Configuring Modular QoS Service Packet Classification

This chapter covers these topics:

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.

Traffic Class Elements

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.
- For egress classification, you must configure all 8 (qos-group) classes including class-default.
- 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
```

Also see, Running Configuration, on page 6.

Also see, Verification, on page 7.

Related Topics

- Traffic Class Elements, on page 2
- Traffic Policy Elements, on page 4

Associated Commands

- class-map
- match access-group
- match cos
- match dscp
- match mpls experimental topmost

- match precedence
- match protocol
- 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.

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

Supported Action Types	Direction supported on Interfaces
bandwidth-remaining*	egress
mark	(See Packet Marking, on page 10)
police	ingress
priority	egress (level 1)
shape	egress

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 2.

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 6.

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 4)

```
Router# configure
Router(config)# policy-map egress_policy2
Router(config-pmap)# class qos-1

/* Configure class-action ('bandwidth remaining' in this example).
Repeat as required, to specify other class-actions */
Router(config-pmap-c)# bandwidth remaining ratio 50
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 6.

See, Verification, on page 7.

Related Topics

- Traffic Policy Elements, on page 4
- Traffic Class Elements, on page 2

Associated Commands

- bandwidth remaining
- class
- police
- policy-map
- priority
- set cos
- set dscp
- 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_16)
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 4)
3. Associating the traffic class with the traffic policy
4. Attaching the traffic policy to an interface, in the ingress or egress direction

```
Router# configure
Router(config)# interface TenGig 0/0/0/0
Router(config-int)# service-policy output egress_policy2
Router(config-int)# commit
```

Running Configuration

```
/* Class-map configuration */

class-map match-any qos-1
 match qos-group 1
 end-class-map
!
- - -
- - -
!
class-map match-any qos-7
 match qos-group 7
 end-class-map

/* Policy-map configuration */
policy-map egress_policy2
 class qos-1
  bandwidth remaining ratio 50
 !
- - -
- - -
 class qos-7
  priority level 1
 !
 class class-default
  bandwidth remaining ratio 2
 !
 end-policy-map

/* Attaching traffic policy to an interface in egress direction */
interface TenGig 0/0/0/0
 service-policy output egress_policy2
 !
```

Verification

```
Router# show policy-map interface tenGigE 0/0/0/0

TenGigE0/0/0/0 direction input: Service Policy not installed

TenGigE0/0/0/0 output: egress_policy2

Class qos-1
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :             4867412/3598770345           0
    Transmitted         :             2901271/2142905575           0
    Total Dropped       :             1966141/1455864770           0
  Queueing statistics
    Queue ID                             : 9
    High watermark                       : N/A
    Inst-queue-len  (cells)              : 10709
    Avg-queue-len                        : N/A
    Taildropped(packets/bytes)           : 1966141/1455864770
    Queue(conform)      :             2901271/2142905575           0
    RED random drops(packets/bytes)      : 0/0

- - -
- - -

Class class-default
  Classification statistics          (packets/bytes)     (rate - kbps)
    Matched             :              262482/237940215            0
    Transmitted         :               97172/88070686             0
    Total Dropped       :              165310/149869529            0
  Queueing statistics
    Queue ID                             : 8
    High watermark                       : N/A
    Inst-queue-len  (cells)              : 10714
    Avg-queue-len                        : N/A
    Taildropped(packets/bytes)           : 165310/149869529
    Queue(conform)      :              97172/88070686             0
    RED random drops(packets/bytes)      : 0/0
```

Related Topics

- Traffic Policy Elements, on page 4
- Traffic Class Elements, on page 2

Associated Commands

- service-policy

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 and absolute rate-based policies are supported. However, for ease of use, it is recommended to use percentage-based policies.

For details, see Configure QoS on Link Bundles.

Class-based Unconditional Packet Marking

The class-based, unconditional packet marking feature provides users with a means to differentiate packets based on the designated markings. This feature allows you to partition your network into multiple priority levels or classes of service.

These tasks can be performed with the packet marking feature:

- Mark packets by setting the IP differentiated services code point (DSCP) in the IP ToS byte.
- Mark packets by setting the Layer 2 class-of-service (CoS) value.
- Mark packets by setting outer CoS tags for an IEEE 802.1Q tunneling (QinQ) configuration.
- Mark packets by setting the value of the *qos-group* argument.

Note *qos-group* is a variable internal to the router, and is not transmitted.

For more details, see Packet Marking, on page 10.

Configure Class-based Unconditional Packet Marking

The Cisco NCS 5000 Series Router support unconditional packet marking in ingress direction.

Guidelines

- Only ingress markings are supported.
- A maximum of only two **set** commands are allowed per class.
- You must configure all 8 classes (including default-class) for a policy-map.

Configuration Example

You have to accomplish the following to complete the unconditional packet marking configuration:

1. Creating or modifying a policy-map that can be attached to one or more interfaces
2. Specifying the traffic class whose policy has to be created or changed
3. Specifying the marking action for the packet
4. Attaching the policy-map to an input interface

```
Router# configure
Router(config)# policy-map policy1
Router(config-pmap)# class class1
/* Specify the marking action. Repeat the set command to specify another marking action */
Router(config-pmap-c)# set dscp 5
Router(config-pmap-c)# exit
```

```
/* Repeat the above steps to configure the remaining classes of the policy-map*/
Router(config-pmap)# exit

Router(config)# interface TenGigE 0/2/0/0
Router(config-if)# service-policy input policy1
Router(config-if)# no shutdown
Router(config-if)# commit
```

Running Configuration

```
policy-map policy1
 class class1
  set dscp 5
  set cos 7
  !
  - - -
  - - -
 !
 end-policy-map

interface TenGigE 0/2/0/0
 service-policy input policy1
!
```

Related Topics

- Class-based Unconditional Packet Marking, on page 8

Associated Commands

- set cos
- set dscp
- set qos-group

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 routers, there is a limit of two 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.
- 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.

Use the **show qos inconsistency** command to view inconsistency in each location. The configuration session is blocked until the modified policy is effective on all interfaces that are using the policy. Output from the show policy-map targets command indicates that the TenGigabit Ethernet interface 0/1/0/0 has one policy map attached as a main policy (as opposed to being attached to a child policy in a hierarchical QoS configuration). Outgoing traffic on this interface is affected if the policy is modified

```
router# show policy-map targets

1) Policymap: policy1    Type: qos
     Targets (applied as main policy):
       TenGigabitEthernet0/1/0/0 output
     Total targets: 1

     Targets (applied as child policy):
     Total targets: 0
```

References for Modular QoS Service Packet Classification

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.

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 dscp	0-63	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.

- 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.

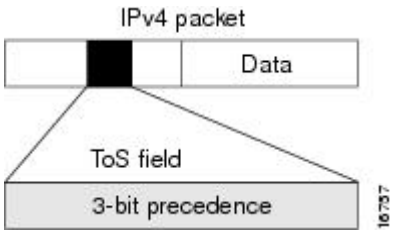
Note

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

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 1: 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.

Dynamic Modification of Interface Bandwidth

Policy States

During the dynamic bandwidth modification process, if the modification is successful, the system does not display policy state information. However, if an error occurs, the system places the interface in one of these states and provides a policy-state error notification:

- Verification—This state indicates an incompatibility of the configured QoS policy with respect to the new interface bandwidth value. The system handles traffic on a best-efforts basis and some traffic drops can occur.

- Hardware programming—This state indicates a hardware programming failure caused by one of these conditions:
  - The modification of the interface default QoS resources encountered hardware update failures.
  - The modification of QoS resources (associated with the applied QoS policy) encountered hardware update failures.

    With either of these failures, hardware programming could be in an inconsistent state, which can impact features such as policing, queueing and marking. Therefore, the system disables QoS policy in hardware for these error conditions.

- Reset—In response to user reconfiguration of QoS policy, the system attempts to apply the new policy but fails; the fallback to the previous QoS policy also fails.

If you receive notification of any of these policy states, you need to reconfigure the QoS policy to clear this condition.

Use the **show qos interface** and **show policy-map interface** commands to query the QoS state of the interface. The system displays the QoS policy status if the interface is in one of the error states (verification, hardware programming, or reset), but does not display it if the state is active.