Configuring RSVP

Last Updated: January 12, 2012

This chapter describes the tasks for configuring the Resource Reservation Protocol (RSVP) feature, which is an IP service that allows end systems or hosts on either side of a router network to establish a reserved-bandwidth path between them to predetermine and ensure Quality of Service (QoS) for their data transmission.

Finding Feature Information

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

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

Prerequisites for Configuring RSVP

RSVP is disabled by default to allow backward compatibility with systems that do not implement RSVP. You must enable RSVP before you make any other RSVP configurations.

Americas Headquarters:
Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA

Restrictions for Configuring RSVP

- RSVP cannot be configured with Versatile Interface Processors (VIP)-distributed Cisco Express Forwarding (dCEF).
- The RSVP over DMVPN feature does not support RSVP over IPsec tunnels without generic routing encapsulation (GRE).
- The ingress call admission control (CAC) functionality does not support RSVP Fast Local Repair; if there are route changes inside the non-RSVP cloud that result in corresponding changes in the ingress interface.

Information About Configuring RSVP

RSVP allows end systems to request QoS guarantees from the network. The need for network resource reservations differs for data traffic versus for real-time traffic, as follows:

- Data traffic seldom needs reserved bandwidth because internetworks provide datagram services for data traffic. This asynchronous packet switching may not need guarantees of service quality. End-to-end controls between data traffic senders and receivers help ensure adequate transmission of bursts of information.
- Real-time traffic (that is, voice or video information) experiences problems when operating over datagram services. Because real-time traffic sends an almost constant flow of information, the network "pipes" must be consistent. Some guarantee must be provided so that service between real-time hosts will not vary. Routers operating on a first-in, first-out (FIFO) basis risk unrecoverable disruption of the real-time information that is being sent.

Data applications, with little need for resource guarantees, frequently demand relatively lower bandwidth than real-time traffic. The almost constant high bit-rate demands of a video conference application and the bursty low bit-rate demands of an interactive data application share available network resources.

RSVP prevents the demands of traffic such as large file transfers from impairing the bandwidth resources necessary for bursty data traffic. When RSVP is used, the routers sort and prioritize packets much like a statistical time-division multiplexer (TDM) would sort and prioritize several signal sources that share a single channel.

RSVP mechanisms enable real-time traffic to reserve resources necessary for consistent latency. A video conferencing application can use settings in the router to propagate a request for a path with the required bandwidth and delay for video conferencing destinations. RSVP will check and repeat reservations at regular intervals. By this process, RSVP can adjust and alter the path between RSVP end systems to recover from route changes.

Real-time traffic (unlike data traffic) requires a guaranteed network consistency. Without consistent QoS, real-time traffic faces the following problems:

- Jitter--A slight time or phase movement in a transmission signal can introduce loss of synchronization or other errors.
- Insufficient bandwidth--Voice calls use a digital signal level 0 (DS-0 at 64 kb/s), video conferencing uses T1/E1 (1.544 Mb/s or 2.048 Mb/s), and higher-fidelity video uses much more.
- Delay variations--If the wait time between when signal elements are sent and when they arrive varies, the real-time traffic will no longer be synchronized, and transmission may fail.
- Information loss--When signal elements drop or arrive too late, lost audio causes distortions with noise or crackle sounds. The lost video causes image blurring, distortions, or blackouts.

RSVP works in conjunction with weighted fair queueing (WFQ) or Random Early Detection (RED). This conjunction of reservation setting with packet queueing uses two key concepts: end-to-end flows with RSVP and router-to-router conversations with WFQ:

- RSVP flow--This is a stream that operates "multidestination simplex," because data travels across it in only one direction: from the origin to the targets. Flows travel from a set of senders to a set of receivers. The flows can be merged or left unmerged, and the method of merging them varies according to the attributes of the application using the flow.
- WFQ conversation--This is the traffic for a single transport layer session or network layer flow that crosses a given interface. This conversation is identified from the source and destination address, protocol type, port number, or other attributes in the relevant communications layer.

RSVP allows for hosts to send packets to a subset of all hosts (multicasting). RSVP assumes that resource reservation applies primarily to multicast applications (such as video conferencing). Although the primary target for RSVP is multimedia traffic, a clear interest exists for the reservation of bandwidth for unicast traffic (such as Network File System (NFS) and Virtual Private Network management). A unicast transmission involves a host sending packets to a single host.

Before configuring RSVP, you should understand the following concepts:

RSVP Reservation Types

There are the two types of multicast flows:

- Distinct reservation--A flow that originates from exactly one sender.
- Shared reservation--A flow that originates from one or more senders.

RSVP describes these reservations as having certain algorithmic attributes.

Distinct Reservation

An example of a distinct reservation is a video application in which each sender emits a distinct data stream that requires admission and management in a queue. Such a flow, therefore, requires a separate reservation per sender on each transmission facility it crosses (such as Ethernet, a High-Level Data Link Control (HDLC) line, a Frame Relay data-link connection identifier (DLCI), or an ATM virtual channel). RSVP refers to this distinct reservation as explicit and installs it using a fixed filter style of reservation.

Use of RSVP for unicast applications is generally a degenerate case of a distinct flow.

Shared Reservation

An example of a shared reservation is an audio application in which each sender emits a distinct data stream that requires admission and management in a queue. However, because of the nature of the application, a limited number of senders are sending data at any given time. Such a flow, therefore, does not require a separate reservation per sender. Instead, it uses a single reservation that can be applied to any sender within a set as needed.

RSVP installs a shared reservation using a Wild Card or Shared Explicit style of reservation, with the difference between the two determined by the scope of application (which is either wild or explicit):

- The Wild Card Filter reserves bandwidth and delay characteristics for any sender and is limited by the list of source addresses carried in the reservation message.
- The Shared Explicit style of reservation identifies the flows for specific network resources.

Planning RSVP Configuration

You must plan carefully to successfully configure and use RSVP on your network. At a minimum, RSVP must reflect your assessment of bandwidth needs on router interfaces. Consider the following questions as you plan for RSVP configuration:

- How much bandwidth should RSVP allow per end-user application flow? You must understand the "feeds and speeds" of your applications. By default, the amount reservable by a single flow can be the entire reservable bandwidth. You can, however, limit individual reservations to smaller amounts using the single flow bandwidth parameter. The reserved bandwidth value may not exceed the interface reservable amount, and no one flow may reserve more than the amount specified.
- How much bandwidth is available for RSVP? By default, 75 percent of the bandwidth available on an interface is reservable. If you are using a tunnel interface, RSVP can make a reservation for the tunnel whose bandwidth is the sum of the bandwidths reserved within the tunnel.
- How much bandwidth must be excluded from RSVP so that it can fairly provide the timely service required by low-volume data conversations? End-to-end controls for data traffic assume that all sessions will behave so as to avoid congestion dynamically. Real-time demands do not follow this behavior. Determine the bandwidth to set aside so bursty data traffic will not be deprived as a side effect of the RSVP QoS configuration.

Note Before entering RSVP configuration commands, you must plan carefully.

RSVP Implementation Considerations

You should be aware of RSVP implementation considerations as you design your reservation system. RSVP does not model all data links likely to be present on the internetwork. RSVP models an interface as having a queueing system that completely determines the mix of traffic on the interface; bandwidth or delay characteristics are deterministic only to the extent that this model holds. Unfortunately, data links are often imperfectly modeled this way. Use the following guidelines:

- Serial line interfaces--PPP; HDLC; Link Access Procedure, Balanced (LAPB); High-Speed Serial Interface (HSSI); and similar serial line interfaces are well modeled by RSVP. The device can, therefore, make guarantees on these interfaces. Nonbroadcast multiaccess (NBMA) interfaces are also most in need of reservations.
- Multiaccess LANs--These data links are not modeled well by RSVP interfaces because the LAN itself represents a queueing system that is not under the control of the device making the guarantees. The device guarantees which load it will offer, but cannot guarantee the competing loads or timings of loads that neighboring LAN systems will offer. The network administrator can use admission controls to control how much traffic is placed on the LAN. The network administrator, however, should focus on the use of admission in network design in order to use RSVP effectively.

The Subnetwork Bandwidth Manager (SBM) protocol is an enhancement to RSVP for LANs. One device on each segment is elected the Designated SBM (DSBM). The DSBM handles all reservations on the segment, which prevents multiple RSVP devices from granting reservations and overcommitting the shared LAN bandwidth. The DSBM can also inform hosts of how much traffic they are allowed to send without valid RSVP reservations.

- Public X.25 networks--It is not clear that rate or delay reservations can be usefully made on public X.25 networks.

You must use a specialized configuration on Frame Relay and ATM networks, as discussed in the next sections.

- Frame Relay Internetwork Considerations, page 5
- ATM Internetwork Considerations, page 5
- Flexible Bandwidth Considerations, page 5

Frame Relay Internetwork Considerations

The following RSVP implementation considerations apply as you design your reservation system for a Frame Relay internetwork:

- Reservations are made for an interface or subinterface. If subinterfaces contain more than one data-link control (DLC), the required bandwidth and the reserved bandwidth may differ. Therefore, RSVP subinterfaces of Frame Relay interfaces must contain exactly one DLC to operate correctly.
- In addition, Frame Relay DLCs have committed information rates (CIR) and burst controls (Committed Burst and Excess Burst) that may not be reflected in the configuration and may differ markedly from the interface speed (either adding up to exceed it or being substantially smaller). Therefore, the **ip rsvp bandwidth** command must be entered for both the interface and the subinterface. Both bandwidths are used as admission criteria.

For example, suppose that a Frame Relay interface runs at a T1 rate (1.544 Mb/s) and supports several DLCs to remote offices served by 128-kb/s and 56-kb/s lines. You must configure the amount of the total interface (75 percent of which is 1.158 Mb/s) and the amount of each receiving interface (75 percent of which would be 96 and 42 kb/s, respectively) that may be reserved. Admission succeeds only if enough bandwidth is available on the DLC (the subinterface) and on the aggregate interface.

ATM Internetwork Considerations

The following RSVP implementation considerations apply as you design your reservation system for an ATM internetwork:

- When ATM is configured, it most likely uses a usable bit rate (UBR) or an available bit rate (ABR) virtual channel (VC) connecting individual routers. With these classes of service, the ATM network makes a "best effort" to meet the bit-rate requirements of the traffic and assumes that the end stations are responsible for information that does not get through the network.
- This ATM service can open separate channels for reserved traffic having the necessary characteristics. RSVP should open these VCs and adjust the cache to make effective use of the VC for this purpose.

Flexible Bandwidth Considerations

RSVP can be enabled on a physical or a logical interface by using the **ip rsvp bandwidth** command. You can either configure an absolute value or a percentage of the interface bandwidth as the RSVP bandwidth or flow bandwidth. That is, you have an option to configure an absolute value for RSVP bandwidth and a percentage of the interface bandwidth as the flow bandwidth or vice versa. Use the **ip rsvp bandwidth** command to configure the absolute values for the RSVP or the flow bandwidth. Use the **ip rsvp bandwidth percent** command to configure a percentage of the interface bandwidth as the RSVP or the flow bandwidth. If you configure a percent of the interface bandwidth as the RSVP bandwidth, the RSVP bandwidth changes in parallel with the changes in the interface bandwidth. The same applies to the flow bandwidth.

The bandwidth on a fixed interface can be changed by making explicit configurations of bandwidth on the fixed interface. Although the same applies to flexible bandwidth interfaces, bandwidth on them can change due to many other reasons such as addition or removal of member links and change in the bandwidth of member links.

RSVP Ingress CAC

The RSVP Ingress CAC feature extends the Cisco IOS RSVP IPv4 implementation to guarantee bandwidth resources not only on a given flow’s outgoing interface, but also on the inbound interfaces.

The figure below presents a deployment scenario where the ingress CAC functionality is implemented. The headquarters and branch office of a company are connected over a non-RSVP Internet service provider (ISP) cloud. In this scenario, the ISP cloud can guarantee the required bandwidth without the need to run RSVP. Therefore, only the customer edge (CE) routers run RSVP, and not the provider edge (PE) routers.

Figure 1 RSVP Ingress CAC

IMAGE MISSING HERE; illos embedded not referenced

Consider a scenario where the CE-PE link used in the headquarters has a bandwidth of 10 Gb/s, whereas the CE-PE link used in the branch office has a bandwidth of 1 Gb/s. Some media traffic from the headquarters to the branch office requires a guaranteed bandwidth of 5 Gb/s. In the RSVP implementation presented in the figure above, the CE-PE link used in the headquarters can participate in the RSVP bandwidth reservation and, therefore can guarantee the required QoS for this 5 Gb/s flow. The CE-PE link used in the branch office is a bottleneck because it has only 1 Gb/s capacity. However, this does not get detected because RSVP CAC is performed only against the egress interface in the branch office (CE to the branch office). Hence, traffic of 5 Gb/s is admitted. This situation can be avoided if RSVP CAC functionality is extended to check the ingress interface bandwidth before admitting this traffic.

The benefits of the RSVP Ingress CAC feature are as follows:

- Extends the bandwidth reservation to perform CAC on inbound interfaces if ingress RSVP bandwidth pools have been configured on those interfaces.
- Extends the preemption logic whenever the ingress interface bandwidth changes (due to link bandwidth changes, ingress bandwidth pool changes, or due to changes in ingress policy), or if a new reservation request is received.
- Extends the RSVP policy to include ingress policy parameters.

This feature is supported over all RSVP-supported transport layers.

The ingress CAC functionality is not enabled by default. Use the **ip rsvp bandwidth** command to enable ingress CAC and to define an ingress RSVP bandwidth pool. The ingress CAC functionality is applicable to only those reservations that are established after the feature is enabled.

- Admission Control on the Intermediate RSVP-Aware Nodes, page 7
- Admission Control on IP Tunnel Interfaces, page 7
- RSVP Preemption, page 7

Admission Control on the Intermediate RSVP-Aware Nodes

For every new or modified RSVP reservation request received on an intermediate RSVP-aware node, the admission control is first performed against the bandwidth pool associated with the egress interface, and then it is performed on the bandwidth pool associated with the ingress interface of that flow.

Admission Control on IP Tunnel Interfaces

If the ingress interface of a flow is an IP tunnel, you must configure the required ingress RSVP bandwidth pools on both the tunnel interface as well as the underlying physical interface. The ingress CAC feature checks against both these bandwidth pools before admitting a request.

RSVP Preemption

RSVP preemption allows the router to preempt one or more existing RSVP bandwidth reservations to accommodate a higher priority reservation, while staying within the RSVP-configured bandwidth pool limit. The dynamic update of the RSVP bandwidth can be made by the RSVP policy to preempt or admit RSVP sessions based on the latest RSVP bandwidth. Use the **ip rsvp policy preempt**commandtoenable RSVP preemption on both egress and ingress interfaces.

RSVP preemption is required for the following reasons:

- The link bandwidth can shrink (either due to custom-made configuration or dynamically, as in case of flexible bandwidth links).
- The user can shrink the RSVP bandwidth pool due to custom-made configuration.
- A new reservation has a higher priority than some of the existing reservations.
- Changes are made to the RSVP local policy such that either the maximum group bandwidth or the maximum single bandwidth (or both) have been reduced and, therefore, all the reservations that match this policy require preemption.

RSVP over DMVPN

Dynamic Multipoint Virtual Private Network (DMVPN) allows users to scale large and small IPsec VPNs by combining GRE tunnels, IPsec encryption, and Next Hop Resolution Protocol (NHRP). For more information on DMVPN, refer to the DMVPN module.

The RSVP over DMVPN feature supports the following types of configuration:

- RSVP over manually configured GRE/multipoint generic routing encapsulation (mGRE) tunnels
- RSVP over manually configured GRE/mGRE tunnels in an IPsec protected mode
- RSVP over GRE/mGRE tunnels (IPsec protected and IPsec unprotected) in a DMVPN environment

The figure below shows a spoke-hub-spoke or phase 1 DMVPN mode. Two static spoke-to-hub tunnels called Tunnel0 have been established. Tunnel0 is presented as a GRE interface on spoke-A and spoke-B. On the hub, Tunnel0 is modeled as an mGRE interface.

Figure 2 RSVP over DMVPN Phase 1

IMAGE MISSING HERE; illos embedded not referenced

There are some differences in the way RSVP operates over tunnels and RSVP operates over a subinterface. If RSVP is configured on a subinterface, Cisco IOS software automatically applies RSVP configuration on the main interface as well. This is possible because the binding between the subinterface and the main interface is static. However, the association between a tunnel interface and a physical interface is dynamic. Therefore, when you configure RSVP over a tunnel, the same configuration cannot be directly applied to any physical interface because the tunnel-to-physical association can change. Hence, you must configure RSVP appropriately on the physical interface (main and/or subinterface) that a tunnel can egress over.

If a device such as an IP phone attached on the 192.168.1.0/24 network has to establish reservation for a call to another device, such as another IP phone, attached on the 192.168.2.0/24 network, spoke A sends out a PATH message directed towards spoke B over tunnel interface 0. The RESV message is intercepted by the hub and forwarded to spoke B. Spoke B responds with a RESV message, which is sent to the hub. The hub attempts to reserve bandwidth over the Tunnel0 mGRE interface and its associated physical interface. If the hub is able reserve the necessary bandwidth, a reservation is installed and the RESV message is forwarded to spoke A. Spoke A receives a RESV message on Tunnel0 and attempts to reserve bandwidth over the Tunnel0 GRE interface and its associated physical interface. If spoke A is successful in reserving the necessary bandwidth, a reservation is installed.

Note RSVP Call Admission Control (CAC) is performed over the new physical interface when there is a change in the tunnel-to-physical interface association for a given session. This might potentially cause the once-established RSVP reservation to fail. In such a case, RSVP removes only the existing reservation. The data flow is determined by other specific applications, such as, Cisco Unified Communications Manager Express (Cisco UCME) in case of voice traffic.

During bandwidth admission control, Cisco IOS software must take into account the additional IP overhead introduced due to tunneling and a possible encryption over these tunnels. Default values are provided for the additional overhead based on the average size of an Internet packet. However, you can use the **ip rsvp tunnel overhead-percent** command to override these values.

Transport Mechanism Support in RSVP

The RSVP Transport for Medianet feature extends the RSVP functionality to act as a transport mechanism for the clients. This is achieved by adding three more parameters to the existing 5-tuple flow that is used to reserve a path from the sender to the receiver for data flow. The 5-tuple flow consists of the destination IP address, source IP address, IP protocol, destination port, and source port.

In this model, for every transport service requested by the clients, RSVP creates a transport protocol (TP) session. Each such transport service request is identified by the 8-tuple flow as shown in the table below:

Table 1 RSVP Transport Protocol Support--8-Tuple Flow

8-Tuple Parameters	Description
Destination-IP	Destination IP address of the flow.
Destination-Port	Destination port of the flow.
IP Protocol	IP protocol number in the IP header.
Source-IP	Source IP address of the flow.
Source-Port	Source port of the flow.

8-Tuple Parameters	Description
Client ID	Identifies a particular client application. The client ID is a globally allocated number identifying a client that uses RSVP transport. It is provided by the client to RSVP when the client registers to RSVP. The client ID enables RSVP to distinguish between different client applications requesting transport service for the same 5-tuple flow.
Initiator ID	Identifies the node initiating the transport service request. The initiator ID enables RSVP distinguish between the transport service request generated by the same client application, for the same 5-tuple flow, but from different initiating nodes. The TP clients need to pass this initiator ID in the 8-tuple flow when they must initiate an RSVP transport session. This ID has to be unique across the network.
Instance ID	Identifies the transport service request from a particular client application and from a particular initiator. The instance ID lets RSVP distinguish between different instances of a transport service request that is generated by the same client application for the same 5-tuple flow and from the same initiating node. The instance ID is passed by the client to RSVP when the client must initiate an RSVP transport session.

The 8-tuple flow identifies RSVP TP sessions and maps them to the specific client transport service requests.

When a TP client requests a transport service from RSVP, RSVP creates a TP session specific to that transport service request, and uses it to transport any other messages being sent by the client for the service request. RSVP also maintains the state of this TP session by refreshing PATH messages periodically.

RSVP provides two types of transport mechanisms to the clients for the transport service requests:

- Path-based transport mechanism--In this mechanism, the initiator node transports a TP client’s message (also referred to as TP-Client-Data) to the destination for a particular flow. RSVP creates TP session specific to the transport service request from the client and uses the PATH message to send the TP-Client-Data. It ensures that the TP-Client-Data is transported in the same path as the data flow for the corresponding 5-tuple. RSVP maintains the state of this transport session on all the intermediate nodes from the initiator to either the destination or to the node on which the TP session will be terminated.
- Transport notify-based transport mechanism--In this mechanism, TP-Client-Data from any node in the path of the flow is transported to any other node in the same path. RSVP uses the Transport-Notify message to send the TP-Client-Data.

In the path-based transport mechanism, RSVP PATH message is used to carry the TP-Client-Data along the path from the sender to the receiver. RSVP hands over the TP-Client-Data to the client stack on each of the RSVP-enabled hops where the client stack is running. The client can then perform one of the following tasks:

- Request RSVP to send out the TP-Client-Data that is modified or not modified further downstream towards the receiver. In this case, RSVP embeds the client's outgoing TP-Client-Data in the PATH message and forwards it towards the receiver.
- Terminate the TP-Client-Data if the client decides to close the transport session on a particular node. In this case, RSVP does not send any PATH message downstream.

In the transport notify-based transport mechanism, RSVP uses Transport-Notify message to send the client's message. In this case, the TP client can request RSVP to perform one of the following tasks:

- Request RSVP to send the TP-Client-Data for the 8-tuple flow to a target IP address. This request works even if the RSVP TP session does not exist for the corresponding 8-tuple flow.
- Request RSVP to send the TP-Client-Data to the previous upstream RSVP hop. This process assumes that an RSVP TP session exists for the corresponding 8-tuple flow. In this case, RSVP derives the previous RSVP-aware hop IP address from the Path State Block (PSB) for the 8-tuple flow and sends the Transport-Notify message to that IP address with TP-Client-Data embedded into it.

RSVP hands over the Transport-Notify message with the embedded transport object to the corresponding TP client running on the router. If the corresponding TP client does not exist on the router, and if there is an existing RSVP TP session for the 8-tuple flow in the RSVP Transport-Notify message, then RSVP further sends this message to the previous upstream RSVP-enabled router. This continues until RSVP is able to deliver this message to the TP client.

If the corresponding TP client does not exist on the router, and if there is no existing RSVP TP session for the 8-tuple flow, RSVP drops the message.

NAT Aware RSVP

The NAT Aware RSVP feature enables the RSVP-Network Address Translation (NAT)-Application Layer Gateway (ALG) functionality. With the RSVP-NAT-ALG functionality enabled, when the RSVP messages pass through a NAT device, the IP addresses embedded in the RSVP payload get translated appropriately. You can use the **show ip nat translations** command to view the active NATs for RSVP messages.

The RSVP-NAT-ALG is present in both pre-routing and post-routing stages. When a packet is travelling from the local to global stage, only the RSVP-NAT-ALG on the post-routing stage will be effective and will perform the local to global address and port translations; where as, if the packet is travelling from the global to local stage, the RSVP-NAT-ALG present in the pre-routing stage will be effective and will perform the global-to-local address and port translations.

With RSVP enabled, the packets are considered after the pre-routing stage. If the packet is travelling from the local to global stage, it acts on the packet before the local to global translations are performed. However, if the packet is travelling from the global to local stage, it acts on the packet after the global to local translations are performed. Hence RSVP states are maintained based on the local addresses.

How to Configure RSVP

- Enabling RSVP to Attach to NetFlow, page 20
- Setting the IP Precedence and ToS Values, page 22
- Configuring Tunnel Bandwidth Overhead, page 23
- Sending RSVP Notifications, page 24
- Verifying RSVP Configuration, page 25

Enabling RSVP

By default, RSVP is disabled so that it is backward compatible with systems that do not implement RSVP. To enable RSVP for IP on an interface, perform the following task. This task starts RSVP and sets the bandwidth and single-flow limits.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-bandwidth*[**percent** *percent-bandwidth* | [*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-bandwidth*[**percent** *percent-bandwidth* | [*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]] **Example:** `Router(config-if)# ip rsvp bandwidth 23 54`	Enables RSVP for IP on an interface.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring RSVP Bandwidth

To configure the RSVP bandwidth, perform the following task. The default maximum bandwidth is up to 75 percent of the bandwidth available on the interface. By default, the amount reservable by a flow can be up to the entire reservable bandwidth.

Reservations on individual circuits that do not exceed 100 kb/s normally succeed. However, if reservations have been made on other circuits adding up to 1.2 Mb/s, and a reservation is made on a subinterface that itself has enough remaining bandwidth, the reservation request will still be refused because the physical interface lacks supporting bandwidth.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
    - **ip rsvp bandwidth** [*interface-bandwidth*[**percent** *percent-bandwidth* | [*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]
    -
    -
    - **ip rsvp bandwidth percent** *rsvp-bandwidth* [*max-flow-bw* | **percent** *flow-bandwidth*]
5. Do one of the following:
    - **ip rsvp bandwidth ingress** *ingress-bandwidth*
    -
    - **ip rsvp bandwidth ingress percent** *percent-bandwidth* [*maximum-ingress-bandwidth* | **percent** *percent-bandwidth*]
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface multilink 2`	Configures an interface and enters interface configuration mode.

Step 4	Do one of the following:
	• **ip rsvp bandwidth** [*interface-bandwidth*[**percent** *percent-bandwidth* | [*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]
	•
	•
	• **ip rsvp bandwidth percent** *rsvp-bandwidth* [*max-flow-bw* | **percent** *flow-bandwidth*]
	Example:
	`Router(config-if)# ip rsvp bandwidth 23 34`
	Example:
	Example:
	`Router(config-if)# ip rsvp bandwidth percent 50 percent 10`
		Configures an absolute value for the RSVP bandwidth and the flow bandwidth.
		Note On subinterfaces, this command applies the more restrictive of the available bandwidths of the physical interface and the subinterface. For example, a Frame Relay interface might have a T1 connector nominally capable of 1.536 Mb/s, and 64-kb/s subinterfaces on 128-kb/s circuits (64-kb/s CIR). RSVP bandwidth can be configured on the main interface up to 1200 kb/s, and on each subinterface up to 100 kb/s.
		or
		Configures a percentage of the interface bandwidth as RSVP bandwidth and flow bandwidth.
		For more examples, refer to Configuration Examples for Configuring RSVP, page 27

	Command or Action	Purpose

Step 5	Do one of the following:
	• **ip rsvp bandwidth ingress** *ingress-bandwidth*
	•
	• **ip rsvp bandwidth ingress percent** *percent-bandwidth* [*maximum-ingress-bandwidth* | **percent** *percent-bandwidth*]
	Example:
	`Router(config-if)# ip rsvp bandwidth ingress 40`
	Example:
	Example:
	`Router(config-if)# ip rsvp bandwidth ingress percent 80`
		(Optional) Configures the RSVP ingress reservable bandwidth.
		or
		Configures a percentage of the interface bandwidth as the ingress bandwidth.

Step 6	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Maximum Bandwidth for Single or Group Flows

Perform this task to configure the maximum bandwidth for single or group flows. As part of the application ID enhancement, maximum bandwidth can be configured for RESV messages. This allows the local policy bandwidth limit to be used by RSVP's admission control process for both shared and nonshared reservations. It also allows a local policy to trigger preemption during the admission control function if there is insufficient policy bandwidth to meet the needs of an incoming RESV message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp policy local identity** *alias1* [*alias2...alias4*]
5. **maximum bandwidth** [**group** | **single**] *bandwidth*
6. **maximum bandwidth ingress** {**group** | **single**} *bandwidth*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface multilink 2`	Configures an interface and enters interface configuration mode.
Step 4	**ip rsvp policy local identity** *alias1* [*alias2...alias4*] **Example:** `Router(config-if)# ip rsvp policy local identity video`	Specifies an application ID alias for an application ID previously configured and enters local policy configuration mode.

Step 5	**maximum bandwidth** [**group** | **single**] *bandwidth*
	Example:
	maximum bandwidth percent {**group** | **single**} *bandwidth-percentage*
	Example:
	`Router(config-rsvp-local-if-policy)# maximum bandwidth group 500`
	Example:
	`Router(config-rsvp-local-if-policy)# maximum bandwidth percent group 50`
		Configures the maximum amount of bandwidth, in kb/s, that can be requested by single or group reservations covered by a local policy.
		or
		Configures a percentage of RSVP bandwidth of an interface as the maximum bandwidth available to single or group reservations covered by a local policy.

	Command or Action	Purpose

Step 6	**maximum bandwidth ingress** {**group** | **single**} *bandwidth*
	Example:
	maximum bandwidth ingress percent {**group** | **single**} *percent*
	Example:
	`Router(config-rsvp-local-policy)# maximum bandwidth ingress group 200`
	Example:
	`Router(config-rsvp-local-if-policy)# maximum bandwidth ingress percent group 50`
		Configures the maximum ingress bandwidth for a group of reservations or for a single reservation in a global-based RSVP policy.
		or
		Configures the maximum percentage of RSVP ingress bandwidth of an interface for a group of reservations or for a single reservation.

Step 7	**end**
	Example:
	`Router(config-rsvp-local-if-policy)# end`
		Exits local policy configuration mode and returns to privileged EXEC mode.

Entering Senders or Receivers in the RSVP Database

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **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*
4. **ip rsvp reservation** *session-ip-address sender-ip-address* [**tcp** | **udp** | *ip-protocol*] *session-dport sender-sport next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**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* **Example:** `Router(config)# ip rsvp sender 10.10.1.1 10.10.2.2 tcp 2 3 10.10.3.1 fastEthernet 0/1 2 3`	Enters the senders in the RSVP database. • Enables a router to behave like it is receiving and processing RSVP PATH messages from the sender or previous hop routes containing the indicated attributes. • The related **ip rsvp sender-host** command enables a router to simulate a host generating RSVP PATH messages. It is used mostly for debugging and testing purposes.
Step 4	**ip rsvp reservation** *session-ip-address sender-ip-address* [**tcp** | **udp** | *ip-protocol*] *session-dport sender-sport next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* **Example:** `Router(config)# ip rsvp reservation 10.0.0.4 10.0.0.5 tcp 2 3 10.0.0.3 fastEthernet 0/1 ff load 2 4`	Enters the receivers in the RSVP database and enables a router to behave like it is receiving and processing RSVP RESV messages. • The related **ip rsvp reservation-host** command enables a router to simulate a host generating RSVP RESV messages. It is used mostly for debugging and testing purposes.

Step 5	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Configuring RSVP as a Transport Protocol

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp transport client** *client-id*
4. **ip rsvp transport sender-host** [**tcp**| **udp**] *destination-address source-address ip-protocol dest-port source-port client-id init-id instance-id*[**vrf** *vrf-name*] [**data** *data-value*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp transport client** *client-id* **Example:** `Router(config)# ip rsvp transport client 2`	Creates an RSVP transport session. It enables a router to simulate a host generating RSVP PATH message. • This command is used for debugging and testing.

Step 4	**ip rsvp transport sender-host** [**tcp**| **udp**] *destination-address source-address ip-protocol dest-port source-port client-id init-id instance-id*[**vrf** *vrf-name*] [**data** *data-value*]
	Example:
	`Router(config)# ip rsvp transport sender-host tcp`
	`10.1.1.1 10.2..1.1 3 4 5 2 3 4 vrf vr1`
		Registers an RSVP transport client ID with RSVP.
		• This command is used for debugging and testing purposes.

Step 5	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Specifying Multicast Destinations

If RSVP neighbors are discovered to be using User Datagram Protocol (UDP) encapsulation, the router will automatically generate UDP-encapsulated messages for consumption by the neighbors.

However, in some cases, a host will not originate such a message until it has first heard from the router, which it can do only via UDP. You must instruct the router to generate UDP-encapsulated RSVP multicasts whenever it generates an IP-encapsulated multicast.

To specify multicast destinations that should receive UDP-encapsulated messages, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp udp-multicasts** [*multicast-address*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp udp-multicasts** [*multicast-address*] **Example:** `Router(config)# ip rsvp udp-multicasts 10.3.4.1`	Specifies multicast destinations that should receive UDP-encapsulated messages.

Step 4	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Controlling RSVP Neighbor Reservations

By default, any RSVP neighbor may offer a reservation request. To control which RSVP neighbors can offer a reservation request, perform the following task. When you perform this task, only neighbors conforming to the access list are accepted. The access list is applied to the IP header.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp neighbor** *access-list-number*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp neighbor** *access-list-number* **Example:** `Router(config)# ip rsvp neighbor 12`	Limits which routers may offer reservations.

Step 4	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Enabling RSVP to Attach to NetFlow

To enable RSVP to attach itself to NetFlow so that it can receive information about packets in order to update its token bucket and set IP precedence as required, perform the following task. This task is optional for the following reason: When the interface is configured with the **ip rsvp svc-required** command to use ATM switched virtual circuits (SVCs), RSVP automatically attaches itself to NetFlow to perform packet flow identification. However, if you want to perform IP Precedence-type of service (ToS) bit setting in every packet without using ATM SVCs, then you must use the **ip rsvp flow-assist** command to instruct RSVP to attach itself to NetFlow.

Note If you use WFQ, then the ToS and IP Precedence bits will be set only on data packets that RSVP sees, due to congestion.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp flow-assist**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp flow-assist** **Example:** `Router(config-if)# ip rsvp flow-assist`	Enables RSVP to attach itself to NetFlow.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Setting the IP Precedence and ToS Values

Note To configure the IP Precedence and ToS values to be used to mark packets in an RSVP reserved path that either conform to or exceed the RSVP flow specification (flowspec), perform the following task.You must configure the **ip rsvp flow-assist** command if you want to set IP Precedence or ToS values in every packet and you are not using ATM SVCs; that is, you have not configured the **ip rsvp svc-required** command.

The ToS byte in the IP header defines the three high-order bits as IP Precedence bits and the five low-order bits as ToS bits.

The router software checks the source and destination addresses and port numbers of a packet to determine if the packet matches an RSVP reservation. If a match exists, as part of its input processing, RSVP checks the packet for conformance to the flowspec of the reservation. During this process, RSVP determines if the packet conforms to or exceeds the flowspec, and it sets the IP header IP Precedence and ToS bits of the packet accordingly. These IP Precedence and ToS bit settings are used by per-VC Distributed Weighted Random Early Detection (DWRED) on the output interface, and they can be used by interfaces on downstream routers.

The combination of scheduling performed by the Enhanced ATM port adapter (PA-A3) and the per-SVC DWRED drop policy ensures that any packet that matches a reservation but exceeds the flowspec (that is, it does not conform to the token bucket for the reservation) is treated as if it were a best-effort packet. It is sent on the SVC for the reservation, but its IP precedence is marked to ensure that it does not interfere with conforming traffic.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp precedence** {**conform**| **exceed**} *precedence-value*
5. **ip rsvp tos** {**conform**| **exceed**} *tos-value*
6. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Router(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp precedence** {**conform**| **exceed**} *precedence-value* **Example:** `Router(config-if)# ip rsvp precedence conform 23`	Sets the IP Precedence conform or exceed values.
Step 5	**ip rsvp tos** {**conform**| **exceed**} *tos-value* **Example:** `Router(config-if)# ip rsvp tos conform 45`	Sets the ToS conform or exceed values.

Step 6	**end**
	Example:
	`Router(config-if)# end`
		Exits interface configuration mode and returns to privileged EXEC mode.

Configuring Tunnel Bandwidth Overhead

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip rsvp tunnel overhead-percent** [*overhead-percent*]
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Router(config)# interface tunnel 0`	Enters interface configuration mode.
Step 4	**ip rsvp tunnel overhead-percent** [*overhead-percent*] **Example:** `Router(config-if)# ip rsvp tunnel overhead-percent 20`	Configures the override value for the percentage bandwidth overhead within the tunnel interface.

Step 5	**end**
	Example:
	`Router(config-if)# end`
		Returns to privileged EXEC mode.

- Troubleshooting Tips, page 24

Troubleshooting Tips

You can use the **show ip rsvp interface detail** command to display the RSVP configuration parameters.

Sending RSVP Notifications

To allow a user on a remote management station to monitor RSVP-related information, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **snmp-server enable traps rsvp**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Router# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps rsvp** **Example:** `Router(config)# snmp-server enable traps rsvp`	Sends RSVP notifications.

Step 4	**end**
	Example:
	`Router(config)# end`
		Exits global configuration mode and returns to privileged EXEC mode.

Verifying RSVP Configuration

Perform this task to verify the resulting RSVP operations, after configuring the RSVP reservations that reflect your network resource policy. You can perform these steps in any order.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp interface** [*type number*]
3. **show ip rsvp installed** [*type number*]
4. **show ip rsvp neighbor** [*type number*]
5. **show ip rsvp sender** [*type number*]
6. **show ip rsvp request** [*type number*]
7. **show ip rsvp reservation** [*type number*]
8. **show ip rsvp ingress interface** [**detail**] [*type number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	Enables privileged EXEC mode. - Enter your password if prompted.
Step 2	**show ip rsvp interface** [*type number*] **Example:** `Router# show ip rsvp interface fastethernet 0/1`	Displays RSVP-related interface information.
Step 3	**show ip rsvp installed** [*type number*] **Example:** `Router# show ip rsvp installed fastethernet 0/1`	Displays RSVP-related filters and bandwidth information.
Step 4	**show ip rsvp neighbor** [*type number*] **Example:** `Router# show ip rsvp neighbor fastethernet 0/1`	Displays current RSVP neighbors.
Step 5	**show ip rsvp sender** [*type number*] **Example:** `Router# show ip rsvp sender fastethernet 0/1`	Displays RSVP sender information.
Step 6	**show ip rsvp request** [*type number*] **Example:** `Router# show ip rsvp request fastethernet 0/1`	Displays RSVP request information.
Step 7	**show ip rsvp reservation** [*type number*] **Example:** `Router# show ip rsvp reservation fastethernet 0/1`	Displays RSVP receiver information.

	Command or Action	Purpose
Step 8	**show ip rsvp ingress interface** [**detail**] [*type number*] **Example:** `Router# show ip rsvp ingress interface detail`	Displays RSVP ingress bandwidth information.

Configuration Examples for Configuring RSVP

Example Configuring RSVP for a Multicast Session

This section describes configuration of RSVP on three Cisco 4500 routers for a multicast session.

For information on how to configure RSVP, see the How to Configure RSVP, page 10.

The three routers form the router network between an RSVP sender application running on an upstream (end system) host and an RSVP receiver application running on a downstream (end system) host--neither host is shown in this example.

The router network includes three routers: Router A, Router B, and Router C. The example presumes that the upstream High-Speed Serial Interface (HSSI) interface 0 of Router A links to the upstream host. Router A and Router B are connected by the downstream Ethernet interface1 of Router A, which links to the upstream interface Ethernet 1 of Router B. Router B and Router C are connected by the downstream HSSI interface 0 of Router B, which links to the upstream HSSI interface 0 of Router C. The example presumes that the downstream Ethernet interface 2 of Router C links to the downstream host.

Typically, an RSVP-capable application running on an end system host on one side of a router network sends either unicast or multicast RSVP PATH (Set Up) messages to the destination end system or host on the other side of the router network with which it wants to communicate. The initiating application is referred to as the sender; the target or destination application is called the receiver. In this example, the sender runs on the host upstream from Router A and the receiver runs on the host downstream from Router C. The router network delivers the RSVP PATH messages from the sender to the receiver. The receiver replies with RSVP RESV messages in an attempt to reserve across the network the requested resources that are required between itself and the sender. The RSVP RESV messages specify the parameters for the requisite QoS that the router network connecting the systems should attempt to offer.

This example does not show the host that would run the sender application and the host that would run the receiver application. Normally, the first router downstream from the sender in the router network--in this case, Router A--would receive the RSVP PATH message from the sender. Normally, the last router in the router network--that is, the next hop upstream from the host running the receiver application, in this case, Router C--would receive an RSVP RESV message from the receiver.

Because this example does not explicitly include the hosts on which the sender and receiver applications run, the routers have been configured to act as if they were receiving PATH messages from a sender and RESV messages from a receiver. The commands used for this purpose, allowing RSVP to be more fully illustrated in the example, are the **ip rsvp sender** command and the **ip rsvp reservation** command. On Router A, the following command has been issued:

```
ip rsvp sender 225.1.1.1 10.1.2.1 UDP 7001 7000 10.1.2.1 Hs0 20 1
```

This command causes the router to act as if it were receiving PATH messages destined to multicast address 225.1.1.1 from a source 10.1.2.1. The previous hop of the PATH message is 10.1.2.1, and the message was received on HSSI interface 0.

On Router C, the following command has been issued:

```
ip rsvp reservation 225.1.1.1 10.1.2.1 UDP 7001 7000 10.1.3.1 Et2 FF LOAD 8 1
```

This command causes the router to act as if it were receiving RESV messages for the session with multicast destination 225.1.1.1. The messages request a Fixed Filter reservation to source 10.1.2.1, and act as if they had arrived from a receiver on Ethernet interface 2 with address 10.1.3.1.

In the example, the RSVP PATH messages flow in one direction: downstream from the sender, which in this example is Router A. (If the host were to initiate the RSVP PATH message, the message would flow from the host to Router A.) Router A sends the message downstream to Router B, and Router B sends it downstream to Router C. (If the downstream host were the actual receiver, Router C would send the RSVP PATH message downstream to the receiver host.) Each router in the router network must process the RSVP PATH message and route it to the next downstream hop.

The RSVP RESV messages flow in one direction: upstream from the receiver (in this example, Router C), upstream from Router C to Router B, and upstream from Router B to Router A. If the downstream host were the receiver, the message would originate with the host, which would send it to Router C. If the upstream host were the sender, the final destination of the RSVP RESV message would be the upstream host. At each hop, the router receiving the RSVP RESV message must determine whether it can honor the reservation request.

The **ip rsvp bandwidth** command both enables RSVP on an interface and specifies the amount of bandwidth on the interface that can be reserved (and the amount of bandwidth that can be allocated to a single flow). To ensure QoS for the RSVP reservation, WFQ is configured on the interfaces enabled for the reservation.

If the router network is capable of offering the specified (QoS) level of service, then an end-to-end reserved path is established. If not, the reservation attempt is rejected and a RESV ERROR message is sent to the receiver. The ability of each router in the network to honor the requested level of service is verified, link by link, as the RSVP RESV messages are sent across the router network to the sender. However, the data itself for which the bandwidth is reserved travels one way only: from the sender to receiver across an established PATH. Therefore, the QoS is effective in only one direction. This is the common case for one-to-many multicast data flows.

After the three routers in the example are configured, the **show ip rsvp sender** and **show ip rsvp reservation** commands will make visible the PATH and RESV state.

Router A Configuration

On Router A, RSVP is enabled on Ethernet interface 1 with 10 kb/s to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router A, RSVP is also enabled on HSSI interface 0 with 1 kb/s reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
```

```
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname routerA
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
!
!
interface Ethernet0
 ip address 172.0.0.193 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 172.1.1.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 10 10
 fair-queue 64 256 1000
 media-type 10BaseT
!
interface Hssi0
 ip address 10.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 10
 network 172.0.0.0 0.255.255.255 area 10
!
ip classless
ip rsvp sender 225.1.1.1 12.1.2.1 UDP 7001 7000 10.1.2.1 Hs0 20 1
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Router B Configuration

On Router B, RSVP is enabled on HSSI interface 0 with 20 kb/s to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router B, RSVP is also enabled on Ethernet interface 1 with 1 kb/s reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname routerB
```

```
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
clock calendar-valid
!
interface Ethernet0
 ip address 172.0.0.194 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 10.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
 media-type 10BaseT
!
interface Hssi0
 ip address 10.1.1.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 20 20
 fair-queue 64 256 1000
 hssi internal-clock
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 10
 network 172.0.0.0 0.255.255.255 area 10
!
ip classless
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Router C Configuration

On Router C, RSVP is enabled on Ethernet interface 2 with 20 kb/s to be reserved for the data transmission. A weighted fair queue is reserved on this interface to ensure RSVP QoS. (On Router C, RSVP is also enabled on HSSI interface 0 with 1 kb/s reserved, but this bandwidth is used simply for passing messages.)

```
!
version 12.0
service config
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service udp-small-servers
service tcp-small-servers
!
hostname routerC
!
ip subnet-zero
no ip domain-lookup
ip multicast-routing
ip dvmrp route-limit 20000
```

```
!
interface Ethernet0
 ip address 172.0.0.195 255.0.0.0
 no ip directed-broadcast
 no ip route-cache
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet2
 ip address 10.1.3.2 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 20 20
 fair-queue 64 256 1000
 media-type 10BaseT
!
interface Ethernet3
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet4
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Ethernet5
 no ip address
 no ip directed-broadcast
 shutdown
 media-type 10BaseT
!
interface Hssi0
 ip address 10.1.1.1 255.0.0.0
 no ip directed-broadcast
 ip pim dense-mode
 ip rsvp bandwidth 1 1
 hssi internal-clock
!
interface ATM0
 no ip address
 no ip directed-broadcast
 shutdown
!
router ospf 100
 network 10.0.0.0 0.255.255.255 area 10
 network 172.0.0.0 0.255.255.255 area 10
!
ip classless
ip rsvp reservation 225.1.1.1 10.1.2.1 UDP 7001 7000 10.1.3.1 Et2 FF LOAD 8 1
!
line con 0
 exec-timeout 0 0
 length 0
 transport input none
line aux 0
line vty 0 4
 login
!
end
```

Examples Configuring RSVP Bandwidth

The following example shows how to configure an absolute value for the RSVP bandwidth and percentage of interface as the flow bandwidth:

```
configure terminal
 interface multilink 2
  ip rsvp bandwidth 1000 percent 50
```

The following example shows how to configure a percentage of interface as the RSVP bandwidth and an absolute value for the flow bandwidth:

```
configure terminal
 interface multilink 2
  ip rsvp bandwidth percent 50 1000
```

The following example shows how to configure an absolute value for the RSVP bandwidth and the flow bandwidth:

```
configure terminal
 interface multilink 2
  ip rsvp bandwidth 23 34
```

The following example shows how to configure a percentage of RSVP bandwidth of an interface that should be the limit for a group of flows in an interface level RSVP policy:

```
configure terminal
 interface multilink 2
 ip rsvp policy local identity id1
 maximum bandwidth percent group 80
 maximum bandwidth percent single 5
 end
```

The following example shows how to verify the configuration of percentage of RSVP bandwidth that should be the limit for a group of flows:

```
Router# show running interface multilink 2
Building configuration...
Current configuration : 298 bytes
!
interface Multilink2
 ip address 30.30.30.1 255.255.255.0
 ip ospf cost 100
 ppp multilink
 ppp multilink group 2
 ppp multilink endpoint ip 30.30.30.2
 ip rsvp policy local identity id1
  maximum bandwidth percent group 80
  maximum bandwidth percent single 5
 ip rsvp bandwidth percent 50 percent 10
end
```

The following example shows how to configure RSVP ingress bandwidth for an interface:

```
enable
 configure terminal
  interface tunnel 0
   ip rsvp bandwidth ingress 200
```

The following example shows how to configure the maximum ingress bandwidth for a group of reservations and for a single reservation respectively, in a global-based RSVP policy:

```
enable
 configure terminal
```

```
  ip rsvp local identity rsvp-video
   maximum bandwidth ingress group 200
   maximum bandwidth ingress single 100
The following example shows how to configure the maximum percentage of RSVP ingress
bandwidth of an interface for a group of reservations and for a single reservation,
respectively:
enable
 configure terminal
  interface tunnel 0
   ip rsvp local identity rsvp-video
    maximum bandwidth ingress percent group 50
    maximum bandwidth ingress single 50
```

The following example shows how to verify the ingress CAC parameters on an interface:

```
Router# show ip rsvp ingress interface detail ethernet 1/0
interface    rsvp  in-allocated  in-i/f max  in-flow max  VRF
Et1/0        ena   0             7500K       7500K        0
```

Example Configuring Tunnel Bandwidth Overhead

The following example shows how to configure tunnel bandwidth overhead:

```
configure terminal
 interface tunnel 0
  ip rsvp overhead-percent 25
```

end

You can use the **show ip rsvp interface**, **show ip rsvp interface detail**and **show ip rsvp reservation**commands to verify the RSVP configuration parameters:

```
Router# show ip rsvp interface detail
Tu0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 10K bits/sec
     Max. allowed (total): 75K bits/sec
     Max. allowed (per flow): 75K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Admission Control:
     Header Compression methods supported:
       rtp (36 bytes-saved), udp (20 bytes-saved)
     Tunnel IP Overhead percent:
       4
     Tunnel Bandwidth considered:
       Yes
   Traffic Control:
     RSVP Data Packet Classification is ON via CEF callbacks
   Signalling:
     DSCP value used in RSVP msgs: 0x3F
     Number of refresh intervals to enforce blockade state: 4
   Authentication: disabled
     Key chain:   <none>
     Type:        md5
     Window size: 1
     Challenge:   disabled
   Hello Extension:
     State: Disabled


Router# show ip rsvp interface
interface    rsvp       allocated  i/f max  flow max sub max  VRF
Et0/0        ena        10400      7500K    7500K    0
Et1/0        ena        20K        7500K    7500K    0
```

Tu0 ena 10400 750K 750K 0

```
Router# show ip rsvp reservation
To             From           Pro DPort Sport Next Hop       I/F       Fi Serv BPS
192.168.2.2    192.168.1.2    TCP 10    10    192.168.2.2    Tu0       SE RATE 10K
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Overview on RSVP	*Signalling Overview*

Standards

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

MIBs

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

RFCs

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

Technical Assistance

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

Feature Information for Configuring RSVP

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

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

Table 2 Feature Information for Configuring RSVP

Feature Name	Releases	Feature Information
RSVP--Resource Reservation Protocol	11.2(1) 12.2(28)SB	RSVP is an IP service that allows end systems or hosts on either side of a router network to establish a reserved-bandwidth path between them to predetermine and ensure QoS for their data transmission. The following commands were introduced or modified: **ip rsvp bandwidth**, **ip rsvp flow-assist**, **ip rsvp neighbor**, **ip rsvp reservation**, **ip rsvp sender**.

Feature Name	Releases	Feature Information
RSVP for Flexible BW Interface	15.1(1)S 15.1(2)T	The RSVP for Flexible BW Interface feature allows you to configure a percentage of the interface bandwidth as the RSVP bandwidth. In Cisco IOS Release 15.1(2)T, this feature was introduced. In Cisco IOS Release 15.1(1)S, this feature was implemented on 7600 Series Routers. The following sections provide information about this feature: The following commands were introduced or modified: **ip rsvp bandwidth percent**, **maximum bandwidth percent**.
RSVP Over DMVPN	15.1(1)S 15.1(2)T	The RSVP over DMVPN feature supports the implementation of RSVP over manually configured and DMVPN IP tunnels. In Cisco IOS Release 15.1(2)T, this feature was introduced. In Cisco IOS Release 15.1(1)S, this feature was implemented on Cisco 7600 series routers. The following sections provide information about this feature: The following commands were introduced or modified: **ip rsvp bandwidth ignore**, **ip rsvp tunnel overhead-percent**, **show ip rsvp interface detail**.

Feature Name	Releases	Feature Information
RSVP Ingress CAC	15.1(1)S 15.1(3)T	The RSVP Ingress CAC feature extends the Cisco IOS RSVP IPv4 implementation to guarantee bandwidth resources not only on a given flow's outgoing interface, but also on the inbound interfaces. In Cisco IOS Release 15.1(3)T, this feature was introduced. In Cisco IOS Release 15.1(1)S, this feature was implemented on Cisco 7600 series routers. The following sections provide information about this feature: The following commands were introduced or modified: **ip rsvp bandwidth**, **maximum bandwidth ingress**, **show ip rsvp ingress**.
RSVP Transport for Medianet	15.1(3)T 15.1(3)S	The RSVP Transport for Medianet feature extends RSVP to act as a transport mechanism for the clients. The following section provides information about this feature: The following commands were introduced or modified: **ip rsvp transport**, **ip rsvp transport sender-host**, **show ip rsvp transport**, **show ip rsvp transport sender**.
NAT Aware RSVP	15.2(2)T	The NAT Aware RSVP feature enables the RSVP-NAT-ALG functionality. With the RSVP-NAT-ALG functionality enabled, when the RSVP messages pass through a NAT device, the IP addresses embedded in the RSVP payload get translated appropriately. The following commands were introduced: **show ip nat translations rsvp**

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.