# ip rsvp reservation-host

To enable a router to simulate a host generating Resource Reservation Protocol (RSVP) RESV messages, use the **ip rsvp reservation-host** global configuration command. To disable this feature, use the **no** form of this command.

**ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*

**no ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*

## Syntax Description

| | |
|---|---|
| *session-ip-address* | For unicast sessions, this is the address of the intended receiver. IP multicast addresses cannot be used with this argument. It must be a logical address configured on an interface on the router you are configuring. |
| *sender-ip-address* | The IP address of the sender. |
| **tcp** \| **udp** \| *ip-protocol* | TCP, User Datagram Protocol UDP, or IP protocol in the range from 0 to 255. |
| *session-dport*<br>*sender-sport* | *session-dport* is the destination port. *sender-sport* is the source port. Port numbers are specified in all cases, because the use of 16-bit ports following the IP header is not limited to UDP or TCP. If destination is zero, source must be zero, and the implication is that ports are not checked. If destination is nonzero, source must be nonzero (except for **wf** reservations, for which the source port is always ignored and can therefore be zero). |
| **ff** \| **se** \| **wf** | Reservation style:<br>• Fixed Filter (**ff**) is single reservation.<br>• Shared Explicit (**se**) is shared reservation, limited scope.<br>• Wild Card Filter (**wf**) is shared reservation, unlimited scope. |
| **rate** \| **load** | QoS guaranteed bit rate service or controlled load service. |
| *bandwidth* | Average bit rate, in kbps, to reserve up to 75 percent of the total on the interface. The range is from 1 to 10000000. |
| *burst-size* | Maximum burst size (KB of data in queue). The range is from 1 to 65535. |

## Defaults

The router does not simulate a host generating RSVP RESV messages by default.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0 | This command was introduced. |

## Usage Guidelines

Use this command to make the router simulate a host generating its own RSVP RESV messages. This command is similar to the **ip rsvp reservation** command, which can cause the router to generate RESV messages on behalf of another host.

The main differences between the **ip rsvp reservation-host** and **ip rsvp reservation** commands follow:

- When you enter the **ip rsvp reservation-host** command, the *session-ip-address* argument must be a local address configured on an interface on the router. Therefore, you cannot proxy a reservation on behalf of a flow destined for another host. Also, you cannot use this command to generate reservation messages for multicast sessions.
- Because the message is assumed to originate from the router you are configuring, you do not specify a next hop or incoming interface for the RSVP RESV message when entering the **ip rsvp reservation-host** command.

Because you cannot use the command to proxy RSVP for non-RSVP-capable hosts or for multicast sessions, the **ip rsvp reservation-host** command is used mostly for debugging and testing purposes.

RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

## Examples

The following example specifies the use of a Shared Explicit style of reservation and the controlled load service, with token buckets of 100 or 150 kbps and 60 or 65 kbps maximum queue depth:

```
ip rsvp reservation-host 10.1.1.1 10.30.1.4 UDP 20 30 se load 100 60
ip rsvp reservation-host 10.40.2.2 10.22.1.1 TCP 20 30 se load 150 65
```

## Related Commands

| Command | Description |
|---|---|
| **fair-queue (WFQ)** | Enables WFQ for an interface. |
| **ip rsvp bandwidth** | Enables RSVP for IP on an interface. |
| **ip rsvp neighbor** | Enables neighbors to request a reservation. |
| **ip rsvp reservation** | Enables a router to simulate receiving and forwarding RSVP RESV messages. |
| **ip rsvp sender** | Enables a router to simulate receiving and forwarding RSVP PATH messages. |
| **ip rsvp sender-host** | Enables a router to simulate a host generating RSVP PATH messages. |
| **ip rsvp udp-multicasts** | Instructs the router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast packet. |
| **random-detect (interface)** | Enables WRED or DWRED. |
| **show ip rsvp installed** | Displays RSVP-related installed filters and corresponding bandwidth information. |
| **show ip rsvp interface** | Displays RSVP-related interface information. |
| **show ip rsvp neighbor** | Displays current RSVP neighbors. |
| **show ip rsvp reservation** | Displays RSVP-related receiver information currently in the database. |
| **show ip rsvp sender** | Displays RSVP PATH-related sender information currently in the database. |

# ip rsvp sender

To enable a router to simulate receiving and forwarding Resource Reservation Protocol (RSVP) PATH messages, use the **ip rsvp sender** global configuration command. To disable this feature, use the **no** form of this command.

**ip rsvp sender** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport previous-hop-ip-address previous-hop-interface bandwidth burst-size*

**no ip rsvp sender** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport previous-hop-ip-address previous-hop-interface bandwidth burst-size*

## Syntax Description

| | |
|---|---|
| *session-ip-address* | For unicast sessions, this is the address of the intended receiver; for multicast sessions, it is the IP multicast address of the session. |
| *sender-ip-address* | The IP address of the sender. |
| **tcp** \| **udp** \| *ip-protocol* | TCP, User Datagram Protocol (UDP), or IP protocol in the range from 0 to 255. |
| *session-dport*<br>*sender-sport* | *session-dport* is the destination port. *sender-sport* is the source port. Port numbers are specified in all cases, because the use of 16-bit ports following the IP header is not limited to UDP or TCP. If destination is zero, source must be zero, and the implication is that ports are not checked. If destination is nonzero, source must be nonzero (except for **wf** reservations, for which the source port is always ignored and can therefore be zero). |
| *previous-hop-ip-address* | Address of the sender or the router closest to the sender. |
| *previous-hop-interface* | Address of the previous hop interface or subinterface. Interface type can be **ethernet**, **loopback**, **null**, or **serial**. |
| *bandwidth* | Average bit rate, in kbps, to reserve up to 75 percent of the total on the interface. The range is from 1 to 10000000. |
| *burst-size* | Maximum burst size (KB of data in queue). The range is from 1 to 65535. |

## Defaults

The router does not simulate receiving and processing RSVP PATH messages by default.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 11.2 | This command was introduced. |

## Usage Guidelines

Use this command to make the router simulate that it is receiving RSVP PATH messages from an upstream host. The command can be used to proxy RSVP PATH messages for non-RSVP-capable senders. By including a local (loopback) previous hop address and previous hop interface, you can also use this command to proxy RSVP for the router you are configuring.

RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

## Examples

The following example sets up the router to act like it is receiving RSVP PATH messages using UDP over loopback interface 1:

```
ip rsvp sender 224.250.0.1 172.16.2.1 udp 20 30 172.16.2.1 loopback 1 50 5
ip rsvp sender 224.250.0.2 172.16.2.1 udp 20 30 172.16.2.1 loopback 1 50 5
```

## Related Commands

| Command | Description |
|---|---|
| **fair-queue (WFQ)** | Enables WFQ for an interface. |
| **ip rsvp bandwidth** | Enables RSVP for IP on an interface. |
| **ip rsvp neighbor** | Enables neighbors to request a reservation. |
| **ip rsvp reservation** | Enables a router to simulate receiving and forwarding RSVP RESV messages. |
| **ip rsvp reservation-host** | Enables a router to simulate a host generating RSVP RESV messages. |
| **ip rsvp sender-host** | Enables a router to simulate a host generating RSVP PATH messages. |
| **ip rsvp udp-multicasts** | Instructs the router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast packet. |
| **random-detect (interface)** | Enables WRED or DWRED. |
| **show ip rsvp installed** | Displays RSVP-related installed filters and corresponding bandwidth information. |
| **show ip rsvp interface** | Displays RSVP-related interface information. |
| **show ip rsvp neighbor** | Displays current RSVP neighbors. |
| **show ip rsvp reservation** | Displays RSVP-related receiver information currently in the database. |
| **show ip rsvp sender** | Displays RSVP PATH-related sender information currently in the database. |

# ip rsvp sender-host

To enable a router to simulate a host generating a Resource Reservation Protocol (RSVP) PATH message, use the **ip rsvp sender-host** global configuration command. To disable this feature, use the **no** form of this command.

**ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport bandwidth burst-size*

**no ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-dport sender-sport bandwidth burst-size*

## Syntax Description

| | |
|---|---|
| *session-ip-address* | For unicast sessions, this is the address of the intended receiver; for multicast sessions, it is the IP multicast address of the session. |
| *sender-ip-address* | The IP address of the sender. It must be a logical address configured on an interface on the router you are configuring. |
| **tcp** \| **udp** \| *ip-protocol* | TCP, User Datagram Protocol (UDP), or IP protocol in the range from 0 to 255. |
| *session-dport sender-sport* | *session-dport* is the destination port. *sender-sport* is the source port. Port numbers are specified in all cases, because the use of 16-bit ports following the IP header is not limited to UDP or TCP. If destination is zero, source must be zero, and the implication is that ports are not checked. If destination is nonzero, source must be nonzero (except for **wf** reservations, for which the source port is always ignored and can therefore be zero). |
| *bandwidth* | Average bit rate, in kbps, to reserve up to 75 percent of the total on the interface. The range is from 1 to 10000000. |
| *burst-size* | Maximum burst size (KB of data in queue). The range is from 1 to 65535. |

## Defaults

The router does not simulate RSVP PATH message generation by default.

## Command Modes

Global configuration

## Command History

| Release | Modification |
|---|---|
| 12.0 | This command was introduced. |

## Usage Guidelines

Use this command to make the router simulate a host generating its own RSVP PATH messages. This command is similar to the **ip rsvp sender** command, which can cause the router to generate RSVP PATH messages on behalf of another host.

The main differences between the **ip rsvp sender-host** and **ip rsvp sender** commands follow:

- When you enter the **ip rsvp sender-host** command, the *sender-ip-address* argument must be a local address configured on an interface on the router.
- Because the message is assumed to originate from the router you are configuring, you do not specify a previous hop or incoming interface for the RSVP PATH message when entering the **ip rsvp sender-host** command.

Because you cannot use the command to proxy RSVP for non-RSVP-capable hosts, the **ip rsvp sender-host** command is used mostly for debugging and testing purposes.

RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

## Examples

The following example sets up the router to act like a host that will send traffic to the given multicast address:

```
ip rsvp sender-host 224.250.0.1 10.24.2.1 udp 20 30 50 5
ip rsvp sender-host 227.0.0.1 10.24.2.1 udp 20 30 50 5
```

## Related Commands

| Command | Description |
|---|---|
| **fair-queue (WFQ)** | Enables WFQ for an interface. |
| **ip rsvp bandwidth** | Enables RSVP for IP on an interface. |
| **ip rsvp neighbor** | Enables neighbors to request a reservation. |
| **ip rsvp reservation** | Enables a router to simulate receiving and forwarding RSVP RESV messages. |
| **ip rsvp reservation-host** | Enables a router to simulate a host generating RSVP RESV messages. |
| **ip rsvp sender** | Enables a router to simulate receiving and forwarding RSVP PATH messages. |
| **ip rsvp udp-multicasts** | Instructs the router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast packet. |
| **random-detect (interface)** | Enables WRED or DWRED. |
| **show ip rsvp installed** | Displays RSVP-related installed filters and corresponding bandwidth information. |
| **show ip rsvp interface** | Displays RSVP-related interface information. |
| **show ip rsvp neighbor** | Displays current RSVP neighbors. |
| **show ip rsvp reservation** | Displays RSVP-related receiver information currently in the database. |
| **show ip rsvp sender** | Displays RSVP PATH-related sender information currently in the database. |

# ip rsvp signalling dscp

To specify the DSCP to be used on all RSVP messages transmitted on an interface, use the **ip rsvp signalling dscp** interface configuration command. To disable the **ip rsvp signalling dscp** interface configuration command, use the **no** form of this command.

**ip rsvp signalling dscp** [*value*]

**no ip rsvp signalling dscp**

## Syntax Description

| | |
|---|---|
| *value* | Indicates a number from 0-63. |

## Defaults

The default value is 0, and the maximum value is 63.

## Command Modes

Interface configuration

## Command History

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

## Usage Guidelines

You configure the DSCP per interface, not per flow. The DSCP determines the priority that a packet receives from various hops as it travels to its destination.

The DSCP applies to all RSVP flows installed on a specific interface. You can configure each interface independently for DSCP.

## Examples

Here is an example of the **ip rsvp signalling dscp** command with a DSCP value of 6:

```
Router(config-if)# ip rsvp signalling dscp 6
Router# show ip rsvp interface detail s2/0

 Se2/0:
   Bandwidth:
     Curr allocated:10K bits/sec
     Max. allowed (total):1536K bits/sec
     Max. allowed (per flow):1536K bits/sec
   Neighbors:
     Using IP enacp:1.  Using UDP encaps:0
   DSCP value used in Path/Resv msgs:0x6
   Burst Police Factor:300%
   RSVP:Data Packet Classification provided by: none
Router#
```

# ip rsvp svc-required

To enable creation of a switched virtual circuit (SVC) to service any new Resource Reservation Protocol (RSVP) reservation made on the interface or subinterface of an Enhanced ATM port adapter (PA-A3), use the **ip rsvp svc-required** interface configuration command. To disable SVC creation for RSVP reservations, use the **no** form of this command.

**ip rsvp svc-required**

**no ip rsvp svc-required**

## Syntax Description

This command has no arguments or keywords.

## Defaults

Disabled. This command applies exclusively to the RSVP-ATM QoS Interworking feature.

## Command Modes

Interface configuration

## Command History

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

## Usage Guidelines

Usually reservations are serviced when RSVP classifies packets and a queueing mechanism schedules them for transmission to manage congestion. Traditionally, RSVP is used with weighted fair queueing (WFQ). When RSVP is coupled with WFQ, all of the packets visible to WFQ are also visible to RSVP, which allows RSVP to identify and take action on packets important to it. In this case, WFQ provides bandwidth guarantees.

However, when the **ip rsvp svc-required** command is used to configure an interface or subinterface, a new SVC is established and used to service each new reservation on the interface. ATM SVCs are used to provide bandwidth guarantees and NetFlow is used on input interfaces to make data packets visible to RSVP.

**Note** When RSVP is enabled, all packets are processed by the Route Switch Processor (RSP).

This command must be executed on both ends of an SVC driven by RSVP. This command is supported only for the Enhanced ATM port adapter (PA-A3) and its subinterfaces.

**Note** For this command to take effect, NetFlow must be enabled. Therefore, the **ip route-cache flow** command must precede this command in the configuration.

Use the **show ip rsvp interface** command to determine whether this command is in effect for any interface or subinterface.

## Examples

The following example signals RSVP that reservations made on ATM interface 2/0/0 will be serviced by creation of an SVC:

```
interface atm2/0/0
 ip rsvp svc-required
```

## Related Commands

| Command | Description |
|---|---|
| **ip route-cache flow** | Enables NetFlow switching for IP routing. |
| **ip rsvp atm-peak-rate-limit** | Sets a limit on the peak cell rate of reservations for all newly created RSVP SVCs established on the current interface or any of its subinterfaces. |
| **ip rsvp precedence** | Allows you to set the IP Precedence values to be applied to packets that either conform to or exceed the RSVP flowspec. |
| **show ip rsvp interface** | Displays RSVP-related interface information. |

# ip rsvp tos

To enable the router to mark the five low-order type of service (ToS) bits of the IP header ToS byte for packets in a Resource Reservation Protocol (RSVP) reserved path using the specified values for traffic that either conforms to or exceeds the RSVP flowspec, use the **ip rsvp tos** interface configuration command. To remove existing settings for the ToS bits, use the **no** form of this command; if neither the **conform** nor **exceed** keyword is specified, all settings for the ToS bits are removed.

**ip rsvp tos** {[**conform** *tos-value*] [**exceed** *tos-value*]}

**no ip rsvp tos** [**conform**] [**exceed**]

## Syntax Description

| | |
|---|---|
| **conform** *tos-value* | (Optional) Specifies a ToS value in the range from 0 to 31 for traffic that conforms to the RSVP flowspec. The ToS value is written to the five low-order bits (bits 0 to 4) of the ToS byte in the IP header of a packet. Either the **conform** or **exceed** keyword is required; both keywords may be specified.<br>When used with the **no** form of the command, the **conform** keyword is optional. |
| **exceed** *tos-value* | (Optional) Specifies a ToS value in the range from 0 to 31 for traffic that exceeds the RSVP flowspec. The ToS byte value is written to the five low-order bits (bits 0 to 4) of the ToS byte in the IP header of a packet. Either the **conform** or **exceed** keyword is required; both keywords may be specified.<br>When used with the **no** form of the command, the **exceed** keyword is optional. |

## Defaults

The ToS bits of the ToS byte are left unmodified when this command is not used. (The default behavior is equivalent to use of the **no ip rsvp tos** command.)

## Command Modes

Interface configuration

## Command History

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

## Usage Guidelines

Packets in an RSVP reserved path are divided into two classes: those that conform to the reservation flowspec and those that correspond to a reservation but that exceed, or are outside, the reservation flowspec.

The **ip rsvp tos** command allows you to set the ToS values to be applied to packets belonging to these two classes. You must specify the ToS value for at least one class of traffic when you use this command. You can use a single instance of the command to specify values for both classes, in which case you can specify the **conform** and **exceed** keywords in either order.

As part of its input processing, RSVP uses the **ip rsvp tos** command configuration to set the ToS bits of the ToS byte on conforming and nonconforming packets. If per-virtual circuit (VC) VIP-distributed Weighted Random Early Detection (DWRED) is configured, the system uses the ToS bit and IP Precedence bit settings on the output interface in its packet drop process. The ToS bit and IP Precedence bit settings of a packet can also be used by interfaces on downstream routers.

Execution of the **ip rsvp tos** command causes ToS bit values for all preexisting reservations on the interface to be modified.

**Note** RSVP must be enabled on an interface before you can use this command; that is, use of the **ip rsvp bandwidth** command must precede use of the **ip rsvp tos** command. RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

**Note** The **ip rsvp tos** command sets bits 0 to 4 so that in combination with the IP Precedence bit settings every bit in the ToS byte is set. Use of these bits is made with full knowledge of the fact that certain canonical texts that address the ToS byte specify that only bits 1 to 4 are used as the ToS bits.

RSVP receives packets from the underlying forwarding mechanism. Therefore, to use the **ip rsvp tos** command to set the ToS bits, one of the following features is required:

- Weighted fair queueing (WFQ) must be enabled on the interface.
- RSVP switched virtual circuits (SVCs) must be used.
- NetFlow must be configured to assist RSVP.

**Note** Use of the **no** form of this command is not equivalent to giving the **ip rsvp tos 0** command, which sets all precedence on the packets to 0, regardless of previous precedence setting.

## Examples

The following example sets the ToS bits value to 4 for all traffic on ATM interface 1 that conforms to the RSVP flowspec. ToS bits on packets exceeding the flowspec are not altered.

```
interface atm1
 ip rsvp tos conform 4
```

## Related Commands

| Command | Description |
|---|---|
| **ip rsvp bandwidth** | Enables RSVP for IP on an interface. |
| **ip rsvp flow-assist** | Enables RSVP to attach itself to NetFlow so that it can leverage NetFlow services. |
| **ip rsvp policy cops minimal** | Lowers the COPS server's load and improves latency times for messages on the governed router. |
| **show ip rsvp** | Displays the IP Precedence and ToS bit values to be applied to packets that either conform to or exceed the RSVP flowspec for a given interface. |

# ip rsvp udp-multicasts

To instruct the router to generate User Datagram Protocol (UDP)-encapsulated Resource Reservation Protocol (RSVP) multicasts whenever it generates an IP-encapsulated multicast packet, use the **ip rsvp udp-multicasts** interface configuration command. To disable this feature, use the **no** form of this command.

**ip rsvp udp-multicasts** [*multicast-address*]

**no ip rsvp udp-multicasts** [*multicast-address*]

## Syntax Description

| | |
|---|---|
| *multicast-address* | (Optional) Host name or UDP multicast address of router. |

## Defaults

The generation of UDP multicasts is disabled. If a system sends a UDP-encapsulated RSVP message to the router, the router begins using UDP for contact with the neighboring system. The router uses multicast address 224.0.0.14 and starts sending to UDP port 1699. If the command is entered with no specifying multicast address, the router uses the same multicast address.

## Command Modes

Interface configuration

## Command History

| Release | Modification |
|---|---|
| 11.2 | This command was introduced. |

## Usage Guidelines

Use this command to instruct a router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast packet. Some hosts require this trigger from the router.

RSVP cannot be configured with VIP-distributed Cisco Express Forwarding (dCEF).

## Examples

The following example reserves up to 7500 kbps on Ethernet interface 2, with up to 1 Mbps per flow. The router is configured to use UDP encapsulation with the multicast address 224.0.0.14.

```
interface ethernet 2
 ip rsvp bandwidth 7500 1000
 ip rsvp udp-multicasts 224.0.0.14
```

## Related Commands

| Command | Description |
|---|---|
| **ip rsvp bandwidth** | Enables RSVP for IP on an interface. |
| **ip rsvp neighbor** | Enables neighbors to request a reservation. |
| **ip rsvp reservation** | Enables a router to simulate receiving and forwarding RSVP RESV messages. |
| **ip rsvp sender** | Enables a router to simulate receiving and forwarding RSVP PATH messages. |

# ip rtp priority

To reserve a strict priority queue for a set of Real-Time Transport Protocol (RTP) packet flows belonging to a range of User Datagram Protocol (UDP) destination ports, use the **ip rtp priority** interface configuration command. To disable the strict priority queue, use the **no** form of this command.

**ip rtp priority** *starting-rtp-port-number port-number-range bandwidth*

**no ip rtp priority**

## Syntax Description

| | |
|---|---|
| *starting-rtp-port-number* | The starting RTP port number. The lowest port number to which the packets are sent. |
| *port-number-range* | The range of UDP destination ports. Number, when added to the *starting-rtp-port-number* argument, that yields the highest UDP port number. |
| *bandwidth* | Maximum allowed bandwidth, in kbps. |

## Defaults

This command has no default behavior or values.

## Command Modes

Interface configuration

## Command History

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

## Usage Guidelines

This command is most useful for voice applications, or other applications that are delay-sensitive.

This command extends and improves on the functionality offered by the **ip rtp reserve** command by allowing you to specify a range of UDP/RTP ports whose voice traffic is guaranteed strict priority service over any other queues or classes using the same output interface. Strict priority means that if packets exist in the priority queue, they are dequeued and sent first—that is, before packets in other queues are dequeued. We recommend that you use the **ip rtp priority** command instead of the **ip rtp reserve** command for voice configurations.

This command can be used in conjunction with either weighted fair queueing (WFQ) or class-based WFQ (CBWFQ) on the same outgoing interface. In either case, traffic matching the range of ports specified for the priority queue is guaranteed strict priority over other CBWFQ classes or WFQ flows; voice packets in the priority queue are always serviced first.

Remember the following guidelines when using the **ip rtp priority** command:

- When used in conjunction with WFQ, the **ip rtp priority** command provides strict priority to voice, and WFQ scheduling is applied to the remaining queues.
- When used in conjunction with CBWFQ, the **ip rtp priority** command provides strict priority to voice. CBWFQ can be used to set up classes for other types of traffic (such as Systems Network Architecture [SNA]) that need dedicated bandwidth and need to be treated better than best effort and not as strict priority; the nonvoice traffic is serviced fairly based on the weights assigned to the enqueued packets. CBWFQ can also support flow-based WFQ within the default CBWFQ class if so configured.

Remember the following guidelines when configuring the *bandwidth* argument:

- It is always safest to allocate to the priority queue slightly more than the known required amount of bandwidth, to allow room for network bursts.
- The IP RTP Priority admission control policy takes RTP header compression into account. Therefore, while configuring the *bandwidth* argument of the **ip rtp priority** command you need to configure only for the bandwidth of the compressed call. Because the *bandwidth* argument is the maximum total bandwidth, you need to allocate enough bandwidth for all calls if there will be more than one call.
- Configure a bandwidth that allows room for Layer 2 headers. The bandwidth allocation takes into account the payload plus the IP, UDP, and RTP headers but does not account for Layer 2 headers. Allowing 25 percent bandwidth for other overhead is conservative and safe.
- The sum of all bandwidth allocation for voice and data flows on an interface cannot exceed 75 percent of the total available bandwidth, unless you change the default maximum reservable bandwidth. To change the maximum reservable bandwidth, use the **max-reserved-bandwidth** command on the interface.

For more information on IP RTP Priority bandwidth allocation, refer to the section “IP RTP Priority” in the chapter “Congestion Management Overview” in the *Cisco IOS Quality of Service Solutions Configuration Guide*.

## Examples

The following example first defines a CBWFQ configuration and then reserves a strict priority queue with the following values: a starting RTP port number of 16384, a range of 16383 UDP ports, and a maximum bandwidth of 40 kbps:

```
! The following commands define a class map:
class-map class1
 match access-group 101
 exit

! The following commands create and attach a policy map:
policy-map policy1
class class1
 bandwidth 3000
 queue-limit 30
 random-detect
 random-detect precedence 0 32 256 100
 exit
interface Serial1
 service-policy output policy1

! The following command reserves a strict priority queue:
 ip rtp priority 16384 16383 40
```

## Related Commands

| Command | Description |
|---|---|
| **bandwidth (policy-map-class)** | Specifies or modifies the bandwidth allocated for a class belonging to a policy map. |
| **fair queue (WFQ)** | Enables WFQ for an interface. |
| **frame-relay ip rtp priority** | Reserves a strict priority queue on a Frame Relay PVC for a set of RTP packet flows belonging to a range of UDP destination ports. |
| **ip rtp reserve** | Reserves a special queue for a set of RTP packet flows belonging to a range of UDP destination ports. |
| **max-reserved-bandwidth** | Changes the percent of interface bandwidth allocated for CBWFQ, LLQ, and IP RTP Priority. |
| **policy-map** | Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy. |
| **ppp multilink** | Enables MLP on an interface and, optionally, enables dynamic bandwidth allocation. |
| **ppp multilink fragment-delay** | Configures a maximum delay allowed for transmission of a packet fragment on an MLP bundle. |
| **ppp multilink interleave** | Enables interleaving of RTP packets among the fragments of larger packets on an MLP bundle. |
| **priority** | Gives priority to a class of traffic belonging to a policy map. |
| **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 for a specified service policy map or all classes for all existing policy maps. |
| **show queue** | Displays the contents of packets inside a queue for a particular interface or VC. |

# match access-group

To configure the match criteria for a class map on the basis of the specified access control list (ACL), use the **match access-group** class-map configuration command. To remove ACL match criteria from a class map, use the **no** form of this command.

**match access-group** {*access-group* | **name** *access-group-name*}

**no match access-group** *access-group*

## Syntax Description

| | |
|---|---|
| *access-group* | A numbered ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. |
| **name** *access-group-name* | A named ACL whose contents are used as the match criteria against which packets are checked to determine if they belong to this class. |

## 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. |

## Usage Guidelines

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

The **match access-group** command specifies a numbered or named ACL whose contents are 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 access-group** 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 acl144 and configures the ACL numbered 144 to be used as the match criteria for this class:

```
class-map acl144
 match access-group 144
```

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **match input-interface** | Configures a class map to use the specified input interface as a match criterion. |
| **match mpls experimental** | Configures a class map to use the specified EXP field value as a match criterion. |
| **match protocol** | Configures the match criteria for a class map on the basis of the specified protocol. |

# match any

To configure the match criteria for a class map to be successful match criteria for all packets, use the **match any** class-map configuration command. To remove all criteria as successful match criteria, use the **no** form of this command.

**match any**

**no match any**

## Syntax Description

This command has no arguments or keywords.

## 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. |

## Examples

In the following configuration, all packets leaving Ethernet interface 1/1 will be policed based on the parameters specified in policy-map class configuration mode.

```
Router(config)# class-map matchany
Router(config-cmap)# match any
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)# exit

Router(config)# interface e1/1
Router(config-if)# service-policy output policy1
```

## Related Commands

| Command | Description |
|---|---|
| **class-map** | Creates a class map to be used for matching packets to a specified class. |
| **match input-interface** | Configures a class map to use the specified input interface as a match criterion. |
| **match protocol** | Configures the match criteria for a class map on the basis of the specified protocol. |

# match class-map

To use a traffic class as a classification policy, use the **match class-map** class-map configuration command. To remove a specific traffic class as a match criterion, use the **no** form of this command.

**match class-map** *class-map-name*

**no match class-map** *class-map-name*

## Syntax Description

| | |
|---|---|
| *class-map-name* | Specifies the name of the traffic class to use 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

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, a traffic class created with the match-any instruction must use a class configured with the match-all instruction as a match criterion (through the **match class-map** command), or vice versa.

You can also use the **match class-map** command to nest traffic classes within one another, saving users the overhead of re-creating a new traffic class when most of the information exists in a previously configured traffic class.

## Examples

In the following example, the traffic class called class1 has the same characteristics as 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, a user 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 the user can simply 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 1.1.1
Router(config-cmap)# exit
```

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 of traffic class called class4 requires a packet to match one of the following three match criteria to be considered a member of traffic class called class 4: IP protocol *and* QoS group 4, destination MAC address 1.1.1, or access group 2.

In this example, only the traffic class called class4 is used with the service 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 1.1.1
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)# exit
```

## Related Commands

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

# match cos

To match a packet based on a Layer 2 class of service (CoS) marking, use the **match cos** command in class-map configuration mode. To remove a specific Layer 2 CoS/Inter-Switch Link (ISL) marking, use the **no** form of this command:

**match cos** *cos-value* [*cos-value cos-value cos-value*]

**no match cos** *cos-value* [*cos-value cos-value cos-value*]

## Syntax Description

| | |
|---|---|
| *cos-value* | (Optional) Specific IEEE 802.1Q/ISL CoS value. The *cos-value* is from 0 to 7; up to four CoS values can be specified in one **match cos** statement. |

## Defaults

This command is disabled by default.

## Command Modes

Class-map configuration

## Command History

| Release | Modification |
|---|---|
| 12.1(5)T | This command was introduced. |

## Examples

In the following example, the CoS-values of 1, 2, and 3 are successful match criteria for the interface containing the classification policy called cos:

```
Router(config)# class-map cos
Router(config-cmap)# match cos 1 2 3
```

In the following example, classes called voice and video-n-data are created to classify traffic based on the CoS values. QoS treatment is then given to the appropriate packets (in this case, the QoS treatment is priority 64 and bandwidth 512) in the CoS-based-treatment policy map.

```
Router(config)# class-map voice
Router(config-cmap)# match cos 7

Router(config)# class-map video-n-data
Router(config-cmap)# match cos 5

Router(config)# policy-map cos-based-treatment
Router(config-pmap)# class voice
Router(config-pmap-c)# priority 64
Router(config-pmap-c)# exit
Router(config-pmap)# class video-n-data
Router(config-pmap-c)# bandwidth 512
Router(config-pmap-c)# exit
Router(config-pmap)# exit
```

```
Router(config)# interface fa0/0.1
Router(config-if)# service-policy output cos-based-treatment
```

The service policy configured in this section is attached to all packets leaving Fast Ethernet interface 0/0.1. The service policy can be attached to any interface that supports service policies.

## 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. |
| **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. |
| **set cos** | Sets the Layer 2 CoS value of an outgoing packet. |
| **show class-map** | Displays all class maps and their matching criteria. |

# match destination-address mac

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

**match destination-address mac** *address*

**no match destination-address mac** *address*

## Syntax Description

| | |
|---|---|
| *address* | Specifies the specific 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. |

## Examples

The following example specifies a class map called macaddress and specifies the destination MAC address to be used as the match criterion for this class.

```
class-map macaddress
match destination-address mac 00:00:00:00:00:00
```

## Related Commands

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

# match input-interface

To configure a class map to use the specified input interface as a match criterion, use the **match input-interface** class-map configuration command. To remove the input interface match criterion from a class map, use the **no** form of this command.

**match input-interface** *interface-name*

**no match input-interface** *interface-name*

## Syntax Description

| | |
|---|---|
| *interface-name* | Name of the input interface to be used as match criteria. |

## 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. |

## 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 input-interface** command specifies the name of an input interface 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 input-interface** 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 input interface named ethernet1to be used as the match criterion for this class:

```
class-map eth1
 match input-interface ethernet1
```

## 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 mpls experimental** | Configures a class map to use the specified EXP field value as a match criterion. |
| **match protocol** | Configures the match criteria for a class map on the basis of the specified protocol. |

# match ip dscp

To identify a specific IP differentiated service code point (DSCP) value as a match criterion, use the **match ip dscp** class-map configuration command. To remove a specific IP DSCP value from a class map, use the **no** form of this command.

**match ip dscp** *ip-dscp-value* [*ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value ip-dscp-value*]

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

## Syntax Description

| | |
|---|---|
| *ip-dscp-value* | Specifies the exact value from 0 to 63 used to identify an IP DSCP 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 in Cisco IOS Release 12.0(9)S. |
| 12.1(2)T | This command was integrated in Cisco IOS Release 12.1(2)T. |

## Usage Guidelines

Up to eight IP DSCP values can be matched in one match statement. For example, if you wanted the IP DCSP values of 0, 1, 2, 3, 4, 5, 6, or 7 (note that only one of the IP DSCP values must be a successful match criterion, not all of the specified IP DSCP values), enter the **match ip dscp 0 1 2 3 4 5 6 7** command.

This command is used by the class map to identify a specific IP DSCP value marking on a packet. The *ip-dscp-value* arguments are used as markings only. The IP DSCP values have no mathematical significance. For instance, the *ip-dscp-value* of 2 is not greater than 1. The value simply indicates that a packet marked with the *ip-dscp-value* of 2 is different than a packet marked with the *ip-dscp-value* of 1. The treatment of these marked 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 ipdscp15 will evaluate all packets entering interface Fast Ethernet 1/0/0 for an IP DSCP value of 15. If the incoming packet has been marked with the IP DSCP value of 15, the packet will be treated with a priority level of 55.

```
Router(config)# class-map ipdscp15
Router(config-cmap)# match ip dscp 15
Router(config)# exit
Router(config)# policy-map priority55
```

```
Router(config-pmap)# class ipdscp15
Router(config-pmap-c)# priority 55
Router(config-pmap-c)# exit
Router(config-pmap)# exit
Router(config)# interface fa1/0/0
Router(config-if)# service-policy input priority55
```

## 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. |
| **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. |
| **set ip dscp** | Marks the IP DSCP value for packets within a traffic class. |
| **show class-map** | Displays all class maps and their matching criteria. |