match ip precedence

To identify IP precedence values as match criteria, use the **match ip precedence** command in class-map configuration mode. To remove IP precedence values from a class map, use the **no** form of this command.

match ip precedence *ip-precedence-value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]

no match ip precedence *ip-precedence value* [*ip-precedence-value ip-precedence-value ip-precedence-value*]

Syntax Description

ip-precedence-value	Specifies the exact value from 0 to 7 used to identify an IP precedence value.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

Up to four precedence values can be matched in one match statement. For example, if you wanted the IP precedence values of 0, 1, 2, or 3 (note that only one of the IP precedence values must be a successful match criterion, not all of the specified IP precedence values), enter the **match ip precedence 0 1 2 3** command.

The *ip-precedence-value* arguments are used as markings only. The IP precedence values have no mathematical significance. For instance, the *ip-precedence-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *ip-precedence-value* of 2 is different than a packet marked with the *ip-precedence-value* of 1. The treatment of these different packets is defined by the user through the setting of QoS policies in policy-map class configuration mode.

Examples

The following example shows how to configure the service policy called priority50 and attach service policy priority50 to an interface. In this example, the class map called ipprec5 will evaluate all packets entering Fast Ethernet interface 1/0/0 for an IP precedence value of 5. If the incoming packet has been marked with the IP precedence value of 5, the packet will be treated with a priority level of 50.

```
Router(config)# class-map ipprec5
Router(config-cmap)# match ip precedence 5
Router(config)# exit
Router(config)# policy-map priority50
Router(config-pmap)# class ipprec5
Router(config-pmap-c)# priority 50
Router(config-pmap-c)# exit
```

```
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy input priority50
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.
policy-map	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
set ip precedence	Sets the precedence value in the IP header.
service-policy	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
show class-map	Displays all class maps and their matching criteria, or a specified class map and its matching criteria.

match ip rtp

To configure a class map to use the Real-Time Protocol (RTP) protocol port as the match criterion, use the **match ip rtp** class-map configuration command. To remove the RTP protocol port match criterion, use the **no** form of this command.

match ip rtp *starting-port-number port-range*

no match ip rtp

Syntax Description

starting-port-number	The starting RTP port number. Values range from 2000 to 65535.
port-range	The RTP port number range. Values range from 0 to 16383.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.1(2)T	This command was introduced.

Usage Guidelines

This command is used to match IP RTP packets that fall within the specified port range. It matches packets destined to all even User Datagram Port (UDP) port numbers in the range <starting port range> <starting port range + port range>.

Use of an RTP port range as the match criterion is particularly effective for applications that use RTP, such as voice or video.

Examples

The following example specifies a class map called eth1 and configures the RTP port number 2024 and range 1000 to be used as the match criteria for this class:

```
class-map eth1
 match ip rtp 2024 1000
```

Related Commands

Command	Description
ip rtp priority	Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.
match access-group	Configures the match criteria for a class map based on the specified ACL number.

match mpls experimental

To configure a class map to use the specified value of the EXP field as a match criterion, use the **match mpls experimental** class-map configuration command. To remove the EXP field match criterion from a class map, use the **no** form of this command.

match mpls experimental *number*

no match mpls experimental *number*

Syntax Description

number	The EXP field value to be used as match criteria. Any number from 0 to 7.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(7)XE1	This command was introduced.
12.0(7)T	This command was integrated into Cisco IOS Release 12.0(7)T.
12.1(1)T	This command was integrated into Cisco IOS Release 12.1(1)T.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

For class-based weighted fair queueing (CBWFQ), you define traffic classes based on match criteria including input interfaces, access control lists (ACLs), protocols, QoS labels, and EXP field values. Packets satisfying the match criteria for a class constitute the traffic for that class.

The **match mpls experimental** command specifies the name of an EXP field value to be used as the match criterion against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match mpls experimental** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use one of the following commands to configure its match criteria:

- **match access-group**
- **match input-interface**
- **match mpls experimental**
- **match protocol**

If you specify more than one command in a class map, only the last command entered applies. The last command overrides the previously entered commands.

Examples

The following example specifies a class map called eth1 and configures the EXP field value 0 to be used as the match criterion for this class:

```
class-map eth1
 match mpls experimental 1
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.
match access-group	Configures the match criteria for a class map based on the specified ACL.
match input-interface	Configures a class map to use the specified input interface as a match criterion.
match qos-group	Configures the match criteria for a class map based on the specified protocol.

match not

To specify the single match criterion value to use as an unsuccessful match criterion, use the **match not** class-map configuration command. To remove a previously specified source value to not use as a match criterion, use the **no** form of this command.

match not *match-criteria*

no match not *match-criteria*

Syntax Description

match-criteria	(Required) Specifies the match criterion value that is an unsuccessful match criterion. All other values of the specified match criterion will be considered successful match criteria.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.
12.0(5)T	This command was integrated into Cisco IOS Release 12.0(5)T.
12.1(1)E	This command was integrated into Cisco IOS Release 12.1(1)E.

Usage Guidelines

The **match not** command is used to specify a QoS policy value that is not used as a match criterion. When the **match not** command is used, all other values of that QoS policy become successful match criteria.

For instance, if the **match not qos-group 4** command is issued in class-map configuration mode, the specified class will accept all QoS group values except 4 as successful match criteria.

Examples

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)# exit
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.

match protocol

To configure the match criteria for a class map on the basis of the specified protocol, use the **match protocol** class-map configuration command. To remove protocol-based match criteria from a class map, use the **no** form of this command.

match protocol *protocol-name*

no match protocol *protocol-name*

Syntax Description

protocol-name	Name of the protocol used as a matching criterion. Supported protocols include the following: • **aarp**—AppleTalk Address Resolution Protocol • **apollo**—Apollo Domain • **arp**—IP Address Resolution Protocol (ARP) • **bridge**—bridging • **bstun**—Block Serial Tunneling • **cdp**—Cisco Discovery Protocol • **clns**—ISO Connectionless Network Service • **clns_es**—ISO CLNS End System • **clns_is**—ISO CLNS Intermediate System • **cmns**—ISO Connection-Mode Network Service • **compressedtcp**—compressed TCP • **decnet**—DECnet • **decnet_node**—DECnet Node • **decnet_router-I1**—DECnet Router L1 • **decnet_router-I2**—DECnet Router L2 • **dlsw**—data-link switching • **ip**—IP • **ipx**—Novell IPX • **llc2**—llc2 • **pad**—packet assembler/disassembler links • **qllc**—Qualified Logical Link Control protocol • **rsrb**—remote source-route bridging • **snapshot**—snapshot routing support • **stun**—serial tunnel • **vines**—Banyan VINES • **xns**—Xerox Network Services

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.
12.0(5)XE	This command was integrated into Cisco IOS Release 12.0(5)XE.
12.0(7)S	This command was integrated into Cisco IOS Release 12.0(7)S.
12.1(1)E	This command was integrated into Cisco IOS Release 12.1(1)E.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

For class-based weighted fair queueing (CBWFQ), you define traffic classes based on match criteria including protocols, access control lists (ACLs), input interfaces, QoS labels, and EXP field values. Packets satisfying the match criteria for a class constitute the traffic for that class.

The **match protocol** command specifies the name of a protocol to be used as the match criteria against which packets are checked to determine if they belong to the class specified by the class map.

To use the **match protocol** command, you must first enter the **class-map** command to specify the name of the class whose match criteria you want to establish. After you identify the class, you can use one of the following commands to configure its match criteria:

- **match access-group**
- **match input-interface**
- **match mpls experimental**
- **match protocol**

If you specify more than one command in a class map, only the last command entered applies. The last command overrides the previously entered commands.

This command can be used to match protocols that are known to the Network-Based Application Recognition (NBAR) feature. For a list of protocols currently supported by NBAR, see the "Classification" section of the *Cisco IOS Quality of Service Solutions Configuration Guide*.

Examples

The following example specifies a class map called ipx and configures the Internetwork Packet Exchange (IPX) protocol as a match criterion for it:

```
class-map ipx
  match protocol ipx
```

The following example configures NBAR to match File Transfer Protocol (FTP) traffic:

```
match protocol ftp
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.
match access-group	Configures the match criteria for a class map based on the specified ACL.
match input-interface	Configures a class map to use the specified input interface as a match criterion.
match qos-group	Configures a class map to use the specified EXP field value as a match criterion.

match protocol citrix

To configure Network-Based Application Recognition (NBAR) to match Citrix traffic, use the **match protocol citrix** class-map configuration command. To disable NBAR from matching Citrix traffic, use the **no** form of this command.

match protocol citrix [**app** *application-name-string*]

no match protocol citrix [**app** *application-name-string*]

Syntax Description

app	(Optional) Specifies matching of an application name string.
application-name-string	(Optional) Specifies string to be used as the subprotocol parameter.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.1(2)E	This command was introduced.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

Entering the **match protocol citrix** command without the **app** keyword establishes all Citrix traffic as successful match criteria.

Examples

The following example configures NBAR to match all Citrix traffic:

```
match protocol citrix
```

The following example configures NBAR to match Citrix traffic with the application name of packet1:

```
match protocol citrix app packet1
```

match protocol http

To configure Network-Based Application Recognition (NBAR) to match Hypertext Transfer Protocol (HTTP) traffic by URL, HOST, or Multipurpose Internet Mail Extension (MIME)-type, use the **match protocol http** class-map configuration command. To disable NBAR from matching HTTP traffic by URL, HOST, or MIME-type, use the **no** form of this command.

match protocol http [**url** *url-string* | **host** *hostname-string* | **mime** *MIME-type*]

no match protocol http [**url** *url-string* | **host** *hostname-string* | **mime** *MIME-type*]

Syntax Description

url	(Optional) Specifies matching by a URL.
url-string	(Optional) User-specified URL of HTTP traffic to be matched.
host	(Optional) Specifies matching by a host name.
hostname-string	(Optional) User-specified host name to be matched.
mime	(Optional) Specifies matching by MIME text string.
MIME-type	(Optional) User-specified MIME text string to be matched.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)XE2	This command was introduced.
12.1(1)E	This command was integrated into Cisco IOS Release 12.1(1)E.
12.1(2)E	The *hostname-string* argument was added.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

When matching by MIME-type, the MIME-type can contain any user-specified text string. Refer to the the Internet Assigned Numbers Authority (IANA) web page (*www.iana.com*) for a list of the IANA-registered MIME types.

When matching by MIME-type is performed, NBAR matches a packet containing the MIME-type and all subsequent packets until the next HTTP transaction.

When matching by HOST is performed, NBAR performs a regular expression match on the host field contents inside an HTTP GET packet and classifies all packets from that host.

When matching by URL is performed, NBAR recognizes the HTTP GET packets containing the URL, and then matches all packets that are part of the HTTP GET request. When specifying a URL for classification, include only the portion of the URL following www.hostname.domain in the match statement. For example, in the URL www.anydomain.com/latest/whatsnew.html, include only /latest/whatsnew.html.

To match the www.anydomain.com portion, use the host name matching feature. The URL or host specification strings can take the form of a regular expression with options shown in Table 8.

Table 8 URL or HOST Specification String Options

Options	Description
*	Match any zero or more characters in this position.
?	Match any one character in this position.
|	Match one of a choice of characters.
(|)	Match one of a choice of characters in a range. For example, xyz.(gif | jpg) matches either xyz.gif or xyz.jpg.
[ ]	Match any character in the range specified, or one of the special characters. For example, [0-9] is all of the digits; [*] is the "*" character, and [[] is the "[" character.

Examples

The following example classifies, within the class map called class1, HTTP packets based on any URL containing the string whatsnew/latest followed by zero or more characters:

```
class-map class1
match protocol http url whatsnew/latest*
```

The following example classifies, within the class map called class2, packets based on any host name containing the string cisco followed by zero or more characters:

```
class-map class2
match protocol http host cisco*
```

The following example classifies, within the class map called class3, packets based on the Joint Photographics Expert Group (JPEG) MIME type:

```
class-map class3
match protocol http mime "*jpeg"
```

match qos-group

To identify a specific QoS group value as a match criterion, use the **match qos-group** command in class-map configuration mode. To remove a specific QoS group value from a class map, use the **no** form of this command.

match qos-group *qos-group-value*

no match qos-group *qos-group-value*

Syntax Description

qos-group-value	Specifies the exact value from 0 to 99 used to identify a QoS group value.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.
12.0(9)S	This command was integrated into Cisco IOS Release 12.0(9)S.

Usage Guidelines

This command is used by the class map to identify a specific QoS group value marking on a packet.

The *qos-group-value* arguments are used as markings only. The QoS group values have no mathematical significance. For instance, the *qos-group-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *qos-group-value* of 2 is different than a packet marked with the *qos-group-value* of 1. The treatment of these packets is defined by the user through the setting of QoS policies in policy-map class configuration mode.

The QoS group value is local to the router, meaning that the QoS group value that is marked on a packet does not leave the router when the packet leaves the router. If you need a marking that resides in the packet, use IP precedence setting, IP Differentiated Services Code Point (DSCP) setting, or another method of packet marking.

Examples

The following example shows how to configure the service policy called priority50 and attach service policy priority50 to an interface. In this example, the class map called qosgroup5 will evaluate all packets entering Fast Ethernet interface 1/0/0 for a QoS group value of 5. If the incoming packet has been marked with the QoS group value of 5, the packet will be treated with a priority level of 50.

```
Router(config)# class-map qosgroup5
Router(config-cmap)# match qos-group 5
Router(config)# exit
Router(config)# policy-map priority50
Router(config-pmap)# class qosgroup5
Router(config-pmap-c)# priority 50
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

```
Router(config)# interface fa1/0/0
Router(config-if)# service-policy output priority50
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.
policy-map	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy.
set ip precedence	Specifies an IP precedence value for packets within a traffic class.
service-policy	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
show class-map	Displays all class maps and their matching criteria, or a specified class map and its matching criteria.

match source-address mac

To use the source MAC address as a match criterion, use the **match source-address mac** class-map configuration command. To remove a previously specified source MAC address as a match criterion in class map configuration mode, use the **no** form of this command.

match source-address mac *address-destination*

no match source-address mac *address-destination*

Syntax Description

address-destination	Specifies the source destination MAC address to be used as a match criterion.

Defaults

This command has no default behavior or values.

Command Modes

Class-map configuration

Command History

Release	Modification
12.0(5)XE	This command was introduced.
12.1(1)E	This command was integrated into Cisco IOS Release 12.1(1)E.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T.

Usage Guidelines

This command can be used only on an input interface with a MAC address. These interfaces include Fast Ethernet and Ethernet interfaces.

This command cannot be used on output interfaces with no MAC address, such as serial and ATM interfaces.

Examples

The following example uses the mac address mac 0.0.0 as a match criterion:

```
class-map matchsrcmac
match source-address mac 0.0.0
```

Related Commands

Command	Description
class-map	Creates a class map to be used for matching packets to a specified class.

max-reserved-bandwidth

To change the percent of interface bandwidth allocated for Resource Reservation Protocol (RSVP), class-based weighted fair queueing (CBWFQ), low latency queueing (LLQ), IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PVC Interface Priority Queueing (PIPQ), use the **max-reserved bandwidth** interface configuration command. To restore the default value, use the **no** form of this command.

max-reserved-bandwidth *percent*

no max-reserved-bandwidth

Syntax Description

percent	Percent of interface bandwidth allocated for RSVP, CBWFQ, LLQ, IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PIPQ.

Defaults

75 percent

Command Modes

Interface configuration

Command History

Release	Modification
12.0(5)T	This command was introduced.

Usage Guidelines

The sum of all bandwidth allocation on an interface should not exceed 75 percent of the available bandwidth on an interface. The remaining 25 percent of bandwidth is used for overhead, including Layer 2 overhead, control traffic, and best-effort traffic.

If you need to allocate more than 75 percent for RSVP, CBWFQ, LLQ, IP RTP Priority, Frame Relay IP RTP Priority, and Frame Relay PIPQ, you can use the **max-reserved-bandwidth** command. The *percent* argument specifies the maximum percentage of the total interface bandwidth that can be used.

If you do use the **max-reserved-bandwidth** command, make sure that not too much bandwidth is taken away from best-effort and control traffic.

The **max-reserved-bandwidth** command is intended for use on main interfaces only; it has no effect on virtual circuits (VCs) or ATM permanent virtual circuits (PVCs).

Examples

In the following example, the policy map called policy1 is configured for three classes with a total of 8 Mbps configured bandwidth, as shown in the output from the **show policy-map** command:

```
Router# show policy-map policy1

 Policy Map policy1
  Weighted Fair Queueing
    Class class1
      Bandwidth 2500 (kbps) Max Threshold 64 (packets)
    Class class2
      Bandwidth 2500 (kbps) Max Threshold 64 (packets)
```

```
    Class class3
      Bandwidth 3000 (kbps) Max Threshold 64 (packets)
```

When you enter the **service-policy** command in an attempt to attach the policy map on a 10-Mbps Ethernet interface, an error message such as the following is produced:

```
I/f Ethernet1/1 class class3 requested bandwidth 3000 (kbps) Available only 2500 (kbps)
```

The error message is produced because the default maximum configurable bandwidth is 75 percent of the available interface bandwidth, which in this example is 7.5 Mbps. To change the maximum configurable bandwidth to 80 percent, use the **max-reserved-bandwidth** command in interface configuration mode, as follows:

```
max-reserved-bandwidth 80
service output policy1
end
```

To verify that the policy map was attached, enter the **show policy-map interface** command:

```
Router# show policy-map interface e1/1

 Ethernet1/1  output :policy1
  Weighted Fair Queueing
    Class class1
      Output Queue:Conversation 265
        Bandwidth 2500 (kbps) Packets Matched 0 Max Threshold 64 (packets)
        (discards/tail drops) 0/0
    Class class2
      Output Queue:Conversation 266
        Bandwidth 2500 (kbps) Packets Matched 0 Max Threshold 64 (packets)
        (discards/tail drops) 0/0
    Class class3
      Output Queue:Conversation 267
        Bandwidth 3000 (kbps) Packets Matched 0 Max Threshold 64 (packets)
        (discards/tail drops) 0/0
```

Virtual Template Configuration Example

The following example configures a strict priority queue in a virtual template configuration with CBWFQ. The **max-reserved-bandwidth** command changes the maximum bandwidth allocated between CBWFQ and IP RTP Priority from the default (75 percent) to 80 percent.

```
multilink virtual-template 1
interface virtual-template 1
 ip address 172.16.1.1 255.255.255.0
 no ip directed-broadcast
 ip rtp priority 16384 16383 25
 service-policy output policy1
 ppp multilink
 ppp multilink fragment-delay 20
 ppp multilink interleave
 max-reserved-bandwidth 80
 end

interface Serial0/1
 bandwidth 64
 ip address 10.1.1.2 255.255.255.0
 no ip directed-broadcast
 encapsulation ppp
 ppp multilink
 end
```

Note To make the virtual access interface function properly, do not configure the **bandwidth** command on the virtual template. Configure it on the actual interface, as shown in the example.

Related Commands

Command	Description
bandwidth (policy-map class)	Specifies or modifies the bandwidth allocated for a class belonging to a policy map.
ip rtp priority	Reserves a strict priority queue for a set of RTP packet flows belonging to a range of UDP destination ports.
service-policy	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
show policy-map	Displays the configuration of all classes comprising the 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.

oam-bundle

To enable end-to-end F5 Operation, Administration, and Maintenance (OAM) loopback cell generation and OAM management for a virtual circuit (VC) class that can be applied to a VC bundle, use the **oam-bundle** vc-class configuration command. To remove OAM management from the class configuration, use the **no** form of this command.

To enable end-to-end F5 OAM loopback cell generation and OAM management for all VC members of a bundle, use the **oam-bundle** bundle configuration command. To remove OAM management from the bundle, use the **no** form of this command.

oam-bundle [**manage**] [*frequency*]

no oam-bundle [**manage**] [*frequency*]

Syntax Description

manage	(Optional) Enables OAM management. If this keyword is omitted, loopback cells are sent but the bundle is not managed.
frequency	(Optional) Number of seconds between sending OAM loopback cells. Values range from 0 to 600 seconds.

Defaults

End-to-end F5 OAM loopback cell generation and OAM management are disabled, but if OAM cells are received, they are looped back. The default value for the *frequency* argument is 10 seconds.

Command Modes

VC-class configuration (for a VC class)

Bundle configuration (for an ATM VC bundle)

Command History

Release	Modification
12.0(3)T	This command was introduced.

Usage Guidelines

This command defines whether a VC bundle is OAM-managed. If this command is configured for a bundle, every VC member of the bundle is OAM-managed. If OAM management is enabled, further control of OAM management is configured using the **oam retry** command.

This command has no effect if the VC class that contains the command is attached to a standalone VC, that is, if the VC is not a bundle member. In this case, the attributes are ignored by the VC.

To use this command in bundle configuration mode, enter the **bundle** subinterface configuration command to create the bundle or to specify an existing bundle before you enter this command.

To use this command in vc-class configuration mode, first enter the **vc-class atm** global configuration command.

VCs in a VC bundle are subject to the following configuration inheritance rules (listed in order of next highest precedence):

- VC configuration in bundle-vc mode
- Bundle configuration in bundle mode (with effect of assigned vc-class configuration)

Examples

The following example enables OAM management for a bundle called chicago:

```
bundle chicago
 oam-bundle manage
```

Related Commands

Command	Description
broadcast	Configures broadcast packet duplication and transmission for an ATM VC class, PVC, SVC, or VC bundle.
class-bundle	Configures a VC bundle with the bundle-level commands contained in the specified VC class.
encapsulation	Sets the encapsulation method used by the interface.
inarp	Configures the Inverse ARP time period for an ATM PVC, VC class, or VC bundle.
oam retry	Configures parameters related to OAM management for an ATM PVC, SVC, VC class, or VC bundle.
protocol (ATM)	Configures a static map for an ATM PVC, SVC, VC class, or VC bundle. Enables Inverse ARP or Inverse ARP broadcasts on an ATM PVC by either configuring Inverse ARP directly on the PVC, on the VC bundle, or in a VC class (applies to IP and IPX protocols only).

police

To configure the Traffic Policing feature, use the **police** QoS policy-map class configuration command. To remove the Traffic Policing feature 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

bps	Average rate, in bits per second.
burst-normal	Normal burst size, in bytes.
burst-max	Excess burst size, in bytes. In Cisco IOS Release 12.1(5)T onward, the excess burst-size need not be specified unless the **violate-action** option is also specified. In Cisco IOS Releases 12.0(5)XE through 12.1(1)E, the excess burst size must be specified.
conform-action	Action to take on packets that conform to the rate limit.
exceed-action	Action to take on packets that exceed the rate limit.
violate-action	(Optional) Action to take on packets that violate the normal and maximum burst sizes. If the **violate-action** option is specified, the token bucket algorithm works with two token buckets. This option is not available in Cisco IOS Release 12.0 XE or Release 12.1 E.
action	Action to take on packets. Specify one of the following keywords: • **drop**—Drops the packet. • **set-prec-transmit** *new-prec*—Sets the IP precedence and sends the packet. • **set-qos-transmit** *new-qos*—Sets the QoS group and sends the packet. • **set-dscp-transmit** *new-dscp*—Sets the differentiated services code point (DSCP) value and transmits the packet. • **transmit**—Sends the packet.

Defaults

This command is disabled by default.

Command Modes

QoS policy-map class configuration

Command History

Release	Modification
11.1 CC	The **rate-limit** command was introduced.
12.0(5)XE	This **police** command, which was closely related to the **rate-limit** command, was introduced.
12.1(1)E	This command was integrated into Cisco IOS Release 12.1(1)E.
12.1(5)T	This command was integrated into Cisco IOS Release 12.1(5)T. The **violate-action** option became available.

Usage Guidelines

The **violate-action** option is not available in Cisco IOS Release 12.0 XE or Release 12.1 E. The **violate-action** option is not available with the **rate-limit** command.

The Traffic 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, see 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:

  (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 completed for the packet.
- If the number of bytes in the conform bucket B (minus the packet size to be limited) 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** 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:

  (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.
- 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

The following example shows 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, Traffic 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
```

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

In this particular example, Traffic 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
```

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.

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	Attaches a policy map to an input interface or VC, or an output interface or VC, to be used as the service policy for that interface or VC.
show policy-map	Displays the contents of a policy map, including the priority setting of a various policy maps.
show policy-map interface	Displays the contents of a policy map, including the priority setting of a specific interface or PVC.