



Restrictions and Other Important Points

Ensure that you read these points before you configure the selective egress policy-based queue-mapping feature.

- There can be only one TC (Traffic Class) mapped class to a PM (Policy Map).
- You cannot use a TC that you used in a mapped class, in a non-mapped class under the same PM.
- You can have a maximum of three unique TC mapped PMs or profiles per platform.
- Every TC mapped class must include **traffic-class 0** in the range values.
- The TC-mapping range is from 0 through 5.
- When a TC-mapped class is present in a PM, the class default becomes a dummy class. This means that the class default statistics and QoS values are not applicable.
- All the class default limitations apply to the TC-mapped class; for example, you cannot configure **priority** command under the TC mapped class.

Note A TC-mapped PM or profile is a PM that contains a TC-mapped class.

Example of a TC-mapped class:

match traffic-class 0 1 2 3

Example of a TC non-mapped class:

match traffic-class 1

Configure Selective Egress Policy-Based Queue Mapping

This section details a sample configuration for the selective egress policy-based queue-mapping feature and a use case to show how this feature works.

Sample configuration

```
class-map match-any <name>
 match traffic-class <value>
commit

policy-map tc_pmap
 class tc035
  shape average percent 1
 !
 class class-default
!
 end-policy-map
!
 class-map match-any tc035
match traffic-class 0 3 5
 end-class-map
!
```

Verification

Run the **show qos interface** and **show policy-map interface** commands.

When TC mapping class is present in a policy map, the class default does not have any values calculated.

show qos interface bundle-Ether 44 output sample

```
NOTE:- Configured values are displayed within parentheses
NPU Id:                        0
Total number of classes:       3
Interface Bandwidth:           100000000 kbps
Policy Name:                   tc_pmap
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                               =   tc1

Level1 Class                               =   tc035

Level1 Class                               =   class-default

Interface HundredGigE0/0/0/30 Ifh 0xf000208 (Member) -- output policy
NPU Id:                        0
Total number of classes:       3
Interface Bandwidth:           100000000 kbps
Policy Name:                   tc_pmap
VOQ Base:                      1264
Accounting Type:               Layer1 (Include Layer 1 encapsulation and above)
------------------------------------------------------------------------------
Level1 Class                               =   tc1
Egressq Queue ID                           =   1265 (LP queue)
Queue Max. BW.                             =   10063882 kbps (10 %)
Queue Min. BW.                             =   0 kbps (default)
Inverse Weight / Weight                    =   1 / (BWR not configured)
Guaranteed service rate                    =   10000000 kbps
TailDrop Threshold                         =   12517376 bytes / 10 ms (default)
WRED not configured for this class
```

```
Level1 Class                                =   tc035
Egressq Queue ID                            =   1264 (LP queue)
Queue Max. BW.                              =   1011732 kbps (1 %)
Queue Min. BW.                              =   0 kbps (default)
Inverse Weight / Weight                     =   1 / (BWR not configured)
Guaranteed service rate                     =   1000000 kbps
TailDrop Threshold                          =   1253376 bytes / 10 ms (default)
WRED not configured for this class

Level1 Class                                =   class-default
Queue Max. BW.                              =   no max (default)
Queue Min. BW.                              =   0 kbps (default)
Inverse Weight / Weight                     =   0 / (BWR not configured)
```

show policy-map interface bundle-Ether 44 output sample

```
Bundle-Ether44 output: tc_pmap

Class tc1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :              429444/53823648             0
    Transmitted         :              429444/53823648             0
    Total Dropped       :                   0/0                    0
  Queueing statistics
    Queue ID                              : None (Bundle)
    Taildropped(packets/bytes)            : 0/0
Class tc035
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             1288331/161470820            0
    Transmitted         :             1288331/161470820            0
    Total Dropped       :                   0/0                    0
  Queueing statistics
    Queue ID                              : None (Bundle)
    Taildropped(packets/bytes)            : 0/0
Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :                   0/0                    0
    Transmitted         :                   0/0                    0
    Total Dropped       :                   0/0                    0
  Queueing statistics
    Queue ID                              : None (Bundle)
    Taildropped(packets/bytes)            : 0/0
Policy Bag Stats time: 1557216940000  [Local Time: 05/07/19 08:15:40.000]
RP/0/RP0/CPU0:BB1#
```

Use Case

With the ingress traffic matching the same match criteria, you can group the egress traffic up to three unique TC mapped profiles. Using this feature, you can provide differentiated services to customers based on the SLAs they have signed up for.

In the example that follows, the ingress policy-map sets the ingress match criteria for the traffic class from 0 through 5. Based on the SLAs, you can group the TC values at the egress PM to deliver differentiated services.

After you group the TC values, you can apply specific egress actions under that class.

Ingress match:

```
class EXP1
  set traffic-class 1
!
class EXP2
  set traffic-class 2
```

```
!
class EXP3
  set traffic-class 3
!
class EXP4
  set traffic-class 4
!
class EXP5
  set traffic-class 5
!
class class-default
!
end-policy-map
!
```

Egress match:

Sample TC mapped class for policy-map PM1

```
class-map match-any TC2:1
match traffic-class 0 1
end-class-map
```

Sample TC mapped class for policy-map PM2

```
class-map match-any TC3:1
match traffic-class 0 1 2
end-class-map
```

Sample TC mapped class for policy-map PM3

```
class-map match-any TC6:1
match traffic-class 0 1 2 3 4 5
end-class-map
```

Configuring QoS Groups with an ACL

You can create QoS groups and configure ACLs to classify traffic into the groups based on a specified match condition. In this example, we match by the QoS group value (0-511).

Prerequisites

Before you can configure QoS groups with an ACL, the QoS peering profile must be enabled on the router or the line card. After enabling QoS peering, the router or line card must be reloaded, as shown in the following configuration.

Enabling QoS Peering Profile on the Router

Enter the global configuration mode and enable the QoS peering profile for the router as shown:

```
RP/0/RP0/CPU0:router(config)# hw-module profile qos ingress-model peering
RP/0/RP0/CPU0:router(config)# exit
RP/0/RP0/CPU0:router# reload
```

Enabling QoS Peering Profile on the Line Card

Enter the global configuration mode and enable the QoS peering profile for the line card as shown:

```
RP/0/RP0/CPU0:router(config)# hw-module profile qos ingress-model peering location 0/0/CPU0
RP/0/RP0/CPU0:router(config)# exit
RP/0/RP0/CPU0:router# reload location 0/0/CPU0
```

Configuration

Use the following set of configuration statements to configure an ACL with QoS groups.

```
/*
 Enter the global configuration mode, and configure an ACL with the required QoS groups.
*/
RP/0/RP0/CPU0:router# configure
RP/0/RP0/CPU0:router(config)# ipv4 access-list qos-acl
RP/0/RP0/CPU0:router(config-ipv4-acl)# 10 permit ipv4 host 5.0.0.1 any set qos-group 1
RP/0/RP0/CPU0:router(config-ipv4-acl)# 11 permit ipv4 host 6.0.0.1 any set qos-group 2
RP/0/RP0/CPU0:router(config-ipv4-acl)# 12 permit ipv4 host 7.0.0.1 any set qos-group 3
RP/0/RP0/CPU0:router(config-ipv4-acl)# 13 deny ipv4 any any


/* Create a policy map with the required classes.
In this example, we also create a default class for traffic that does not belong to any of
 the specified
classes. */
RP/0/RP0/CPU0:router(config)# policy-map qos-acl-map
RP/0/RP0/CPU0:router(config-pmap)# class qos1
RP/0/RP0/CPU0:router(config-pmap-c)# set dscp af43
RP/0/RP0/CPU0:router(config-pmap-c)# set traffic-class 2
RP/0/RP0/CPU0:router(config-pmap-c)# exit

RP/0/RP0/CPU0:router(config-pmap)# class qos2
RP/0/RP0/CPU0:router(config-pmap-c)# set precedence critical
RP/0/RP0/CPU0:router(config-pmap-c)# set traffic-class 7
RP/0/RP0/CPU0:router(config-pmap-c)# exit

RP/0/RP0/CPU0:router(config-pmap)# class qos3
RP/0/RP0/CPU0:router(config-pmap-c)# set precedence 2
RP/0/RP0/CPU0:router(config-pmap-c)# set traffic-class 2
RP/0/RP0/CPU0:router(config-pmap-c)# exit

RP/0/RP0/CPU0:router(config-pmap)# class qos4
RP/0/RP0/CPU0:router(config-pmap-c)# set traffic-class 4
RP/0/RP0/CPU0:router(config-pmap-c)# set dscp cs4
RP/0/RP0/CPU0:router(config-pmap-c)# exit

RP/0/RP0/CPU0:router(config-pmap)# class class-default
RP/0/RP0/CPU0:router(config-pmap-c)# police rate percent 20
RP/0/RP0/CPU0:router(config-pmap-c-police)# exit


/* Create the class maps for specifying the match conditions. */
RP/0/RP0/CPU0:router(config)# class-map match-any qos1
RP/0/RP0/CPU0:router(config-cmap)# match qos-group 1
RP/0/RP0/CPU0:router(config-cmap)# end-class-map

RP/0/RP0/CPU0:router(config)# class-map match-any qos2
RP/0/RP0/CPU0:router(config-cmap)#  match qos-group 2
RP/0/RP0/CPU0:router(config-cmap)# end-class-map

RP/0/RP0/CPU0:router(config)# class-map match-any qos3
RP/0/RP0/CPU0:router(config-cmap)# match qos-group 3
RP/0/RP0/CPU0:router(config-cmap)# end-class-map

RP/0/RP0/CPU0:router(config)# class-map match-any qos4
RP/0/RP0/CPU0:router(config-cmap)# match qos-group 4
RP/0/RP0/CPU0:router(config-cmap)# end-class-map
```

```
/* Apply the access list and the QoS map to the Gigabit interface, and commit your
configuration. */
RP/0/RP0/CPU0:router(config)# interface TenGigE0/0/0/1
RP/0/RP0/CPU0:router(config-if)# ipv4 address 12.0.0.1/24
RP/0/RP0/CPU0:router(config-if)# no shut
RP/0/RP0/CPU0:router(config-if)# service-policy input qos-acl-map
RP/0/RP0/CPU0:router

RP/0/RP0/CPU0:router(config-if)# commit
Tue Mar 28 10:23:34.106 IST

RP/0/0/CPU0:Mar 28 10:37:48.570 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/1, changed state to Down
RP/0/0/CPU0:Mar 28 10:37:48.608 : ifmgr[397]: %PKT_INFRA-LINK-3-UPDOWN : Interface
TenGigE0/0/0/1, changed state to Up

RP/0/RP0/CPU0:router(config-if)# exit
```

Running Configuration

Confirm your configuration.

```
RP/0/RP0/CPU0:router(config)# show run
Tue Mar 28 10:37:55.737 IST

Building configuration...
!! IOS XR Configuration 0.0.0

ipv4 access-list qos-acl
10 permit ipv4 host 5.0.1.1 any set qos-group 1
11 permit ipv4 host 6.0.1.1 any set qos-group 2
12 permit ipv4 host 7.0.1.1 any set qos-group 3
13 deny ipv4 any any

class-map match-any qos1
match qos-group 1
end-class-map
!
class-map match-any qos2
match qos-group 2
end-class-map
!
class-map match-any qos3
match qos-group 3
end-class-map
!
class-map match-any qos4
match qos-group 4
end-class-map
!

policy-map qos-acl-map
class qos1
  set dscp af43
  set traffic-class 2
!
class qos2
  set precedence critical
  set traffic-class 7
!
class qos3
```

```
  set precedence 2
  set traffic-class 2
!
class qos4
  set traffic-class 4
  set dscp cs4
!
class class-default
  police rate percent 20
  !
!
end-policy-map
!

interface TenGigE0/0/0/1
service-policy input qos-acl-map
ipv4 address 12.0.0.1 255.255.255.0
ipv4 access-group qos-acl ingress compress level 3

!
```

You have successfully configured an ACL with QoS groups.

QoS Egress Marking and Queuing Using Dual Policy-Map

To achieve QoS Egress marking/queuing, the router utilizes the dual policy model on the Egress with independent policies for marking and queuing.

Egress marking can be achieved by applying a policy-map on the ingress interface by setting qos-group/discard-class. Then the qos-group which is set by the ingress policy-map is used by the egress-policy map along with DP (drop-precedence or discard class) value to remark the cos/dei bits of the outgoing L2 packet. Similarly Egress queuing can be achieved by applying a policy-map on the ingress interface by setting the traffic-class. Then the traffic-class is used by the egress-policy map to perform queuing actions.

Benefits

- This feature enables the users to make the marking decision based on the DP (drop precedence) field.
- In case of MPLS-to-Layer 2 traffic stream, the Layer 2 packet is within the MPLS data packet; therefore marking of the Layer 2 header is possible only at Egress after data transmission.
- In case of Egress rewrite operations, where the VLAN tags are modified or added, the cos or the dei fields can be marked with Egress marking.

QoS Egress Marking and Queueing can be summarized in the following three steps—

1. Configure a Ingress Policy-Map— classifying the incoming packet and setting the qos-group/discard-class or the traffic class.
2. Configure a Egress Policy-Map:
    - Configure Egress Marking Policy—
        - Create class-map to classify on qos-group/discard-class.
        - Create policy-map to mark cos/dei field in the L2 header.

- Configure Egress Queuing Policy—
  - Create class-map to classify on traffic-class.
  - Create policy-map to perform the queuing actions (for example, bandwidth, shaping, priority).

3. Attaching the policies to the Interfaces.

Note While marking QinQ traffic, only outer dot1q header is effected and the inner header remains as is. However, in case of few rewrite operations where the new QinQ tags are added, the inner header is marked.

Example— Ingress Policy-Map Configuration:

```
/*Create class-map/*
Router#config
Router(config)#class-map match-any cos2
Router(config-cmap)#match cos 2
Router(config-cmap)#commit
Router(config)#class-map match-any cos3
Router(config-cmap)#match cos 3
Router(config-cmap)#commit
Router(config)#class-map match-any cos4
Router(config-cmap)#match cos 4
Router(config-cmap)#commit

/*Create classification policies*/
Router#config
Router(config)#policy-map ingress-classification
Route(config-pmap)#class cos2
Router(config-pmap-c)#set qos-group 1
Router(config-pmap-c)#set traffic-class 3
Router(config-pmap-c)#class cos3
Router(config-pmap-c)#set qos-group 2
Router(config-pmap-c)#set traffic-class 5
Router(config-pmap-c)#class cos4
Router(config-pmap-c)#set qos-group 3
Router(config-pmap-c)#set traffic-class 4
Router(config-pmap-c)#class class-default
Router(config-pmap-c)#set qos-group 7
Router(config-pmap-c)#set traffic-class 6
Router(config-pmap-c)#commit
```

Example— Egress Policy-Map Configuration:

```
*/Egress Marking Policy/*
Router#config
Router(config)#class-map match-any qos1
Router(config-cmap)#match qos-group 1
Router(config-cmap)#commit
Router(config)#class-map match-any qos2
Router(config-cmap)#match qos-group 2
Router(config-cmap)#commit
Router(config)#class-map match-any qos3
Router(config-cmap)#match qos-group 3
Router(config-cmap)#commit
Router#config
Router(config)#policy-map egress-marking
Route(config-pmap)#class qos1
```

```
Router(config-pmap-c)#set cos 1
Router(config-pmap-c)#class qos2
Router(config-pmap-c)#set cos 2
Router(config-pmap-c)#set dei 1
Router(config-pmap-c)#class qos3
Router(config-pmap-c)#set cos 3
Router(config-pmap-c)#class class-default
Router(config-pmap-c)#set cos 7
Router(config-pmap-c)#commit

*/Egress Queuing Policy/*
Router#config
Router(config)#class-map match-any tc3
Router(config-cmap)#match traffic-class 3
Router(config-cmap)#commit
Router(config)#class-map match-any tc4
Router(config-cmap)#match traffic-class 3
Router(config-cmap)#commit
Router(config)#class-map match-any tc5
Router(config-cmap)#match traffic-class 3
Router(config-cmap)#commit
Router#config
Router(config)#policy-map egress-queuing
Route(config-pmap)#class tc3
Router(config-pmap-c)#shape average 2 mbps
Router(config-pmap-c)#class tc4
Router(config-pmap-c)#shape average 5 mbps
Router(config-pmap-c)#class tc5
Router(config-pmap-c)#shape average 7 mbps
Router(config-pmap-c)#class class-default
Router(config-pmap-c)#commit
```

Example— Attaching the policies to the Interface

```
Router#config
Router(config)#interface tenGigE 0/0/0/1
Router(config-if)#service-policy input ingress-classification
Router(config-if)#service-policy output egress-marking
Router(config-if)#service-policy output egress-queuing
Router(config-if)#commit
```

Restrictions

- Statistics for marking policy is not supported, that is, the show policy-map interface command does not display any output.
- Statistics output is displayed only when the queuing policy is applied.
- Egress marking policy can classify only on qos-group/discard-class.
- Egress queueing policy can classify only on traffic-class.
- Egress marking policy can mark only the cos/dei field in L2 header.

Ingress QoS Scale Limitation

Refer to the below table for Ingress QoS Scale Limitation.

Table 1: Ingress QoS Scale Limitation

Class-Map Size	Maximum number of Policers with Ingress QoS Applied	
	Per Core	**Per NPU**
4	6999	13999
8	3499	6999
16	1749	3499
32	874	1749

Note If you apply an ingress policy map to a bundle that has bundle members only from a single core of an NPU, then the QoS resources are consumed on both cores of that NPU.

Restrictions

- If you have a **set traffic class** statement explicitly configured in ingress service policy, it's mandatory to have a corresponding **match traffic class** on egress for the traffic to be correctly matched and the stats (statistics) to be accounted in **show policy-map interface <> output** command. To match the ingress traffic to egress class-default, traffic class should be set to 0 on ingress.
- If you have a **set traffic class** configured in Ingress service policy, and no corresponding **match traffic class** on egress, the traffic doesn't proceed to class default and the stats for this traffic flow won't be available in **show policy-map interface <> output** command.
- If you don't have any **set traffic class** statement in ingress, then traffic applies to the default-class on egress.
- If you have a **set discard-class** statement configured in ingress service policy, it's mandatory to have a corresponding **match discard-class** on egress for the traffic to be correctly matched and the stats to be accounted in **show policy-map interface <> output** command.
- If you have a **set discard-class** statement configured in ingress service policy and don't have a corresponding **match discard-class** on egress, the traffic won't hit the class-default and the stats for this flow won't be accounted in **show policy-map interface <> output** command.
- The router doesn't support class-map size on peering mode.

Restrictions for QoS on BVI

- The router doesn't support the egress policy on Bridge-Group Virtual Interface (BVI), but BVI (CoS, DEI) marking is supported by applying the policy to its corresponding Layer 2 interface, which is part of the same bridge domain.
- If you apply L3 ingress QoS policy on L2 interface, which is a part of the same bridge-domain as BVI, the packet classification might not work if packets are destined to the BVI MAC address.
- If a QoS policy is attached to BVI, the policy is inherited by the L2 interfaces, which are part of the same bridge-domain. Hence, any other policy can't be applied on the L2 interfaces. Similarly, if a QoS policy

is attached to any of the L2 interfaces, any QoS policy can't be applied on the BVI, which is part of the same bridge-domain.

Restrictions for TCAM

- The creation of 250 ingress unique policy-maps is supported. However, you may be able to create up to 254 unique policy maps after which the error message "Out of ACLID resource" may display. However, you must avoid creating more than 250 ingress unique policy maps because the additional map sizes are reserved for internal purposes.
- The 250 policy-maps scale is based on the internal TCAM space available for each router. The available TCAM space differs for every PID, and is dependent upon TCAM bank sharing.

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.
- The system does not support the show policy-map statistics for marking policies.
- An in-place modification of an ACL does not reset the policy-map statistics counter.

Note

- For QOS EXP-Egress marking applied on a Layer 3 interface on Cisco NCS550x and NCS55Ax routers, there is a limit of two unique policy-maps per NPU. This limit is three unique policy maps per NPU for routers that have the Cisco NC57 line cards installed. When the maximum limit for policy-maps is reached and you try to modify a policy-map which is shared between different interfaces, you may get an error.
- For QOS egress marking (CoS, DEI) applied on a Layer 2 interface, there is a limit of 13 unique policy-maps per NPU. When the maximum limit for policy-maps is reached and you try to modify a policy-map which is shared between different interfaces, you may get an error.

Verification

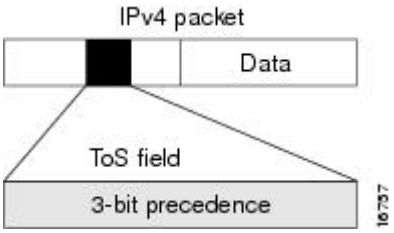
If unrecoverable errors occur during in-place policy modification, the policy is put into an inconsistent state on target interfaces. No new configuration is possible until the configuration session is unblocked. It is recommended to remove the policy from the interface, check the modified policy and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

References for Modular QoS Service Packet Classification

Specification of the CoS for a Packet with IP Precedence

Use of IP precedence allows you to specify the CoS for a packet. You can create differentiated service by setting precedence levels on incoming traffic and using them in combination with the QoS queuing features. So that, each subsequent network element can provide service based on the determined policy. IP precedence is usually deployed as close to the edge of the network or administrative domain as possible. This allows the rest of the core or backbone to implement QoS based on precedence.

Figure 2: IPv4 Packet Type of Service Field



You can use the three precedence bits in the type-of-service (ToS) field of the IPv4 header for this purpose. Using the ToS bits, you can define up to eight classes of service. Other features configured throughout the network can then use these bits to determine how to treat the packet in regard to the ToS to grant it. These other QoS features can assign appropriate traffic-handling policies, including congestion management strategy and bandwidth allocation. For example, queuing features such as LLQ can use the IP precedence setting of the packet to prioritize traffic.

IP Precedence Bits Used to Classify Packets

Use the three IP precedence bits in the ToS field of the IP header to specify the CoS assignment for each packet. You can partition traffic into a maximum of eight classes and then use policy maps to define network policies in terms of congestion handling and bandwidth allocation for each class.

Each precedence corresponds to a name. IP precedence bit settings 6 and 7 are reserved for network control information, such as routing updates. These names are defined in RFC 791.

IP Precedence Value Settings

By default, the routers leave the IP precedence value untouched. This preserves the precedence value set in the header and allows all internal network devices to provide service based on the IP precedence setting. This policy follows the standard approach stipulating that network traffic should be sorted into various types of service at the edge of the network and that those types of service should be implemented in the core of the network. Routers in the core of the network can then use the precedence bits to determine the order of transmission, the likelihood of packet drop, and so on.

Because traffic coming into your network can have the precedence set by outside devices, we recommend that you reset the precedence for all traffic entering your network. By controlling IP precedence settings, you prohibit users that have already set the IP precedence from acquiring better service for their traffic simply by setting a high precedence for all of their packets.

The class-based unconditional packet marking and LLQ features can use the IP precedence bits.

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.