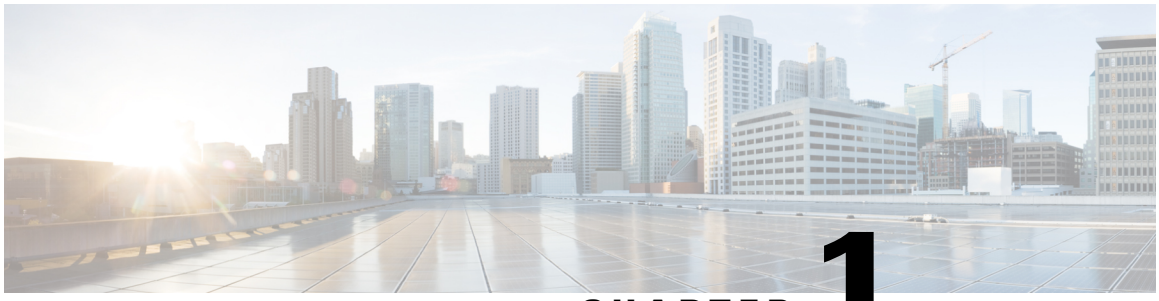
# QoS Modular QoS Command-Line Interface Configuration Guide Cisco IOS XE Release 3S

**Americas Headquarters**
Cisco Systems, Inc.
170 West Tasman Drive
San Jose, CA 95134-1706
USA
http://www.cisco.com
Tel: 408 526-4000
800 553-NETS (6387)
Fax: 408 527-0883

# CONTENTS



CHAPTER 1

# Applying QoS Features Using the MQC

This module contains the concepts about applying QoS features using the Modular Quality of Service (QoS) Command-Line Interface (CLI) (MQC) and the tasks for configuring the MQC. The MQC allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the QoS feature that will be applied to the traffic class.



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Restrictions for Applying QoS Features Using the MQC

The MQC does not support Internetwork Packet Exchange (IPX) packets.

The number of QoS class maps supported in a single policy map varies by release, as follows:

- For Cisco IOS XE Release 2.1 and 2.2, the MQC supports a maximum of eight class maps in a single policy map.

- For Cisco IOS XE Release 2.3, the MQC supports a maximum of 256 class maps in a single policy map.
- For Cisco IOS XE Release 3.5, the MQC supports a maximum of 1000 class maps in a single policy map.

The number of QoS policy maps supported on a router varies by release, as follows:

- For Cisco IOS XE Release 2.1 and 2.2, the MQC supports no more than 1024 (or 1K) unique policy map definitions.
- For Cisco IOS XE Release 2.3, the MQC supports no more than 4096 (or 4K) unique policy map definitions.

**Note** The policy map limitations do not refer to the number of applied policy map instances, only to the definition of the policy maps.

The following restrictions apply to Cisco IOS XE release 3.5S for the Cisco ASR 903 router:

- QoS policy maps are not supported in sessions.
- Nested traffic maps are not supported.

# Information About Applying QoS Features Using the MQC

## The MQC Structure

The MQC structure allows you to define a traffic class, create a traffic policy, and attach the traffic policy to an interface.

The MQC structure consists of the following three high-level steps:

1. Define a traffic class by using the **class-map** command. A traffic class is used to classify traffic.
2. Create a traffic policy by using the **policy-map** command. (The terms *traffic policy* and *policy map* are often synonymous.) A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.
3. Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

## Elements of a Traffic Class

A traffic class contains three major elements: a traffic class name, a series of **match** commands, and, if more than one **match** command is used in the traffic class, instructions on how to evaluate these **match** commands.

The **match** commands are used for classifying packets. Packets are checked to determine whether they meet the criteria specified in the **match**commands; if a packet meets the specified criteria, that packet is considered a member of the class. Packets that fail to meet the matching criteria are classified as members of the default traffic class.

### Available match Commands

The table below lists *some* of the available **match** commands that can be used with the MQC. The available **match** commands vary by Cisco IOS XE release. For more information about the commands and command syntax, see the *Cisco IOS Quality of Service Solutions* Command Reference.

***Table 1: match Commands That Can Be Used with the MQC***

| Command | Purpose |
|---|---|
| **match access-group** | Configures the match criteria for a class map on the basis of the specified access control list (ACL). |
| **match any** | Configures the match criteria for a class map to be successful match criteria for all packets. |
| **match cos** | Matches a packet based on a Layer 2 class of service (CoS) marking. |
| **match destination-address mac** | Uses the destination MAC address as a match criterion. |
| **match discard-class** | Matches packets of a certain discard class. |
| **match** [**ip**] **dscp** | Identifies a specific IP differentiated service code point (DSCP) value as a match criterion. Up to eight DSCP values can be included in one match statement. |
| **match fr-dlci** | Specifies the Frame Relay data-link connection identifier (DLCI) number as a match criterion in a class map. |
| **match input-interface** | Configures a class map to use the specified input interface as a match criterion. |
| **match ip rtp** | Configures a class map to use the Real-Time Transport Protocol (RTP) port as the match criterion. |
| **match mpls experimental** | Configures a class map to use the specified value of the Multiprotocol Label Switching (MPLS) experimental (EXP) field as a match criterion. |
| **match mpls experimental topmost** | Matches the MPLS EXP value in the topmost label. |

| Command | Purpose |
|---|---|
| **match not** | Specifies the single match criterion value to use as an unsuccessful match criterion.<br>**Note** The **match not** command, rather than identifying the specific match parameter to use as a match criterion, is used to specify a match criterion that prevents a packet from being classified as a member of the class. For instance, if the **match not qos-group 6** command is issued while you configure the traffic class, QoS group 6 becomes the only QoS group value that is not considered a successful match criterion. All other QoS group values would be successful match criteria. |
| **match packet length** | Specifies the Layer 3 packet length in the IP header as a match criterion in a class map. |
| **match port-type** | Matches traffic on the basis of the port type for a class map. |
| **match** [**ip**] **precedence** | Identifies IP precedence values as match criteria. |
| **match protocol** | Configures the match criteria for a class map on the basis of the specified protocol.<br>**Note** A separate **match protocol** (NBAR) command is used to configure network-based application recognition (NBAR) to match traffic by a protocol type known to NBAR. |
| **match protocol fasttrack** | Configures NBAR to match FastTrack peer-to-peer traffic. |
| **match protocol gnutella** | Configures NBAR to match Gnutella peer-to-peer traffic. |
| **match protocol http** | Configures NBAR to match Hypertext Transfer Protocol (HTTP) traffic by URL, host, Multipurpose Internet Mail Extension (MIME) type, or fields in HTTP packet headers. |
| **match protocol rtp** | Configures NBAR to match RTP traffic. |
| **match qos-group** | Identifies a specific QoS group value as a match criterion. |
| **match source-address mac** | Uses the source MAC address as a match criterion. |

### Multiple match Commands in One Traffic Class

If the traffic class contains more than one **match** command, you need to specify how to evaluate the **match** commands. You specify this by using either the **match-any** or **match-all** keyword of the **class-map** command. Note the following points about the **match-any** and **match-all** keywords:

- If you specify the **match-any** keyword, the traffic being evaluated by the traffic class must match *one* of the specified criteria.
- If you specify the **match-all** keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria.
- If you do not specify either keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria (that is, the behavior of the **match-all** keyword is used).

# Elements of a Traffic Policy

A traffic policy contains three elements: a traffic policy name, a traffic class (specified with the **class** command), and the command used to enable the QoS feature.

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the interface (by using the **service-policy** command).

**Note** A packet can match only *one* traffic class within a traffic policy. If a packet matches more than one traffic class in the traffic policy, the *first* traffic class defined in the policy will be used.

### Commands Used to Enable QoS Features

The commands used to enable QoS features vary by Cisco IOS XE release. The table below lists *some* of the available commands and the QoS features that they enable. For complete command syntax, see the *Cisco IOS QoS Command Reference*.

For more information about a specific QoS feature that you want to enable, see the appropriate module of the Cisco IOS XE Quality of Service Solutions Configuration Guide.

***Table 2: Commands Used to Enable QoS Features***

| Command | Purpose |
|---|---|
| **bandwidth** | Configures a minimum bandwidth guarantee for a class. |
| **bandwidth remaining** | Configures an excess weight for a class. |
| **fair-queue** | Enables the flow-based queueing feature within a traffic class. |
| **drop** | Discards the packets in the specified traffic class. |
| **police** | Configures traffic policing. |

| Command | Purpose |
|---|---|
| **police (percent)** | Configures traffic policing on the basis of a percentage of bandwidth available on an interface. |
| **police (two rates)** | Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR). |
| **priority** | Gives priority to a class of traffic belonging to a policy map. |
| **queue-limit** | Specifies or modifies the maximum number of packets the queue can hold for a class configured in a policy map. |
| **random-detect** | Enables Weighted Random Early Detection (WRED). |
| **random-detect discard-class** | Configures the WRED parameters for a discard-class value for a class in a policy map. |
| **random-detect discard-class-based** | Configures WRED on the basis of the discard class value of a packet. |
| **random-detect exponential-weighting-constant** | Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class. |
| **random-detect precedence** | Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map. |
| **service-policy** | Specifies the name of a traffic policy used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another). |
| **set atm-clp** | Sets the cell loss priority (CLP) bit when a policy map is configured. |
| **set cos** | Sets the Layer 2 class of service (CoS) value of an outgoing packet. |
| **set discard-class** | Marks a packet with a discard-class value. |
| **set** [**ip**] **dscp** | Marks a packet by setting the differentiated services code point (DSCP) value in the type of service (ToS) byte. |
| **set fr-de** | Changes the discard eligible (DE) bit setting in the address field of a Frame Relay frame to 1 for all traffic leaving an interface. |

| Command | Purpose |
|---|---|
| **set mpls experimental** | Designates the value to which the MPLS bits are set if the packets match the specified policy map. |
| **set precedence** | Sets the precedence value in the packet header. |
| **set qos-group** | Sets a QoS group identifier (ID) that can be used later to classify packets. |
| **shape** | Shapes traffic to the indicated bit rate according to the algorithm specified. |

# Nested Traffic Classes

The MQC does not necessarily require that you associate only one traffic class to one traffic policy. When packets meet more than one match criterion, multiple traffic classes can be associated with a single traffic policy.

Similarly, the MQC allows multiple traffic classes (nested traffic classes, which are also called nested class maps or MQC Hierarchical class maps) to be configured as a single traffic class. This nesting can be achieved with the use of the **match class-map** command. The only method of combining match-any and match-all characteristics within a single traffic class is with the **match class-map** command.

# match-all and match-any Keywords of the class-map Command

One of the commands used when you create a traffic class is the **class-map**command. The command syntax for the **class-map** command includes two keywords: **match-all** and **match-any**. The **match-all** and **match-any** keywords need to be specified only if more than one match criterion is configured in the traffic class. Note the following points about these keywords:

- The **match-all** keyword is used when *all* of the match criteria in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- The **match-any** keyword is used when only *one* of the match criterion in the traffic class must be met in order for a packet to be placed in the specified traffic class.
- If neither the **match-all** keyword nor **match-any** keyword is specified, the traffic class will behave in a manner consistent with the **match-all** keyword.

# input and output Keywords of the service-policy Command

As a general rule, the QoS features configured in the traffic policy can be applied to packets entering the interface or to packets leaving the interface. Therefore, when you use the **service-policy** command, you need to specify the direction of the traffic policy by using the **input** or **output** keyword.

For instance, the **service-policy output policy-map1** command would apply the QoS features in the traffic policy to the interface in the output direction. All packets leaving the interface (output) are evaluated according to the criteria specified in the traffic policy named policy-map1.

**Note** For Cisco IOS XE Release 2.1 and later releases, queueing mechanisms are not supported in the input direction. Nonqueueing mechanisms (such as traffic policing and traffic marking) are supported in the input direction. Also, classifying traffic on the basis of the source MAC address (using the **match source-address mac** command) is supported in the input direction only.

## Benefits of Applying QoS Features Using the MQC

The MQC structure allows you to create the traffic policy (policy map) once and then apply it to as many traffic classes as needed. You can also attach the traffic policies to as many interfaces as needed.

# How to Apply QoS Features Using the MQC

## Creating a Traffic Class

To create a traffic class, use the **class-map** command to specify the traffic class name. Then use one or more **match** commands to specify the appropriate match criteria. Packets matching the criteria that you specify are placed in the traffic class. For more information about the **match-all** and **match-any** keywords of the class-map comand, see the “match-all and match-any Keywords of the class-map Command” section.

**Note** The **match cos** command is shown in Step 4. The **match cos** command is simply an example of one of the **match** commands that you can use. For information about the other available **match** commands, see the “match-all and match-any Keywords of the class-map Command” section.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-map-name*
4. **match cos** *cos-number*
5. Enter additional match commands, if applicable; otherwise, continue with step 6.
6. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router> enable` | • Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map** [**match-all** \| **match-any**] *class-map-name*<br>**Example:**<br>`Router(config)# class-map match-any class1` | Creates a class to be used with a class map and enters class-map configuration mode.<br>• The class map is used for matching packets to the specified class.<br>• Enter the class name.<br>**Note** The **match-all** keyword specifies that all match criteria must be met. The **match-any** keyword specifies that one of the match criterion must be met. Use these keywords only if you will be specifying more than one **match** command. |
| **Step 4** | **match cos** *cos-number*<br>**Example:**<br>`Router(config-cmap)# match cos 2` | Matches a packet on the basis of a Layer 2 class of service (CoS) number.<br>• Enter the CoS number.<br>**Note** The **match cos** command is an example of the **match** commands you can use. For information about the other **match** commands that are available, see the "match-all and match-any Keywords of the class-map Command" section. |
| **Step 5** | Enter additional match commands, if applicable; otherwise, continue with step 6. | -- |
| **Step 6** | **end**<br>**Example:**<br>`Router(config-cmap)# end` | (Optional) Exits QoS class-map configuration mode and returns to privileged EXEC mode. |

# Creating a Traffic Policy

**Note** The **bandwidth** command is shown in Step 5. The **bandwidth** command is an example of the commands that you can use in a policy map to enable a QoS feature (in this case, Class-based Weighted Fair Queuing (CBWFQ). For information about other available commands, see the "Elements of a Traffic Policy" section.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name* | **class-default**}
5. **bandwidth** {*bandwidth-kbps* | **percent** *percent*}
6. Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Step 7.
7. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br>`Router(config)# policy-map policy1` | Creates or specifies the name of the traffic policy and enters QoS policy-map configuration mode.<br>• Enter the policy map name. |
| **Step 4** | **class** {*class-name* | **class-default**}<br><br>**Example:**<br>`Router(config-pmap)# class class1` | Specifies the name of a traffic class and enters QoS policy-map class configuration mode.<br>**Note** This step associates the traffic class with the traffic policy. |
| **Step 5** | **bandwidth** {*bandwidth-kbps* | **percent** *percent*}<br><br>**Example:**<br>`Router(config-pmap-c)# bandwidth 3000` | (Optional) Specifies a minimum bandwidth guarantee to a traffic class in periods of congestion.<br>• A minimum bandwidth guarantee can be specified in kb/s or by a percentage of the overall available bandwidth.<br>**Note** The **bandwidth** command enables CBWFQ. The **bandwidth** command is an example of the commands that you can use in a policy map to enable a QoS feature. For information about the other commands available, see the "Elements of a Traffic Policy" section. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Step 7. | -- |
| **Step 7** | **end**<br><br>**Example:**<br><br>`Router(config-pmap-c)# end` | (Optional) Exits QoS policy-map class configuration mode and returns to privileged EXEC mode. |

# Attaching a Traffic Policy to an Interface Using the MQC

**Note** A traffic policy containing a queueing mechanism or feature cannot be attached to a physical interface or subinterface.

**Note** Cisco IOS XE Release 2.3.0 and later releases do not support the attachment of policies for ATM interfaces that have unspecified bit rate (UBR) configured as the default mode on their VC or virtual path (VP). An attempt to use this configuration results in an error message: CBWFQ: Not supported on ATM interfaces with UBR configuration. You can also specify UBR with a rate in the UBR configuration, if you do not want to use the default UBR value.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **service-policy** {**input** | **output**} *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Router> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *type number*<br><br>**Example:**<br>`Router(config)# interface serial 0/0/1` | Configures an interface type and enters interface configuration mode.<br>• Enter the interface type and interface number. |
| **Step 4** | **service-policy** {**input** \| **output**} *policy-map-name*<br><br>**Example:**<br>`Router(config-if)# service-policy input policy1` | Attaches a policy map to an interface.<br>• Enter either the **input** or **output** keyword and the policy map name. |
| **Step 5** | **end**<br><br>**Example:**<br>`Router(config-if)# end` | (Optional) Exits interface configuration mode and returns to privileged EXEC mode. |

# Verifying the Traffic Class and Traffic Policy Information

The show commands described in this section are optional and can be entered in any order.

## SUMMARY STEPS

1. **enable**
2. **show class-map**
3. **show policy-map** *policy-map-name* **class** *class-name*
4. **show policy-map**
5. **show policy-map interface** *type number*
6. **exit**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router> enable` | • Enter your password if prompted. |
| **Step 2** | **show class-map**<br>**Example:**<br>`Router# show class-map` | (Optional) Displays all class maps and their matching criteria. |
| **Step 3** | **show policy-map** *policy-map-name* **class** *class-name*<br>**Example:**<br>`Router#`<br>`show policy-map policy1 class class1` | (Optional) Displays the configuration for the specified class of the specified policy map.<br>• Enter the policy map name and the class name. |
| **Step 4** | **show policy-map**<br>**Example:**<br>`Router# show policy-map` | (Optional) Displays the configuration of all classes for all existing policy maps. |
| **Step 5** | **show policy-map interface** *type number*<br>**Example:**<br>`Router# show policy-map interface serial 0/0/1` | (Optional) Displays the statistics and the configurations of the input and output policies that are attached to an interface.<br>• Enter the interface type and number. |
| **Step 6** | **exit**<br>**Example:**<br>`Router# exit` | (Optional) Exits privileged EXEC mode. |

# Configuration Examples for Applying QoS Features Using the MQC

## Example: Creating a Traffic Class

In the following example, two traffic classes are created and their match criteria are defined. For the first traffic class called class1, access control list (ACL) 101 is used as the match criterion. For the second traffic class called class2, ACL 102 is used as the match criterion. Packets are checked against the contents of these ACLs to determine if they belong to the class.

```
Router(config)# class-map class1
Router(config-cmap)# match access-group 101
Router(config-cmap)# exit
Router(config)# class-map class2
Router(config-cmap)# match access-group 102
Router(config-cmap)# end
```

## Example Creating a Traffic Policy

In the following example, a traffic policy called policy1 is defined. The traffic policy contains the QoS features to be applied to two classes--class1 and class2. The match criteria for these classes were previously defined (as described in the Example Creating a Traffic Class).

For class1, the policy includes a bandwidth allocation request and a maximum packet count limit for the queue reserved for the class. For class2, the policy specifies only a bandwidth allocation request.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class1
Router(config-pmap-c)# bandwidth 3000
Router(config-pmap-c)# queue-limit 30
Router(config-pmap-c)# exit
Router(config-pmap)# class class2
Router(config-pmap-c)# bandwidth 2000
Router(config-pmap-c)# end
```

## Example: Attaching a Traffic Policy to an Interface

The following example shows how to attach an existing traffic policy to an interface. After you define a traffic policy with the **policy-map** command, you can attach it to one or more interfaces by using the **service-policy** command in interface configuration mode. Although you can assign the same traffic policy to multiple interfaces, each interface can have only one traffic policy attached in the input direction and only one traffic policy attached in the output direction.

```
Router(config)# interface fastethernet 1/1/1
Router(config-if)# service-policy output policy1
Router(config-if)# exit
Router(config)# interface fastethernet 1/0/0
Router(config-if)# service-policy output policy1
Router(config-if)# end
```

# Example: match not Command

The **match not**command is used to specify a specific QoS policy value that is not used as a match criterion. If the **match not** command is issued, all other values of that QoS policy become successful match criteria. For instance, if the **match not qos-group 4** command is issued in QoS class-map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

In the following traffic class, all protocols except IP are considered successful match criteria:

```
Router(config)# class-map noip
Router(config-cmap)# match not protocol ip
Router(config-cmap)# end
```

# Example: Default Traffic Class Configuration

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 you do not configure a default class, packets are still treated as members of the default class. However, by default, the default class has no QoS features enabled. Therefore, packets belonging to a default class have no QoS functionality. These packets are placed into a first-in, first-out (FIFO) queue managed by tail drop. Tail drop is a means of avoiding congestion that treats all traffic equally and does not differentiate between classes of service. Queues fill during periods of congestion. When the output queue is full and tail drop is in effect, packets are dropped until the congestion is eliminated and the queue is no longer full.

The following example configures a traffic policy for the default class of the traffic policy called policy1. The default class (which is always called class-default) has these characteristics: 10 queues for traffic that does not meet the match criteria of other classes whose policy is defined by the traffic policy policy1, and a maximum of 20 packets per queue before tail drop is enacted to handle additional queued packets.

```
Router(config)# policy-map policy1
Router(config-pmap)# class class-default
Router(config-pmap-c)# fair-queue
Router(config-pmap-c)# queue-limit 20
```

# Example: class-map match-any and class-map match-all Commands

This example illustrates the difference between the **class-map match-any** command and the **class-map match-all** command. The **match-any** and **match-all** keywords determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all**) or meet one of the match criteria (**match-any**) to be considered a member of the traffic class.

The following example shows a traffic class configured with the **class-map match-all**c ommand:

```
Router(config)# class-map match-all cisco1
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

If a packet arrives on a router with the traffic class called cisco1 configured on the interface, the packet is evaluated to determine if it matches the IP protocol, QoS group 4, *and* access group 101. If all three of these match criteria are met, the packet is classified as a member of the traffic class cisco1.

The following example shows a traffic class that is configured with the **class-map match-any** command:

```
Router(config)# class-map match-any cisco2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# match access-group 101
```

In the traffic class called cisco2, the match criteria are evaluated consecutively until a successful match criterion is located. The packet is first evaluated to determine whether the IP protocol can be used as a match criterion. If the IP protocol can be used as a match criterion, the packet is matched to traffic class cisco2. If the IP protocol is not a successful match criterion, then QoS group 4 is evaluated as a match criterion. Each criterion is evaluated to see if the packet matches that criterion. Once a successful match occurs, the packet is classified as a member of traffic class cisco2. If the packet matches none of the specified criteria, the packet is classified as a member of the default traffic class (class default-class).

Note that the **class-map match-all** command requires that *all* of the match criteria be met in order for the packet to be considered a member of the specified traffic class (a logical AND operator). In the first example, protocol IP AND QoS group 4 AND access group 101 must be successful match criteria. However, only one match criterion must be met in order for the packet in the **class-map match-any** command to be classified as a member of the traffic class (a logical OR operator). In the second example, protocol IP OR QoS group 4 OR access group 101 must be successful match criterion.

# Example: Traffic Class as a Match Criterion (Nested Traffic Classes)

There are two reasons to use the **match class-map** command. One reason is maintenance; if a large traffic class currently exists, using the traffic class match criterion is easier than retyping the same traffic class configuration. The more common reason for the **match class-map** command is to allow users to use match-any and match-all statements in the same traffic class. If you want to combine match-all and match-any characteristics in a traffic policy, create a traffic class using one match criterion evaluation instruction (either match-any or match-all) and then use this traffic class as a match criterion in a traffic class that uses a different match criterion type.

Here is a possible scenario: Suppose A, B, C, and D were all separate match criterion, and you wanted traffic matching A, B, or C and D (A or B or [C and D]) to be classified as belonging to the traffic class. Without the nested traffic class, traffic would either have to match all four of the match criterion (A and B and C and D) or match any of the match criterion (A or B or C or D) to be considered part of the traffic class. You would not be able to combine “and” (match-all) and “or” (match-any) statements within the traffic class, and you would therefore be unable to configure the desired configuration.

The solution: Create one traffic class using match-all for C and D (which we will call criterion E), and then create a new match-any traffic class using A, B, and E. The new traffic class would have the correct evaluation sequence (A or B or E, which would also be A or B or [C and D]). The desired traffic class configuration has been achieved.

The only method of mixing match-all and match-any statements in a traffic class is through the use of the traffic class match criterion.

## Example: Nested Traffic Class for Maintenance

In the following example, the traffic class called class1 has the same characteristics as the traffic class called class2, with the exception that traffic class class1 has added a destination address as a match criterion. Rather than configuring traffic class class1 line by line, you can enter the **match class-map class2** command. This command allows all of the characteristics in the traffic class called class2 to be included in the traffic class

called class1, and you can add the new destination address match criterion without reconfiguring the entire traffic class.

```
Router(config)# class-map match-any class2
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 3
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# class-map match-all class1
Router(config-cmap)# match class-map class2
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# exit
```

## Example: Nested Traffic Class to Combine match-any and match-all Characteristics in One Traffic Class

The only method of including both match-any and match-all characteristics in a single traffic class is to use the **match class-map** command. To combine match-any and match-all characteristics into a single class, use the match-any instruction to create a traffic class that uses a class configured with the match-all instruction as a match criterion (through the **match class-map** command).

The following example shows how to combine the characteristics of two traffic classes, one with match-any and one with match-all characteristics, into one traffic class with the **match class-map** command. The result requires a packet to match one of the following three match criteria to be considered a member of traffic class class4: IP protocol *and* QoS group 4, destination MAC address 00.00.00.00.00.00, or access group 2.

In this example, only the traffic class called class4 is used with the traffic policy called policy1.

```
Router(config)# class-map match-all class3
Router(config-cmap)# match protocol ip
Router(config-cmap)# match qos-group 4
Router(config-cmap)# exit
Router(config)# class-map match-any class4
Router(config-cmap)# match class-map class3
Router(config-cmap)# match destination-address mac 00.00.00.00.00.00
Router(config-cmap)# match access-group 2
Router(config-cmap)# exit
Router(config)# policy-map policy1
Router(config-pmap)# class class4
Router(config-pmap-c)# police 8100 1500 2504 conform-action transmit exceed-action
set-qos-transmit 4
Router(config-pmap-c)# end
```

# Example: Traffic Policy as a QoS Policy (Hierarchical Traffic Policies)

A traffic policy can be included in a QoS policy when the **service-policy** command is used in QoS policy-map class configuration mode. A traffic policy that contains a traffic policy is called a hierarchical traffic policy.

A hierarchical traffic policy contains a child policy and a parent policy. The child policy is the previously defined traffic policy that is being associated with the new traffic policy through the use of the **service-policy** command. The new traffic policy using the preexisting traffic policy is the parent policy. In the example in this section, the traffic policy called child is the child policy and traffic policy called parent is the parent policy.

Hierarchical traffic policies can be attached to subinterfaces and ATM PVCs. When hierarchical traffic policies are used, a single traffic policy (with a child and a parent policy) can be used to shape and prioritize permanent virtual connection (PVC) traffic. In the following example, the child policy is responsible for prioritizing

traffic and the parent policy is responsible for shaping traffic. In this configuration, the parent policy allows packets to be sent from the interface, and the child policy determines the order in which the packets are sent.

```
Router(config)# policy-map child
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 50
Router(config)# policy-map parent
Router(config-pmap)# class class-default
Router(config-pmap-c)# shape average 10000000
Router(config-pmap-c)# service-policy child
```

The value used with the **shape** command is provisioned from the committed information rate (CIR) value from the service provider.

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Quality of Service Solutions Command Reference* |
| Packet classification | "Classifying Network Traffic" module |
| Frame Relay Fragmentation (FRF) PVCs | "FRF .20 Support" module |
| Selective Packet Discard | "IPv6 Selective Packet Discard" module |
| Scaling and performance information | "Broadband Scalability and Performance" module of the Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide. |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Applying QoS Features Using the MQC

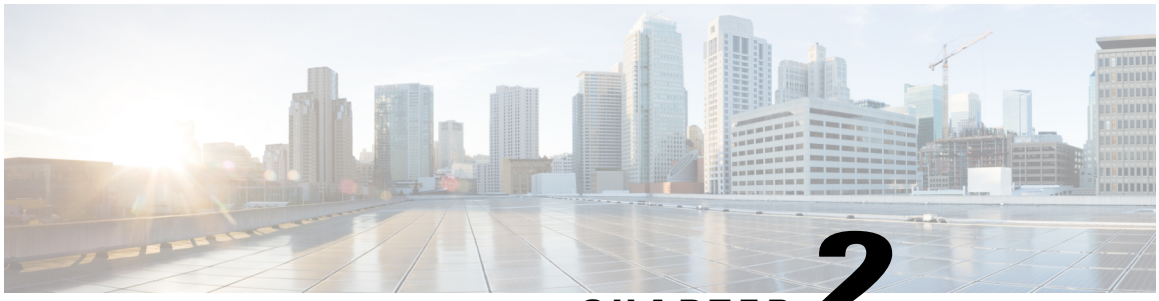
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 3: Feature Information for Applying QoS Features Using the MQC***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Class-Based Weighted Fair Queueing (CBWFQ) | Cisco IOS XE Release 2.1<br>Cisco IOS XE Release 3.5S | This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.<br>In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router. |
| Modular QoS CLI (MQC) | Cisco IOS XE Release 2.1<br>Cisco IOS XE Release 3.5S | This module describes how to apply and configure quality of service (QoS) features using the modular QoS CLI (MQC). The MQC allows you to define a traffic class, create a traffic policy (policy map), and attach the traffic policy to an interface. The traffic policy contains the QoS feature that will be applied to the traffic class.<br>This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.<br>This feature was enhanced to provide infrastructure support for additional features included with Cisco IOS XE Release 2.3.<br>In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 router. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| MQC Hierarchical Class Map | Cisco IOS XE Release 3.2 | MQC allows multiple traffic classes (nested traffic classes, which are also called nested class maps or MQC hierarchical class maps) to be configured as a single traffic class. This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. |
| Priority Queueing | Cisco IOS XE Release 2.1 | This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. |
| Weighted Random Early Detection | Cisco IOS XE Release 2.1 | This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers. |



CHAPTER 2

# QoS: Policies Aggregation

The QoS: Policies Aggregation feature for the Cisco ASR 1000 Series Aggregation Services Routers supports Modular QoS CLI (MQC) configuration of default traffic classes in policy maps on different subinterfaces to be queued as a single, user-defined traffic class at the main-interface policy map. It is most useful in quality of service (QoS) configurations where you have several subinterface policy maps on the same physical interface and you want identical treatment of the default traffic classes on those subinterfaces.

Beginning in Cisco IOS XE Release 2.6, the QoS: Policies Aggregation feature is enhanced to support queueing aggregation at the primary interface for other traffic classes, including Differentiated Services Code Point (DSCP) traffic classes such as the expedited forwarding (EF), Assured Forwarding 1 (AF1), and AF4 traffic classes. With this enhancement, any traffic classes from VLAN subinterfaces can share a common queue for that traffic class at the main-interface policy map. Other enhancements include the ability to configure and show drop statistics that occur at the aggregate level for these classes.



# Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Prerequisites for QoS: Policies Aggregation

- This feature is configured using the MQC.
- All traffic over the main interface should come through one or more subinterfaces.

# Restrictions for QoS: Policies Aggregation

- Applies only when multiple subinterfaces with policy maps are attached to the same physical interface. This feature cannot be used to collectively classify default traffic classes or other traffic classes of policy maps on different physical interfaces.
- Certain traffic class configuration prior to Cisco IOS XE Release 2.6 at the subinterface policy map and main-interface policy map will have different behavior and queueing results. See the "Understanding the QoS Policies Aggregation MQC" section on page 3 and the "Differences Between the Original Feature and the MQC Support for Multiple Queue Aggregation" section on page 4.
- The **service-fragment** keyword is only supported on the Gigabit Ethernet interfaces and not on Fast Ethernet interfaces.

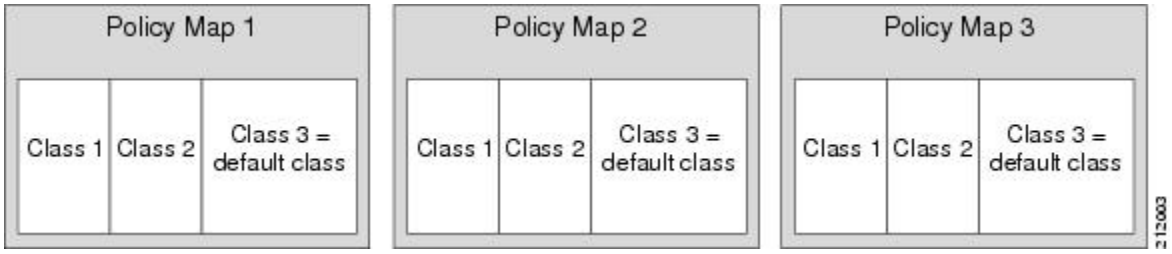
# Information About QoS: Policies Aggregation

## Understanding Fragments in Class Definition Statements

QoS: Policies Aggregation introduces the idea of fragments in class definition statements. A default traffic class definition statement can be marked as a fragment within a policy map. Other policy maps on the same interface can also define their default traffic class statements as fragments, if desired. A separate policy map can then be created with a service-fragment class definition statement that will be used to apply QoS to all of the fragments as a single group.

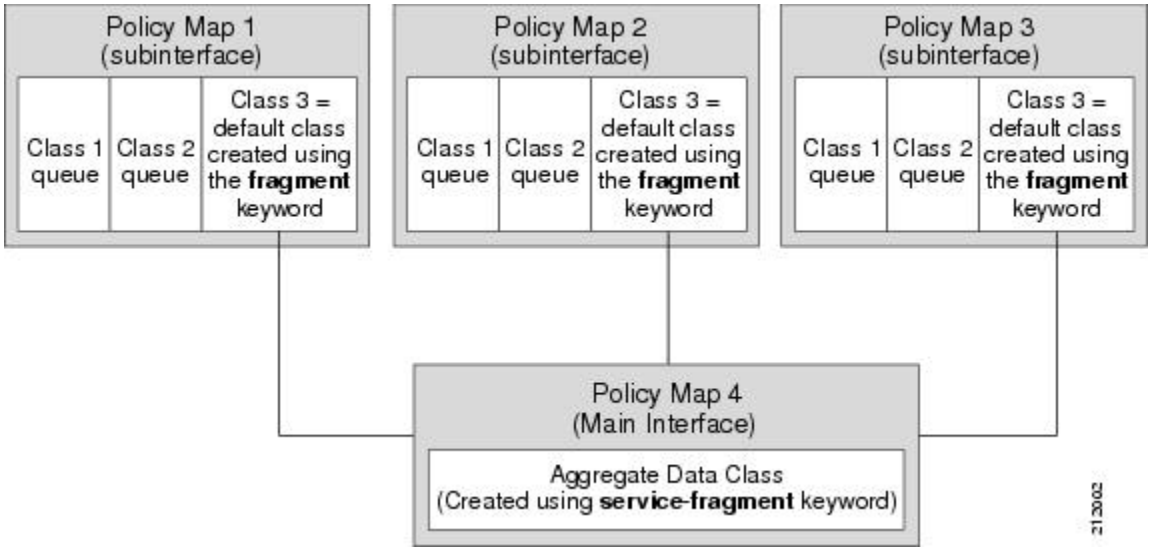
The figure below provides an example of one physical interface with three attached policy maps that is not using fragments. Note that each policy map has a default traffic class that can only classify traffic for the default traffic within its own policy map.

***Figure 1: Three Policy Maps Configured Without Fragments***




The figure below shows the same configuration configured with fragments and adds a fourth policy map with a class definition statement that classifies the fragments collectively. The default traffic classes are now classified as one service-fragment group rather than three separate default traffic classes within the individual policy maps.

***Figure 2: Three Policy Maps Configured Using Fragments***




# Understanding Fragments for Gigabit Etherchannel Bundles

When fragments are configured for Gigabit Etherchannel bundles, the policy maps that have a default traffic class configured using the **fragment** keyword are attached to the member subinterface links, and the policy maps that have a traffic class configured with the **service-fragment** keyword to collectively classify the fragments is attached to the physical interface.

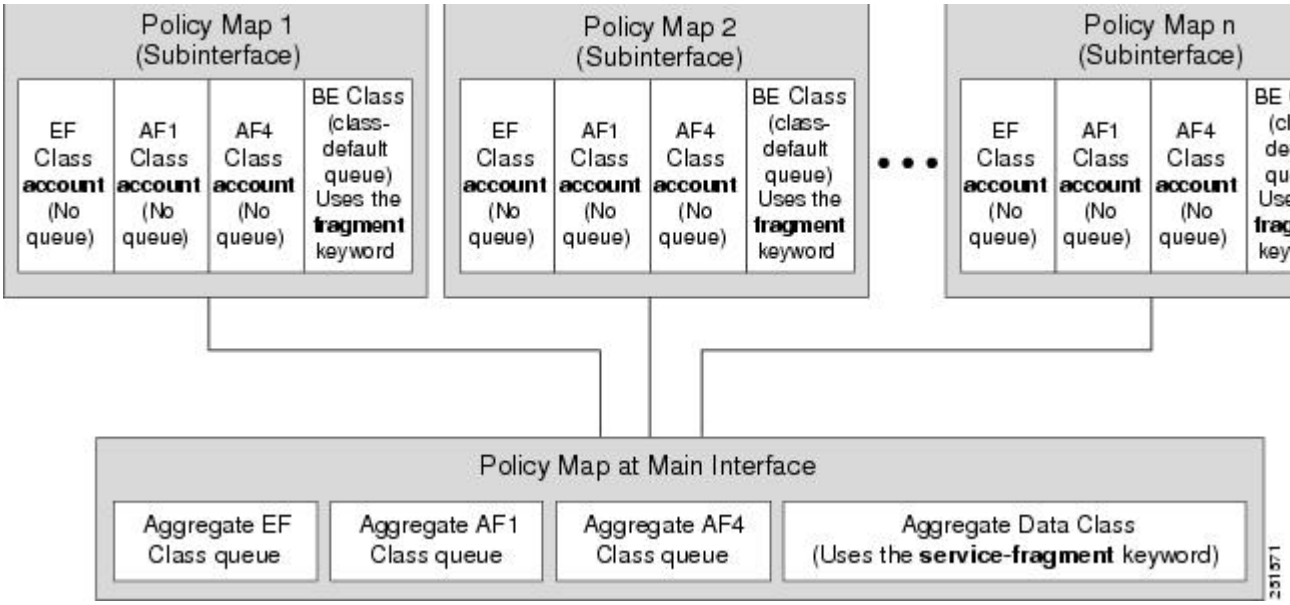
All port-channel subinterfaces configured with fragments that are currently active on a given port-channel member link will use the aggregate service fragment class on that member link. If a member link goes down, the port-channel subinterfaces that must switch to the secondary member link will then use the aggregate service fragment on the new interface.

# Understanding the QoS: Policies Aggregation MQC

The QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature extends the previous support of aggregation of class-default traffic using the **fragment** and **service-fragment** configurations, to other user-defined traffic classes in a subinterface policy map, such as DSCP-based traffic classes, that are aggregated at the main-interface policy map as shown in the figure below.

When no queueing is configured on a traffic class in the subinterface policy map, the **account** command can be used to track queueing drops that occur at the aggregate level for these classes, and can be displayed using the **show policy-map interface** command.

***Figure 3: Policy Map Overview for the MQC Support for Multiple Queue Aggregation at Main Interface Feature***




## Differences Between the Original Feature and the MQC Support for Multiple Queue Aggregation

Although some of the configuration between the original QoS policies aggregation feature and enhancements in the MQC Support for Multiple Queue Aggregation at Main Interface feature appears similar, there are some important differences in the queueing behavior and the internal data handling.

For example, both configurations share and require the use of the **fragment** keyword for the **class class-default** command in the subscriber policy map, as well as configuration of the **service-fragment** keyword for a user-defined class in the main-interface policy map to achieve common policy treatment for aggregate traffic. However, the use of this configuration results in different behavior between the original and enhanced QoS policies aggregation implementation:

- In the original implementation using the fragment and service-fragment architecture, all default class traffic and any traffic for classes without defined queueing features at the subinterface goes to the class-default queue and is aggregated into a common user-defined queue and policy defined at the main policy map. Subinterface traffic aggregation (for example, from multiple subscribers on the same physical interface) ultimately occurs only for a single class, which is the default class.

- In the enhanced implementation of the MQC Support for Multiple Queue Aggregation at Main Interface feature also using the fragment and service-fragment architecture, all default class traffic also goes to the class-default queue and is aggregated into a common user-defined queue and policy defined at the main policy map. However, other classes, such as DSCP-based subscriber traffic classes, are also supported for an aggregate policy. These traffic classes do not support any queues or queueing features other than **account** at the subscriber policy map. The use of the fragment and service-fragment architecture enables these other subscriber traffic classes (from multiple subscribers on the same physical interface) to achieve common policy treatment for aggregate traffic that is defined for those same classes at the main policy map.

The following sections summarize the key behavioral differences between the original QoS: Policies Aggregation feature and the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature.

### QoS: Policies Aggregation Feature Prior to Cisco IOS XE Release 2.6

- All subinterface traffic classes have queues. However, when a traffic class in the subinterface policy-map is not configured with any queueing feature (commands such as **priority**, **shape**, **bandwidth**, **queue-limit**, **fair-queue**, **random-detect**, and so on, are not configured), the traffic is assigned to the class-default queue.
- Default class traffic from multiple subinterfaces can be aggregated into a common policy map at the main interface when you use the **fragment** keyword at the subinterface **class class-default** configuration, and **service-fragment** configuration at the main-interface class.
- No classification occurs or is supported at the main-interface policy map for any subinterface traffic classes that do not use the **fragment** and **service-fragment** configuration.
- Queueing occurs at the subinterface for other traffic classes defined with queueing features in the subinterface policy map.

### QoS: Policies Aggregation - MQC Support for Multiple Queue Aggregation at Main Interface Feature Beginning in Cisco IOS XE Release 2.6

- Subinterface traffic classes without configured queueing features do not have queues at the subscriber level.
- Default class traffic from multiple subinterfaces can be aggregated into a common policy map at the main interface when you use the **fragment** keyword at the subinterface **class class-default** configuration, and **service-fragment** configuration at the main-interface class. This configuration additionally enables support for other subinterface traffic classes (such as DSCP-based classes) to be aggregated into a common policy-map at the main interface.
- Other class traffic from multiple subinterfaces can be aggregated into a common policy map at the main interface, according to the following configuration requirements:
- You enable this behavior by using the **fragment** keyword at the subinterface **class class-default** configuration, and **service-fragment** configuration at the main-interface class (this also enables aggregation of the default class).
- You do not configure any queueing features at the subinterface policy-map for the other traffic classes.
- Queueing occurs at the main-interface policy map for other subinterface traffic classes as an aggregate.

- Optional tracking of statistics is supported using the **account** command for other traffic classes in the subinterface policy map.

### Changes in Queue Limit and WRED Thresholds

In Cisco IOS XE Release 2.6 the Cisco ASR 1000 Series Routers support the addition of bytes as a unit of configuration for both queue limits and WRED thresholds. Therefore, as of this release, packet-based and byte-based limits are configurable, with some restrictions.

# How to Configure QoS: Policies Aggregation

## Configuring QoS: Policies Aggregation for an Interface

### Configuring a Fragment Traffic Class in a Policy Map

#### Before You Begin

This procedure shows only how to configure the default traffic class as a fragment within a policy map. It does not include steps on configuring other classes within the policy map, or other policy maps on the device.

**Note** Only the default class statement in a policy map can be configured as a fragment.

Fragments work only when multiple policy maps are attached to the same physical interface. This process cannot be used to classify default traffic classes as fragments on policy maps on different physical interfaces.

Only queuing features are allowed in classes where the **fragment** keyword is entered, and at least one queuing feature must be entered in classes where the **fragment** keyword is used.

A policy map with a class using the **fragment** keyword can only be applied to traffic leaving the interface (policy maps attached to interfaces using the **service-policy output** command).

The **fragment** keyword cannot be entered in a child policy map.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class class-default fragment** *fragment-class-name*
5. **shape average percent** *percent*
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br>`Device(config)# policy-map subscriber1` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class class-default fragment** *fragment-class-name*<br><br>**Example:**<br>`Device(config-pmap)# class class-default fragment BestEffort` | Specifies the default traffic class as a fragment, and names the fragment traffic class. |
| **Step 5** | **shape average percent** *percent*<br><br>**Example:**<br>`Device(config-pmap-c)# shape average percent 50` | Enters a QoS configuration command. Only queuing features are supported in default traffic classes configured as fragments.<br>The queuing features supported are **bandwidth**, **shape**, and **random-detect exponential-weighting-constant**.<br>Multiple QoS queuing commands can be entered. |
| **Step 6** | **end**<br><br>**Example:**<br>`Device(config-pmap-c)# end` | Exits policy map class configuration mode and returns to privileged EXEC mode. |

### Example

**Note** This example shows a sample configuration that is supported in releases prior to Cisco IOS XE Release 2.6.

In the following example, a fragment named BestEffort is created in policy map subscriber1 and policy map subscriber 2. In this example, queuing features for other traffic classes are supported at the subinterface policy map.

```
policy-map subscriber1
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber 2
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

**Note** This example shows a sample configuration that is supported in Cisco IOS XE Release 2.6 and later releases.

The following example also shows how to configure a fragment named BestEffort for the default class in a policy map on a subinterface using the QoS Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface implementation. In this example, notice that queuing features are not supported for the other classes in the policy map:

```
policy-map subscriber1
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

## What to Do Next

After configuring default class statements as fragments in multiple subinterface policy maps, a separate policy map with a class statement using the **service-fragment** keyword must be configured to apply QoS to the class statements configured as fragments.

This task is documented in the "Configuring a Service Fragment Traffic Class" section on page 8.

## Configuring a Service Fragment Traffic Class

### Before You Begin

This task describes how to configure a service fragment traffic class statement within a policy map. A service fragment traffic class is used to apply QoS to a collection of default class statements that have been configured previously in other policy maps as fragments.

This procedure assumes that fragment default traffic classes were already created. The procedure for creating fragment default traffic classes is documented in the "Configuring a Fragment Traffic Class in a Policy Map" section.

Like any policy map, the configuration does not manage network traffic until it has been attached to an interface. This procedure does not cover the process of attaching a policy map to an interface.

**Note** A service fragment can be used to collectively classify fragments only from the same physical interface. Fragments from different interfaces cannot be classified using the same service fragment.

Only queueing features are allowed in classes where the **service-fragment** keyword is entered, and at least one queueing feature must be entered in classes when the **service-fragment** keyword is used.

A policy map with a class using the **service-fragment** keyword can be applied only to traffic leaving the interface (policy maps attached to interfaces using the **service-policy output** command).

A class configured using the **service-fragment** keyword cannot be removed when it is being used to collectively apply QoS to fragments that are still configured on the interface. If you wish to remove a class configured using the **service-fragment** keyword, remove the fragment traffic classes before removing the service fragment.

The **service-fragment** keyword cannot be entered in a child policy map.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name* **service-fragment** *fragment-class-name*
5. **shape average percent** *percent*
6. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br><br>`Device(config)# policy-map BestEffortFragments` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class** *class-name* **service-fragment** *fragment-class-name*<br><br>**Example:**<br><br>`Device(config-pmap)# class data service-fragment BestEffort` | Specifies a class of traffic that is the composite of all fragments matching the *fragment-class-name*. The *fragment-class-name* when defining the fragments in other policy maps must match the *fragment-class-name* in this command line to properly configure the service fragment class. |
| **Step 5** | **shape average percent** *percent*<br><br>**Example:**<br><br>`Device(config-pmap-c)# shape average percent 50` | Enters a QoS configuration command. Only queueing features are supported in default traffic classes configured as fragments.<br><br>The queueing features that are supported are **bandwidth**, **shape**, and **random-detect exponential-weighting-constant**.<br><br>Multiple QoS queueing commands can be entered. |
| **Step 6** | **end**<br><br>**Example:**<br><br>`Device(config-pmap-c)# end` | Exits policy map class configuration mode and returns to privileged EXEC mode. |

### Examples

**Note** This example shows a sample configuration that is supported in releases prior to Cisco IOS XE Release 2.6.

In the following example, a policy map is created to apply QoS to all fragments named BestEffort.

```
policy-map main-interface
 class data service-fragment BestEffort
  shape average 400000000
```

In the following example, two fragments are created and then classified collectively using a service fragment.

```
policy-map subscriber1
 class voice
  set cos 5
```

```
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber 2
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

**Note** This example shows a sample configuration that is supported in Cisco IOS XE Release 2.6 and later releases.

The following example shows the creation of two fragments called BestEffort in the subinterface policy maps, followed by a sample configuration for the **service-fragment** called BestEffort to aggregate the queues at the main interface policy map:

```
policy-map subscriber1
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber2
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map main-interface
 class voice
  priority level 1
 class video
  priority level 2
 class AF1
  bandwidth remaining ratio 90
 class data service-fragment BestEffort
  shape average 400000000
  bandwidth remaining ratio 1
```

## Troubleshooting Tips

Ensure that all class statements that are supposed to be part of the same service fragment share the same *fragment-class-name*.

### What to Do Next

The policy map (traffic policy) must be attached to an interface. This task is documented in the "Attaching a Traffic Policy to an Interface Using the MQC" section in chapter "Applying QoS Features Using the MQC."

# Configuring QoS: Policies Aggregation on Gigabit Etherchannels

To properly configure QoS: Policies Aggregation on a Gigabit Etherchannel bundle, the following actions must be completed:

- Service-fragment traffic classes must be configured and attached to the main physical interfaces.
- Fragment traffic classes must be configured and attached to the member link subinterfaces.

## Configuring Service Fragments on a Physical Interface Supporting a Gigabit Etherchannel Bundle

### Before You Begin

This procedure assumes that a service fragment traffic class has already been created. A service fragment traffic class cannot be configured without configuring a fragment class. The procedure for creating a fragment class is documented in the “Configuring a Fragment Traffic Class in a Policy Map” section. The procedure for creating a service fragment traffic classes is documented in the “Configuring a Service Fragment Traffic Class” section.

These instructions do not provide any details about the options that can be configured for Gigabit Etherchannel member link subinterfaces. These instructions document only the procedure for attaching a policy map that already has a fragment traffic class to a member link subinterface.

**Note** For proper behavior, when a port-channel member link goes down, all member links should have the same policy map applied.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *card/bay/port*
4. **service-policy output** *service-fragment-class-name*
5. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable** | Enables privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Device> enable` | • Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface GigabitEthernet** *card/bay/port*<br>**Example:**<br>`Device(config)# interface GigabitEthernet 0/1/0` | Specifies the member link physical interface that receives the service-policy configuration. |
| **Step 4** | **service-policy output** *service-fragment-class-name*<br>**Example:**<br>`Device(config-if)# service-policy output aggregate-member-link` | Attaches a service policy that contains a service fragment default traffic class to the physical Gigabit Ethernet interface. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

### Examples

In the following example, the policy map aggregate-member-link is attached to the physical interface.

```
interface GigabitEthernet1/1/1
 service-policy output aggregate-member-link
!
interface GigabitEthernet1/1/2
 service-policy output aggregate-member-link
```

### What to Do Next

Ensure that the fragment class name is consistent across service-fragment and fragment class definitions. Continue to the “Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces” section.

## Troubleshooting Tips

Ensure that the *fragment-class-name* is consistent across service-fragment and fragment-class definitions.

### What to Do Next

Attach the fragment service policy on the Gigabit Etherchannel member link subinterfaces. This task is documented in the "Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces" section on page 14.

## Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces

### Before You Begin

This task assumes that a service-fragment traffic class has already been created. A service-fragment traffic class cannot be configured without configuring a fragment class. The procedure for creating a fragment class is documented in the "Configuring a Fragment Traffic Class in a Policy Map" section on page 6. The procedure for creating a service-fragment traffic classes is documented in the "Configuring a Service Fragment Traffic Class" section on page 8.

These instructions do not provide any details about the options that can be configured for Gigabit Etherchannel member link subinterfaces. These instructions only document the procedure for attaching a policy map that already has a fragment traffic class to a member link subinterface.

**Note** Fragments cannot be used for traffic on two or more physical interfaces. The GEC must all be on the same physical interface for this configuration to work properly.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-interface-number.port-channel-subinterface-number*
4. **service-policy output** *fragment-class-name*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **interface port-channel** *port-channel-interface-number.port-channel-subinterface-number*<br><br>**Example:**<br>`Router(config)# interface port-channel 1.100` | Enters subinterface configuration mode to configure a Etherchannel member link subinterface. |
| **Step 4** | **service-policy output** *fragment-class-name*<br><br>**Example:**<br>`Router(config-subif)# service-policy output subscriber` | Attaches a service policy that contains a fragment default traffic class to the Etherchannel member link subinterface. |

### Example

**Note** This example shows a sample configuration that is supported for the original QoS: Policies Aggregation feature in releases prior to Cisco IOS XE Release 2.6. By following the newer policy-map configuration guidelines for the updates in Cisco IOS XE Release 2.6, it can be adapted to the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature.

In the following example, the service policy named subscriber has a fragment default traffic class and is attached to the member link subinterface of a Gigabit Etherchannel bundle.

**Note** This example only shows how to attach a fragment default traffic class to the member link subinterface of a Gigabit Etherchannel bundle. This configuration is incomplete and would not classify default traffic appropriately until the physical interface was configured to support a service-fragment traffic class.

```
policy-map subscriber
 class voice
  priority level 1
 class video
  priority level 2
 class class-default fragment BE
  shape average 100000000
  bandwidth remaining ratios 80
policy-map aggregate-member-link
 class BestEffort service-fragment BE
  shape average 100000000
!
interface Port-channel1
 ip address 172.16.2.3 255.255.0.0
!
interface Port-channel1.100
 encapsulation dot1Q 100
 ip address 192.168.2.100 255.255.255.0
 service-policy output subscriber
!
```

### Troubleshooting Tips

This configuration will not work until a service-fragment default traffic class is created to classify the default traffic classes marked as fragments. This service-fragment traffic class must be configured for this configuration to have any affect on network traffic.

# How to Configure QoS: Policies Aggregation MQC

Some backward-compatibility exists between support of policies aggregation feature configuration in Cisco IOS XE Release 2.6 and prior Cisco IOS XE software releases. However, we recommend that you follow these upgrade guidelines for any physical interface where you want to move to the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature configuration.

For best results, you should upgrade any service policies configuration that you implemented prior to Cisco IOS XE Release 2.6, to the latest supported configuration.

The original and enhanced QoS: Policies Aggregation feature configuration can only reside on the same Cisco ASR 1000 Series Router if the mixed configuration does not reside on the same physical interface. In other words, you can support the original configuration for one physical interface, and the enhanced configuration on a different physical interface.

The QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature requires the same configuration of a fragment traffic class as the original feature, using the **class class-default fragment** command to enable and then define all subinterface policies aggregation, both for the default traffic class and the other traffic classes.

In the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature, the queueing features for the aggregate class queues (with traffic from the corresponding classes identified at the subinterfaces), are configured at the main-interface policy map.

# Upgrading Your Service Policies for QoS: Policies Aggregation MQC

## Before You Begin

Upgrading your service policies to support the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature assumes the following network conditions:

- The corresponding class-map statements appropriate for your network traffic are already configured.
- QoS service policies aggregation has been previously configured and applied for the main-interface policy map for a given physical interface and its corresponding subinterfaces, or subscriber interfaces, prior to Cisco IOS XE Release 2.6 for the default traffic class.
- A port on the same physical interface where you have previously configured the service policies aggregation feature prior to Cisco IOS XE Release 2.6 needs to support the configuration for the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface.

## Upgrade Tasks

### SUMMARY STEPS

1. Configure the service policies for the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature.
2. Remove any service policies configured prior to Cisco IOS XE Release 2.6 for any prior configured policies aggregation features using the **no service-policy** and **no policy-map** commands as follows:
3. Apply the new service policies for the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature at the appropriate interfaces using the **service-policy output** command as follows:

### DETAILED STEPS

**Step 1** Configure the service policies for the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature.
See the tasks described in the "Configuring QoS Policies Aggregation MQC Traffic Classes" section on page 18.

**Step 2** Remove any service policies configured prior to Cisco IOS XE Release 2.6 for any prior configured policies aggregation features using the **no service-policy** and **no policy-map** commands as follows:

a) At each of the subinterfaces, configure the **no service-policy** command. Be sure to remove the policies at the subinterfaces first.
b) At the physical interface, configure the **no service-policy**command.

**Step 3** Apply the new service policies for the QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature at the appropriate interfaces using the **service-policy output** command as follows:

a) At the physical interface, configure the **service-policy output** command.
b) At each of the subinterfaces, configure the **service-policy output** command.

# Configuring QoS: Policies Aggregation MQC Traffic Classes

## Configuring Traffic Classes on the Subscriber Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **account** [**drop**]

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Router> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br><br>`Router(config)# policy-map subscriber1` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class** *class-name*<br><br>**Example:**<br><br>`Router(config-pmap)# class EF` | Specifies the name of the traffic class to be aggregated at the main-interface policy map, and enters policy-map class configuration mode.<br><br>**Note** Do not configure any queueing features for this class. Queueing is configured and aggregated at the main-interface policy map for all subinterfaces associated with this class and physical interface. |
| **Step 5** | **account** [**drop**]<br><br>**Example:**<br><br>`Router(config-pmap-c)# account` | (Optional) Enables collection of statistics for packets matching the traffic class where this command is configured, where the **drop** keyword collects all packet drop statistics. Collection of drop statistics is the default. |

### Example

The following example configures the EF traffic class for policies aggregation at the subscriber subinterface with collection of drop statistics:

```
policy-map subscriber1
 class EF
 account
```

### What to Do Next

Perform this task for all traffic classes that you want to aggregate, then perform the task in the "Configuring the Fragment Traffic Class on a Subinterface" section on page 19.

## Configuring the Fragment Traffic Class on a Subinterface

### What to Do Next

If you are upgrading your subinterface policy-map configuration from an earlier implementation of the QoS: Policies Aggregation feature, then remove the current service-policy from the subinterface using the **no service-policy** command.

Apply the new policy map to outbound traffic on the subinterface using the **service-policy output** command.

## Configuring Traffic Classes at the Main Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name*
5. **priority level** *level*

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br>`Router(config)# policy-map main-interface` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class** *class-name*<br><br>**Example:**<br>`Router(config-pmap)# class EF` | Specifies the name of the traffic class to be aggregated at the main-interface policy map, and enters policy-map class configuration mode. |
| **Step 5** | **priority level** *level* | Enters a QoS configuration command. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router(config-pmap-c)# priority level 1` | The queueing features that are currently supported are **bandwidth**, **priority**, **shape**, and **random-detect exponential-weighting-constant**.<br>Multiple QoS queueing commands can be entered. |

#### Example

The following example configures three traffic classes at the main-interface policy map, along with the aggregate service-fragment data class:

```
policy-map main-interface
 class voice
  priority level 1
 class video
  priority level 2
 class AF1
  bandwidth remaining ratio 90
 class data service-fragment BestEffort
  shape average 400000000
  bandwidth remaining ratio 1
```

### What to Do Next

Perform this task to define queueing features for all traffic classes that you want to aggregate, then perform the task in the "Configuring the Service Fragment Traffic Class at the Main Interface" section on page 21.

## Configuring the Service Fragment Traffic Class at the Main Interface

### What to Do Next

After configuring multiple default class statements as fragments in a policy map, a separate policy map with a class statement using the **service-fragment** keyword must be configured to apply QoS to the class statements configured as fragments.

This process is documented in the “Configuring a Service Fragment Traffic Class” section.

# Configuring QoS: Policies Aggregation MQC Support

The QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature also supports configuration of the enhanced service policies on Gigabit Etherchannels according to the subscriber and main-interface configuration guidelines described for this enhancement.

For more information, see the following sections:

# Verifying the Traffic Policy Class Policy Information and Drop Statistics

To display information about policy-map configuration and subscriber drop statistics enabled using the account command, use the **show policy-map interface** command:

```
Router# show policy-map interface port-channel 1.1
Port-channel1.1
   Service-policy input: input_policy
     Class-map: class-default (match-any)
       0 packets, 0 bytes
       5 minute offered rate 0000 bps, drop rate 0000 bps
       Match: any
       QoS Set
       dscp default
       No packet marking statistics available
   Service-policy output: Port-channel_1_subscriber
     Class-map: EF (match-any)
       105233 packets, 6734912 bytes
       5 minute offered rate 134000 bps, drop rate 0000 bps
       Match: dscp ef (46)
       Match: access-group name VLAN_REMARK_EF
       Match: qos-group 3
       Account QoS statistics
         Queueing
           Packets dropped 0 packets/0 bytes
       QoS Set
       cos 5
       No packet marking statistics available
       dscp ef
       No packet marking statistics available
     Class-map: AF4 (match-all)
       105234 packets, 6734976 bytes
       5 minute offered rate 134000 bps, drop rate 0000 bps
       Match: dscp cs4 (32)
       Account QoS statistics
         Queueing
           Packets dropped 0 packets/0 bytes
       QoS Set
       cos 4
       No packet marking statistics available
     Class-map: AF1 (match-any)
       315690 packets, 20204160 bytes
       5 minute offered rate 402000 bps, drop rate 0000 bps
       Match: dscp cs1 (8)
       Match: dscp af11 (10)
       Match: dscp af12 (12)
       Account QoS statistics
         Queueing
           Packets dropped 0 packets/0 bytes
       QoS Set
       cos 1
       No packet marking statistics available
     Class-map: class-default (match-any) fragment Port-channel_BE
       315677 packets, 20203328 bytes
       5 minute offered rate 402000 bps, drop rate 0000 bps
       Match: any
       Queueing
         queue limit 31250 bytes
         (queue depth/total drops/no-buffer drops) 0/0/0
         (pkts output/bytes output) 315679/20203482
         bandwidth remaining ratio 1
```

# Configuration Examples for QoS: Policies Aggregation

## Example: QoS: Policies Aggregation

**Note** This example shows a sample configuration that is supported in the original QoS: Policies Aggregation feature prior to Cisco IOS XE Release 2.6.

In the following example, QoS: Policies Aggregation is used to define a fragment class of traffic to classify default traffic using the default traffic class named BestEffort. All default traffic from the policy maps named subscriber1 and subscriber2 is part of the fragment default traffic class named BestEffort. This default traffic is then shaped collectively by creating a class called data that uses the **service-fragment** keyword and the **shape** command.

Note the following about this example:

- The *class-name* for each fragment default traffic class is "BestEffort."
- The *class-name* of "BestEffort" is also used to define the class where the **service-fragment** keyword is entered. This class applies a shaping policy to all traffic forwarded using the fragment default traffic classes named "BestEffort."

```
policy-map subscriber1
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber 2
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map input_policy
 class class-default
  set dscp default
policy-map main-interface
 class data service-fragment BestEffort
  shape average 400000000
interface portchannel1.1001
 encapsulation dot1q 1001
 service-policy output subscriber1
 service-policy input input_policy
interface portchannel1.1002
 encapsulation dot1q 1002
 service-policy output subscriber2
 service-policy input input_policy
interface gigabitethernet 0/1
 description member-link1
 port channel 1
```

```
 service-policy output main-interface
interface gigabitethernet 0/2
 description member-link2
 port channel 1
service-policy output main-interface
```

# Example: Gigabit Etherchannel QoS Policies Aggregation

**Note** This example shows a sample configuration that is supported in the original QoS: Policies Aggregation feature prior to Cisco IOS XE Release 2.6.

In the following example, policy map subscriber is configured with a fragment class named BE. The fragment is then configured as part of a policy map named aggregate-member-link. Policy map subscriber is then attached to the bundle subinterfaces while policy map aggregate-member-link is attached to the physical interface.

```
port-channel load-balancing vlan-manual
class-map match-all BestEffort
!
class-map match-all video
!
class-map match-all voice
!
policy-map subscriber
 class voice
  priority level 1
 class video
  priority level 2
 class class-default fragment BE
  shape average 100000000
  bandwidth remaining ratios 80
policy-map aggregate-member-link
 class BestEffort service-fragment BE
  shape average 100000000
!
interface Port-channel1
 ip address 10.1.1.3 255.255.0.0
!
interface Port-channel1.100
 encapsulation dot1Q 100
 ip address 10.1.2.1 255.255.255.0
 service-policy output subscriber
!
interface Port-channel1.200
 encapsulation dot1Q 200
 ip address 10.1.2.2 255.255.255.0
 service-policy output subscriber
!
interface Port-channel1.300
 encapsulation dot1Q 300
 ip address 10.1.2.3 255.255.255.0
 service-policy output subscriber
!
interface GigabitEthernet1/1/1
 no ip address
 channel-group 1 mode on
 service-policy output aggregate-member-link
!
interface GigabitEthernet1/1/2
 no ip address
 channel-group 1 mode on
 service-policy output aggregate-member-link
```

# Example: QoS: Policies Aggregation MQC Support at Main Interface

**Note** This example shows a sample configuration that is supported beginning in Cisco IOS XE Release 2.6.

At the main-interface policy map called Port-channel_1_main_policy, the queueing features for the DSCP-based subscriber traffic classes are configured. You can also see the use of byte-based queue limits and random-detect thresholds implemented at the main-interface queues.

The service fragment called Port-channel_BE is also configured to aggregate the traffic from the subscriber class-default fragment class.

```
policy-map Port-channel_1_main_policy
 class EF
  priority level 1
  queue-limit 547500 bytes
 class AF4
  priority level 2
  queue-limit 4037500 bytes
 class AF1
  bandwidth remaining ratio 90
  queue-limit 750000 bytes
  random-detect dscp-based
  random-detect dscp 8 750000 bytes 750000 bytes
  random-detect dscp 10 750000 bytes 750000 bytes
  random-detect dscp 12 600000 bytes 675000 bytes
 class data service-fragment Port-channel_BE
  shape average 250000000
  bandwidth remaining ratio 1
!
```

In this example, the policy map Port-channel_1_subscriber is configured with a fragment class named Port-channel_BE. (For simplicity, only a single subinterface policy is shown.) This enable queueing and policies aggregation for the subscriber traffic classes at the main-interface policy map.

The Port-channel_1_subscriber policy map identifies the DSCP-based traffic classes of EF, AF4, and AF1 and enables collection of drop statistics for those classes.

```
policy-map Port-channel_1_subscriber
 class EF
  account
  set cos 5
  set dscp ef
 class AF4
  account
  set cos 4
 class AF1
  account
  set cos 1
 class class-default fragment Port-channel_BE
  bandwidth remaining ratio 1
  queue-limit 31250 bytes
!
port-channel load-balancing vlan-manual
!
interface Port-channel1
 no ip address
 no negotiation auto
!
```

The service policies are applied first to the physical interface, and then to the subinterfaces as shown:

```
interface GigabitEthernet1/2/0
 no ip address
```

```
 negotiation auto
 no cdp enable
 service-policy output Port-channel_1_main_policy
 channel-group 1
!
interface GigabitEthernet2/2/0
 no ip address
 negotiation auto
 service-policy output Port-channel_1_main_policy
 channel-group 1
!
interface Port-channel1.1
 encapsulation dot1Q 2 primary GigabitEthernet1/2/0 secondary GigabitEthernet2/2/0
 ip address 10.0.0.2 255.255.255.0
 service-policy output Port-channel_1_subscriber
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Quality of Service Solutions Command Reference* |
| Modular Quality of Service Command-Line Interface | "Applying QoS Features Using the MQC" module |
| Distribution of Remaining Bandwidth Using Ratio | "Distribution of Remaining Bandwidth Using Ratio" module |
| Class-Based Shaping | "Regulating Packet Flow--Using Class-Based Traffic Shaping" module |

### Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported, and support for existing standards has not been modified by this feature. | |

### MIBs

| MIB | MIBs Link |
|---|---|
| CISCO-CLASS-BASED-QOS-MIB | To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for QoS: Policies Aggregation

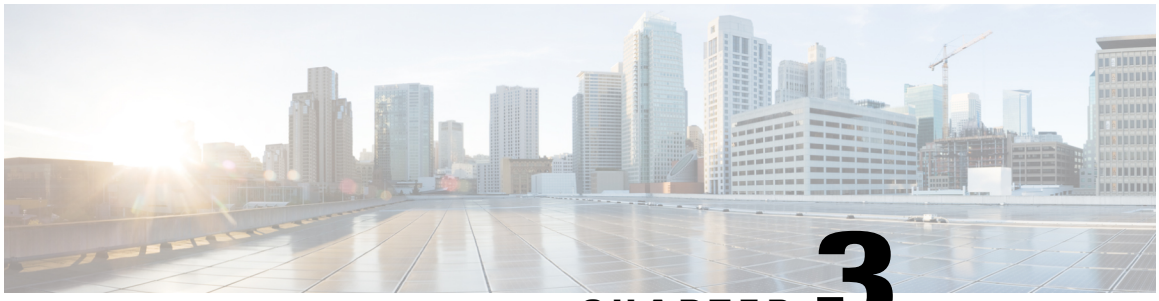
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 4: Feature Information for QoS: Policies Aggregation***

| Feature Name | Releases | Feature Information |
|---|---|---|
| QoS: Policies Aggregation | Cisco IOS XE Release 2.1 | This feature was introduced on Cisco ASR 1000 Series Routers.<br>The following command was modified: **class (policy-map)**. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface | Cisco IOS XE Release 2.6 | This feature was enhanced to support queueing aggregation at the primary interface for other traffic classes, including DSCP-based classes such as EF, AF1, and AF4 traffic classes. With this enhancement, other traffic classes from different subinterfaces share a common queue for that traffic class. Other enhancements include the ability to configure and show per-subscriber drop statistics on the aggregate queues and byte-based queue limits and WRED thresholds.<br>In Cisco IOS XE Release 2.6, support for the CISCO-CLASS-BASED-QOS-MIB was added.<br>The following commands are new or modified: **account**, **show policy-map interface**. |



CHAPTER 3

# Legacy QoS Command Deprecation

The functionality provided by these hidden commands has been replaced by similar functionality provided via the modular QoS CLI (MQC). The MQC is a set of a platform-independent commands for configuring QoS on Cisco platforms. This means that you must now provision QoS by defining traffic classes, creating traffic policies containing those classes, and attaching those policies to the desired interfaces. This document lists the hidden commands and their replacement MQC commands.



# Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Information About Legacy QoS Command Deprecation

## QoS Features Applied Using the MQC

The MQC structure lets you define a traffic class (also called a class map), create a traffic policy (also called a policy map), and attach the traffic policy to an interface. This comprises the following three high-level steps.

1 Define a traffic class by using the **class-map** command. A traffic class is used to classify traffic.

2 Create a traffic policy by using the **policy-map** command. A traffic policy contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.

3 Attach the traffic policy to the interface by using the **service-policy** command.

Steps 1 and 3 do not involve legacy QoS hidden commands, which means that they are not within the scope of this document. For more information about these two steps, see the " Applying QoS Features Using the MQC " module in the *Quality of Service Solutions Configuration Guide* .

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Defining traffic classes; attaching traffic policies to interfaces | " Applying QoS Features Using the MQC " module in the *Quality of Service Solutions Configuration Guide* |
| Reference pages for QoS commands | *Cisco IOS Quality of Service Solutions Command Reference* |
| Reference pages for wide-area networking commands | *Cisco IOS Wide-Area Networking Command Reference* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Legacy QoS Command Deprecation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

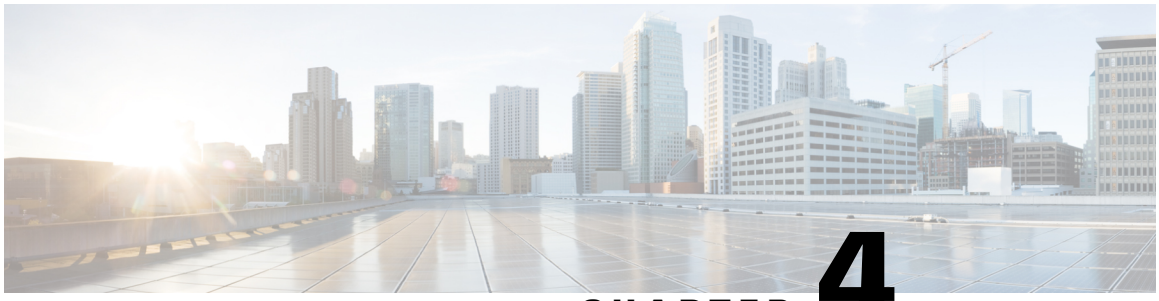
*Table 5: Feature Information for Legacy QoS Command Deprecation*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Legacy QoS Command Deprecation: Hidden Commands | 15.0(1)S 15.1(3)T | |

| Feature Name | Releases | Feature Information |
|---|---|---|
| | | To streamline Cisco IOS QoS, certain commands have been hidden, which means that if you try to view a hidden command by entering a question mark (**?**) at the command line, the command does not appear. However, if you know the command syntax, you can enter it. These commands will be removed in a future release.<br><br>The functionality provided by these hidden commands is replaced by similar functionality from the modular QoS CLI (MQC), which is a set of a platform-independent commands for configuring QoS.<br><br>The following commands were modified: **custom-queue-list**, **fair-queue (WFQ)**, **frame-relay adaptive-shaping** (**becn** keyword), **frame-relay adaptive-shaping** (**foresight** keyword), **frame-relay bc**, **frame-relay be**, **frame-relay cir**, **frame-relay congestion threshold de**, **frame-relay congestion threshold ecn**, **frame-relay custom-queue-list**, **frame-relay fair-queue**, **frame-relay fecn-adapt**, **frame-relay ip rtp priority**, **frame-relay priority-group**, **frame-relay qos-autosense**, **ip rtp priority**, **max-reserved-bandwidth**, **priority-group**, **random-detect**, **random-detect dscp**, **random-detect**(**dscp-based** keyword), **random-detect exponential-weighting-constant**, **random-detect flow**, **random-detect flow average-depth-factor**, **random-detect flow count**, **random-detect**(**prec-based** keyword), **random-detect precedence**, **random-detect-group**, **show interfaces fair-queue**, **show** |

| Feature Name | Releases | Feature Information |
|---|---|---|
| | | **interfaces random-detect**, **show queue**, **show queueing**, **show random-detect-group**, **show traffic-shape**, **show traffic-shape queue**, **show traffic-shape statistics**. |
| Legacy QoS Command Deprecation: Hidden Commands | Cisco IOS XE Release 2.6 | To streamline Cisco IOS XE QoS, certain commands have been hidden, which means that if you try to view a hidden command by entering a question mark (**?**) at the command line, the command does not appear. However, if you know the command syntax, you can enter it. These commands will be removed in a future release.<br>The functionality provided by these hidden commands is replaced by similar functionality from the modular QoS CLI (MQC), which is a set of a platform-independent commands for configuring QoS.<br>The following commands were modified: **custom-queue-list**, **fair-queue (WFQ)**, **frame-relay adaptive-shaping** (**becn** keyword), **frame-relay adaptive-shaping** (**foresight** keyword), **frame-relay bc**, **frame-relay be**, **frame-relay cir**, **frame-relay congestion threshold de**, **frame-relay congestion threshold ecn**, **frame-relay custom-queue-list**, **frame-relay fair-queue**, **frame-relay fecn-adapt**, **frame-relay ip rtp priority**, **frame-relay priority-group**, **frame-relay qos-autosense**, **ip rtp priority**, **max-reserved-bandwidth**, **show interfaces fair-queue**, **show interfaces random-detect**, **show queue**, **show queueing**, **show traffic-shape**, **show traffic-shape queue**, **show traffic-shape statistics**. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| Legacy QoS Command Deprecation: Removed Commands | Cisco IOS XE Release 3.2S | The legacy QoS commands were removed. This means that you must use the appropriate replacement MQC commands.<br><br>The following commands were removed: **custom-queue-list**, **fair-queue (WFQ)**, **frame-relay adaptive-shaping** (**becn** keyword), **frame-relay adaptive-shaping** (**foresight** keyword), **frame-relay bc**, **frame-relay be**, **frame-relay cir**, **frame-relay congestion threshold de**, **frame-relay congestion threshold ecn**, **frame-relay custom-queue-list**, **frame-relay fair-queue**, **frame-relay fecn-adapt**, **frame-relay ip rtp priority**, **frame-relay priority-group**, **frame-relay qos-autosense**, **ip rtp priority**, **max-reserved-bandwidth**, **show interfaces fair-queue**, **show interfaces random-detect**, **show queue**, **show queueing**, **show traffic-shape**, **show traffic-shape queue**, **show traffic-shape statistics**. |



CHAPTER 4

# QoS Packet Marking Statistics

The QoS: Packet Marking Statistics feature allows you to display the number of packets that have:

- Modified headers
- Been classified into a category for local router processing



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Prerequisites for QoS Packet Marking Statistics

You cannot enable or disable the QoS: Packet Marking Statistics feature if a policy map is associated with any interface on the system.

# Restrictions for QoS Packet Marking Statistics

Enabling the Qos: Packet Marking Statistics feature may increase CPU utilization on a scaled configuration. Before enabling the Qos: Packet Marking Statistics feature, weigh the benefits of the statistics information against the increased CPU utilization for your system.

# Information About QoS Packet Marking Statistics

## QoS Packet Marking Statistics Feature Overview

The QoS: Packet Marking Statistics feature allows you to display the number of packets that have:

- Modified headers
- Been classified into a category for local router processing

Use the QoS: Packet Marking Statistics feature to display traffic types. Using this information you can do the following:

- Compare the amount of voice traffic to data traffic on a segment of your network
- Adjust bandwidth availability
- Accurately determine billing
- Troubleshoot service problems

The system collects packet marking statistics on a 10-second cycle. If there are many interfaces or sessions then the system collects statistics for about 8000 of them during each cycle. In a scaled configuration several 10-second cycles may be required to gather all statistics.

# How to Use QoS Packet Marking Statistics

## Configuring QoS Packet Marking Statistics

### Before You Begin

Before enabling the QoS: Packet Marking Statistics feature, ensure no policy maps are associated with interfaces on the system. If there are, the system returns the following message:

```
Either a) A system RELOAD or
       b) Remove all service-policies, re-apply the change
            to the statistics, re-apply all service-policies
   is required before this command will be activated.
```

**Note** Enabling the Qos: Packet Marking Statistics feature may increase CPU utilization on a scaled configuration. Before enabling the Qos: Packet Marking Statistics feature, weigh the benefits of the statistics information against the increased CPU utilization for your system.

>

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform qos marker-statistics**
4. **end**
5. Do one of the following:
   - **show policy-map interface** *interface-name*
   - [**vc** [*vpi* /] *vci*] [**dlci** *dlci*] [**input** | **output**]
6. **configure terminal**
7. **no platform qos marker-statistics**
8. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **platform qos marker-statistics**<br>**Example:**<br>`Router(config)# platform qos marker-statistics` | Enables the QoS: Packet Marking Statistics feature. |
| **Step 4** | **end**<br>**Example:**<br>`Router# end` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | Do one of the following:<br>• **show policy-map interface** *interface-name*<br>• [**vc** [*vpi* /] *vci*] [**dlci** *dlci*] [**input** \| **output**]<br>**Example:**<br>`Router# show policy-map interface serial4/0/0`<br>**Example:**<br>**Example:**<br>**show policy-map session**<br>**Example:**<br>`Router#`<br>`show policy-map interface` | Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific Permanent Virtual Circuit (PVC) on the interface.<br>or<br>Displays the quality of service (QoS) policy map in effect for a Point-to-Point Protocol over Ethernet (PPPoE) session. |
| **Step 6** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 7** | **no platform qos marker-statistics**<br>**Example:**<br>`Router(config)# no platform qos marker-statistics` | Disables the QoS: Packet Marking Statistics feature. |
| **Step 8** | **end**<br>**Example:**<br>`Router# end` | Exits configuration mode. |

### Example

Use the **show policy-map interface** command to display the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface.

```
Router#
show policy-map interface

ATM1/0/0.1: VC 1/110 -
  Service-policy output: m_asr1000_atm_out
    Class-map: m_asr1000_atm_out (match-all)
      6644555 packets, 784057490 bytes
      5 minute offered rate 9024000 bps, drop rate 0000 bps
      Match:  precedence 4
      Match: qos-group 4
      QoS Set
        atm-clp

 Packets marked 6649123 <-------- The interface transmitted 6644555 packets matching class-map
 m_asr1000_atm_out. Of these packets, 6649123 had the ATM CLP bit marked. These two numbers
 are often the same, but a time difference in when the statistics were gathered may cause
the numbers to be different.
        precedence 3
          Packets marked 6649123 <--------
The interface transmitted 6644555 packets
matching
 class-map m_asr1000_atm_out. Of these packets, 6649123 had the IP Precedence field set to
 3.
These two numbers are often the same, but a time difference in when the statistics were
gathered may cause the numbers to be different.
 Class-map: class-default (match-any)

  10 packets, 1080 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
POS2/0/1.1
  Service-policy input: m_asr1000_policy

 Class-map: m_asr1000_class (match-all)
      6644560 packets, 757479840 bytes
      5 minute offered rate 8720000 bps, drop rate 0000 bps
      Match:  precedence 5
      QoS Set
        precedence 4
          Packets marked 6644560
<-------- The interface received 6644560 packets matching class-map m_asr1000_class. Of
these packets, 6644560 had the IP Precedence set to 4.
 mpls experimental imposition 4
          Packets marked 6644560
<-------- The interface received 6644560 packets matching class-map m_asr1000_class. Of
these packets, 6644560 had the MPLS Experimental Imposition set to 4.
 qos-group 4
          Packets marked 6644560
<-------- The interface received 6644560 packets matching class-map m_asr1000_class. Of
these packets, 6644560 had the QoS-group set to 4.
    Class-map: class-default (match-any)
      18 packets, 1612 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
Virtual-Template2
  Service-policy input: m_pppoe_policy
    Service policy content is displayed for cloned interfaces only such as vaccess and
sessions
Router# show policy-map interface Virtual-Access 2.1

Virtual-Access2.1
 SSS session identifier 10 -
  Service-policy input: m_pppoe_policy
    Class-map: m_pppoe_class (match-all)
      4563 packets, 538434 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
```

```
      Match:  precedence 5
      QoS Set
        precedence 6
         Packets marked 4563 <-------- The virtual interface received 4563 packets matching
 class-map m_pppoe_class. Of these packets, 4563 had the IP Precedence set to 6.
 Class-map: class-default (match-any)
      4 packets, 152 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

Use the **show policy-map session** command to display the QoS policy map in effect for a PPPoE session.

```
Router# show policy-map session uid 10

 SSS session identifier 10 -
  Service-policy input: m_pppoe_policy
    Class-map: m_pppoe_class (match-all)
      4563 packets, 538434 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match:  precedence 5
      QoS Set
        precedence 6
         Packets marked 4563 <-------- The virtual interface received 4563 packets matching
 class-map m_pppoe_class. Of these packets, 4563 had the IP Precedence set to 6.
 Class-map: class-default (match-any)
 53 packets, 2014 bytes
      30 second offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

### Troubleshooting Tips

To confirm that the QoS: Packet Marking Statistics feature is enabled, use the **show platform hardware qfp active feature qos config global** command.

# Configuration Examples for QoS Packet Marking Statistics

## Example Configuring a Policy on an Ingress Interface

This example shows how to do the following:

- Enable the QoS: Packet Marking Statistics feature
- Configure an input service policy on an ingress interface
- Classify traffic to a configured class
- Configure marking in the class to set the IP precedence to 1
- Display the **show policy-map interface** command output

```
Router# platform qos marker-statistics

class-map test_class
    match access-group 101
  policy-map test_policy
    class test_class
       set ip precedence 1
Interface POS2/0/1
  service-policy input test_policy
Router#
```

```
show policy-map interface
POS2/0/1
  Service-policy input: test_policy
    Class-map: test_class (match-all)
      6644560 packets, 757479840 bytes
      5 minute offered rate 8720000 bps, drop rate 0000 bps
      Match:  precedence 5
      QoS Set
        precedence 1
          Packets marked 6644560
    Class-map: class-default (match-any)
      18 packets, 1612 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Quality of service commands | *Cisco IOS Quality of Service Command Reference* |

### Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported, and support for existing standards has not been modified. | -- |

### MIBs

| MIB | MIBs Link |
|---|---|
| CISCO-CLASS-BASED-QOS-MIB | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | -- |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

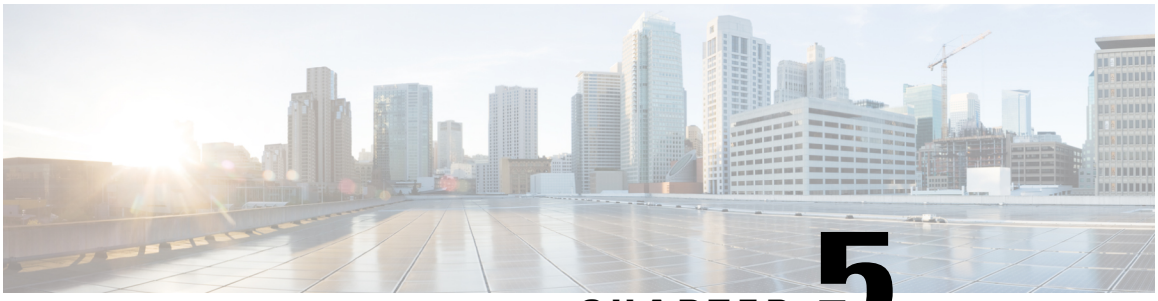
# Feature Information for QoS Packet Marking Statistics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 6: Feature Information for QoS: Packet Marking Statistics***

| Feature Name | Releases | Feature Information |
|---|---|---|
| QoS: Packet Marking Statistics | Cisco IOS XE Release 3.3S | The QoS: Packet Marking Statistics feature allows you to display the number of packets that have:<br>• Modified headers<br>• Been classified into a category for local router processing<br><br>The following commands were introduced or modified: **platform qos marker-statistics**, **no platform qos marker-statistics,**and **show platform hardware qfp active feature qos config global**. |



CHAPTER 5

# QoS Packet Matching Statistics

The QoS: Packet Matching Statistics feature allows you to define a quality of service (QoS) packet filter, then display the number of packets and bytes matching that filter.



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Prerequisites for QoS Packet Matching Statistics

You cannot enable or disable the QoS: Packet Matching Statistics feature if a policy map is associated with any interface on the system.

# Restrictions for QoS Packet Matching Statistics

Enabling the Qos: Packet Matching Statistics feature may increase CPU utilization on a scaled configuration. Before enabling the Qos: Packet Matching Statistics feature, weigh the benefits of the statistics information against the increased CPU utilization for your system.

# Information About QoS Packet Matching Statistics

## QoS Packet Matching Statistics Feature Overview

The QoS: Packet Matching Statistics feature allows you to define a QoS packet filter, then display the number of packets and bytes matching that filter.

To define a filter, use the **class-map** command with the **match-any** keyword, for example:

```
class-map match-any my_class
   match ip precedence 4 <-------- User-defined filter
   match qos-group 10 <-------- User-defined filter
```

Using this information you can do the following:

- Compare the amount of voice traffic to data traffic on a segment of your network
- Adjust bandwidth availability
- Accurately determine billing
- Troubleshoot service problems

The system collects packet matching statistics on a 10-second cycle. If there are many interfaces or sessions then the system collects statistics for about 8000 of them during each cycle. In a scaled configuration several 10-second cycles may be required to gather all statistics.

# How to Use QoS Packet Matching Statistics

## Configuring QoS Packet Matching Statistics

### Before You Begin

Before enabling the QoS: Packet Matching Statistics feature, ensure no policy maps are associated with interfaces on the system. If there are, the system returns the following message:

```
Either a) A system RELOAD or
       b) Remove all service-policies, re-apply the change
            to the statistics, re-apply all service-policies
   is required before this command will be activated.
```

Before enabling the QoS: Packet Matching Statistics feature, ensure you have defined a filter is using **class-map** command with the **match-any** keyword.

**Note** Enabling the Qos: Packet Matching Statistics feature may increase CPU utilization on a scaled configuration. Before enabling the Qos: Packet Matching Statistics feature, weigh the benefits of the statistics information against the increased CPU utilization for your system.

>

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **platform qos match-statistics per-filter**
4. **end**
5. **show policy-map interface** *interface-name*
6. **configure terminal**
7. no platform qos match-statistics per-filter
8. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **platform qos match-statistics per-filter**<br><br>**Example:**<br>`Router(config)# platform qos match-statistics per-filter` | Enables the QoS: Packet Matching Statistics feature. |
| **Step 4** | **end**<br><br>**Example:**<br>`Router# end` | Exits configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **show policy-map interface** *interface-name*<br><br>**Example:**<br><br>[**vc** [*vpi* /] *vci*] [**dlci** *dlci*] [**input** \| **output**]<br><br>**Example:**<br><br>`Router# show policy-map interface serial4/0/0` | Displays the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific permanent virtual circuit (PVC) on the interface. |
| **Step 6** | **configure terminal**<br><br>**Example:**<br><br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 7** | no platform qos match-statistics per-filter<br><br>**Example:**<br><br>`Router(config)# no platform qos match-statistics per-filter` | Disables the QoS: Packet Matching Statistics feature. |
| **Step 8** | **end**<br><br>**Example:**<br><br>`Router# end` | Exits configuration mode. |

### Example

Use the **show policy-map interface** command to display the packet statistics of all classes that are configured for all service policies either on the specified interface or subinterface or on a specific PVC on the interface:

```
Router# show policy-map interface gig1/1/0

 GigabitEthernet1/1/0
  Service-policy input: pol1     ! target = gig1/1/0,input
    Class-map: class1 (match-any)
      1000 packets, 40000 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 1 <-------- User-defined filter
        800 packets, 32000 bytes <-------- Filter matching results
      Match: ip precedence 2 <-------- User-defined filter
        200 packets, 8000 bytes <-------- Filter matching results
      QoS Set
        ip precedence 7
          No packet marking statistics available
    Class-map: class-default (match-any)
      500 packets, 20000 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any <-------- User-defined filter
        500 packets, 20000 bytes <-------- Filter matching results
```

### Troubleshooting Tips

To confirm that the QoS: Packet Matching Statistics feature is enabled, use the **show platform hardware qfp active feature qos config global** command. If the feature is disabled, you should see a message similar to the following:

Router# **show platform hardware qfp active feature qos config global**

```
Marker statistics are: enabled
Match per filter statistics are: enabled
```

# Configuration Examples for QoS Packet Matching Statistics

## Example Configuring a QoS Packet Matching Filter

This example shows how to do the following:

- Define a QoS packet matching filter
- Display the **show policy-map interface** command output

```
Router# show policy-map interface Tunnel1

  Service-policy output: DATA-OUT-PARENT
    Class-map: class-default (match-any)
      4469 packets, 4495814 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any <-------- User-defined filter
      Queueing
      queue limit 416 packets
      (queue depth/total drops/no-buffer drops) 0/0/0
      (pkts output/bytes output) 4469/4558380
      shape (average) cir 100000000, bc 400000, be 400000
      target shape rate 100000000
      Service-policy : DATA-OUT
        queue stats for all priority classes:
          Queueing
          queue limit 200 packets
          (queue depth/total drops/no-buffer drops) 0/0/0
          (pkts output/bytes output) 4469/4558380
        Class-map: ATM-VTI-RIP-SPK1-DATA (match-any)
          4469 packets, 4495814 bytes <-------- Filter matching results
          5 minute offered rate 0000 bps, drop rate 0000 bps
          Match: access-group 121 <-------- User-defined filter
            4469 packets, 4495814 bytes <-------- Filter matching results
            5 minute rate 0 bps
          QoS Set
            ip precedence 3
              Packets marked 4469
          Priority: 100 kbps, burst bytes 2500, b/w exceed drops: 0
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Quality of service commands | *Cisco IOS Quality of Service Command Reference* |

### Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported, and support for existing standards has not been modified. | -- |

### MIBs

| MIB | MIBs Link |
|---|---|
| CISCO-CLASS-BASED-QOS-MIB | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | -- |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

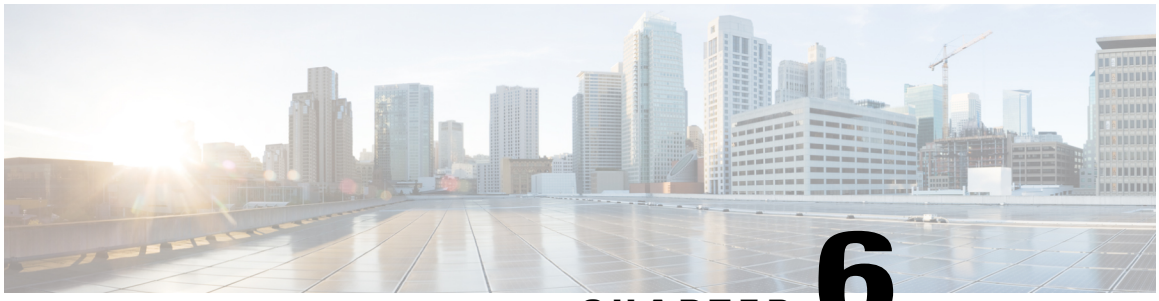
# Feature Information for QoS Packet Matching Statistics

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 7: Feature Information for QoS: Packet Matching Statistics***

| Feature Name | Releases | Feature Information |
|---|---|---|
| QoS: Packet Matching Statistics | Cisco IOS XE Release 3.3S | The QoS: Packet Matching Statistics feature allows you to display the number of packets that have:<br>• Modified headers<br>• Been classified into a category for local router processing<br><br>The following commands were introduced or modified: **platform qos match-statistics per-filter**, **no platform qos match-statistics per-filter,** and **show platform hardware qfp active feature qos config global**. |



CHAPTER 6

# Set ATM CLP Bit Using Policer

The Set ATM CLP Bit Using Policer feature allows you to police and then mark outbound PPP over ATM (PPPoA) traffic. You can set the ATM cell loss priority (CLP) bit using either of the following methods:

- A policed threshold
- Matching a class



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Prerequisites for Set ATM CLP Bit Using Policer

If you are setting the ATM CLP bit by a policed threshold, ensure that a policy map includes the **set-clp-transmit**action. The new policer action conditionally marks PPPoA traffic in the matched class for a higher drop probability in the ATM network when traffic exceeds a given rate.

If you are setting the ATM CLP bit strictly by matching a class, ensure that a policy map includes the **set atm-clp** action. The set directive marks all traffic in the matched class for higher drop probability in the ATM network.

You can attach policy maps with the **set-clp-transmit**or **set atm-clp**actions to a virtual template. This template is cloned when PPPoA sessions are created or by dynamic assignment.

# Information About Set ATM CLP Bit Using Policer

## ATM CLP Bit

The ATM CLP bit shows the drop priority of the ATM cell. During ATM network congestion, the router discards ATM cells with the CLP bit set to 1 before discarding cells with a CLP bit setting of 0.

Using the Set ATM CLP Bit Using Policer feature, you can configure the **police** command to enable the ATM CLP bit in cell headers. The ATM CLP bit can be explicitly marked by a set directive.

The Set ATM CLP Bit Using Policer feature supports the **set-clp-transmit** policing action in the following types of policies:

- Single-rate policing
- Dual-rate policing
- Hierarchical

# How to Set the ATM CLP Bit Using Policer

## Configuring PPPoA Broadband Traffic Policing

### Before You Begin

Before configuring the policy map, ensure that you have defined any class maps used to classify traffic.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name*| **class-default**]
5. **police** [**cir** *cir*] [**conform-action** *action*] [**exceed-action** *action*]
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Router> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br><br>`Router(config)# policy-map parent-policy` | Enters policy-map configuration mode and creates a policy map. |
| **Step 4** | **class** {*class-name*\| **class-default**]<br><br>**Example:**<br><br>`Router(config-pmap)# class class-default` | Enters policy-map class configuration mode.<br><br>Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy. Repeat this command as many times as necessary to specify the child or parent classes that you are creating or modifying:<br><br>• *class name* --Name of the class to be configured or whose policy is to be modified. The class name is used for both the class map and to configure a policy for the class in the policy map.<br><br>• **class-default** --Specifies the default class so that you can configure or modify its policy. |
| **Step 5** | **police** [**cir** *cir*] [**conform-action** *action*] [**exceed-action** *action*]<br><br>**Example:**<br><br>`Router(config-pmap-c)# police 1000000`<br><br>**Example:**<br><br>`Router(config-pmap-c-police)# conform-action`<br><br>**Example:**<br><br>`transmit` | Configures traffic policing and specifies multiple actions applied to packets marked as conforming to, exceeding, or violating a specific rate.<br><br>• Enters policy-map class police configuration mode. Use one line per action that you want to specify:<br><br>• **cir**--(Optional) Committed information rate. Indicates that the CIR will be used for policing traffic.<br><br>• **conform-action**--(Optional) Action to take on packets when the rate is less than the conform burst.<br><br>• **exceed-action**--(Optional) Action to take on packets whose rate is within the conform and conform plus exceed burst. |

| | Command or Action | Purpose |
|---|---|---|
| | **Example:**<br>`Router(config-pmap-c-police)# exceed-action`<br>**Example:**<br>`set-clp-transmit` | |
| **Step 6** | **end**<br>**Example:**<br>`Router(config-pmap-c)# end` | (Optional) Returns to privileged EXEC mode. |

### Example

The following example shows you how to set the ATM CLP using a policer:

```
policy-map egress_atm_clp_policer
 class prec0
  police cir 5000000
 class prec1
  police cir 3000000 conform-action transmit exceed-action set-clp-transmit
 class class-default
  police cir 1000000 conform-action transmit exceed-action set-clp-transmit
```

# Marking the ATM CLP Bit

### Before You Begin

Before configuring the policy map, ensure that you have defined any class maps used to classify traffic.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map policy-map-name**
4. **class** {*class-name*| **class-default**]
5. **set atm-clp**
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Router> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map policy-map-name**<br><br>**Example:**<br><br>`Router(config)# policy-map parent-policy` | Enters policy-map configuration mode and creates a policy map. |
| **Step 4** | **class** {*class-name*\| **class-default**]<br><br>**Example:**<br><br>`Router(config-pmap)# class class-default` | Enters policy-map class configuration mode.<br><br>Specifies the name of the class whose policy you want to create or change or specifies the default class (commonly known as the class-default class) before you configure its policy. Repeat this command as many times as necessary to specify the child or parent classes that you are creating or modifying:<br><br>• *class name* --Name of the class to be configured or whose policy is to be modified. The class name is used for both the class map and to configure a policy for the class in the policy map.<br><br>• **class-default** --Specifies the default class so that you can configure or modify its policy. |
| **Step 5** | **set atm-clp**<br><br>**Example:**<br><br>`Router(config-pmap-c)# set atm-clp` | Configures marking of the ATM CLP bit for all traffic matching this class. |
| **Step 6** | **end**<br><br>**Example:**<br><br>`Router(config-pmap-c)# end` | (Optional) Returns to privileged EXEC mode. |

### Example

The following example shows you how to set the ATM CLP using explicit marking:

```
policy-map egress_atm_clp_policer
 class prec0
  police cir 5000000
 class class-default
  set atm-clp
```

# Configuration Examples for Set ATM CLP Bit Using Policer

## Example Marking the ATM CLP by Policer Action Matching a Class

This example shows how to do the following:

- Define traffic classes.
- Configure a two-layer policy map.
- Apply the policy map to PPPoA sessions.

This policy conditionally marks the ATM CLP bit on the traffic in the matching low_interest class once traffic on the class exceeds a given rate.

```
class-map voice
  match precedence 4
!
class-map web
  match precedence 3
!
class low_interest
  match precedence 1 0
!
policy-map child
 child class voice
  police cir 256000
  priority level 1
 class web
  bandwidth remaining ratio 10
 class low_interest
  police cir 1000000 conform-action transmit exceed-action set-clp-transmit
 class class-default
  bandwidth remaining ratio 1
!
policy-map parent
 class class-default
  shape average 15000000
  service-policy child
```

policy maps attached to virtual templates are cloned and used to create a virtual access interface for each PPPoA session:

```
interface Virtual-Template1
 ip unnumbered Loopback1
 load-interval 30
 peer default ip address pool POOL1
 ppp authentication chap ppp
 ipcp address required
 service-policy output parent
```

# Example Marking the ATM CLP by Policer Action Policed Threshold

This example shows how to do the following:

- Define traffic classes.
- Configure a two-layer policy map.
- Apply the policy map to PPPoA sessions.

This policy marks all non-essential traffic with the ATM CLP bit so that it is eligible for dropping if the ATM network becomes congested.

```
class-map video
  match precedence 5
!
class-map voice
  match precedence 4
!
class-map web
  match precedence 3
!
policy-map child
 child class voice
  police cir 256000
  priority level 1
 class video
  police cir 4000000
  priority level 2
 class web
  set atm-clp
  bandwidth remaining ratio 10
 class class-default
  bandwidth remaining ratio 1
  set atm-clp
!
interface Virtual-Template1
 ip unnumbered Loopback1
 load-interval 30
 peer default ip address pool POOL1
 ppp authentication chap ppp
 ipcp address required
 service-policy output parent
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |
| Quality of Service commands | *Cisco IOS Quality of Service Command Reference* |

### Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature. | -- |

### MIBs

| MIB | MIBs Link |
|---|---|
| No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature. | To locate and download MIBs for selected platforms, Cisco software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature. | -- |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

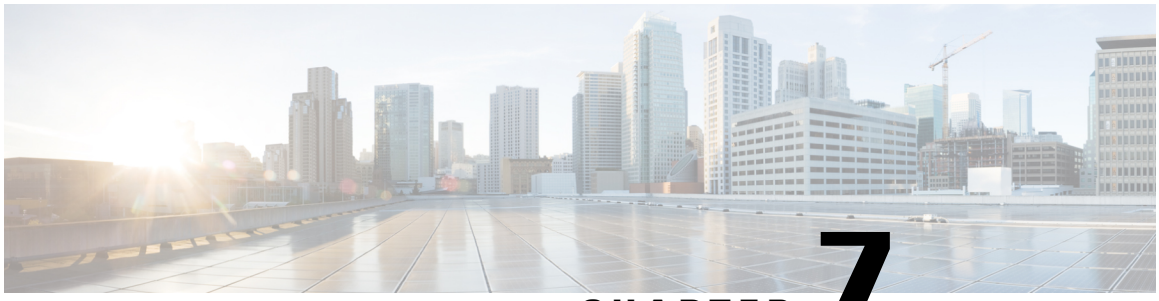
# Feature Information for Set ATM CLP Bit Using Policer

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 8: Feature Information for Set ATM CLP Bit Using Policer***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Set ATM CLP Bit Using Policer | Cisco IOS XE Release 3.3S | The Set ATM CLP Bit Using Policer feature allows you to police and then mark outbound PPPoA traffic.<br>The following commands were introduced or modified: **set atm-clp**and **police**. |



CHAPTER 7

# EVC Quality of Service

This document contains information about how to enable quality of service (QoS) features (such as traffic classification and traffic policing) for use on an Ethernet virtual circuit (EVC).

An EVC as defined by the Metro Ethernet Forum is a port-level point-to-point or multipoint-to-multipoint circuit. It is an end-to-end representation of a single instance of a service being offered by a provider to a customer. It embodies the different parameters on which the service is being offered.



# Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Information About Quality of Service on an EVC

## EVC Quality of Service and the MQC

QoS functionality is typically applied using traffic classes, class maps, and policy maps. For example, you can specify that traffic belonging to a particular class be grouped into specific categories, and receive a specific

QoS treatment (such as classification or policing). The QoS treatment the traffic is to receive is specified in a policy map and the policy map is attached to an interface. The mechanism used for applying QoS in this manner is the modular QoS CLI (MQC.)

The policy map can be attached to an interface in either the incoming (ingress) or outgoing (egress) direction with the **service-policy** command.

The MQC structure allows you to define a traffic class, create a traffic policy, and attach the traffic policy to an interface (in this case, an EVC).

The MQC structure consists of the following three high-level steps.

1. Define a traffic class by using the **class-map**command. A traffic class is used to classify traffic.
2. Create a traffic policy by using the **policy-map** command. (The terms *traffic policy* and *policy map* are often synonymous.) A traffic policy (policy map) contains a traffic class and one or more QoS features that will be applied to the traffic class. The QoS features in the traffic policy determine how to treat the classified traffic.
3. Attach the traffic policy (policy map) to the interface by using the **service-policy** command.

**Note** For more information about the MQC, including information about hierarchical policy maps and class maps, see the "Applying QoS Features Using the MQC" module.

# QoS-Aware Ethernet Flow Point (EFP)

As described in the EVC Quality of Service and the MQC, on page 83, the MQC is used to apply one or more QoS features to network traffic. The last step in using the MQC is to attach the traffic policy (policy map) to an interface (in this case, an EVC) by using the **service-policy** command.

With the EVC Quality of Service feature, the **service-policy** command can be used to attach the policy map to an Ethernet Flow Point (EFP) in either the incoming (ingress) *or* outgoing (egress) direction of an EVC. This way, the EFP is considered to be "QoS-aware."

# QoS Functionality and EVCs

The specific QoS functionality available on an EVC varies by Cisco IOS XE release but can include the following:

- Packet classification (for example, based on differentiated services code point (DSCP) value and QoS group identifier)
- Packet marking (for example, based on Class of Service (CoS) value)
- Traffic policing (two- and three-color and multiple actions)
- Bandwidth sharing
- Priority queueing (in the outbound direction on the EVC only)
- Weighted Random Early Detection (WRED)

The QoS functionality is enabled by using the appropriate commands listed in the following sections.

## match Commands Supported by EVC QoS for Classifying Traffic

The table below lists *some* of the available **match** commands that can be used when classifying traffic on an EVC. The available **match** commands vary by Cisco IOS XE release. For more information about the commands and command syntax, see the Cisco IOS Quality of Service Solutions Command Reference.

***Table 9: match Commands That Can Be Used with the MQC***

| Command | Purpose |
|---|---|
| **match access-group** | Configures the match criteria for a class map on the basis of the specified access control list (ACL). |
| **match any** | Configures the match criteria for all packets. |
| **match cos** | Matches a packet based on a Layer 2 CoS marking. |
| **match cos inner** | Matches the inner CoS of QinQ packets on a Layer 2 CoS marking. |
| **match** [**ip**] **dscp** | Identifies a specific IP DSCP value as a match criterion. Up to eight DSCP values can be included in one match statement. |
| **match not** | Specifies the single match criterion value to use as an unsuccessful match criterion.<br>**Note** The **match not** command, rather than identifying the specific match parameter to use as a match criterion, is used to specify a match criterion that prevents a packet from being classified as a member of the class. For instance, if the **match not qos-group 6**command is issued while you configure the traffic class, QoS group 6 becomes the only QoS group value that is not considered a successful match criterion. All other QoS group values would be successful match criteria. |
| **match** [**ip**] **precedence** | Identifies IP precedence values as match criteria. |
| **match qos-group** | Identifies a specific QoS group value as a match criterion. |
| **match source-address mac** | Uses the source MAC address as a match criterion.<br>**Note** Classifying traffic using the **match source-address mac**command is supported in the input direction only. |

| Command | Purpose |
|---|---|
| **match vlan (QoS)** | Matches and classifies traffic on the basis of the VLAN identification number. |
| match vlan inner | Configures a class map to match the innermost VLAN ID in an 802.1q tagged frame. |

### Multiple match Commands in One Traffic Class

If the traffic class contains more than one **match** command, you need to specify how to evaluate the **match** commands. You specify this by using either the **match-any** or **match-all**keyword of the **class-map** command. Note the following points about the **match-any** and **match-all** keywords:

- If you specify the **match-any**keyword, the traffic being evaluated by the traffic class must match *one* of the specified criteria.
- If you specify the **match-all** keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria.
- If you do not specify either keyword, the traffic being evaluated by the traffic class must match *all* of the specified criteria (that is, the behavior of the **match-all** keyword is used).

## Commands Used to Enable QoS Features on the EVC

The commands used to enable QoS features vary by Cisco IOS XE release. The table below lists *some* of the available commands and the QoS features that they enable. For complete command syntax, see the Cisco IOS Quality of Service Solutions Command Reference.

For more information about a specific QoS feature that you want to enable, see the appropriate module of the Cisco IOS Quality of Service Solutions Configuration Guide.

***Table 10: Commands Used to Enable QoS Features***

| Command | Purpose |
|---|---|
| **bandwidth** | Configures a minimum bandwidth guarantee for a class. |
| **bandwidth remaining** | Configures an excess weight for a class. |
| **drop** | Discards the packets in the specified traffic class. |
| **fair-queue** | Enables the flow-based queueing feature within a traffic class. |
| **police** | Configures traffic policing. Allows specifying of multiple policing actions. |

| Command | Purpose |
|---|---|
| **police (percent)** | Configures traffic policing on the basis of a percentage of bandwidth available on an interface. |
| **police (two rates)** | Configures traffic policing using two rates, the committed information rate (CIR) and the peak information rate (PIR). |
| **priority** | Gives priority to a class of traffic belonging to a policy map. |
| **queue-limit** | Specifies or modifies the maximum number of packets the queue can hold for a class configured in a policy map. |
| **random-detect** | Enables Weighted Random Early Detection (WRED). |
| **random-detect discard-class** | Configures the WRED parameters for a discard-class value for a class in a policy map. |
| **random-detect discard-class-based** | Configures WRED on the basis of the discard class value of a packet. |
| **random-detect exponential-weighting-constant** | Configures the exponential weight factor for the average queue size calculation for the queue reserved for a class. |
| **random-detect precedence** | Configure the WRED parameters for a particular IP Precedence for a class policy in a policy map. |
| **service-policy** | Specifies the name of a traffic policy used as a matching criterion (for nesting traffic policies [hierarchical traffic policies] within one another). |
| **set cos** | Sets the Layer 2 CoS value of an outgoing packet. |
| **set discard-class** | Marks a packet with a discard-class value. |
| **set** [**ip**] **dscp** | Marks a packet by setting the DSCP value in the type of service (ToS) byte. |
| **set mpls experimental** | Designates the value to which the Multiprotocol Label Switching (MPLS) bits are set if the packets match the specified policy map. |
| **set precedence** | Sets the precedence value in the packet header. |
| **set qos-group** | Sets a QoS group identifier (ID) that can be used later to classify packets. |

| Command | Purpose |
|---|---|
| **shape** | Shapes traffic to the indicated bit rate according to the algorithm specified. |

## input and output Keywords of the service-policy Command

As a general rule, the QoS features configured in the traffic policy can be applied to packets entering the interface or to packets leaving the interface. Therefore, when you use the **service-policy** command, you need to specify the direction of the traffic policy by using the **input** or **output** keyword.

For instance, the **service-policy output policy-map1** command would apply the QoS features in the traffic policy to the interface in the output direction. All packets leaving the interface (output) are evaluated according to the criteria specified in the traffic policy named policy-map1.

**Note** For Cisco IOS XE Release 2.1 and later releases, queueing mechanisms are not supported in the input direction. Nonqueueing mechanisms (such as traffic policing and traffic marking) are supported in the input direction. Also, classifying traffic on the basis of the source MAC address (using the **match source-address mac** command) is supported in the input direction only.

# How to Configure a Quality of Service Feature on an EVC

## Creating a Traffic Class for Use on the EVC

To create a traffic class, use the **class-map** command to specify the traffic class name. Then use one or more **match** commands to specify the appropriate match criteria. Packets matching the criteria that you specify are placed in the traffic class.

To create the traffic class for use on the EVC, complete the following steps.

**Note** The **match cos**command shown in Step Creating a Traffic Class for Use on the EVC is an example of a **match** command that you can use. For information about the other available **match** commands, see Creating a Traffic Class for Use on the EVC, on page 88.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **class-map** [**match-all** | **match-any**] *class-name*
4. **match cos** *cos-number*
5. Enter additional **match** commands, if applicable; otherwise, continue with Creating a Traffic Class for Use on the EVC .
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **class-map** [**match-all** \| **match-any**] *class-name*<br><br>**Example:**<br>`Router(config)# class-map match-any class1` | Creates a class map and enters class-map configuration mode.<br>• The class map is used for matching packets to the specified class.<br><br>**Note** The **match-all** keyword specifies that all match criteria must be met. The **match-any** keyword specifies that one of the match criteria must be met. Use these keywords only if you will be specifying more than one **match** command. |
| **Step 4** | **match cos** *cos-number*<br><br>**Example:**<br>`Router(config-cmap)# match cos 2` | Matches a packet on the basis of a Layer 2 CoS number.<br><br>**Note** The **match cos** command is an example of a **match** command you can use. For information about the other **match** commands that are available, see Creating a Traffic Class for Use on the EVC, on page 88. |
| **Step 5** | Enter additional **match** commands, if applicable; otherwise, continue with Creating a Traffic Class for Use on the EVC . | -- |
| **Step 6** | **end**<br><br>**Example:**<br>`Router(config-cmap)# end` | (Optional) Exits class map configuration mode and returns to privileged EXEC mode. |

# Creating a Policy Map for Use on the EVC

To create a traffic policy (or policy map) for use on the EVC, complete the following steps.

**Note** The **police**command shown in Creating a Policy Map for Use on the EVC is an example of one of the commands that you can use in a policy map. For information about other available commands, see Creating a Policy Map for Use on the EVC, on page 90.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** {*class-name*| **class-default**}
5. **police** *bps* [*burst-normal*] [*burst-max*] [**conform-action** *action*] [**exceed-action** *action*] [**violate-action** *action*]
6. Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Creating a Policy Map for Use on the EVC.
7. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br>**Example:**<br>`Router(config)#`<br>`policy-map policy1` | Creates or specifies the name of the traffic policy and enters QoS policy-map configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **class** {*class-name*\| **class-default**}<br><br>**Example:**<br>`Router(config-pmap)#`<br>`class class1` | Specifies the name of a class and enters QoS policy-map class configuration mode.<br><br>**Note** This step associates the traffic class with the traffic policy. |
| **Step 5** | **police** *bps* [*burst-normal*] [*burst-max*] [**conform-action** *action*] [**exceed-action** *action*] [**violate-action** *action*]<br><br>**Example:**<br>`Router(config-pmap-c)# police 3000` | (Optional) Configures traffic policing.<br><br>**Note** The **police**command is an example of a command that you can use in a policy map to enable a QoS feature. For information about the other commands available, see Creating a Policy Map for Use on the EVC, on page 90. |
| **Step 6** | Enter the commands for any additional QoS feature that you want to enable, if applicable; otherwise, continue with Creating a Policy Map for Use on the EVC. | -- |
| **Step 7** | **end**<br><br>**Example:**<br>`Router(config-pmap-c)# end` | (Optional) Exits QoS policy-map class configuration mode and returns to privileged EXEC mode. |

# Configuring the EVC and Attaching a Traffic Policy to the EVC

The traffic policy (policy map) applies the enabled QoS feature to the traffic class once you attach the policy map to the EVC.

To configure the EVC and attach a traffic policy to the EVC, complete the following steps.

**Note** One of the commands used to attach the traffic policy to the EVC is the **service-policy** command. When you use this command, you must specify either the **input** or **output** keyword along with the policy map name. The policy map contains the QoS feature you want to use. Certain QoS features can only be used in either the input or output direction. For more information about these keywords and the QoS features supported, see the input and output Keywords of the service-policy Command, on page 7. Also, if you attach a traffic policy to an interface containing multiple EVCs, the traffic policy will be attached to *all* of the EVCs on the interface.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface-type interface-number*
4. **service instance** *id* **ethernet** [*evc-name*]
5. **encapsulation dot1q** *vlan-id* [,*vlan-id*[-*vlan-id*]] [**native**]
6. **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric**
7. **bridge domain** *domain-number*
8. **service-policy** {**input** | **output**} *policy-map-name*
9. **end**
10. **show policy-map interface** *type number* **service instance** *service-instance-number*

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface** *interface-type interface-number*<br><br>**Example:**<br>`Router(config)# interface gigabitethernet 0/0/1` | Configures an interface type and enters interface configuration mode.<br>• Enter the interface type and interface number. |
| **Step 4** | **service instance** *id* **ethernet** [*evc-name*]<br><br>**Example:**<br>`Router(config-if)# service instance 333 ethernet evc1` | Configures an Ethernet service instance on an interface and enters Ethernet service configuration mode.<br>• Enter the service instance identification number and, if applicable, the EVC name (optional). |
| **Step 5** | **encapsulation dot1q** *vlan-id* [,*vlan-id*[-*vlan-id*]] [**native**]<br><br>**Example:**<br>`Router(config-if-srv)# encapsulation dot1q 10` | Defines the matching criteria to map 802.1Q frames ingress on an interface to the appropriate service instance. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **rewrite ingress tag translate 1-to-1 dot1q** *vlan-id* **symmetric**<br>**Example:**<br>`Router(config-if-srv)# rewrite ingress tag translate 1-to-1 dot1q 300 symmetric` | Specifies the encapsulation adjustment to be performed on a frame ingressing a service instance. |
| **Step 7** | **bridge domain** *domain-number*<br>**Example:**<br>`Router(config-if-srv)# bridge domain 1` | Configures a bridge domain.<br>• Enter the bridge domain number. |
| **Step 8** | **service-policy** {**input** \| **output**} *policy-map-name*<br>**Example:**<br>`Router(config-if-srv)#`<br>`service-policy input policy1` | Attaches a policy map to an interface.<br>• Enter either the **input** or **output** keyword and the policy map name. |
| **Step 9** | **end**<br>**Example:**<br>`Router(config-if-srv)# end` | (Optional) Returns to privileged EXEC mode. |
| **Step 10** | **show policy-map interface** *type number* **service instance** *service-instance-number*<br>**Example:**<br>`Router# show policy-map interface gigabitethernet 1/0/0 service instance 30` | (Optional) Displays the statistics and the configurations of the input and output policies that are attached to an interface.<br>• Enter the interface type, interface number, and service instance number. |

# Configuration Examples for EVC Quality of Service

## Example Creating a Traffic Class for Use on the EVC

In this example, traffic with a CoS value of 2 is placed in the traffic class called class1:

```
Router> enable
Router# configure terminal
Router(config)# class-map match-any class1
Router(config-cmap)# match cos 2
Router(config-cmap)# end
```

## Example Creating a Policy Map for Use on the EVC

In this example, traffic policing has been configured in the policy map called policy1. Traffic policing is the QoS feature applied to the traffic in class1:

```
Router> enable
Router# configure terminal
Router(config)#
 policy-map policy1
Router(config-pmap)#
 class class1
Router(config-pmap-c)# police 3000

Router(config-pmap-c)# end
```

# Example Configuring the EVC and Attaching a Traffic Policy to the EVC

In this example, an EVC has been configured and a traffic policy called policy1 has been attached to the EVC:

```
Router> enable
Router# configure terminal
Router(config)# interface gigabitethernet 0/0/1
Router(config-if)# service instance 333 ethernet evc1
Router(config-if-srv)# encapsulation dot1q 10
Router(config-if-srv)# rewrite ingress tag translate 1-to-1 dot1q 300 symmetric
Router(config-if-srv)# bridge domain 1
Router(config-if-srv)# service-policy input policy1

Router(config-if-srv)# end
```

# Example Verifying the Traffic Class and Traffic Policy Information for the EVC

The following is sample output of the **show policy-map interface service instance**command. It displays the QoS features configured for and attached to the EFP on the GigabitEthernet interface 1/1/7.

```
Router# show policy-map interface gigabitethernet 1/1/7 service instance 10
 GigabitEthernet1/1/7: EFP 10
  Service-policy input: multiaction
    Class-map: c1 (match-all)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: ip precedence 3
      police:
          cir 300000 bps, bc 2000 bytes
        conformed 0 packets, 0 bytes; actions:
          set-prec-transmit 7
          set-qos-transmit 10
        exceeded 0 packets, 0 bytes; actions:
          drop
        conformed 0000 bps, exceed 0000 bps
    Class-map: class-default (match-any)
      0 packets, 0 bytes
      5 minute offered rate 0000 bps, drop rate 0000 bps
      Match: any
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Commands List, All Releases |

| Related Topic | Document Title |
|---|---|
| QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Quality of Service Solutions Command Reference* |
| Packet classification | "Classifying Network Traffic" module |
| Selective Packet Discard | “IPv6 Selective Packet Discard” module |

### Standards

| Standard | Title |
|---|---|
| No new or modified standards are supported, and support for existing standards has not been modified. | -- |

### MIBs

| MIB | MIBs Link |
|---|---|
| No new or modified MIBs are supported, and support for existing MIBs has not been modified. | To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### RFCs

| RFC | Title |
|---|---|
| No new or modified RFCs are supported, and support for existing RFCs has not been modified. | -- |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

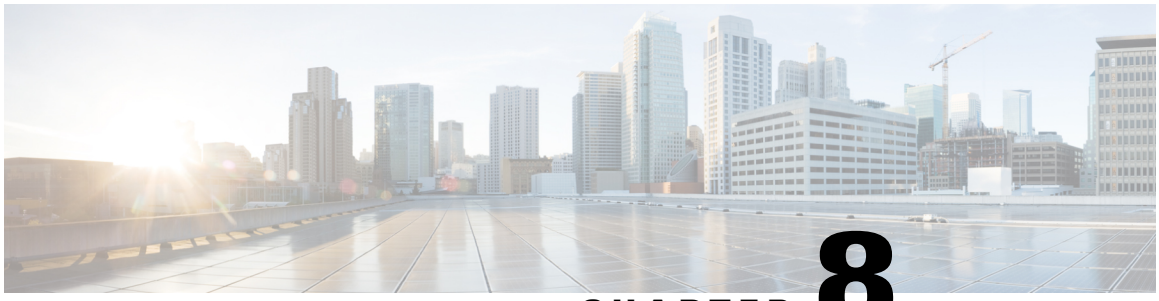
# Feature Information for Configuring EVC Quality of Service

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 11: Feature Information for EVC Quality of Service***

| Feature Name | Releases | Feature Information |
|---|---|---|
| EVC Quality of Service | Cisco IOS XE Release 3.3 | This document contains information about how to enable quality of service (QoS) features (such as traffic classification and traffic policing) for use on an Ethernet virtual circuit (EVC).<br>The EVC Quality of Service feature was introduced on the Cisco ASR 1000 Series Aggregation Services Router.<br>The following commands were introduced or modified: **service-policy**, **show policy-map interface service instance**. |



CHAPTER 8

# Quality of Service for Etherchannel Interfaces

Quality of Service (QoS) is supported on Ethernet Channel (Etherchannel) interfaces on Cisco ASR 1000 Series Routers. The QoS functionality has evolved over several Cisco IOS XE releases and has different capabilities based on software level, Etherchannel configuration, and configured Modular QoS CLI (MQC) features.



## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

## Information About QoS for Etherchannels

### Etherchannel with QoS Feature Evolution

An Etherchannel is a port-channel architecture that allows grouping of several physical links to create one logical Ethernet link for the purpose of providing fault tolerance, and high-speed links between switches, routers, and servers. An Etherchannel can be created from between two and eight active Fast, Gigabit, or

10-Gigabit Ethernet ports, with an additional one to eight inactive (failover) ports, which become active as the other active ports fail.

QoS for Etherchannel interfaces has evolved over several Cisco IOS XE releases. It is important to understand what level of support is allowed for your current level of Cisco IOS XE software and underlying Etherchannel configuration. Various combinations of QoS are supported based on how Etherchannel is configured. There are three different modes in which Etherchannel can be configured:

- Etherchannel VLAN-based load balancing via port-channel subinterface encapsulation CLI
- Etherchannel Active/Standby with LACP (no Etherchannel load balancing)
- Etherchannel with LACP with load balancing

Each of these models has specific restrictions regarding which levels of Cisco IOS XE software include support and the possible QoS configurations with each.

The following summarizes the various Etherchannel and QoS configuration combinations that are supported. Example configurations will be provided later in this document. Unless specifically mentioned together, the combination of service policies in different logical and physical interfaces for a given Etherchannel configuration is not supported.

**Etherchannel VLAN-Based Load Balancing via Port-Channel Subinterface Encapsulation CLI**

Supported in Cisco IOS XE Release 2.1 and subsequent releases:

- Egress MQC Queuing Configuration on Port-Channel Subinterface
- Egress MQC Queuing Configuration on Port-Channel Member Link
- QoS Policies Aggregation - Egress MQC Queuing at Subinterface
- Ingress Policing and Marking on Port-Channel Subinterface
- Egress Policing and Marking on Port-Channel Member Link

Supported in Cisco IOS XE Release 2.6 and subsequent releases:

- QoS Policies Aggregation - MQC Support for Multiple Queue Aggregation at Main Interface - Egress MQC Queuing at Main Interface

**Etherchannel Active/Standby with LACP (No Etherchannel Load Balancing)**

Supported in Cisco IOS XE Release 2.4 and subsequent releases:

- Egress MQC Queuing on Port-Channel Member Link - No Etherchannel Load Balancing

**Etherchannel with LACP and Load Balancing**

Supported in Cisco IOS XE Release 2.5 and subsequent releases:

- Egress MQC Queuing Configuration on Port-Channel Member Link - Etherchannel Load Balancing

There is no support for ingress QoS features in any release.

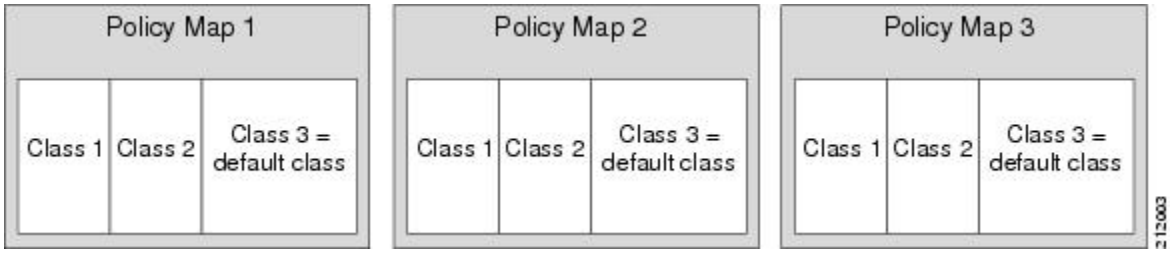
# Understanding Fragments in Class Definition Statements

The QoS Policies Aggregation feature introduces the idea of fragments in class definition statements. A default traffic class definition statement can be marked as a fragment within a policy map. Other policy maps on the

same interface can also define their default traffic class statements as fragments, if desired. A separate policy map can then be created with a service fragment class definition statement that will be used to apply QoS to all of the fragments as a single group.

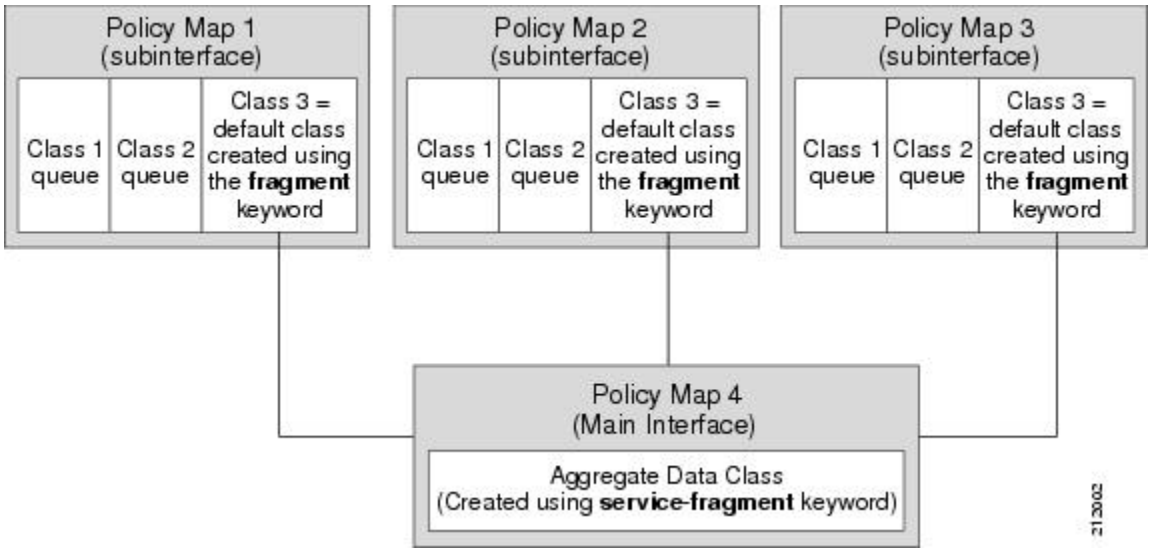
The figure below provides an example of one physical interface with three attached policy maps that is not using fragments. Note that each policy map has a default traffic class that can classify traffic only for the default traffic within its own policy map.

***Figure 4: Physical Interface with Policy Maps—Not Using Fragments***




The figure below shows the same configuration configured with fragments, and adds a fourth policy map with a class definition statement that classifies the fragments collectively. The default traffic classes are now classified as one service fragment group rather than three separate default traffic classes within the individual policy maps.

***Figure 5: Physical Interface with Policy Maps—Using Fragments***




# Understanding Fragments for Gigabit Etherchannel Bundles

When fragments are configured for Gigabit Etherchannel bundles, the policy maps that have a default traffic class configured using the **fragment** keyword are attached to the member subinterface links, and the policy maps that have a traffic class configured with the **service-fragment** keyword to collectively classify the fragments is attached to the physical interface.

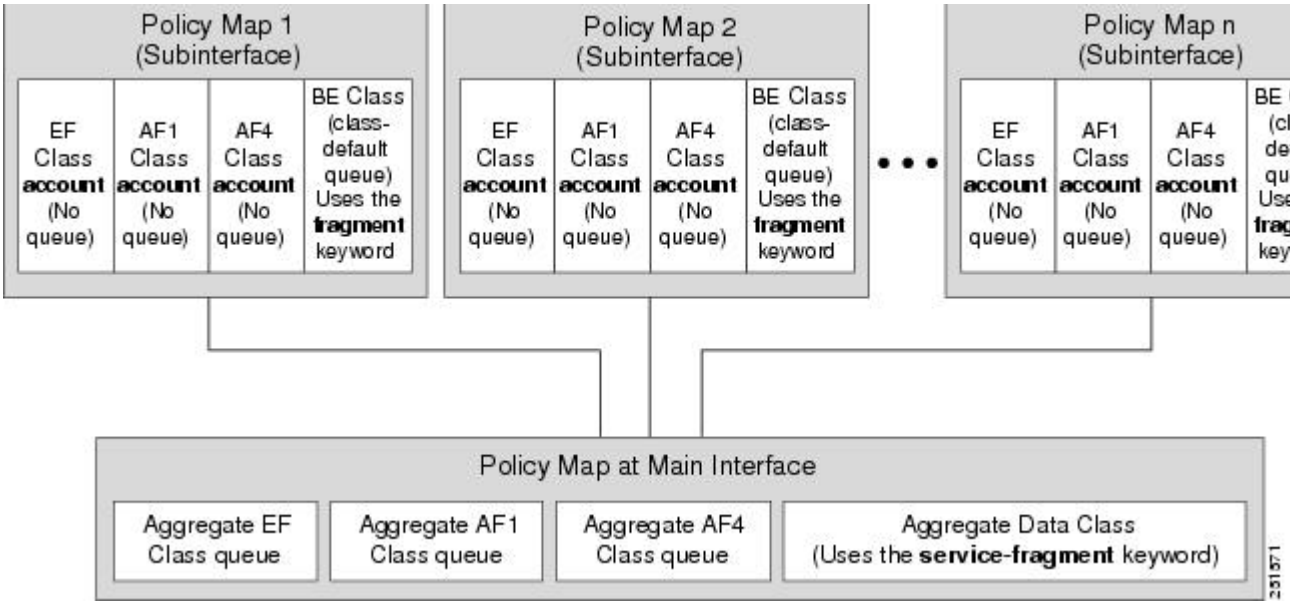
All port-channel subinterfaces configured with fragments that are currently active on a given port-channel member link will use the aggregate service fragment class on that member link. If a member link goes down, the port-channel subinterfaces that must switch to the secondary member link will then use the aggregate service fragment on the new interface.

# Understanding the QoS: Policies Aggregation MQC

The QoS: Policies Aggregation MQC Support for Multiple Queue Aggregation at Main Interface feature extends the previous support of aggregation of class-default traffic using the **fragment** and **service-fragment** configurations, to other user-defined traffic classes in a subinterface policy map, such as DSCP-based traffic classes, that are aggregated at the main-interface policy map as shown in the figure below.

When no queueing is configured on a traffic class in the subinterface policy map, the **account** command can be used to track queueing drops that occur at the aggregate level for these classes, and can be displayed using the **show policy-map interface** command.

***Figure 6: Policy Map Overview for the MQC Support for Multiple Queue Aggregation at Main Interface Feature***




# Differences Between the Original Feature and the MQC Support for Multiple Queue AggregationDifferences Between Policy Aggregation—Egress MQC Queuing at Subinterface and the MQC Support for Multiple Queue Aggregation at Main Interface

Although some of the configuration between the "Policy Aggregation – Egress MQC Queuing at Subinterface" scenario and the "MQC Support for Multiple Queue Aggregation at Main Interface - Egress MQC Queuing at Main Interface" scenario appear similar, there are some important differences in the queuing behavior and the internal data handling. See the figure in the "Understanding the QoS: Policies Aggregation MQC" section.

For example, both configurations share and require the use of the **fragment** keyword for the **class class-default** command in the subscriber policy map, as well as configuration of the **service-fragment** keyword for a user-defined class in the main-interface policy map to achieve common policy treatment for aggregate traffic. However, the use of this configuration results in different behavior between the original and enhanced QoS policies aggregation implementation:

- In the original implementation using the fragment and service-fragment architecture, all default class traffic and any traffic for classes without defined queueing features at the subinterface goes to the class-default queue and is aggregated into a common user-defined queue and policy defined at the main policy map. Subinterface traffic aggregation (for example, from multiple subscribers on the same physical interface) ultimately occurs only for a single class, which is the default class.
- In the enhanced implementation of the MQC Support for Multiple Queue Aggregation at Main Interface feature also using the fragment and service-fragment architecture, all default class traffic also goes to the class-default queue and is aggregated into a common user-defined queue and policy defined at the main policy map. However, other classes, such as DSCP-based subscriber traffic classes, are also supported for an aggregate policy. These traffic classes do not support any queues or queueing features other than **account** at the subscriber policy map. The use of the fragment and service-fragment architecture enables these other subscriber traffic classes (from multiple subscribers on the same physical interface) to achieve common policy treatment for aggregate traffic that is defined for those same classes at the main policy map.

# How to Configure QoS for Etherchannels

## Configuring Egress MQC Queuing on Port-Channel Subinterface

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one- or two-level hierarchical policy map should be configured using previously defined class maps. The port-channel subinterface should have been previously configured with the appropriate encapsulation subcommand to match the select primary and secondary physical interfaces on the Etherchannel. Cisco IOS XE Release 2.1 or later software is required. The global configuration must contain the **port-channel load-balancing vlan-manual** command, or the port-channel main-interface configuration must contain the **load-balancing vlan** command. It is assumed that these commands have already been executed.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-number.subinterface-number*
4. **service-policy output** *policy-map-name*
5. **end**

**DETAILED STEPS**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface port-channel** *port-channel-number.subinterface-number*<br><br>**Example:**<br>`Device(config)# interface port-channel 1.200` | Specifies the port-channel subinterface that receives the service policy configuration. |
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br>`Device(config-subif)#`<br>`service-policy output WAN-GEC-sub-Out` | Specifies the name of the service policy that is applied to output traffic. |
| **Step 5** | **end**<br><br>**Example:**<br>`Device(config-subif)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuring Egress MQC queuing on Port-Channel Member Links

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one- or two-level hierarchical policy-map that uses queuing features should be configured using previously defined class maps. The Etherchannel member link interface should already be configured to be part of the channel group (Etherchannel group). No policy maps that contain queuing commands should be configured on any port-channel subinterfaces. Cisco IOS XE Release 2.1 or later software is required. The global configuration must contain the **port-channel load-balancing vlan-manual** command, or the port-channel main-interface configuration must contain the **load-balancing vlan** command. It is assumed that these commands have already been executed.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *card/bay/port*
4. **service-policy output** *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface GigabitEthernet** *card/bay/port*<br><br>**Example:**<br>`Device(config)# interface GigabitEthernet 0/1/0` | Specifies the member link physical interface that receives the service policy configuration. |
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br>`Device(config-if)# service-policy output WAN-GEC-sub-Out` | Specifies the name of the service policy that is applied to output traffic for this physical interface that is part of the Etherchannel. |
| **Step 5** | **end**<br><br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuring QoS Policies Aggregation—Egress MQC Queuing at Subinterface

### Before You Begin

Default class traffic from multiple Port-channel subinterfaces can be aggregated into a common policy map at the main interface when you use the **fragment** keyword at the subinterface **class class-default** configuration, and the **service-fragment** configuration at the main interface class. Queuing occurs at the subinterface for other traffic classes that are defined with queuing features in the subinterface policy-map.

This feature is configured using Modular QoS CLI (MQC). It is most useful in QoS configurations where several policy maps attached to the same physical interface want aggregated treatment of multiple default traffic classes from multiple port-channel sub-interfaces. Cisco IOS XE Release 2.1 or later software is required. The global configuration must contain the **port-channel load-balancing vlan-manual** command, or the port-channel main-interface must have the **load-balancing vlan** command. It is assumed that these commands have already been executed.

**Note** This feature is supported when policy maps are attached to multiple port-channel subinterfaces and the port-channel member link interfaces. This feature cannot be used to collectively classify default traffic classes of policy maps on different physical interfaces. It can collectively classify all traffic directed toward a given port-channel member link when designated by the **primary** or **secondary** directives on the subinterface **encapsulation** command. All subinterface traffic classes should have queues. However, when a traffic class in the subinterface policy-map is not configured with any queuing feature (commands such as **priority**, **shape**, **bandwidth**, **queue-limit**, **fair-queue**, or **random-detect**), the traffic is assigned to the class-default queue. No classification occurs or is supported at the main interface policy-map for any subinterface traffic classes that do not use the **fragment** and **service-fragment** configuration.

A multistep process is involved with the complete configuration of the QoS Policies Aggregation feature. The following sections detail those steps.

Note the following about attaching and removing a policy map:

- To configure QoS Policies Aggregation, you must attach the policy map that contains the **service-fragment** keyword to the main interface first, and then you must attach the policy map that contains the **fragment** keyword to the subinterface.
- To disable QoS Policies Aggregation, you must remove the policy map that contains the fragment keyword from the subinterface first, and then you must remove the policy map that contains the service-fragment keyword from the main interface.

## Configuring a Fragment Traffic Class in a Policy Map

### Before You Begin

This procedure shows only how to configure the default traffic class as a fragment within a policy map. It does not include steps on configuring other classes within the policy map, or other policy maps on the device.

**Note** Only the default class statement in a policy map can be configured as a fragment.

Fragments work only when multiple policy maps are attached to the same physical interface. This process cannot be used to classify default traffic classes as fragments on policy maps on different physical interfaces.

Only queuing features are allowed in classes where the **fragment** keyword is entered, and at least one queuing feature must be entered in classes where the **fragment** keyword is used.

A policy map with a class using the **fragment** keyword can only be applied to traffic leaving the interface (policy maps attached to interfaces using the **service-policy output** command).

The **fragment** keyword cannot be entered in a child policy map.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class class-default fragment** *fragment-class-name*
5. **shape average percent** *percent*
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br>**Example:**<br>`Device(config)# policy-map subscriber1` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class class-default fragment** *fragment-class-name*<br>**Example:**<br>`Device(config-pmap)# class class-default fragment BestEffort` | Specifies the default traffic class as a fragment, and names the fragment traffic class. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 5** | **shape average percent** *percent*<br><br>**Example:**<br><br>`Device(config-pmap-c)# shape average percent 50` | Enters a QoS configuration command. Only queuing features are supported in default traffic classes configured as fragments.<br><br>The queuing features supported are **bandwidth**, **shape**, and **random-detect exponential-weighting-constant**.<br><br>Multiple QoS queuing commands can be entered. |
| **Step 6** | **end**<br><br>**Example:**<br><br>`Device(config-pmap-c)# end` | Exits policy map class configuration mode and returns to privileged EXEC mode. |

### Example

**Note** This example shows a sample configuration that is supported in releases prior to Cisco IOS XE Release 2.6.

In the following example, a fragment named BestEffort is created in policy map subscriber1 and policy map subscriber 2. In this example, queuing features for other traffic classes are supported at the subinterface policy map.

```
policy-map subscriber1
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber 2
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

**Note** This example shows a sample configuration that is supported in Cisco IOS XE Release 2.6 and later releases.

The following example also shows how to configure a fragment named BestEffort for the default class in a policy map on a subinterface using the QoS Policies Aggregation MQC Support for Multiple Queue Aggregation

at Main Interface implementation. In this example, notice that queuing features are not supported for the other classes in the policy map:

```
policy-map subscriber1
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

### What to Do Next

After configuring multiple default class statements as fragments in a policy map, a separate policy map with a class statement using the **service-fragment** keyword must be configured to apply QoS to the class statements configured as fragments.

This process is documented in the "Configuring a Service Fragment Traffic Class" section.

## Configuring a Service Fragment Traffic Class

### Before You Begin

This task describes how to configure a service fragment traffic class statement within a policy map. A service fragment traffic class is used to apply QoS to a collection of default class statements that have been configured previously in other policy maps as fragments.

This procedure assumes that fragment default traffic classes were already created. The procedure for creating fragment default traffic classes is documented in the "Configuring a Fragment Traffic Class in a Policy Map" section.

Like any policy map, the configuration does not manage network traffic until it has been attached to an interface. This procedure does not cover the process of attaching a policy map to an interface.

**Note** A service fragment can be used to collectively classify fragments only from the same physical interface. Fragments from different interfaces cannot be classified using the same service fragment.

Only queueing features are allowed in classes where the **service-fragment** keyword is entered, and at least one queueing feature must be entered in classes when the **service-fragment** keyword is used.

A policy map with a class using the **service-fragment** keyword can be applied only to traffic leaving the interface (policy maps attached to interfaces using the **service-policy output** command).

A class configured using the **service-fragment** keyword cannot be removed when it is being used to collectively apply QoS to fragments that are still configured on the interface. If you wish to remove a class configured using the **service-fragment** keyword, remove the fragment traffic classes before removing the service fragment.

The **service-fragment** keyword cannot be entered in a child policy map.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **policy-map** *policy-map-name*
4. **class** *class-name* **service-fragment** *fragment-class-name*
5. **shape average percent** *percent*
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **policy-map** *policy-map-name*<br><br>**Example:**<br>`Device(config)# policy-map BestEffortFragments` | Specifies the name of the traffic policy to configure and enters policy map configuration mode. |
| **Step 4** | **class** *class-name* **service-fragment** *fragment-class-name*<br><br>**Example:**<br>`Device(config-pmap)# class data service-fragment BestEffort` | Specifies a class of traffic that is the composite of all fragments matching the *fragment-class-name*. The *fragment-class-name* when defining the fragments in other policy maps must match the *fragment-class-name* in this command line to properly configure the service fragment class. |
| **Step 5** | **shape average percent** *percent*<br><br>**Example:**<br>`Device(config-pmap-c)# shape average percent 50` | Enters a QoS configuration command. Only queueing features are supported in default traffic classes configured as fragments.<br>The queueing features that are supported are **bandwidth**, **shape**, and **random-detect exponential-weighting-constant**.<br>Multiple QoS queueing commands can be entered. |
| **Step 6** | **end**<br><br>**Example:**<br>`Device(config-pmap-c)# end` | Exits policy map class configuration mode and returns to privileged EXEC mode. |

### Examples

**Note** This example shows a sample configuration that is supported in releases prior to Cisco IOS XE Release 2.6.

In the following example, a policy map is created to apply QoS to all fragments named BestEffort.

```
policy-map main-interface
 class data service-fragment BestEffort
  shape average 400000000
```

In the following example, two fragments are created and then classified collectively using a service fragment.

```
policy-map subscriber1
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber 2
 class voice
  set cos 5
  priority level 1
 class video
  set cos 4
  priority level 2
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
```

**Note** This example shows a sample configuration that is supported in Cisco IOS XE Release 2.6 and later releases.

The following example shows the creation of two fragments called BestEffort in the subinterface policy maps, followed by a sample configuration for the **service-fragment** called BestEffort to aggregate the queues at the main interface policy map:

```
policy-map subscriber1
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map subscriber2
 class voice
  set cos 5
  account
 class video
```

```
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
policy-map main-interface
 class voice
  priority level 1
 class video
  priority level 2
 class AF1
  bandwidth remaining ratio 90
 class data service-fragment BestEffort
  shape average 400000000
  bandwidth remaining ratio 1
```

### Troubleshooting Tips

Ensure that all class statements that should be part of the same service fragment share the same *fragment-class-name*.

### What to Do Next

Attach the service fragment traffic classes to the main physical interfaces.

Attach the fragment traffic classes to the member-link subinterfaces.

## Configuring Service Fragments on a Physical Interface Supporting a Gigabit Etherchannel Bundle

### Before You Begin

This procedure assumes that a service fragment traffic class has already been created. A service fragment traffic class cannot be configured without configuring a fragment class. The procedure for creating a fragment class is documented in the “Configuring a Fragment Traffic Class in a Policy Map” section. The procedure for creating a service fragment traffic classes is documented in the “Configuring a Service Fragment Traffic Class” section.

These instructions do not provide any details about the options that can be configured for Gigabit Etherchannel member link subinterfaces. These instructions document only the procedure for attaching a policy map that already has a fragment traffic class to a member link subinterface.

**Note** For proper behavior, when a port-channel member link goes down, all member links should have the same policy map applied.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *card/bay/port*
4. **service-policy output** *service-fragment-class-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface GigabitEthernet** *card/bay/port*<br>**Example:**<br>`Device(config)# interface GigabitEthernet 0/1/0` | Specifies the member link physical interface that receives the service-policy configuration. |
| **Step 4** | **service-policy output** *service-fragment-class-name*<br>**Example:**<br>`Device(config-if)# service-policy output aggregate-member-link` | Attaches a service policy that contains a service fragment default traffic class to the physical Gigabit Ethernet interface. |
| **Step 5** | **end**<br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

### Examples

In the following example, the policy map aggregate-member-link is attached to the physical interface.

```
interface GigabitEthernet1/1/1
 service-policy output aggregate-member-link
!
```

```
interface GigabitEthernet1/1/2
 service-policy output aggregate-member-link
```

### What to Do Next

Ensure that the fragment class name is consistent across service-fragment and fragment class definitions. Continue to the "Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces" section.

## Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces

### Before You Begin

This procedure assumes that a service fragment traffic class has already been created. A service fragment traffic class cannot be configured without configuring a fragment class. The procedure for creating a fragment class is documented in the "Configuring a Fragment Traffic Class in a Policy Map" section. The procedure for creating a service fragment traffic class is documented in the "Configuring a Service Fragment Traffic Class" section.

These instructions do not provide any details about the options that can be configured for Gigabit Etherchannel member link subinterfaces. These instructions only document the procedure for attaching a policy map that already has a fragment traffic class to a member link subinterface.

Fragments cannot be used for traffic on two or more physical interfaces.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-interface-number* **.** *port-channel-subinterface-number*
4. **service-policy output** *fragment-class-name*
5. **end**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 3** | **interface port-channel** *port-channel-interface-number* **.** *port-channel-subinterface-number*<br><br>**Example:**<br><br>`Device(config)# interface port-channel 1.100` | Enters subinterface configuration mode to configure an Etherchannel member link subinterface. |
| **Step 4** | **service-policy output** *fragment-class-name*<br><br>**Example:**<br><br>`Device(config-subif)# service-policy output subscriber` | Attaches a service policy that contains a fragment default traffic class to the Etherchannel member link subinterface |
| **Step 5** | **end**<br><br>**Example:**<br><br>`Device(config-subif)# end` | Exits subinterface configuration mode and returns to privileged EXEC mode. |

### Example

In the following example, the service policy named subscriber has a fragment default traffic class and is attached to the port-channel subinterface of an Etherchannel bundle.

```
interface port-channel 1.100
 service-policy output subscriber
```

# Configuring Ingress Policing and Marking on Port-Channel Subinterface

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one- or two-level hierarchical policy-map should be configured using previously defined class maps. The Etherchannel member link interface should already be configured to be part of the channel group (Etherchannel group). Cisco IOS XE Release 2.1 or later software is required. The global configuration must contain the **port-channel load-balancing vlan-manual** command or the port-channel main-interface configuration must contain the **load-balancing vlan** command. It is assumed that these commands have already been executed.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-number.port-channel-interface-number.sub-interface-number*
4. **service-policy input** *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface port-channel** *port-channel-number.port-channel-interface-number.sub-interface-number*<br><br>**Example:**<br>`Device(config)# interface port-channel 1.100.100` | Enters subinterface configuration mode to configure an Etherchannel member link subinterface. |
| **Step 4** | **service-policy input** *policy-map-name*<br><br>**Example:**<br>`Device(config-subif)# service-policy input sub-intf-input` | Specifies the name of the service policy that is applied to input traffic for the port-channel subinterface previously specified. |
| **Step 5** | **end**<br><br>**Example:**<br>`Device(config-subif)# end` | Exits subinterface configuration mode and returns to privileged EXEC mode. |

### Example

In the following example, the service policy named sub-intf-input is defined and attached to the port-channel subinterface in the input direction.

```
policy-map sub-intf-input
  class voice
```

```
    set precedence 5
  class video
    set precedence 6
  class class-default
    set precedence 3
!
interface Port-channel 1.100
 service-policy input sub-intf-input
```

# Configuring Egress Policing and Marking on Port-Channel Member Links

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one- or two-level hierarchical policy-map should be configured using previously defined class maps. The Etherchannel member link interface should already be configured to be part of the channel group (Etherchannel group). Cisco IOS XE Release 2.1 or later software is required. The global configuration must contain the **port-channel load-balancing vlan-manual** command or the port-channel main-interface configuration must contain the **load-balancing vlan** command. It is assumed that these commands have already been executed.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface port-channel** *port-channel-number.port-channel-interface-number.sub-interface-number*
4. **service-policy output** *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface port-channel** *port-channel-number.port-channel-interface-number.sub-interface-number*<br><br>**Example:**<br>`Device(config)# interface port-channel 1.100.100` | Enters subinterface configuration mode to configure an Etherchannel member link subinterface. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br><br>`Device(config-subif)#`<br>`service-policy output WAN-GEC-member-Out-police` | Specifies the name of the service policy that is applied to output traffic for the Etherchannel member link subinterface specified in the previous step. |
| **Step 5** | **end**<br><br>**Example:**<br><br>`Device(config-subif)# end` | Exits subinterface configuration mode and returns to privileged EXEC mode. |

### Example

In the following example, the service policy named WAN-GEC-member-Out-police is defined and attached to the port-channel subinterface in the output direction.

```
policy-map WAN-GEC-member-Out-police
  class voice
    set precedence 5
  class video
    set precedence 6
  class class-default
    set precedence 3
!
interface port-channel 1.100
 service-policy output WAN-GEC-member-Out-police
```

# Configuring Policies Aggregation—MQC Support for Multiple Queue Aggregation at Main Interface

### Before You Begin

This feature is configured using the MQC. It is most useful in QoS configurations where several policy maps attached to the same physical interface want aggregated treatment of multiple user-defined traffic classes from multiple port-channel subinterfaces. Cisco IOS XE Release 2.6 or later software is required. The global configuration must contain the following command: **port-channel load-balancing vlan-manual** or the main interface of the port-channel being configured must have the following command: **port-channel load-balancing vlan**. It is assumed that these commands have already been executed.

This feature is supported when policy maps are attached to multiple port-channel subinterfaces and the port-channel member link interfaces. This feature cannot be used to collectively classify default traffic classes of policy maps on different physical interfaces. It can collectively classify all traffic directed towards a given Port-channel member-link when designated by the **primary** or **secondary** directives on the sub-interface **encapsulation** command. The following items describe the behavior and restrictions on configuring this type of QoS Policy Aggregation with Etherchannel:

- Subinterface traffic classes without configured queuing features do not have queues at the subscriber level
- Default class traffic from multiple subinterfaces can be aggregated into a common policy-map at the main interface when you use the **fragment** keyword at the subinterface **class class-default** configuration, and **service-fragment** configuration at the main interface class
- This configuration additionally enables support for other subinterface traffic classes (such as DSCP-based classes) to be aggregated into a common policy-map at the main interface.
- This feature is enabled by using the **fragment** keyword in the subinterface **class-default** class, and **service-fragment** configuration in the main interface class (this also enables aggregation of the default class.
- Queuing features are not configured at the subinterface policy-map for the other traffic classes.
- Queuing occurs at the main interface policy-map for other subinterface traffic classes as an aggregate.
- Optional tracking of statistics is supported using the **account** command for other traffic classes in the subinterface policy map.

A multistep process is involved with the complete configuration of QoS multiple queue aggregation at a main interface feature, as follows:

1. Configure default class statements as fragments in multiple subinterface policy maps as described in the "Configuring a Fragment Traffic Class in a Policy Map" section.
2. Configure a separate policy map with a class statement using the **service-fragment** keyword in order to apply QoS to the class statements configured as fragments as described in the "Configuring a Service Fragment Traffic Class" section.
3. Configure service fragment traffic classes and attach them to the main physical interfaces as described in the "Configuring Service Fragments on a Physical Interface Supporting a Gigabit Etherchannel Bundle" section.
4. Configure fragment traffic classes and attach them to the member link subinterfaces as described in the "Configuring Fragments on Gigabit Etherchannel Member Link Subinterfaces" section.

# Configuring MQC Queuing on Port-Channel Member Link—No Etherchannel Load Balancing

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one or two level hierarchical policy-map should be configured using previously defined class maps.

Cisco IOS XE Release 2.4 or later software is required.

The port-channel main interface should also contain the following commands that create an active/standby scenario. Such a configuration will allow only a single interface to be active and forwarding traffic at any time.

- **interface Port-channel1**
- **lacp fast-switchover**

- **lacp max-bundle 1**

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *card/bay/port*
4. **service-policy output** *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Device> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface GigabitEthernet** *card/bay/port*<br><br>**Example:**<br>`Device(config)# interface GigabitEthernet 0/1/0` | Specifies the member link physical interface that receives the service policy configuration. |
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br>`Device(config-if)#`<br>`service-policy output WAN-GEC-member-Out` | Specifies the name of the service policy that is applied to output traffic. |
| **Step 5** | **end**<br><br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

### Example

In the following example, the service policy named main-intf is defined and attached to the port-channel member links in the output direction.

```
interface Port-channel 1
 lcap fast-switchover
 lacp max-bundle 1
!
policy-map main-intf
 class voice
  priority
  police cir 10000000
 class video
  bandwidth remaining ratio 10
 class class-default
  bandwidth remaining ratio 3
!
interface GigabitEthernet0/0/0
 channel-group 1 mode active
 service-policy output main-intf
!
interface GigabitEthernet0/0/1
 channel-group 1 mode active
 service-policy output main-intf
```

# Configuring MQC Queuing Configuration on Port-Channel Member Link—Etherchannel Load Balancing

### Before You Begin

Traffic classes must be configured using the **class-map** command. A one- or two-level hierarchical policy-map should be configured using previously defined class maps. The port-channel subinterface should have been previously configured with the appropriate encapsulation subcommand to match the select primary and secondary physical interfaces on the Etherchannel. Cisco IOS XE Release 2.5 or later software is required.

The Etherchannel setup may have multiple active interfaces with flow-based load balancing enabled.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface GigabitEthernet** *card*/*bay*/*port*
4. **service-policy output** *policy-map-name*
5. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Device> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface GigabitEthernet** *card/bay/port*<br><br>**Example:**<br>`Device(config)# interface GigabitEthernet 0/1/0` | Specifies the member link physical interface that receives the service policy configuration. |
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br>`Device(config-if)#`<br>`service-policy output WAN-GEC-member-Out` | Specifies the name of the service policy that is applied to output traffic. |
| **Step 5** | **end**<br><br>**Example:**<br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

### Example

In the following example, the service policy named main-intf is defined and attached to the port-channel member links in the output direction.

```
  class voice
   priority
   police cir 10000000
  class video
   bandwidth remaining ratio 10
  class class-default
   bandwidth remaining ratio 3
!
interface GigabitEthernet0/0/0
 channel-group 1 mode active
 service-policy output main-intf
!
interface GigabitEthernet0/0/1
 channel-group 1 mode active
 service-policy output main-intf
```

# Configuration Examples for QoS for Etherchannels

## Example: Configuring QoS Policies Aggregation—Egress MQC Queuing at Subinterface

```
port-channel load-balancing vlan-manual
!
class-map match-all BestEffort
!
class-map match-all video
 match precedence 4
!
class-map match-all voice
 match precedence 5
!
policy-map subscriber
 class voice
  priority level 1
 class video
  priority level 2
 class class-default fragment BE
  shape average 100000000
  bandwidth remaining ratios 80

policy-map aggregate-member-link
 class BestEffort service-fragment BE
 shape average 100000000
!
interface Port-channel1
 ip address 209.165.200.225 255.255.0.0
!
interface Port-channel1.100
 encapsulation dot1Q 100
 ip address 209.165.200.226 255.255.255.0
 service-policy output subscriber
!
interface Port-channel1.200
 encapsulation dot1Q 200
 ip address 209.165.200.227 255.255.255.0
 service-policy output subscriber
!
interface Port-channel1.300
 encapsulation dot1Q 300
 ip address 209.165.200.228 255.255.255.0
 service-policy output subscriber
!
interface GigabitEthernet1/1/1
 no ip address
 channel-group 1 mode on
 service-policy output aggregate-member-link
!
interface GigabitEthernet1/1/2
 no ip address
 channel-group 1 mode on
 service-policy output aggregate-member-link
```

# Example: Configuring QoS Policies Aggregation—MQC Support for Multiple Queue Aggregation at Main Interface

```
port-channel load-balancing vlan-manual
!
policy-map subscriber1
 class voice
  set cos 5
  account
 class video
  set cos 4
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
!
policy-map subscriber2
 class voice
  set cos 2
  account
 class video
  set cos 3
  account
 class AF1
  account
 class class-default fragment BestEffort
  shape average 200000000
  bandwidth remaining ratio 10
!
policy-map main-interface-out
 class voice
  priority level 1
 class video
  priority level 2
 class AF1
  bandwidth remaining ratio 90
 class data service-fragment BestEffort
  shape average 400000000
  bandwidth remaining ratio 1
!
interface GigabitEthernet1/1/1
 no ip address
 channel-group 1 mode on
 service-policy output main-interface-out
!
interface GigabitEthernet1/1/2
 no ip address
 channel-group 1 mode on
 service-policy output main-interface-out
!
interface Port-channel1.100
 encapsulation dot1Q 100
 ip address 10.0.0.1 255.255.255.0
 service-policy output subscriber1
!
interface Port-channel1.200
 encapsulation dot1Q 200
 ip address 10.0.0.2 255.255.255.0
 service-policy output subscriber2
!
interface Port-channel1.300
 encapsulation dot1Q 300
 ip address 10.0.0.4 255.255.255.0
 service-policy output subscriber2
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Command List, All Releases |
| QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Quality of Service Solutions Command Reference* |
| Modular Quality of Service Command-Line Interface | "Applying QoS Features Using the MQC" module |
| Configuring RADIUS-based policing | *Intelligent Services Gateway Configuration Guide* |
| CISCO ASR 1000 Series software configuration | *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for Quality of Service for Etherchannel Interfaces

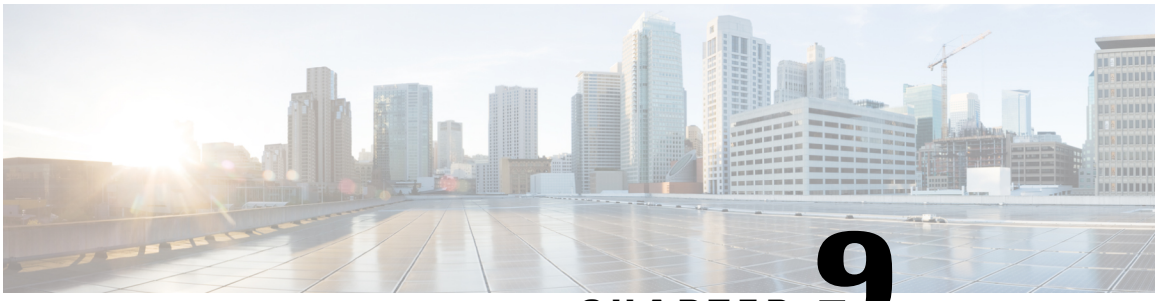
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

*Table 12: Feature Information for Quality of Service for Etherchannel Interfaces*

| Feature Name | Releases | Feature Information |
|---|---|---|
| Egress MQC Queuing Configuration on Port-Channel Subinterface | Cisco IOS XE Release 2.1 | This feature supports the configuration of Egress MQC queuing on port-channel subinterface.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| Egress MQC Queuing Configuration on Port-Channel Member Link | Cisco IOS XE Release 2.1 | This feature supports the configuration of Egress MQC queuing on port-channel member link.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| QoS Policies Aggregation—Egress MQC Queuing at Subinterface | Cisco IOS XE Release 2.1 | This feature supports the configuration of QoS Policies Aggregation - Egress MQC queuing at subinterface.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| Ingress Policing and Marking on Port-Channel Subinterface | Cisco IOS XE Release 2.1 | This feature supports the configuration of Ingress Policing and Marking on port-channel subinterface.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| Egress Policing and Marking on Port-Channel Member Link | Cisco IOS XE Release 2.1 | This feature supports the configuration of Egress policing and marking on port-channel member link.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| Egress MQC Queuing Configuration on Port-Channel Member Link - No Etherchannel Load Balancing | Cisco IOS XE Release 2.4 | This feature supports the configuration of Egress MQC Queuing on Port-Channel Member Link - no Etherchannel Load Balancing.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| Egress MQC Queuing Configuration Supported on Port-Channel Member Link - Etherchannel Load Balancing | Cisco IOS XE Release 2.5 | This feature supports the configuration of Egress MQC Queuing on Port-Channel Member Link - Etherchannel Load Balancing.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |
| QoS Policies Aggregation - MQC Support for Multiple Queue Aggregation at Main Interface - Egress MQC Queuing at Main Interface | Cisco IOS XE Release 2.6 | This feature supports the configuration of QoS Policies Aggregation - MQC Support for Multiple Queue Aggregation at Main Interface - Egress MQC Queuing at Main Interface.<br>This feature was introduced on Cisco ASR 1000 Series Routers. |



CHAPTER 9

# PPPoGEC Per Session QoS

The PPPoGEC Per Session QoS feature supports the configuration of specific QoS policies on PPPoE sessions on the PPP Termination and Aggregation (PTA), L2TP Access Concentrator (LAC), or L2TP Network Server (LNS) devices in a PPPoE /L2TP environment (broadband deployments). PPPoE sessions with Etherchannel Active/Standby functionality is also supported on Cisco ASR 1000 Series Routers acting as PTA, LAC, or LNS devices in a PPPoE/L2TP environment.



# Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Information About PPPoGEC Per Session QoS

## Restrictions for PPPoGEC Per Session QoS

- QoS policy maps cannot be configured on member links, a port-channel main interface, or a port-channel subinterface that is associated with the transmit path for PPPoE sessions with QoS.

## PPPoGEC Sessions with Active/Standby Etherchannel

PPPoE sessions with active/standby Etherchannel support one-level or two-level hierarchical output policy maps (with queueing settings) also support flat input policy maps (without queueing settings). The policy maps are configured using previously defined class maps. The traffic classes must be configured using the **class-map** command.

The output hierarchical policy map and the input policy map can be associated with the PPPoE sessions in one of the following ways:

- Configuration settings on a virtual template interface
- Dynamic configuration settings via external tools configured in the authentication, authorization, and accounting (AAA) model (for example, a radius server). For more information, see the *Intelligent Services Gateway Configuration Guide* and the *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide*.

The port-channel main interface must contain the following commands that create an active/standby scenario. Such a configuration will allow only a single interface to be active and forwarding traffic at any time.

- **interface port-channel1**
- **lacp fast-switchover**
- **lacp max-bundle 1**

# How to Configure PPPoGEC Per Session QoS

## Configuring QoS on PPPoE Sessions with Etherchannel Active/Standby

To configure QoS on PPPoE sessions, you must specify the virtual template to use for PPP sessions on the Etherchannel interface, specify the name of the service policy that is applied to input traffic, and specify the output traffic. This configuration shows how to associate the output hierarchical policy map and the input policy map with the PPPoE sessions by defining a virtual template interface.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface virtual-template** *number*
4. **service-policy output** *policy-map-name*
5. **service-policy input** *policy-map-name*
6. **end**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br><br>`Device> enable` | Enables privileged EXEC mode.<br><br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br><br>`Device# configure terminal` | Enters global configuration mode. |
| **Step 3** | **interface virtual-template** *number*<br><br>**Example:**<br><br>`Device(config)# interface virtual-template 99` | Creates a virtual template interface that can be configured and applied dynamically in creating virtual access interfaces, and enters interface configuration mode.<br><br>• Specify the virtual template to use for PPP sessions on the Etherchannel interface. |
| **Step 4** | **service-policy output** *policy-map-name*<br><br>**Example:**<br><br>`Device(config-if)# service-policy output session_parent` | Specifies the name of the service policy that is applied to output traffic. |
| **Step 5** | **service-policy input** *policy-map-name*<br><br>**Example:**<br><br>`Device(config-if)# service-policy input session_ingress` | Specifies the name of the service policy that is applied to input traffic. |
| **Step 6** | **end**<br><br>**Example:**<br><br>`Device(config-if)# end` | Exits interface configuration mode and returns to privileged EXEC mode. |

# Configuration Examples for PPPoGEC Per Session QoS

## Example: QoS on PPPoE Sessions with Etherchannel Active/Standby

The following example shows the session_parent hierarchical policy map and the session_ingress policy map. These policy maps are attached to a virtual template interface using the **service-policy** command.

```
policy-map session_child
  class voice
    priority level 1
    police cir 256000
    set precedence 5
  class web
    bandwidth remaining ratio 10
  class p2p
    bandwidth remaining ratio 1
    set precedence 1
  class class-default
    set precedence 2
    bandwidth remaining ratio 5
!
policy-map session_parent
  class class-default
    bandwidth remaining ratio 1
    shape average 25000000
    service-policy session_child
!
policy-map session_ingress
  class voip
    police cir 256000
  class p2p
    police cir 256000 pir 512000
      conform-action set-prec-transmit 1
      exceed  set-prec-transmit 0
      violate drop
  class class-default
    police cir 5000000
      conform-action set-prec-transmit 2
      exceed  drop
!
interface Virtual-template 99
 service-policy output session_parent
 service-policy input session_ingress
```

# Additional References for PPPoGEC Per Session QoS

### Related Documents

| Related Topic | Document Title |
|---|---|
| Cisco IOS commands | Cisco IOS Master Command List, All Releases |
| QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples | *Cisco IOS Quality of Service Solutions Command Reference* |

| Related Topic | Document Title |
|---|---|
| Modular Quality of Service Command-Line Interface | "Applying QoS Features Using the MQC" module |
| Configuring RADIUS-based policing | *Intelligent Services Gateway Configuration Guide* |
| CISCO ASR 1000 Series software configuration | *Cisco ASR 1000 Series Aggregation Services Routers Software Configuration Guide* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

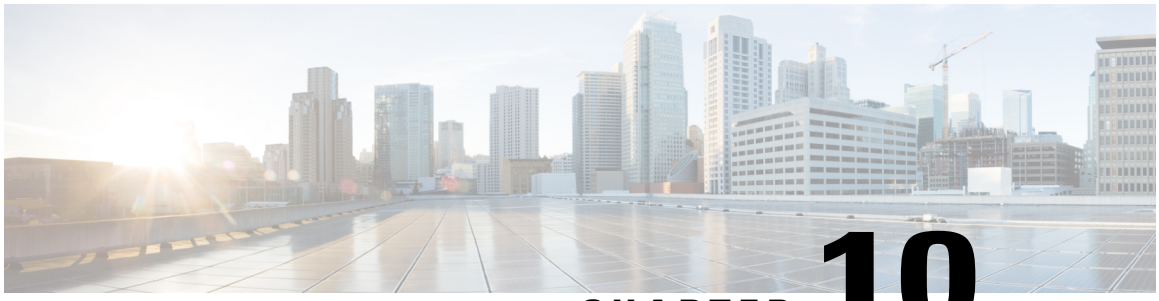
# Feature Information for PPPoGEC Per Session QoS

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

*Table 13: Feature Information for PPPoGEC Per Session QoS*

| Feature Name | Releases | Feature Information |
|---|---|---|
| PPPoGEC: Per Session QoS | Cisco IOS XE Release 3.7S<br>Cisco IOS XE Release 3.8S | This feature supports the configuration of specific QoS policies on PPPoE sessions on the PTA, LAC, and LNS for broadband deployments.<br>This feature was introduced on Cisco ASR 1000 Series Routers.<br>In Cisco IOS XE Release 3.8S support was added for per-session QoS in vlan mapping mode and per-session QoS in 1:1 mode for PPPoGEC. |



CHAPTER 10

# IPv6 Selective Packet Discard

The selective packet discard (SPD) mechanism manages the process level input queues on the RP. SPD provides priority to routing protocol packets and other important traffic control Layer 2 keepalives during periods of process level queue congestion



# Finding Feature Information

Your software release may not support all the features documented in this module. For the latest feature information and caveats, see the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the Feature Information Table at the end of this document.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

# Information About IPv6 Selective Packet Discard

## SPD in IPv6 Overview

The SPD mechanism manages the process level input queues on the RP. SPD provides priority to routing protocol packets and other important traffic control Layer 2 keepalives during periods of process level queue congestion.

## SPD State Check

The SPD state check is performed on the IPv6 process input queue on the RP. High-priority packets, such as those of IP precedence 7, are not applied to SPD and are never dropped. All remaining packets, however, can be dropped depending on the length of the IPv6 packet input queue and the SPD state. The possible SPD states are as follows:

- Normal: The process input queue is less than the SPD minimum threshold.
- Random drop: The process input queue is between the SPD minimum and maximum thresholds.
- Max: The process input queue is equal to the SPD maximum threshold.

The size of the process input queue governs the SPD state: normal (no drop), random drop, or max. When the process input queue is less than the SPD minimum threshold, SPD takes no action and enters normal state. In the normal state, no packets are dropped. When the input queue reaches the maximum threshold, SPD enters max state, in which normal priority packets are discarded. If the input queue is between the minimum and maximum thresholds, SPD enters the random drop state, in which normal packets may be dropped.

## SPD Mode

Three IPv6 SPD modes are supported: none (which is the default), aggressive drop, and OSPF mode. The aggressive drop mode discards incorrectly formatted packets when the IPv6 is in the random drop state. OSPF mode provides a mechanism whereby OSPF packets are handled with SPD priority.

## SPD Headroom

With SPD, the behavior of normal IPv6 packets is not changed. However, routing protocol packets are given higher priority, because SPD recognizes routing protocol packets by the IPv6 precedence field. Therefore, if the IPv6 precedence is set to 7, then the packet is given priority.

SPD prioritizes IPv6 packets with a precedence of 7 by allowing the Cisco IOS software to queue them into the process level input queue above the normal input queue limit. The number of packets allowed in excess of the normal limit is called the SPD headroom. The SPD headroom default is 100, which means that a high precedence packet is not dropped if the size of the input hold queue is lower than 175 (which is the input queue default size + SPD headroom size).

Because Interior Gateway Protocols (IGPs) and link stability are tenuous and crucial, such packets are given the highest priority and are given extended SPD headroom with a default of 10 packets. These packets are not dropped if the size of the input hold queue is lower than 185 (input queue default size + SPD headroom size + SPD extended headroom).

Non-IPv6 packets such as Connectionless Network Service Intermediate System-to-Intermediate System (CLNS IS-IS) packets, PPP packets, and High-Level Data Link Control (HDLC) keepalives are treated as normal priority as a result of being Layer 2 instead of Layer 3. In addition, IGPs operating at Layer 3 or higher are given priority over normal IPv6 packets, but are given the same priority as Border Gateway Protocol (BGP) packets. Therefore, during BGP convergence or during times of very high BGP activity, IGP hellos and keepalives often are dropped, causing IGP adjacencies to fail.

# How to Configure IPv6 Selective Packet Discard

## Configuring the SPD Process Input Queue

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 spd queue max-threshold** *value*
4. **ipv6 spd queue min-threshold** *value*
5. exit
6. **show ipv6 spd**

### DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ipv6 spd queue max-threshold** *value*<br><br>**Example:**<br>`Router(config)# ipv6 spd queue max-threshold 60000` | Configures the maximum number of packets in the SPD process input queue. |
| **Step 4** | **ipv6 spd queue min-threshold** *value*<br><br>**Example:**<br>`Router(config)# ipv6 spd queue max-threshold 4094` | Configures the minimum number of packets in the IPv6 SPD process input queue. |
| **Step 5** | exit<br><br>**Example:**<br>`Router(config)# exit` | Returns the router to privileged EXEC mode. |

| | Command or Action | Purpose |
|---|---|---|
| **Step 6** | **show ipv6 spd**<br><br>**Example:**<br>`Router# show ipv6 spd` | Displays IPv6 SPD configuration. |

# Configuring an SPD Mode

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 spd mode** {**aggressive** | **tos protocol ospf**}

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br><br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br><br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **ipv6 spd mode** {**aggressive** | **tos protocol ospf**}<br><br>**Example:**<br>`Router(config)# ipv6 spf mode aggressive` | Configures an IPv6 SPD mode. |

# Configuring SPD Headroom

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **spd headroom** *size*
4. **spd extended-headroom** *size*
5. exit
6. **show ipv6 spd**

## DETAILED STEPS

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | **enable**<br>**Example:**<br>`Router> enable` | Enables privileged EXEC mode.<br>• Enter your password if prompted. |
| **Step 2** | **configure terminal**<br>**Example:**<br>`Router# configure terminal` | Enters global configuration mode. |
| **Step 3** | **spd headroom** *size*<br>**Example:**<br>`Router(config)# spd headroom 200` | Configures SPD headroom. |
| **Step 4** | **spd extended-headroom** *size*<br>**Example:**<br>`Router(config)# spd extended-headroom 11` | Configures extended SPD headroom. |
| **Step 5** | exit<br>**Example:**<br>`Router(config)# exit` | Returns the router to privileged EXEC mode. |
| **Step 6** | **show ipv6 spd**<br>**Example:**<br>`Router# show ipv6 spd` | Displays the IPv6 SPD configuration. |

# Configuration Examples for IPv6 Selective Packet Discard

## Example: Configuring the SPD Process Input Queue

The following example shows the SPD process input queue configuration. The maximum process input queue threshold is 60,000, and the SPD state is normal. The headroom and extended headroom values are the default:

```
Router# ipv6 spd queue max-threshold 5000
Router# show ipv6 spd

Current mode: normal
Queue max threshold: 60000, Headroom: 100, Extended Headroom: 10
IPv6 packet queue: 0
```

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| IPv6 addressing and connectivity | *IPv6 Configuration Guide* |
| Cisco IOS commands | Master Commands List, All Releases |
| IPv6 commands | IPv6 Command Reference |
| Cisco IOS IPv6 features | IPv6 Feature Mapping |

### Standards and RFCs

| Standard/RFC | Title |
|---|---|
| RFCs for IPv6 | *IPv6 RFCs* |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |

# Feature Information for IPv6 Selective Packet Discard

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

***Table 14: Feature Information for IPv6 Selective Packet Discard***

| Feature Name | Releases | Feature Information |
|---|---|---|
| IPv6: Full Selective Packet Discard Support | Cisco IOS XE Release 2.6 | The SPD mechanism manages the process level input queues on the RP. SPD provides priority to routing protocol packets and other important traffic control Layer 2 keepalives during periods of process level queue congestion.<br>The following commands were introduced or modified: **clear ipv6 spd**, **debug ipv6 spd**, **ipv6 spd mode**, **ipv6 spd queue max-threshold**, **ipv6 spd queue min-threshold**, **monitor event-trace ipv6 spd**, **show ipv6 spd**, **spd extended-headroom**, **spd headroom**. |