



Class-Based Policing

Feature History

Release	Modification
12.0(5)XE	This feature was introduced.
12.1(5)T	This command was introduced for Cisco IOS Release 12.1 T. A new Class-Based Policing algorithm was introduced. The violate-action option became available. This feature became available on Cisco 2600, 3600, 4500, 7200, and 7500 series routers.
12.2(2)T	The <code>set-clp-transmit</code> option for the <i>action</i> argument was added to the police command. The <code>set-frde-transmit</code> option for the <i>action</i> argument was added to the police command. The <code>set-mpls-exp-transmit</code> option for the <i>action</i> argument was added to the police command.
12.0(26)S	This feature was integrated into Cisco IOS Release 12.0(26)S for the Cisco 7200 and 7500 series routers. The name of the feature changed from <i>Traffic Policing</i> to <i>Class-Based Policing</i> .

Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

Contents

- [Feature Overview, page 2](#)
- [Prerequisites, page 3](#)
- [Configuration Tasks, page 3](#)
- [Monitoring and Maintaining Traffic Policing, page 5](#)
- [Configuration Examples, page 5](#)
- [Additional References, page 7](#)
- [Command Reference, page 8](#)
- [Glossary, page 15](#)

Feature Overview

This feature module describes the Class-Based Policing feature. It includes information on the benefits of the feature, supported platforms, related documents, and so forth.

The Class-Based Policing feature performs the following functions:

- Limits the input or output transmission rate of a class of traffic based on user-defined criteria
- Marks packets by setting the ATM Cell Loss Priority (CLP) bit, Frame Relay Discard Eligibility (DE) bit, IP precedence value, IP differentiated services code point (DSCP) value, MPLS experimental value, and Quality of Service (QoS) group.

Class-Based Policing allows you to control the maximum rate of traffic transmitted or received on an interface. The Class-Based Policing feature is applied when you attach a traffic policy contain the Class-Based Policing configuration to an interface. A traffic policy is configured using the Modular Quality of Service Command-Line Interface (Modular QoS CLI). For information on configuring the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface Overview* on Cisco Connection Online (CCO) and the Documentation CD-ROM.

Benefits

Bandwidth Management Through Rate Limiting

Class-Based Policing allows you to control the maximum rate of traffic transmitted or received on an interface. Class-Based Policing is often configured on interfaces at the edge of a network to limit traffic into or out of the network. In most Class-Based Policing configurations, traffic that falls within the rate parameters is transmitted, whereas traffic that exceeds the parameters is dropped or transmitted with a different priority.

Packet Marking

Packet marking allows you to partition your network into multiple priority levels or classes of service (CoS). A packet is marked and these markings can be used to identify and classify traffic for downstream devices. In some cases, such as ATM Cell Loss Priority (CLP) marking or Frame Relay Discard Eligibility (DE) marking, the marking is used to classify traffic.

- Use Class-Based Policing to set the IP precedence or DSCP values for packets entering the network. Networking devices within your network can then use the adjusted IP precedence values to determine how the traffic should be treated. For example, the Weighted Random Early Detection (WRED) feature uses the IP precedence values to determine the probability that a packet will be dropped.
- Use Class-Based Policing to assign packets to a QoS group. The router uses the QoS group to determine how to prioritize packets within the router.

Traffic can be marked without using the Class-Based Policing feature. If you want to mark traffic but do not want to use Traffic Policing, see the *Class-Based Marking* feature module.

Packet Prioritization for Frame Relay Frames

The Class-Based Policing feature allows users to mark the Frame Relay DE bit of the Frame Relay frame. The Frame Relay DE bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, frames with the DE bit set to 1 are discarded before frames with the DE bit set to 0.

Packet Prioritization for ATM Cells

The Class-Based Policing feature allows users to mark the ATM CLP bit in ATM cells. The ATM CLP bit is used to prioritize packets in ATM networks. The ATM CLP bit is one bit and, therefore, can be set to either 0 or 1. In congested environments, cells with the ATM CLP bit set to 1 are discarded before cells with the ATM CLP bit set to 0.

Restrictions

- To use the *set-clp-transmit* action available with this feature, the Enhanced ATM Port Adapter (PA-A3) is required. Therefore, the *set-clp-transmit* action is not supported on any platform that does not support the PA-A3 adapter (such as the Cisco 2600 series router, the Cisco 3640 router, and the 4500 series router). For more information, refer to the documentation for your specific router.
- On a Cisco 7500 series router, Class-Based Policing can monitor Cisco Express Forwarding (CEF) switching paths only. In order to use the Class-Based Policing feature, Cisco Express Forwarding must be configured on both the interface receiving the packet and the interface sending the packet.
- On a Cisco 7500 series router, Class-Based Policing cannot be applied to packets that originated from or are destined to a router.
- Class-Based Policing can be configured on an interface or a subinterface.
- Class-Based Policing is not supported on the following interfaces:
 - Fast EtherChannel
 - Tunnel



Note

Class-Based Policing is supported on tunnels that are using the Cisco generic routing encapsulation (GRE) tunneling protocol.

- PRI
- Any interface on a Cisco 7500 series router that does not support Cisco Express Forwarding

Prerequisites

On a Cisco 7500 series router, Cisco Express Forwarding (CEF) must be configured on the interface before Class-Based Policing can be used.

For additional information on Cisco Express Forwarding, see the *Cisco Express Forwarding* and *Distributed Cisco Express Forwarding* feature modules.

Configuration Tasks

See the following sections for configuration tasks for the Class-Based Policing feature. Each task in the list indicates if the task is optional or required.

- [Configuring Traffic Policing](#) (Required)
- [Verifying Traffic Policing](#) (Optional)

Configuring Traffic Policing

To successfully configure the Class-Based Policing feature, a traffic class and a traffic policy must be created, and the traffic policy must be attached to a specified interface. These tasks are performed using the Modular QoS CLI. For information on the Modular QoS CLI, see the *Modular Quality of Service Command-Line Interface* document on CCO or the Documentation CD-ROM.

The Class-Based Policing feature is configured in the traffic policy. To configure the Class-Based Policing feature, use the following command in policy map configuration mode:

Command	Purpose
Router(config-pmap-c)# police <i>bps burst-normal burst-max conform-action action exceed-action action violate-action action</i>	Specifies a maximum bandwidth usage by a traffic class.

The Class-Based Policing feature works with a token bucket mechanism. There are currently two types of token bucket algorithms: a single token bucket algorithm and a two token bucket algorithm. A single token bucket system is used when the **violate-action** option is not specified, and a two token bucket system is used when the **violate-action** option is specified.

For a description of a single token bucket algorithm and an explanation of how it works, see the “What is a Token Bucket?” section of the *Policing and Shaping Overview* document. An example of how the single token bucket algorithm works is also given in the [Command Reference](#) section of this document.

For a description of the two token bucket algorithm and an explanation of how it works, see the “[Command Reference](#)” section of this document.

Verifying Traffic Policing

Use the **show policy-map interface EXEC** command to verify that the Class-Based Policing feature is configured on your interface. If the feature is configured on your interface, the **show policy-map interface** command output displays policing statistics:

```
Router# show policy-map interface
Ethernet1/7
  service-policy output: x
    class-map: a (match-all)
      0 packets, 0 bytes
      5 minute rate 0 bps
      match: ip precedence 0
      police:
        1000000 bps, 10000 limit, 10000 extended limit
        conformed 0 packets, 0 bytes; action: transmit
        exceeded 0 packets, 0 bytes; action: drop
        conformed 0 bps, exceed 0 bps, violate 0 bps
```

Troubleshooting Tips

- Check the interface type. Verify that your interface is not mentioned in the nonsupported interface description in the “[Restrictions](#)” section of this document.
- For input Class-Based Policing on a Cisco 7500 series router, verify that CEF is configured on the interface where Class-Based Policing is configured.

- For output Class-Based Policing on a Cisco 7500 series router, ensure that the incoming traffic is CEF-switched. Class-Based Policing cannot be used on the switching path unless CEF switching is enabled.

Monitoring and Maintaining Traffic Policing

To monitor and maintain the Class-Based Policing feature, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# <code>show policy-map</code>	Displays all configured policy maps.
Router# <code>show policy-map policy-map-name</code>	Displays the user-specified policy map.
Router# <code>show policy-map interface</code>	Displays statistics and configurations of all input and output policies that are attached to an interface.

Configuration Examples

This section provides the following configuration example:

- [Configuring a Service Policy that Includes Traffic Policing](#)

Configuring a Service Policy that Includes Traffic Policing

In the following example, Class-Based Policing is configured with the average rate at 8000 bits per second, the normal burst size at 1000 bytes, and the excess burst size at 1000 bytes for all packets leaving Fast Ethernet interface 0/0.

For additional information on configuring traffic classes and traffic policies, see the *Modular Quality of Service Command-Line Interface* document on CCO and the Documentation CD-ROM.

For a description of a token bucket and an explanation of how a token bucket works, see the “What is a Token Bucket?” section of the *Policing and Shaping Overview* document. An example of how the token bucket works is also given in the [Command Reference](#) section of this document.

```
class-map access-match
  match access-group 1
  exit
policy-map police-setting
  class access-match
    police 8000 1000 1000 conform-action transmit exceed-action set-qos-transmit 1
    violate-action drop
  exit
exit
service-policy output police-setting
```

The treatment of a series of packets leaving Fast Ethernet interface 0/0 depends on the size of the packet and the number of bytes remaining in the conform and exceed token buckets. The series of packets are policed based on the following rules:

- If the previous arrival of the packet was at T1 and the current arrival of the packet is at T, the bucket is updated with $T - T1$ worth of bits based on the token arrival rate. The refill tokens are placed in the conform bucket. If the tokens overflow the conform bucket, the overflow tokens are placed in the exceed bucket. The token arrival rate is calculated as follows:
(time between packets <which is equal to $T - T1$ > * policer rate)/8 bytes
- If the number of bytes in the conform bucket B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is taken. The exceed bucket is unaffected in this scenario.
- If the number of bytes in the conform bucket B is less than 0, the excess token bucket is checked for bytes by the packet. If the number of bytes in the exceed bucket B is greater than or equal to 0, the exceed action is taken and B bytes are removed from the exceed token bucket. No bytes are removed from the conform bucket in this scenario.
- If the number bytes in the exceed bucket B is fewer than 0, the packet violates the rate and the violate action is taken. The action is complete for the packet.

In this example, the initial token buckets starts full at 1000 bytes. If a 450-byte packet arrives, the packet conforms because enough bytes are available in the conform token bucket. The conform action (send) is taken by the packet and 450 bytes are removed from the conform token bucket (leaving 550 bytes).

If the next packet arrives 0.25 seconds later, 250 bytes are added to the conform token bucket

((0.25 * 8000)/8), leaving 800 bytes in the conform token bucket. If the next packet is 900 bytes, the packet does not conform because only 800 bytes are available in the conform token bucket.

The exceed token bucket, which starts full at 1000 bytes (as specified by the excess burst size) is then checked for available bytes. Because enough bytes are available in the exceed token bucket, the exceed action (set the QoS transmit value of 1) is taken and 900 bytes are taken from the exceed bucket (leaving 100 bytes in the exceed token bucket).

If the next packet arrives 0.40 seconds later, 400 bytes are added to the token buckets ((.40 * 8000)/8). Therefore, the conform token bucket now has 1000 bytes (the maximum number of tokens available in the conform bucket) and 200 bytes overflow the conform token bucket (because it only 200 bytes were needed to fill the conform token bucket to capacity). These overflow bytes are placed in the exceed token bucket, giving the exceed token bucket 300 bytes.

If the arriving packet is 1000 bytes, the packet conforms because enough bytes are available in the conform token bucket. The conform action (transmit) is taken by the packet, and 1000 bytes are removed from the conform token bucket (leaving 0 bytes).

If the next packet arrives 0.20 seconds later, 200 bytes are added to the token bucket ((.20 * 8000)/8). Therefore, the conform bucket now has 200 bytes. If the arriving packet is 400 bytes, the packet does not conform because only 200 bytes are available in the conform bucket. Similarly, the packet does not exceed because only 300 bytes are available in the exceed bucket. Therefore, the packet violates and the violate action (drop) is taken.

Additional References

The following sections provide references related to Traffic Policing.

Related Documents

Related Topic	Document Title
Traffic Policing	<ul style="list-style-type: none"> • <i>Policer Enhancement: Multiple Actions</i> • <i>QoS: Color-Aware Policer</i> • <i>Two-Rate Policer</i>
Modular Quality of Service	<i>Modular Quality of Service Command Line Interface</i>

Standards

Standards	Title
None	—

MIBs

MIBs	MIBs Link
<i>Class-Based Quality of Service MIB</i> <ul style="list-style-type: none"> • CISCO-CLASS-BASED-QOS-MIB • CISCO-CLASS-BASED-QOS-CAPABILITY-MIB 	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 2697	<i>A Single Rate Three Color Marker</i>

Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/public/support/tac/home.shtml

Command Reference

This section documents new or modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.3 command reference publications.

- [police](#)

police

To configure traffic policing, use the **police** command in policy-map class configuration mode or policy-map class police configuration mode. To remove Class-Based Policing from the configuration, use the **no** form of this command.

```
police bps [burst-normal] [burst-max] conform-action action exceed-action action
[violate-action action]
```

```
no police bps [burst-normal] [burst-max] conform-action action exceed-action action
[violate-action action]
```

Syntax Description

<i>bps</i>	Average rate in bits per second. Valid values are 8000 to 200000000.
<i>burst-normal</i>	(Optional) Normal burst size in bytes. Valid values are 1000 to 51200000. The default normal burst size is 1500 bytes.
<i>burst-max</i>	(Optional) Excess burst size in bytes. Valid values are 1,000 to 51200000.
conform-action <i>action</i>	Action to take on packets that conform to the rate limit.
exceed-action <i>action</i>	Action to take on packets that exceed the rate limit.
violate-action <i>action</i>	(Optional) Action to take on packets that violate the normal and maximum burst sizes.
<i>action</i>	Action to take on packets. Specify one of the following keywords: <ul style="list-style-type: none"> • drop—Drops the packet. • set-clp-transmit <i>value</i>—Sets the ATM Cell Loss Priority (CLP) bit from 0 to 1 on the ATM cell and transmits the packet with the ATM CLP bit set to 1. • set-discard-class-transmit—Sets the discard class attribute of a packet and transmits the packet with the new discard class setting. • set-dscp-transmit <i>value</i>—Sets the IP differentiated services code point (DSCP) value and transmits the packet with the new IP DSCP value setting. • set-frde-transmit <i>value</i>—Sets the Frame Relay Discard Eligibility (DE) bit from 0 to 1 on the frame relay frame and transmits the packet with the DE bit set to 1. • set-mpls-experimental-imposition-transmit <i>value</i>—Sets the Multiprotocol Label Switching (MPLS) experimental (EXP) bits (0 to 7) in the imposed label headers and transmits the packet with the new MPLS EXP bit value setting. • set-prec-transmit <i>value</i>—Sets the IP precedence and transmits the packet with the new IP precedence value setting. • set-qos-transmit <i>value</i>—Sets the qos-group value and transmits the packet with the new qos-group value setting. • transmit—Transmits the packet. The packet is not altered.

Defaults Disabled

Command Modes Policy-map class configuration (when specifying a single action to be applied to a marked packet)
Policy-map class police configuration (when specifying multiple actions to be applied to a marked packet)

Release	Modification
12.0(5)XE	This police command was introduced.
12.1(1)E	This command was integrated in Cisco IOS Release 12.1(1)E.
12.1(5)T	This command was integrated in Cisco IOS Release 12.1(5)T. The violate-action keyword was added.
12.2(2)T	The set-clp-transmit option for the <i>action</i> argument was added. The set-frde-transmit keyword for the <i>action</i> argument was added. The set-mpls-exp-transmit keyword for the <i>action</i> argument was added.
12.2(8)T	The command was modified for the Policer Enhancement — Multiple Actions feature. This command can now accommodate multiple actions for packets marked as conforming to, exceeding, or violating a specific rate.
12.2(13)T	In the <i>action</i> argument, the set-mpls-experimental-transmit keyword was renamed to set-mpls-experimental-imposition-transmit .
12.0(26)S	This command was integrated into Cisco IOS Release 12.0(26)S.

Usage Guidelines Use the **police** command to mark a packet with different quality of service (QoS) values based on conformance to the service-level agreement.

Class-Based Policing will not be executed for traffic that passes through an interface.

Specifying Multiple Actions

The **police** command allows you to specify multiple policing actions. When specifying multiple policing actions when configuring the **police** command, note the following points:

- You can specify a maximum of four actions at one time.
- You cannot specify contradictory actions such as **conform-action transmit** and **conform-action drop**.

Using the Police Command with the Class-Based Policing Feature

The **police** command can be used with the Class-Based Policing feature. The Class-Based Policing feature works with a token bucket algorithm. Two types of token bucket algorithms are in Cisco IOS Release 12.1(5)T: a single-token bucket algorithm and a two-token bucket algorithm. A single-token bucket system is used when the **violate-action** option is not specified, and a two-token bucket system is used when the **violate-action** option is specified.

The token bucket algorithm for the **police** command that was introduced in Cisco IOS Release 12.0(5)XE is different from the token bucket algorithm for the **police** command introduced in Cisco IOS Release 12.1(5)T. For information on the token bucket algorithm introduced in

Release 12.0(5)XE, refer to the *Traffic Policing* document for Release 12.0(5)XE. This document is available on the *New Features for 12.0(5)XE* feature documentation index (under Modular QoS CLI-related feature modules) at www.cisco.com.

The following are explanations of how the token bucket algorithms introduced in Cisco IOS Release 12.1(5)T work.

Token Bucket Algorithm with One Token Bucket

The one token bucket algorithm is used when the **violate-action** option is not specified in the **police** command command-line interface (CLI).

The conform bucket is initially set to the full size (the full size is the number of bytes specified as the normal burst size).

When a packet of a given size (for example, “B” bytes) arrives at specific time (time “T”) the following actions occur:

- Tokens are updated in the conform bucket. If the previous arrival of the packet was at T1 and the current time is T, the bucket is updated with (T - T1) worth of bits based on the token arrival rate. The token arrival rate is calculated as follows:
$$(\text{time between packets} <\text{which is equal to } T - T1 > * \text{policer rate})/8 \text{ bytes}$$
- If the number of bytes in the conform bucket B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is completed for the packet.
- If the number of bytes in the conform bucket B is fewer than 0, the exceed action is taken.

Token Bucket Algorithm with Two Token Buckets

The two-token bucket algorithm is used when the **violate-action** option is specified in the **police** command CLI.

The conform bucket is initially full (the full size is the number of bytes specified as the normal burst size).

The exceed bucket is initially full (the full exceed bucket size is the number of bytes specified in the maximum burst size).

The tokens for both the conform and exceed token buckets are updated based on the token arrival rate, or committed information rate (CIR).

When a packet of given size (for example, “B” bytes) arrives at specific time (time “T”) the following actions occur:

- Tokens are updated in the conform bucket. If the previous arrival of the packet was at T1 and the current arrival of the packet is at t, the bucket is updated with T - T1 worth of bits based on the token arrival rate. The refill tokens are placed in the conform bucket. If the tokens overflow the conform bucket, the overflow tokens are placed in the exceed bucket.

The token arrival rate is calculated as follows:

$$(\text{time between packets} <\text{which is equal to } T - T1 > * \text{policer rate})/8 \text{ bytes}$$

- If the number of bytes in the conform bucket - B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is taken. The exceed bucket is unaffected in this scenario.

- If the number of bytes in the conform bucket B is less than 0, the excess token bucket is checked for bytes by the packet. If the number of bytes in the exceed bucket B is greater than or equal to 0, the exceed action is taken and B bytes are removed from the exceed token bucket. No bytes are removed from the conform bucket.
- If the number bytes in the exceed bucket B is fewer than 0, the packet violates the rate and the violate action is taken. The action is complete for the packet.

Examples

Token Bucket Algorithm with One Token Bucket Example

The token bucket algorithm for the **police** command that was introduced in Cisco IOS Release 12.0(5)XE is different from the token bucket algorithms introduced in Cisco IOS Release 12.1(5)T. The following example is for the token bucket algorithm with one token bucket introduced in Cisco IOS Release 12.1(5)T.

If the **violate-action** option is not specified when you configure a policy with the **police** command in Cisco IOS Release 12.1(5)T onward, the token bucket algorithm uses one token bucket. If the **violate-action** option is specified, the token bucket algorithm uses two token buckets. In the following example, the **violate-action** option is not specified, so the token bucket algorithm only uses one token bucket.

The following configuration shows users how to define a traffic class (using the **class-map** command) and associate the match criteria from the traffic class with the traffic policing configuration, which is configured in the service policy (using the **policy-map** command). The **service-policy** command is then used to attach this service policy to the interface.

In this particular example, Class-Based Policing is configured with the average rate at 8000 bits per second and the normal burst size at 1000 bytes for all packets leaving Fast Ethernet interface 0/0:

```
Router(config)# class-map access-match
Router(config-cmap)# match access-group 1
Router(config-cmap)# exit
Router(config)# policy-map police-setting
Router(config-pmap)# class access-match
Router(config-pmap-c)# police 8000 1000 conform-action transmit exceed-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fastethernet 0/0
Router(config-if)# service-policy output police-setting
```

The treatment of a series of packets leaving Fast Ethernet interface 0/0 depends on the size of the packet and the number of bytes remaining in the conform bucket. These packets are policed based on the following rules:

- Tokens are updated in the conform bucket. If the previous arrival of the packet was at t_1 and the current time is t , the bucket is updated with $T - T_1$ worth of bits based on the token arrival rate. The token arrival rate is calculated as follows:
(time between packets <which is equal to $T - T_1$ > * policer rate)/8 bytes
- If the number of bytes in the conform bucket B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is completed for the packet.
- If the number of bytes in the conform bucket B is fewer than 0, the exceed action is taken.

In this example, the initial token buckets starts full at 1000 bytes. If a 450-byte packet arrives, the packet conforms because enough bytes are available in the conform token bucket. The conform action (send) is taken by the packet and 450 bytes are removed from the conform token bucket (leaving 550 bytes).

If the next packet arrives 0.25 seconds later, 250 bytes are added to the token bucket $((0.25 * 8000)/8)$, leaving 800 bytes in the token bucket. If the next packet is 900 bytes, the packet exceeds and the exceed action (drop) is taken. No bytes are taken from the token bucket.

Token Bucket Algorithm with Two Token Buckets Example

If the **violate-action** option is specified when you configure a policy with the **police** command in Cisco IOS Release 12.1(5)T onward, the token bucket algorithm uses two token buckets. The following example uses the token bucket algorithm with two token buckets.

The following configuration shows users how to define a traffic class (using the **class-map** command) and associate the match criteria from the traffic class with the Class-Based Policing configuration, which is configured in the service policy (using the **policy-map** command). The **service-policy** command is then used to attach this service policy to the interface.

In this particular example, Class-Based Policing is configured with the average rate at 8000 bits per second, the normal burst size at 1000 bytes, and the excess burst size at 1000 bytes for all packets leaving Fast Ethernet interface 0/0.

```
Router(config)# class-map access-match
Router(config-cmap)# match access-group 1
Router(config-cmap)# exit
Router(config)# policy-map police-setting
Router(config-pmap)# class access-match
Router(config-pmap-c)# police 8000 1000 1000 conform-action transmit exceed-action
set-qos-transmit 1 violate-action drop
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fastethernet 0/0
Router(config-if)# service-policy output police-setting
```

The treatment of a series of packets leaving Fast Ethernet interface 0/0 depends on the size of the packet and the number of bytes remaining in the conform and exceed token buckets. The series of packets are policed based on the following rules:

- If the previous arrival of the packet was at T1 and the current arrival of the packet is at T, the bucket is updated with $T - T1$ worth of bits based on the token arrival rate. The refill tokens are placed in the conform bucket. If the tokens overflow the conform bucket, the overflow tokens are placed in the exceed bucket. The token arrival rate is calculated as follows:
(time between packets <which is equal to $T - T1$ > * policer rate)/8 bytes
- If the number of bytes in the conform bucket B is greater than or equal to 0, the packet conforms and the conform action is taken on the packet. If the packet conforms, B bytes are removed from the conform bucket and the conform action is taken. The exceed bucket is unaffected in this scenario.
- If the number of bytes in the conform bucket B is less than 0, the excess token bucket is checked for bytes by the packet. If the number of bytes in the exceed bucket B is greater than or equal to 0, the exceed action is taken and B bytes are removed from the exceed token bucket. No bytes are removed from the conform bucket in this scenario.
- If the number bytes in the exceed bucket B is fewer than 0, the packet violates the rate and the violate action is taken. The action is complete for the packet.

In this example, the initial token buckets starts full at 1000 bytes. If a 450-byte packet arrives, the packet conforms because enough bytes are available in the conform token bucket. The conform action (send) is taken by the packet and 450 bytes are removed from the conform token bucket (leaving 550 bytes).

If the next packet arrives 0.25 seconds later, 250 bytes are added to the conform token bucket $((0.25 * 8000)/8)$, leaving 800 bytes in the conform token bucket. If the next packet is 900 bytes, the packet does not conform because only 800 bytes are available in the conform token bucket.

The exceed token bucket, which starts full at 1000 bytes (as specified by the excess burst size) is then checked for available bytes. Because enough bytes are available in the exceed token bucket, the exceed action (set the QoS transmit value of 1) is taken and 900 bytes are taken from the exceed bucket (leaving 100 bytes in the exceed token bucket).

If the next packet arrives 0.40 seconds later, 400 bytes are added to the token buckets $((.40 * 8000)/8)$. Therefore, the conform token bucket now has 1000 bytes (the maximum number of tokens available in the conform bucket) and 200 bytes overflow the conform token bucket (because it only 200 bytes were needed to fill the conform token bucket to capacity). These overflow bytes are placed in the exceed token bucket, giving the exceed token bucket 300 bytes.

If the arriving packet is 1000 bytes, the packet conforms because enough bytes are available in the conform token bucket. The conform action (transmit) is taken by the packet, and 1000 bytes are removed from the conform token bucket (leaving 0 bytes).

If the next packet arrives 0.20 seconds later, 200 bytes are added to the token bucket $((.20 * 8000)/8)$. Therefore, the conform bucket now has 200 bytes. If the arriving packet is 400 bytes, the packet does not conform because only 200 bytes are available in the conform bucket. Similarly, the packet does not exceed because only 300 bytes are available in the exceed bucket. Therefore, the packet violates and the violate action (drop) is taken.

Conforming to the MPLS EXP Value Example

The following example shows that if packets conform to the rate limit, the MPLS EXP field is set to 5. If packets exceed the rate limit, the MPLS EXP field is set to 3.

```
policy-map input-IP-dscp
  class dscp24
    police 8000 1500 1000
      conform-action set-mpls-experimental-imposition-transmit 5
      exceed-action set-mpls-experimental-imposition-transmit 3
      violate-action drop
```

Related Commands

Command	Description
policy-map	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
service-policy	Specifies the name of the service policy to be attached to the interface.
show policy-map	Displays the configuration of all classes for a specified service policy map or all classes for all existing policy maps.
show policy-map interface	Displays the configuration of all classes configured for all service policies on the specified interface or displays the classes for the service policy for a specific PVC on the interface.

Glossary

average rate—Maximum long-term average rate of conforming traffic.

conform action—Action to take on packets with a burst size below the rate allowed by the rate limit.

DSCP—differentiated services code point

exceed action—Action to take on packets that exceed the rate limit.

excess burst size—Bytes allowed in a burst before all packets will exceed the rate limit.

normal burst size—Bytes allowed in a burst before some packets will exceed the rate limit. Larger bursts are more likely to exceed the rate limit.

QoS group—Internal QoS group ID for a packet used to determine weighted fair queuing characteristics for that packet.

policing policy—Rate limit, conform actions, and exceed actions that apply to traffic matching a certain criteria.

Versatile Interface Processor (VIP)—Interface card used by Cisco 7500 series and Cisco 7000 series with RSP7000 routers.

CCIP, CCSP, the Cisco Arrow logo, the Cisco *Powered* Network mark, Cisco Unity, Follow Me Browsing, FormShare, and StackWise are trademarks of Cisco Systems, Inc.; Changing the Way We Work, Live, Play, and Learn, and iQuick Study are service marks of Cisco Systems, Inc.; and Aironet, ASIST, BPX, Catalyst, CCDA, CCDP, CCIE, CCNA, CCNP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS, the Cisco IOS logo, Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Empowering the Internet Generation, Enterprise/Solver, EtherChannel, EtherSwitch, Fast Step, GigaStack, Internet Quotient, IOS, IP/TV, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, LightStream, MGX, MICA, the Networkers logo, Networking Academy, Network Registrar, *Packet*, PIX, Post-Routing, Pre-Routing, RateMUX, Registrar, ScriptShare, SlideCast, SMARTnet, StrataView Plus, Stratm, SwitchProbe, TeleRouter, The Fastest Way to Increase Your Internet Quotient, TransPath, and VCO are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and certain other countries.

All other trademarks mentioned in this document or Web site are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (0304R)