QoS: RSVP Configuration Guide Cisco IOS Release 15M&T

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

CONTENTS

Signalling Overview

In the most general sense, QoS signalling is a form of network communication that allows an end station or network node to communicate with, or signal, its neighbors to request special handling of certain traffic. QoS signalling is useful for coordinating the traffic handling techniques provided by other QoS features. It plays a key role in configuring successful overall end-to-end QoS service across your network.

True end-to-end QoS requires that every element in the network path--switch, router, firewall, host, client, and so on--deliver its part of QoS, and that all of these entities be coordinated with QoS signalling.

Many viable QoS signalling solutions provide QoS at some places in the infrastructure; however, they often have limited scope across the network. To achieve end-to-end QoS, signalling must span the entire network.

Cisco IOS QoS software takes advantage of IP to meet the challenge of finding a robust QoS signalling solution that can operate over heterogeneous network infrastructures. It overlays Layer 2 technology-specific QoS signalling solutions with Layer 3 IP QoS signalling methods of the Resource Reservation Protocol (RSVP) and IP Precedence features.

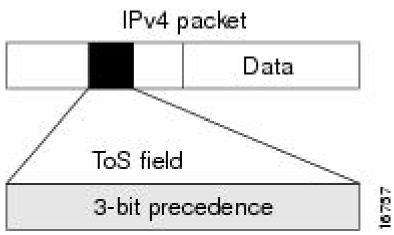
An IP network can achieve end-to-end QoS, for example, by using part of the IP packet header to request special handling of priority or time-sensitive traffic. Given the ubiquity of IP, QoS signalling that takes advantage of IP provides powerful end-to-end signalling. Both RSVP and IP Precedence fit this category.

Either in-band (IP Precedence, 802.1p) or out-of-band (RSVP) signalling is used to indicate that a particular QoS is desired for a particular traffic classification. IP Precedence signals for differentiated QoS, and RSVP for guaranteed QoS.

IP Precedence

As shown in the figure below, the IP Precedence feature utilizes the three precedence bits in the type of service (ToS) field of the IP version 4 (IPv4) header to specify class of service for each packet. You can partition traffic in up to six classes of service using IP precedence. The queueing technologies throughout the network can then use this signal to provide the appropriate expedited handling.

Figure 1 IP Precedence ToS Field



You can use features such as policy-based routing (PBR) and committed access rate (CAR) to set precedence based on extended access list classification. Use of these features allows considerable flexibility of precedence assignment, including assignment by application or user, or by destination or source subnet. Typically, you deploy these features as close to the edge of the network or the administrative domain as possible, so that each subsequent network element can provide service based on the determined policy. IP precedence can also be set in the host or the network client; however, IP precedence can be overridden by policy within the network.

IP precedence enables service classes to be established using existing network queueing mechanisms, such as weighted fair queueing (WFQ) and Weighted Random Early Detection (WRED), with no changes to existing applications and with no complicated network requirements.

Resource Reservation Protocol

RSVP is the first significant industry-standard protocol for dynamically setting up end-to-end QoS across a heterogeneous network. RSVP, which runs over IP, allows an application to dynamically reserve network bandwidth. Using RSVP, applications can request a certain level of QoS for a data flow across a network.

The Cisco IOS QoS implementation allows RSVP to be initiated within the network using configured proxy RSVP. Using this capability, you can take advantage of the benefits of RSVP in the network even for non-RSVP enabled applications and hosts. RSVP is the only standard signalling protocol designed to guarantee network bandwidth from end-to-end for IP networks.

RSVP does not perform its own routing; instead it uses underlying routing protocols to determine where it should carry reservation requests. As routing changes paths to adapt to topology changes, RSVP adapts its reservation to the new paths wherever reservations are in place. This modularity does not prevent RSVP from using other routing services. RSVP provides transparent operation through device nodes that do not support RSVP.

RSVP works in conjunction with, not in place of, current queueing mechanisms. RSVP requests the particular QoS, but it is up to the particular interface queueing mechanism, such as WFQ or WRED, to implement the reservation.

You can use RSVP to make two types of dynamic reservations: controlled load and guaranteed rate services, both of which are briefly described in the chapter "Quality of Service Overview" in this book.

A primary feature of RSVP is its scalability. RSVP scales well using the inherent scalability of multicast. RSVP scales to very large multicast groups because it uses receiver-oriented reservation requests that merge as they progress up the multicast tree. Although RSVP is designed specifically for multicast applications, it may also make unicast reservations. However, it does not scale as well with a large number of unicast reservations.

RSVP is an important QoS feature, but it does not solve all problems addressed by QoS, and it imposes a few hindrances, such as the time required to set up end-to-end reservation.

How It Works

Hosts and devices use RSVP to deliver QoS requests to the devices along the paths of the data stream and to maintain device and host state to provide the requested service, usually bandwidth and latency. RSVP uses a mean data rate--the largest amount of data the device will keep in the queue--and minimum QoS (that is, guarantee of the requested bandwidth specified when you made the reservation using RSVP) to determine bandwidth reservation.

A host uses RSVP to request a specific QoS service from the network on behalf of an application data stream. RSVP requests the particular QoS, but it is up to the interface queueing mechanism to implement the reservation. RSVP carries the request through the network, visiting each node the network uses to carry the stream. At each node, RSVP attempts to make a resource reservation for the stream using its own admission control module, exclusive to RSVP, which determines whether the node has sufficient available resources to supply the requested QoS.

Note For RSVP, an application could send traffic at a rate higher than the requested QoS, but the application is guaranteed only the minimum requested rate. If bandwidth is available, traffic surpassing the requested rate will go through if sent; if bandwidth is not available, the exceeding traffic will be dropped.

If the required resources are available and the user is granted administrative access, the RSVP daemon sets arguments in the packet classifier and packet scheduler to obtain the desired QoS. The classifier determines the QoS class for each packet and the scheduler orders packet transmission to achieve the promised QoS for each stream. If either resource is unavailable or the user is denied administrative permission, the RSVP program returns an error notification to the application process that originated the request.

WFQ or WRED sets up the packet classification and the scheduling required for the reserved flows. Using WFQ, RSVP can deliver an integrated services Guaranteed Rate Service. Using WRED, it can deliver a Controlled Load Service.

For information on how to configure RSVP, see the chapter "Configuring RSVP" in this book.

RSVP Support for Low Latency Queueing

RSVP is a network-control protocol that provides a means for reserving network resources--primarily bandwidth--to guarantee that applications sending end-to-end across networks achieve the desired QoS.

RSVP enables real-time traffic (which includes voice flows) to reserve resources necessary for low latency and bandwidth guarantees.

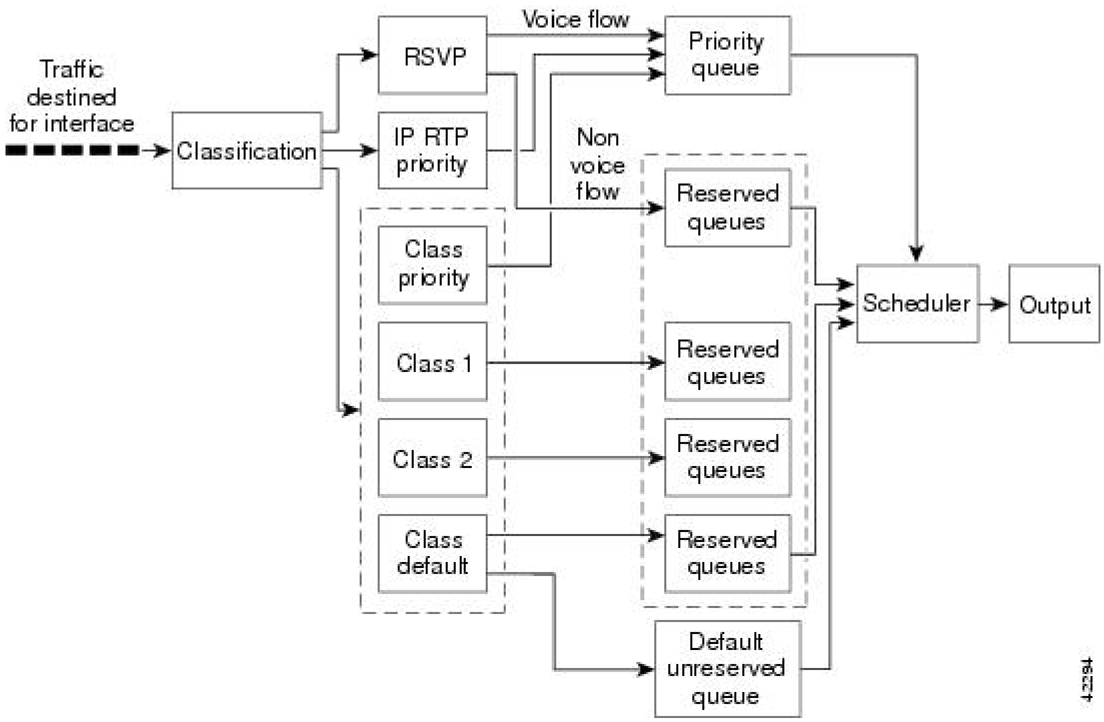
Voice traffic has stringent delay and jitter requirements. It must have very low delay and minimal jitter per hop to avoid degradation of end-to-end QoS. This requirement calls for an efficient queueing implementation, such as low latency queueing (LLQ), that can service voice traffic at almost strict priority in order to minimize delay and jitter.

RSVP uses WFQ to provide fairness among flows and to assign a low weight to a packet to attain priority. However, the preferential treatment provided by RSVP is insufficient to minimize the jitter because of the nature of the queueing algorithm itself. As a result, the low latency and jitter requirements of voice flows might not be met in the prior implementation of RSVP and WFQ.

RSVP provides admission control. However, to provide the bandwidth and delay guarantees for voice traffic and get admission control, RSVP must work with LLQ. The RSVP Support for LLQ feature allows RSVP to classify voice flows and queue them into the priority queue within the LLQ system while simultaneously providing reservations for nonvoice flows by getting a reserved queue.

The figure below shows how RSVP operates with other Voice over IP (VoIP) features, such as **ip rtp priority**, using the same queueing mechanism, LLQ.

Figure 2 RSVP Support for LLQ



RSVP is the only protocol that provides admission control based on the availability of network resources such as bandwidth. LLQ provides a means to forward voice traffic with strict priority ahead of other data traffic. When combined, RSVP support for LLQ provides admission control and forwards voice flows with the lowest possible latency and jitter.

High priority nonvoice traffic from mission-critical applications can continue to be sent without being adversely affected by voice traffic.

Nonconformant traffic receives best-effort treatment, thereby avoiding any degradation that might otherwise occur for all traffic.

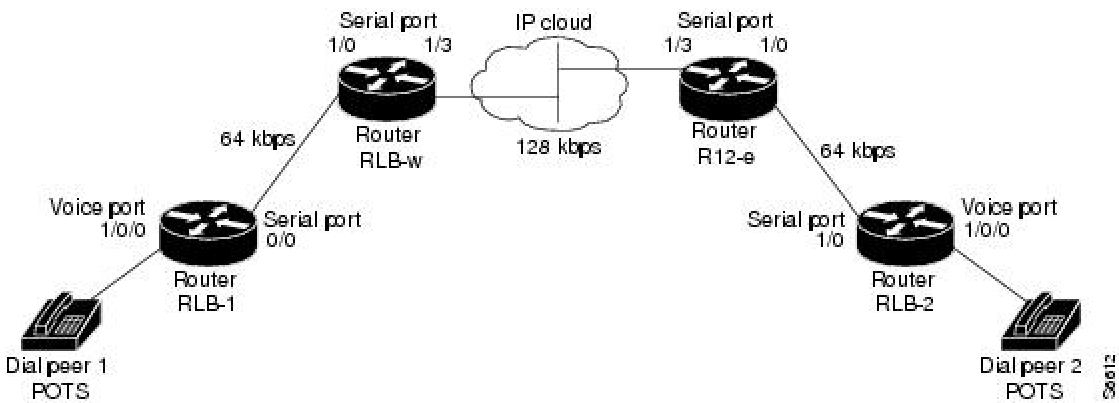
The RSVP Support for LLQ feature supports the following RFCs:

- RFC 2205, *Resource Reservation Protoco* l
- RFC 2210, *RSVP with IETF Integrated Services*
- RFC 2211, *Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

The figure below shows a sample network topology with LLQ running on each interface. This configuration guarantees QoS for voice traffic.

Note If the source is incapable of supporting RSVP, then the device can proxy on behalf of the source.

Figure 3 Topology Showing LLQ on Each Interface



For information on how to configure the RSVP Support for LLQ feature, see the "Configuring RSVP Support for LLQ" module.

- Restrictions, page 5
- Prerequisites, page 5

Restrictions

The following restrictions apply to the RSVP Support for LLQ feature:

- The LLQ is not supported on any tunnels.
- RSVP support for LLQ is dependent on the priority queue. If LLQ is not available on any interface or platform, then RSVP support for LLQ is not available.

Prerequisites

The network must support the following Cisco IOS features before RSVP support for LLQ is enabled:

- RSVP
- WFQ or LLQ (WFQ with priority queue support)

RSVP Support for Frame Relay

Network administrators use queueing to manage congestion on a device interface or a virtual circuit (VC). In a Frame Relay environment, the congestion point might not be the interface itself, but the VC because of the committed information rate (CIR). For real-time traffic (voice flows) to be sent in a timely manner, the data rate must not exceed the CIR or packets might be dropped, thereby affecting voice quality. Frame Relay Traffic Shaping (FRTS) is configured on the interfaces to control the outbound traffic rate by preventing the device from exceeding the CIR. This type of configuration means that fancy queueing such as class-based WFQ (CBWFQ), LLQ, or WFQ, can run on the VC to provide the QoS guarantees for the traffic.

Previously, RSVP reservations were not constrained by the CIR of the outbound VC of the flow. As a result, oversubscription could occur when the sum of the RSVP traffic and other traffic exceeded the CIR.

The RSVP Support for Frame Relay feature allows RSVP to function with per-VC (data-link connection identifier (DLCI) queueing for voice-like flows. Traffic shaping must be enabled in a Frame Relay environment for accurate admission control of resources (bandwidth and queues) at the congestion point, that is, the VC itself. Specifically, RSVP can function with VCs defined at the interface and subinterface levels. There is no limit to the number of VCs that can be configured per interface or subinterface.

- RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI), page 6
- Benefits, page 6
- Restrictions, page 7
- Prerequisites, page 7

RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI)

RSVP can use an interface (or a PVC) queueing algorithm, such as WFQ, to ensure QoS for its data flows.

- Admission Control, page 6
- Data Packet Classification, page 6

Admission Control

When WFQ is running, RSVP can co-exist with other QoS features on an interface (or PVC) that also reserve bandwidth and enforce QoS. When you configure multiple bandwidth-reserving features (such as RSVP, LLQ, CB-WFQ, and **ip rtp priority**), portions of the interface's (or PVC's) available bandwidth may be assigned to each of these features for use with flows that they classify.

An internal interface-based (or PVC-based) bandwidth manager prevents the amount of traffic reserved by these features from oversubscribing the interface (or PVC). You can view this pool of available bandwidth using the **show queue** command.

When you configure features such as LLQ and CB-WFQ, any classes that are assigned a bandwidth reserve their bandwidth at the time of configuration, and deduct this bandwidth from the bandwidth manager. If the configured bandwidth exceeds the interface's capacity, the configuration is rejected.

When RSVP is configured, no bandwidth is reserved. (The amount of bandwidth specified in the **ip rsvp bandwidth** command acts as a strict upper limit, and does **not** guarantee admission of any flows.) Only when an RSVP reservation arrives does RSVP attempt to reserve bandwidth out of the remaining pool of available bandwidth (that is, the bandwidth that has not been dedicated to traffic handled by other features.)

Data Packet Classification

By default, RSVP performs an efficient flow-based, datapacket classification to ensure QoS for its reserved traffic. This classification runs prior to queueing consideration by **ip rtp priority** or CB-WFQ. Thus, the use of a CB-WFQ class or **ip rtp priority** command is **not**required in order for RSVP data flows to be granted QoS. Any **ip rtp priority** or CB-WFQ configuration will not match RSVP flows, but they will reserve additional bandwidth for any non-RSVP flows that may match their classifiers.

Benefits

The benefits of this feature include the following:

- RSVP now provides admission control based on the VC minimum acceptable outgoing (minCIR) value, if defined, instead of the amount of bandwidth available on the interface.
- RSVP provides QoS guarantees for high priority traffic by reserving resources at the point of congestion, that is, the Frame Relay VC instead of the interface.
- RSVP provides support for point-to-point and multipoint interface configurations, thus enabling deployment of services such as VoIP in Frame Relay environments with QoS guarantees.
- RSVP, CBWFQ, and the **ip rtp priority** command do not oversubscribe the amount of bandwidth available on the interface or the VC even when they are running simultaneously. Prior to admitting a reservation, these features (and the **ip rtp priority**command) consult with an internal bandwidth manager to avoid oversubscription.
- IP QoS features can now be integrated seamlessly from IP into Frame Relay environments with RSVP providing admission control on a per-VC (DLCI) basis.

The RSVP Support for Frame Relay feature supports the following MIB and RFCs:

- RFC 2206, *RSVP Management Information Base using SMIv2*
- RFC 220, *Resource Reservation Protocol*
- RFC 2210, *RSVP with IETF Integrated Services*
- RFC 221, *Controlled-Load Network Element Service*
- RFC 2212, *Specification of Guaranteed Quality of Service*
- RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

For information on how to configure RVSP Support for Frame Relay, see the "Configuring RSVP Support for Frame Relay" module.

Restrictions

The following restrictions apply to the RSVP Support for Frame Relay feature:

- Interface-level Generic Traffic Shaping (GTS) is not supported.
- VC-level queueing and interface-level queueing on the same interface are not supported.
- Nonvoice RSVP flows are not supported.
- Multicast flows are not supported.

Prerequisites

The network must support the following Cisco IOS features before RSVP support for Frame Relay is enabled:

- RSVP
- WFQ on the VC
- LLQ
- Frame Relay Forum (FRF).12 on the interface

RSVP-ATM QoS Interworking

The RSVP-ATM QoS Interworking feature provides support for Controlled Load Service using RSVP over an ATM core network. This feature requires the ability to signal for establishment of switched virtual circuits (SVCs) across the ATM cloud in response to RSVP reservation request messages. To meet this requirement, RSVP over ATM supports mapping of RSVP sessions to ATM SVCs.

The RSVP-ATM QoS Interworking feature allows you to perform the following tasks:

- Configure an interface or subinterface to dynamically create SVCs in response to RSVP reservation request messages. To ensure defined QoS, these SVCs are established having QoS profiles consistent with the mapped RSVP flow specifications (flowspecs).
- Attach Distributed Weighted Random Early Detection (DWRED) group definitions to the Enhanced ATM port adapter (PA-A3) interface to support per-VC DWRED drop policy. Use of per-VC DWRED ensures that if packets must be dropped, then best-effort packets are dropped first and not those that conform to the appropriate QoS determined by the token bucket of RSVP.
- Configure the IP Precedence and ToS values to be used for packets that conform to or exceed QoS profiles. As part of its input processing, RSVP uses the values that you specify to set the ToS and IP Precedence bits on incoming packets. If per-VC DWRED is configured, it then uses the ToS and IP Precedence bit settings on the output interface of the same device in determining which packets to drop. Also, interfaces on downstream devices use these settings in processing packets.

This feature is supported on Cisco 7500 series devices with a VIP2-50 and Enhanced ATM port adapter (PA-A3). The hardware provides the traffic shaping required by the feature and satisfies the OC-3 rate performance requirement.

How It Works

Traditionally, RSVP has been coupled with WFQ. WFQ provides bandwidth guarantees to RSVP and gives RSVP visibility to all packets visible to it. This visibility allows RSVP to identify and mark packets pertinent to it.

The RSVP-ATM QoS Interworking feature allows you to decouple RSVP from WFQ, and instead associate it with ATM SVCs to handle reservation request messages (and provide bandwidth guarantees) and NetFlow to make packets visible to RSVP.

To configure an interface or subinterface to use the RSVP-ATM QoS Interworking feature, use the **ip rsvp svc-required** command. Then, whenever a new RSVP reservation is requested, the device software establishes a new ATM SVC to service the reservation.

To ensure correspondence between RSVP and ATM SVC values, the software algorithmically maps the rate and burst size parameters in the RSVP flowspec to the ATM sustained cell rate (SCR) and maximum burst size (MBS). For the peak cell rate (PCR), it uses the value you configure or it defaults to the line rate. RSVP-ATM QoS Interworking requires an Enhanced ATM port adapter (PA-A3) with OC-3 speed.

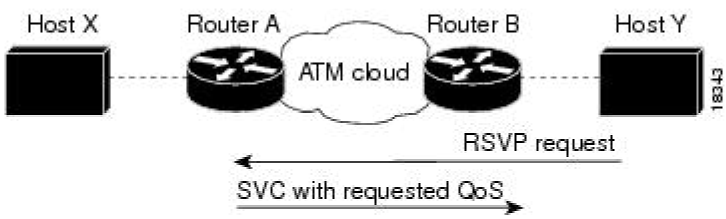
When a packet belonging to a reserved flow arrives on the interface or subinterface, the RSVP-ATM QoS Interworking software uses a token bucket to manage bandwidth guarantees. It measures actual traffic rates against the reservation flowspec to determine if the packet conforms to or exceeds the flowspec. Using values you configure for conformant or exceeding traffic, it sets the IP Precedence and ToS bits in the ToS byte of the header of the packet and delivers the packet to the appropriate virtual circuit (VC) for transmission. For the RSVP-ATM QoS Interworking feature, packets are shaped before they are sent on the ATM SVC. Shaping creates back pressure to the Versatile Interface Processor (VIP) when the offered load exceeds the rate.

The RSVP-ATM QoS Interworking software uses per-SVC DWRED to drop packets when shaping causes a queue to build up on the VIP. Use of per-SVC DWRED allows RSVP to deliver Controlled Load Service class, which requires that reserved packets experience performance equivalent to that of an unloaded network (which is one with very low loss and moderate delay). For a more detailed account of how the RSVP-ATM QoS Interworking feature works, see the following example scenario.

An Example Scenario

To understand the behavior of the RSVP-ATM QoS Interworking feature, consider the following example, which uses a Cisco 7500 router with VIP ingress and egress interfaces and RSVP ingress functionality implemented on the Route Switch Processor (RSP). The figure below illustrates this example; it shows a pair of routers that communicate over the ATM cloud. In this example, a single PVC is used for RSVP request messages and an ATM SVC is established to handle each new reservation request message.

Figure 4 Two Routers Connected over an ATM Core Network



Host X, which is upstream from Router A, is directly connected to Router A using FDDI. Host Y, which is downstream from Router B, is directly connected to Router B using FDDI. (In an alternative configuration, these host-device connections could use ATM VCs.)

For the RSVP-ATM QoS Interworking feature, reservations are needed primarily between devices across the ATM backbone network. To limit the number of locations where reservations are made, you can enable RSVP selectively only at subinterfaces corresponding to device-to-device connections across the ATM backbone network. Preventing reservations from being made between the host and the device both limits VC usage and reduces load on the device.

RSVP RESV messages flow from receiving host to sending host. In this example, Host Y is the sending host and Host X is the receiving host. (Host Y sends a RESV message to Host X.) Router B, which is at the edge of the ATM cloud, receives the RESV message and forwards it upstream to Router A across the PVC used for control messages. The example configuration shown in the figure above uses one PVC; as shown, it carries the RSVP request.

The ingress interface on Router A is configured for RSVP-ATM, which enables it to establish for each request an SVC to service any new RSVP RESV reservations made on the interface. When it receives a reservation request, the interface on Router A creates a new nonreal-time variable bit rate (nRTVBR) SVC with the appropriate QoS characteristics. The QoS characteristics used to establish the SVC result from algorithmic mapping of the flowspec in the RSVP RESV message to the appropriate set of ATM signalling parameters.

In this example, Controlled Load Service is used as the QoS class. The ATM PCR parameter is set to the line rate. If the **ip rsvp atm-peak-rate-limit** command is used on the interface to configure a rate limiter, the PCR is set to the peak rate limiter. The ATM SCR parameter is set to the RSVP flowspec rate and the ATM MBS is set to the RSVP flowspec burst size. Packets are shaped before they are sent on the ATM SVC. Shaping creates back pressure to the VIP when the offered load exceeds the rate.

When a new SVC is set up to handle a reservation request, another state is also set up including a classifier state that uses a source and destination addresses and port numbers of the packet to determine which, if any, reservation the packet belongs to. Also, a token bucket is set up to ensure that if a source sends more data than the data rate and MBS parameters of its flowspec specify, the excess traffic does not interfere with other reservations.

The following section describes more specifically, how data traverses the path.

When a data packet destined for Router B arrives at Router A, before they traverse the ATM cloud, the source and destination addresses and port numbers of the packet are checked against the RSVP filter specification (filterspec) to determine if the packet matches a reservation.

If the packet does not match a reservation, it is sent out the best-effort PVC to Router B. If a packet matches a reservation, it is further processed by RSVP. The packet is checked against the token bucket of the reservation to determine whether it conforms to or exceeds the token bucket parameters. (All packets matching a reservation are sent out on the SVC of the reservation to prevent misordering of packets.)

To introduce differentiation between flowspec-conformant and flowspec-exceeding packets, you can specify values for RSVP-ATM to use in setting the IP Precedence and ToS bits of the packets. To specify these values, you use the **ip rsvp precedence** and **ip rsvp tos** commands. When you set different precedence values for conformant and exceeding packets and use a preferential drop policy such as DWRED, RSVP-ATM ensures that flowspec-exceeding packets are dropped prior to flowspec-conformant packets when the VC is congested.

For information on how to configure the RSVP-ATM QoS Interworking feature, see the "Configuring RSVP-ATM QoS Interworking" module.

COPS for RSVP

Common Open Policy Service (COPS) is a protocol for communicating network traffic policy information to network devices. RSVP is a means for reserving network resources--primarily bandwidth--to guarantee that applications sending end-to-end across the Internet will perform at the desired speed and quality.

Combined, COPS with RSVP gives network managers centralized monitoring and control of RSVP, including the following abilities:

- Ensure adequate bandwidth and jitter and delay bounds for time-sensitive traffic such as voice transmission
- Ensure adequate bandwidth for multimedia applications such as video conferencing and distance learning
- Prevent bandwidth-hungry applications from delaying top priority flows or harming the performance of other applications customarily run over the same network

In so doing, COPS for RSVP supports the following crucial RSVP features:

- Admission control. The RSVP reservation is accepted or rejected based on end-to-end available network resources.
- Bandwidth guarantee. The RSVP reservation, if accepted, will guarantee that those reserved resources will continue to be available while the reservation is in place.
- Media-independent reservation. An end-to-end RSVP reservation can span arbitrary lower layer media types.
- Data classification. While a reservation is in place, data packets belonging to that RSVP flow are separated from other packets and forwarded as part of the reserved flow.
- Data policing. Data packets belonging to an RSVP flow that exceed the reserved bandwidth size are marked with a lower packet precedence.

Note In order to use the COPS for RSVP feature, your network must be running Cisco IOS 12.1(1)T or later releases. Moreover, a compatible policy server must be connected to the network, such as the Cisco COPS QoS Policy Manager.

Note The Cisco IOS 12.1(2)T release of COPS for RSVP does not support RSVP+.

COPS for RSVP functions on the following interfaces:

- Ethernet
- Fast Ethernet
- High-Speed Serial Interface (HSSI): V.35, EIA/TIA-232
- T1

The COPS for RSVP feature supports the following RFCs:

- RFC 2749, *COPS Usage for RSVP*
- RFC 2205, Resource ReSerVation Protocol (RSVP)
- RFC 2748, The COPS (Common Open Policy Service) Protocol

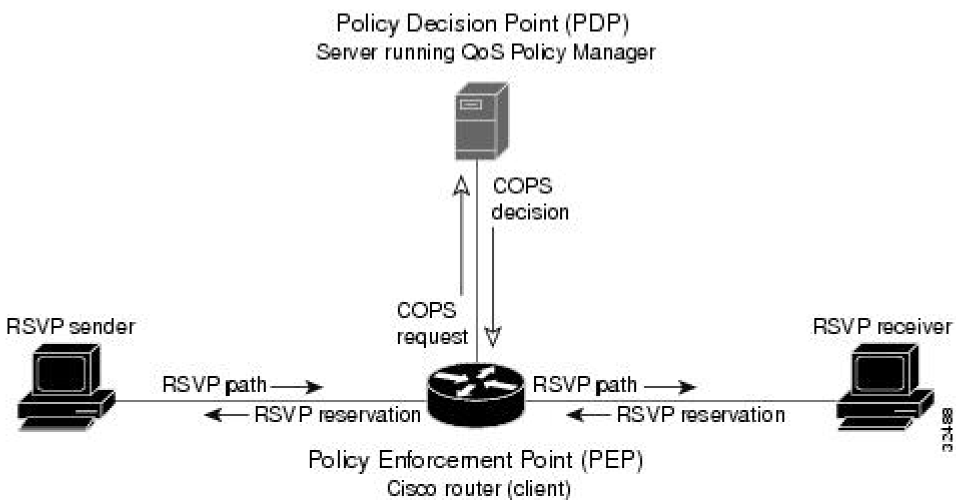
- How It Works, page 11

How It Works

This section provides a high-level overview of how the COPS for RSVP feature works on your network, and provides the general steps for configuring the COPS for RSVP feature.

The figure below is a sample arrangement of COPS with RSVP.

Figure 5 Sample Arrangement of COPS with RSVP



To configure a device to process all RSVP messages coming to it according to policies stored on a particular policy server (called the Policy Decision Point, or PDP), perform the following steps:

1. At the PDP server enter the policies using the Cisco COPS QoS Policy Manager or a compatible policy manager application.
2. Configure the device (through its command-line interface) to request decisions from the server regarding RSVP messages.

After that configuration, network flows are processed by the device designated as the Policy Enforcement Point (PEP), as follows:

1. When an RSVP signalling message arrives at the device, the device asks the PDP server how to process the message, either to accept, reject, forward, or install the message.
2. The PDP server sends its decision to the device, which then processes the message as instructed.
3. Alternatively, you may configure the device to make those decisions itself ("locally") without it needing to consult first with the PDP server. (The local feature is not supported in this release but will be in a future release.)

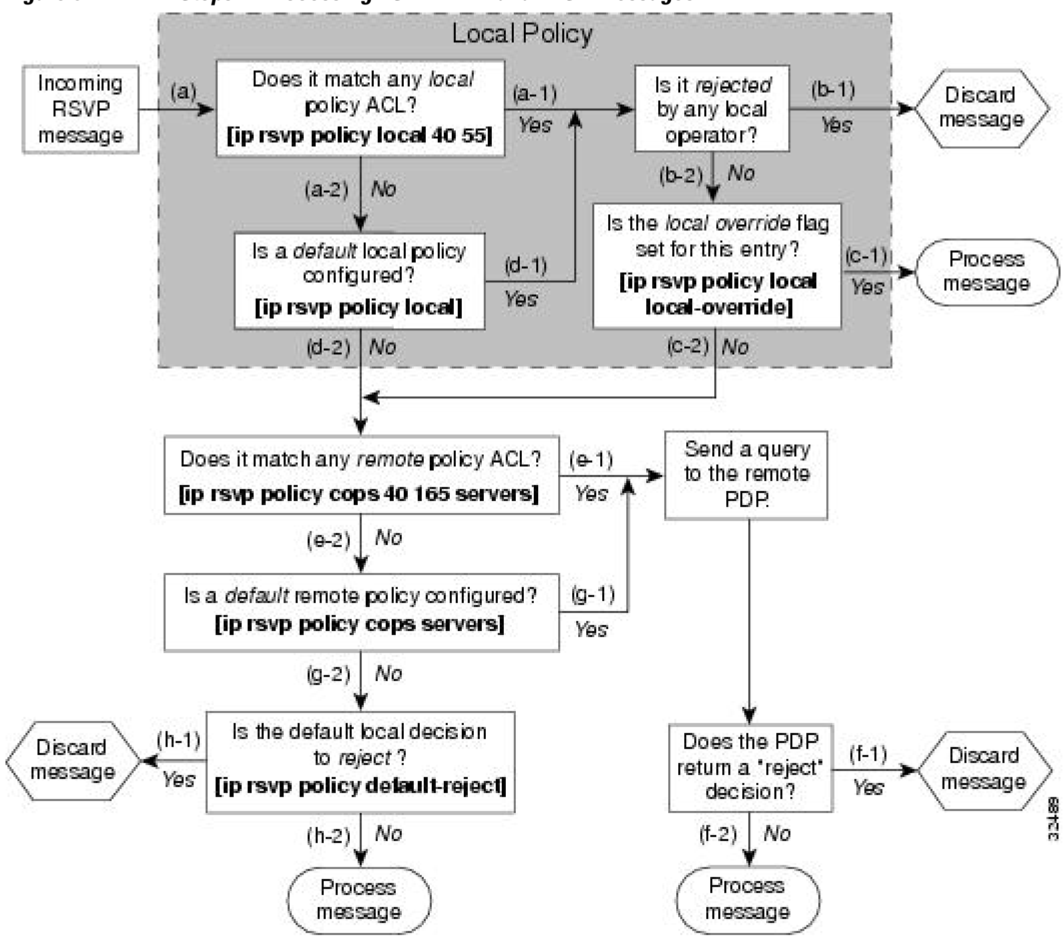
- A Detailed Look at COPS for RSVP Functioning, page 12

A Detailed Look at COPS for RSVP Functioning

The figure below traces options available in policy management of RSVP message flows. For each option, an example of the device configuration command used for setting that option is given in brackets and boldface type.

The shaded area covers local policy operations; the remainder of the figure illustrates remote policy operation. (Configuring local policy will be available in a future release.)

Figure 6 Steps in Processing RSVP PATH and RESV Messages



The following information is keyed to the figure:

1. The device receives a PATH or RESV message and first tries to adjudicate it locally (that is, without referring to the policy server). If the device has been configured to adjudicate specific access control lists (ACLs) locally and the message matches one of those lists (a-1), the policy module of the device applies the operators with which it had been configured. Otherwise, policy processing continues (a-2).
2. For each message rejected by the operators, the device sends an error message to the sender and removes the PATH or RESV message from the database (b-1). If the message is not rejected, policy processing continues (b-2).
3. If the local override flag is set for this entry, the message is immediately accepted with the specified policy operators (c-1). Otherwise, policy processing continues (c-2).
4. If the message does not match any ACL configured for local policy (a-2), the device applies the default local policy (d-1). However, if no default local policy has been configured, the message is directed toward remote policy processing (d-2).
5. If the device has been configured with specific ACLs against specific policy servers (PDPs), and the message matches one of these ACLs, the device sends that message to the specific PDP for adjudication (e-1). Otherwise, policy processing continues (e-2).
6. If the PDP specifies a "reject" decision (f-1), the message is discarded and an error message is sent back to the sender, indicating this condition. If the PDP specifies an "accept" decision (f-2), the message is accepted and processed using normal RSVP processing rules.
7. If the message does not match any ACL configured for specific PDPs (e-2), the device applies the *default* PDP configuration. If a default COPS configuration has been entered, policy processing continues (g-1). Otherwise, the message is considered to be unmatched (g-2).

If the default policy decision for unmatched messages is to reject (h-1), the message is immediately discarded and an ERROR message is sent to the sender indicating this condition. Otherwise, the message is accepted and processed using normal RSVP processing rules (h-2).

Here are additional details about PDP-PEP communication and processing:

- Policy request timer. Whenever a request for adjudication (of any sort) is sent to a PDP, a 30-second timer associated with the PATH or RESV message is started. If the timer runs out before the PDP replies to the request, the PDP is assumed to be down and the request is given to the default policy (step g-2 in the figure above).
- PDP tracking of PEP reservations. When the PDP specifies that a reservation can be installed, this reservation must then be installed on the device. Once bandwidth capacity has been allocated and the reservation installed, the policy module of the PEP sends a COMMIT message to the PDP. But if the reservation could not be installed because of insufficient resources, the reservation is folded back to the noninstalled state and a NO-COMMIT message is sent to the PDP. If the reservation was also new (no previous state), then a DELETE REQUEST message instead is sent to the PDP. In these ways, the PDP can keep track of reservations on the PEP.
- Resynchronization. If the PDP sends a SYNCHRONIZE-REQUEST message to the PEP, the policy module of the PEP scans its database for all paths and reservations that were previously adjudicated by this PDP, and resends requests for them. The previously adjudicated policy information is retained until a new decision is received. When all the PATH or RESV states have been reported to the PDP, a SYNCHRONIZE-COMPLETE message is sent by the policy module to the PDP. The PEP also sends queries concerning all flows that were locally adjudicated while the PDP was down.
- Readjudication:
    - So long as flows governed by the RSVP session continue to pass through the PEP device, the PDP can unilaterally decide to readjudicate any of the COPS decisions of that session. For example, the PDP might decide that a particular flow that was earlier granted acceptance now

needs to be rejected (due perhaps to a sudden preemption or timeout). In such cases, the PDP sends a new decision message to the PEP, which then adjusts its behavior accordingly.
    - If the PEP device receives a RESV message in which an object has changed, the policy decision needs to be readjudicated. For example, if the sender wants to increase or decrease the bandwidth reservation, a new policy decision must be made. In such cases, the policy flags previously applied to this session are retained, and the session is readjudicated.
- Tear-downs. The policy module of the PEP is responsible for notifying the PDP whenever a reservation or path that was previously established through policy is torn down for any reason. The PEP notifies the PDP by sending the PDP a DELETE REQUEST message.
- Connection management:
    - If the connection to the PDP is closed (either because the PDP closed the connection, a TCP/IP error occurred, or the keepalives failed), the PEP issues a CLIENT-CLOSE message and then attempts to reconnect to the same PDP. If the PEP receives a CLIENT-CLOSE message containing a PDP redirect address, the PEP attempts to connect to the redirected PDP.
    - If either attempt fails, the PEP attempts to connect to the PDPs previously specified in the configuration **ip rsvp policy cops servers** command, obeying the sequence of servers given in that command, always starting with the first server in that list.
    - If the PEP reaches the end of the list of servers without connecting, it waits a certain time (called the "reconnect delay") before trying again to connect to the first server in the list. This reconnect delay is initially 30 seconds, and doubles each time the PEP reaches the end of the list without having connected, until the reconnect delay becomes its maximum of 30 minutes. As soon as a connection is made, the delay is reset to 30 seconds.
- Replacement objects--The matrix in the table below identifies objects that the PDP can replace within RSVP messages passing through the PEP. An x in the column indicates that the PDP can replace the particular object within RSVP messages.

Table 1 Matrix for Objects the PDP Can Replace Within RSVP Messages

Message Context	Objects	Items Affected			
Policy	TSpec	Flowspec	Errorspec		
Path In	x	x	•--	•--	• Installed PATH state. • All outbound PATH messages for this PATH.
Path Out	x	x	•--	•--	This refresh of the PATH (but not the installed PATH state).

Message Context	Objects	Items Affected			
Resv In	x	•--	x	•--	• Installed RESV state (incoming and traffic control installation). • All outbound RESV messages for this RESV.
Resv Alloc	•--	•--	x	•--	Installed resources for this session.
Resv Out	x	•--	x	•--	This particular refresh of the RESV message (but not the installed RESV state nor the traffic control allocation).
PathError In	x	•--	•--	x	The forwarded PATHERROR message.
PathError Out	x	•--	•--	x	The forwarded PATHERROR message.
ResvError In	x	•--	•--	x	All RESVERROR messages forwarded by this device.
ResvError Out	x	•--	•--	x	This particular forwarded RESVERROR message.

If an RSVP message whose object was replaced is later refreshed from upstream, the PEP keeps track of both the old and new versions of the object, and does not wrongly interpret the refresh as a change in the PATH or RESV state.

For information on how to configure COPS for RSVP, see the chapter "Configuring COPS for RSVP" in this book.

Subnetwork Bandwidth Manager

RSVP and its service class definitions are largely independent of the underlying network technologies. This independence requires that a user define the mapping of RSVP onto subnetwork technologies.

The Subnetwork Bandwidth Manager (SBM) feature answers this requirement for RSVP in relation to IEEE 802-based networks. SBM specifies a signalling method and protocol for LAN-based admission control for RSVP flows. SBM allows RSVP-enabled devices and Layer 2 and Layer 3 devices to support reservation of LAN resources for RSVP-enabled data flows. The SBM signalling method is similar to that of RSVP itself. SBM protocol entities have the following features:

- Reside in Layer 2 or Layer 3 devices.
- Can manage resources on a segment. A segment is a Layer 2 physical segment shared by one or more senders, such as a shared Ethernet or Token Ring wire.
- Can become candidates in a dynamic election process that designates one SBM as the segment manager. The elected candidate is called the Designated Subnetwork Bandwidth Manager (DSBM). The elected DSBM is responsible for exercising admission control over requests for resource reservations on a managed segment.

A managed segment includes those interconnected parts of a shared LAN that are not separated by DSBMs. The presence of a DSBM makes the segment a managed one. One or more SBMs may exist on a managed segment, but there can be only one DSBM on each managed segment.

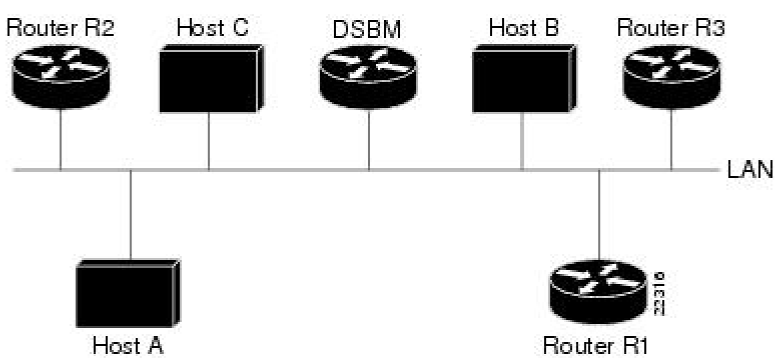
You can configure an interface on devices connected to the segment to participate in the DSBM election process. The contender configured with the highest priority becomes the DSBM for the managed segment.

If you do not configure a device as a DSBM candidate and RSVP is enabled, then the system interacts with the DSBM if a DSBM is present on the segment. In fact, if a DSBM, identifying itself as such, exists on the segment, the segment is considered a managed segment and all RSVP message forwarding will be based on the SBM message forwarding rules. This behavior exists to allow cases in which you might not want an RSVP-enabled interface on a device connected to a managed segment interface to become a DSBM, but you want it to interact with the DSBM if one is present managing the segment.

Note SBM is not supported currently on Token Ring LANs.

The figure below shows a managed segment in a Layer 2 domain that interconnects a set of hosts and devices.

Figure 7 DSBM Managed Segment



When a DSBM client sends or forwards an RSVP PATH message over an interface attached to a managed segment, it sends the PATH message to the DSBM of the segment instead of to the RSVP session destination address, as is done in conventional RSVP processing. As part of its message processing procedure, the DSBM builds and maintains a PATH state for the session and notes the previous Layer 2 or Layer 3 hop from which it received the PATH message. After processing the PATH message, the DSBM forwards it toward its destination address.

The DSBM receives the RSVP RESV message and processes it in a manner similar to how RSVP itself handles reservation request processing, basing the outcome on available bandwidth. The procedure is as follows:

- If it cannot grant the request because of lack of resources, the DSBM returns a RESVERROR message to the requester.
- If sufficient resources are available and the DSBM can grant the reservation request, it forwards the RESV message toward the previous hops using the local PATH state for the session.

For information on how to configure SBM, see the "Configuring Subnetwork Bandwidth Manager" module.

Configuring RSVP

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 device 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 caveats and feature information, see Bug Search Tool and 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 module.

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.

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. Devices 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 devices 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 device 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 device-to-device 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 device 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

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 devices. 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) devices run RSVP, and not the provider edge (PE) devices.

Figure 8 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

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 device 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 9 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 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 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 2 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.
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.

8-Tuple Parameters	Description
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 device. If the corresponding TP client does not exist on the device, 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 device. This continues until RSVP is able to deliver this message to the TP client.

If the corresponding TP client does not exist on the device, 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

This task starts RSVP and sets the bandwidth and single-flow limits. 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.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-bandwidth* [**percent** *percent-bandwidth*	[*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]] **Example:** `Device(config-if)# ip rsvp bandwidth 23 54`	Enables RSVP for IP on an interface.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(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:
	
	`Device(config-if)# ip rsvp bandwidth 23 34`
	
	Example:
	
	Example:
	
	`Device(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 45

	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:
	`Device(config-if)# ip rsvp bandwidth ingress 40`
	
	Example:
	
	Example:
	`Device(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:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface multilink 2`	Configures an interface and enters interface configuration mode.
Step 4	**ip rsvp policy local identity** *alias1* [*alias2...alias4*] **Example:** `Device(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:
	
	`Device(config-rsvp-local-if-policy)# maximum bandwidth group 500`
	
	Example:
	
	`Device(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:
	
	`Device(config-rsvp-local-policy)# maximum bandwidth ingress group 200`
	
	Example:
	
	`Device(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:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# 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:** `Device(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 device 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 device 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:** `Device(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 device to behave like it is receiving and processing RSVP RESV messages. • The related **ip rsvp reservation-host** command enables a device to simulate a host generating RSVP RESV messages. It is used mostly for debugging and testing purposes.
Step 5	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp transport client** *client-id* **Example:** `Device(config)# ip rsvp transport client 2`	Creates an RSVP transport session. It enables a device 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:** `Device(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:** `Device(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 device 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 device, which it can do only via UDP. You must instruct the device 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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp udp-multicasts** [*multicast-address*] **Example:** `Device(config)# ip rsvp udp-multicasts 10.3.4.1`	Specifies multicast destinations that should receive UDP-encapsulated messages.
Step 4	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp neighbor** *access-list-number* **Example:** `Device(config)# ip rsvp neighbor 12`	Limits which devices may offer reservations.
Step 4	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp flow-assist** **Example:** `Device(config-if)# ip rsvp flow-assist`	Enables RSVP to attach itself to NetFlow.
Step 5	**end** **Example:** `Device(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 device 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 devices.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp precedence** {**conform**| **exceed**} *precedence-value* **Example:** `Device(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:** `Device(config-if)# ip rsvp tos conform 45`	Sets the ToS conform or exceed values.
Step 6	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 0`	Enters interface configuration mode.
Step 4	**ip rsvp tunnel overhead-percent** [*overhead-percent*] **Example:** `Device(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:** `Device(config-if)# end`	Returns to privileged EXEC mode.

• Troubleshooting Tips, page 42

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**snmp-server enable traps rsvp** **Example:** `Device(config)# snmp-server enable traps rsvp`	Sends RSVP notifications.
Step 4	**end** **Example:** `Device(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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp interface** [*type number*] **Example:** `Device# show ip rsvp interface fastethernet 0/1`	Displays RSVP-related interface information.
Step 3	**show ip rsvp installed** [*type number*] **Example:** `Device# show ip rsvp installed fastethernet 0/1`	Displays RSVP-related filters and bandwidth information.
Step 4	**show ip rsvp neighbor** [*type number*] **Example:** `Device# show ip rsvp neighbor fastethernet 0/1`	Displays current RSVP neighbors.
Step 5	**show ip rsvp sender** [*type number*] **Example:** `Device# show ip rsvp sender fastethernet 0/1`	Displays RSVP sender information.
Step 6	**show ip rsvp request** [*type number*] **Example:** `Device# show ip rsvp request fastethernet 0/1`	Displays RSVP request information.
Step 7	**show ip rsvp reservation** [*type number*] **Example:** `Device# 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:** `Device# 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 28.

The three devices form the 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 network includes three devices: 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 network sends either unicast or multicast RSVP PATH (Set Up) messages to the destination end system or host on the other side of the 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 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 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 device downstream from the sender in the network--in this case, Router A--would receive the RSVP PATH message from the sender. Normally, the last router in the 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 devices 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 device 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 device 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 device in the 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 device 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 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 device in the network to honor the requested level of service is verified, link by link, as the RSVP RESV messages are sent across the 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 devices 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:

```
device# 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:

```
device# 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:

```
Device# 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


Device# 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

```
Device# 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 3 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 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 15.1(1)SY	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**

RSVP over UDP

The Resource Reservation Protocol (RSVP) over UDP feature provides the capability for routers to enable neighbor routers to process and send RSVP control packets over UDP. With the implementation of the RSVP over UDP feature, the RSVP protocol stack is enhanced to support processing of RSVP control messages over UDP and raw IP.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Over UDP

- You must enable RSVP before you enable the RSVP over UDP feature.
- The RSVP stack running on the client host must support sending and receiving the RSVP control messages with the first hop routers they are connected to.

Information About RSVP over UDP

RSVP over UDP

The RSVP over UDP feature addresses the following scenarios:

- A router intends to communicate to the first hop router over UDP but not raw IP.
- A firewall that is located in between two routers drops raw IP packets due to security concerns, but allows UDP packets.

How to Configure RSVP over UDP

- Enabling RSVP, page 58
- Configuring RSVP over UDP, page 59

Enabling RSVP

This task starts RSVP and sets the bandwidth and single-flow limits. 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.

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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface fastethernet 0/1`	Configures the specified interface and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-bandwidth* [**percent** *percent-bandwidth* | [*single-flow-bandwidth*] [**sub-pool** *bandwidth*]]] **Example:** `Device(config-if)# ip rsvp bandwidth 23 54`	Enables RSVP for IP on an interface.
Step 5	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring RSVP over UDP

To enable RSVP over UDP, perform the following task:

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp udp neighbor** *neighbor-IP-address* **router** [**vrf** *vrf-name*]
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp udp neighbor** *neighbor-IP-address* **router** [**vrf** *vrf-name*] **Example:** `Device(config)# ip rsvp udp neighbor 10.1.1.1 router vrf vrf-1`	Configures the RSVP over UDP feature for the neighbor router.
Step 4	**end** **Example:** `Device(config)# end`	Returns to privileged EXEC mode.

Configuration examples for RSVP over UDP

Example: Enabling RSVP

The following example shows how to enable RSVP for IP on an interface:

```
enable
 configure terminal
 interface  fastethernet 0/1
  ip rsvp bandwidth 23 54
  end
```

Example: Configuring RSVP over UDP

The following example shows how to configure the RSVP over UDP feature on a neighbor router:

```
enable
 configure terminal
 ip rsvp udp neighbor 10.1.1.1 router vrf vrf-1
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Master Command List, All Releases*
RSVP commands	*Quality of Service Solutions Command Reference*
Overview on RSVP	*Signaling Overview*

Standards and RFCs

Standard/RFC	Title
RFC 2205	*RSVP—Version 1 Function Specification*
RFC 2209	*RSVP—Version 1 Message Processing Rules*

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 RSVP over UDP

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

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

Table 4 Feature Information for RSVP over UDP

Feature Name	Releases	Feature Information
RSVP over UDP	15.2(4)M	The RSVP over UDP feature allows a router to enable a neighbor router to process and send RSVP control packets over UDP. The following commands were introduced or modified: **ip rsvp udp neighbor**.

Control Plane DSCP Support for RSVP

Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This document describes the Cisco control plane differentiated services code point (DSCP) support for Resource Reservation Protocol (RSVP) feature. It identifies the supported platforms, provides configuration examples, and lists related IOS command line interface (CLI) commands.

This document includes the following major sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Feature Overview

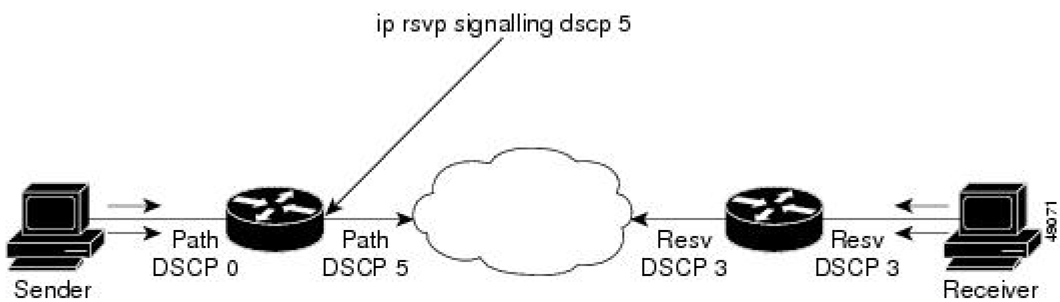
Typically, networks operate on a best-effort delivery basis, which means that all traffic has equal priority and an equal chance of being delivered in a timely manner. When congestion occurs, all traffic has an equal chance of being dropped.

Before traffic can be handled according to its unique requirements, it must be identified or labeled. There are numerous classification techniques for doing this. These include Layer 3 schemes such as IP precedence or the differentiated services code point (DSCP), Layer 2 schemes such as 802.1P, and implicit characteristics of the data itself, such as the traffic type using the Real-Time Transport Protocol (RTP) and a defined port range.

The control plane DSCP support for RSVP feature allows you to set the priority value in the type of service (ToS) byte/differentiated services (DiffServ) field in the Internet Protocol (IP) header for RSVP messages. The IP header functions with resource providers such as weighted fair queueing (WFQ), so that voice frames have priority over data fragments and data frames. When packets arrive in a device's output queue, the voice packets are placed ahead of the data frames.

The figure below shows a path message originating from a sender with a DSCP value of 0 (the default) that is changed to 5 to give the message a higher priority and a reservation (resv) message originating from a receiver with a DSCP of 3.

Figure 10 *Control Plane DSCP Support for RSVP*



Raising the DSCP value reduces the possibility of packets being dropped, thereby improving call setup time in VoIP environments.

- Benefits, page 64
- Restrictions, page 65

Benefits

Faster Call Setup Time

The control plane DSCP support for RSVP feature allows you to set the priority for RSVP messages. In a DiffServ QoS environment, higher priority packets get serviced before lower priority packets, thereby improving the call setup time for RSVP sessions.

Improved Message Delivery

During periods of congestion, devices drop lower priority traffic before they drop higher priority traffic. Since RSVP messages can now be marked with higher priority, the likelihood of these messages being dropped is significantly reduced.

Faster Recovery after Failure Conditions

When heavy congestion occurs, many packets are dropped. Network resources attempt to retransmit almost instantaneously resulting in further congestion. This leads to a considerable reduction in throughput.

Previously, RSVP messages were marked best effort and subject to being dropped by congestion avoidance mechanisms such as weighted random early detection (WRED). However, with the control plane DSCP support for RSVP feature, RSVP messages are likely to be dropped later, if at all, thereby providing faster recovery of RSVP reservations.

Restrictions

Control plane DSCP support for RSVP can be configured on interfaces and subinterfaces only. It affects all RSVP messages sent out the interface or that are on any logical circuit of the interface, including subinterfaces, permanent virtual circuits (PVCs), and switched virtual circuits (SVCs).

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series
- Cisco 7500 route/switch processor (RSP) only
- Cisco 12000 series Gigabit Switch Router (GSR)

Prerequisites

The network must support the following Cisco IOS feature before control plane DSCP support for RSVP is enabled:

- Resource Reservation Protocol (RSVP)

Configuration Tasks

- Enabling RSVP on an Interface, page 65
- Specifying the DSCP, page 66
- Verifying Control Plane DSCP Support for RSVP Configuration, page 66

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]	Enables RSVP on an interface.

Specifying the DSCP

To specify the DSCP, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp signalling dscp** `[value]`	Specifies the DSCP to be used on all RSVP messages transmitted on an interface.

Verifying Control Plane DSCP Support for RSVP Configuration

To verify control plane DSCP support for RSVP configuration, enter the **show ip rsvp interface detail**command to display RSVP-related interface information.

In the following sample output from the **show ip rsvp interface detail**command, only the Se2/0 interface has DSCP configured. Interfaces that are not configured for DSCP do not show the DSCP value, which is 0 by default.

```
Device# show ip rsvp interface detail
Et1/1:
   Bandwidth:
     Curr allocated:0M bits/sec
     Max. allowed (total):7500K bits/sec
     Max. allowed (per flow):7500K bits/sec
   Neighbors:
     Using IP enacp:1.  Using UDP encaps:0
 Et1/2:
   Bandwidth:
     Curr allocated:0M bits/sec
     Max. allowed (total):7500K bits/sec
     Max. allowed (per flow):7500K bits/sec
   Neighbors:
     Using IP enacp:0.  Using UDP encaps: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
Device#
```

Monitoring and Maintaining Control Plane DSCP Support for RSVP

To monitor and maintain control plane DSCP support for RSVP, use the following command in EXEC mode:

Command	Purpose
`Device#` **show ip rsvp interface detail**	Displays RSVP-related information about interfaces.

Configuration Examples

This section provides a configuration example for the control plane DSCP support for RSVP feature.

```
 Device(config-if)# ip rsvp sig ?
 dscp  DSCP for RSVP signalling messages
 Device(config-if)# ip rsvp sig dscp ?
 <0-63>  DSCP value
Device(config-if)# ip rsvp sig dscp 48
Device# show run int e3/0
interface Ethernet3/0
ip address 50.50.50.1 255.255.255.0
fair-queue 64 256 235
ip rsvp signalling dscp 48
ip rsvp bandwidth 7500 7500
```

Additional References

The following sections provide references related to the Control Plane DSCP Support for RSVP feature.

Related Documents

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

Standards

Standard	Title
None	--

MIBs

MIB	MIBs Link
RFC 2206 (RSVP Management Information Base using SMIv2)	To locate and download MIBs for selected platforms, software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource Reservation Protocol*

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

Glossary

CBWFQ-- Class-based weighted fair queueing. A queueing mechanism that extends the standard WFQ functionality to provide support for user-defined traffic classes.

class-based weighted fair queueing --See CBWFQ.

differentiated services --See DiffServ.

differentiated services code point --See DSCP.

DiffServ --An architecture based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network. The class of traffic is then identified with a DS codepoint or bit marking in the IP header. Within the core of the network, packets are forwarded according to the per-hop behavior associated with the DS code point.

DSCP --Differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

IP precedence --The three most significant bits of the 1-byte type of service (ToS) field. IP precedence values range between zero for low priority and seven for high priority.

latency --The delay between the time a device receives a packet and the time that packet is forwarded out the destination port.

marking --The process of setting a Layer 3 DSCP value in a packet.

QoS --Quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

quality of service --See QoS.

Resource Reservation Protocol --See RSVP.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

ToS --Type of service. An 8-bit value in the IP header field.

type of service --See ToS.

Voice over IP --See VoIP.

VoIP --Voice over IP. The ability to carry normal telephony-style voice over an IP-based internet maintaining telephone-like functionality, reliability, and voice quality.

weighted fair queueing --See WFQ.

weighted random early detection --See WRED.

WFQ --Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on relative bandwidth applied to each of the queues.

WRED --Weighted random early detection. A congestion avoidance mechanism that slows traffic by randomly dropping packets when there is congestion.

Configuring RSVP Support for Frame Relay

This chapter describes the tasks for configuring the RSVP Support for Frame Relay feature.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

How to Configure RSVP Support for Frame Relay

Enabling Frame Relay Encapsulation on an Interface

SUMMARY STEPS

1. Device(config)# **interface s3/0**
2. Device(config-if)# **encapsulation frame-relay**[**cisco**| **ietf**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **interface s3/0**	Enables an interface (for example, serial interface 3/0) and enters configuration interface mode.
Step 2	Device(config-if)# **encapsulation frame-relay**[**cisco**| **ietf**]	Enables Frame Relay and specifies the encapsulation method.

Configuring a Virtual Circuit

Command	Purpose
`Device(config-if)#` **frame-relay interface-dlci** *dlci*	Assigns a data-link connection identifier (DLCI) to a specified Frame Relay subinterface on a router or access server.

Enabling Frame Relay Traffic Shaping on an Interface

Command	Purpose
`Device(config-if)#` **frame-relay traffic-shaping**	Enables traffic shaping and per-VC queueing for all permanent virtual circuits (PVCs) and switched virtual circuits (SVCs) on a Frame Relay interface.

Enabling Enhanced Local Management Interface

Command	Purpose
`Router(config-if)#` **frame-relay lmi-type**	Selects the LMI type.

Enabling RSVP on an Interface

Command	Purpose
`Device(config-if)#` **ip rsvp bandwidth**	Enables RSVP on an interface.

Specifying a Traffic Shaping Map Class for an Interface

Command	Purpose
`Device(config-if)#` **frame-relay class** *name*	Associates a map class with an interface or subinterface.

Defining a Map Class with WFQ and Traffic Shaping Parameters

Command	Purpose
`Device(config)#` **map-class frame-relay** *map-class-name*	Defines parameters for a specified class.

Specifying the CIR

Command	Purpose
`Device(config-map-class)#` **frame-relay cir** {**in** | **out**} *bps*	Specifies the maximum incoming or outgoing CIR for a Frame Relay VC.

Specifying the Minimum CIR

Command	Purpose
`Device(config-map-class)#` **frame-relay mincir** {**in** | **out**} *bps*	Specifies the minimum acceptable incoming or outgoing CIR for a Frame Relay VC. **Note** If the minCIR is not configured, then the admission control value is the CIR/2.

Enabling WFQ

Command	Purpose
Device(config-map-class)# **frame-relay fair-queue**	Enables WFQ on a PVC.

Enabling FRF.12

Command	Purpose
Device(config-map-class)# **frame-relay fragment** *fragment-size*	Enables Frame Relay fragmentation on a PVC.

Configuring a Path

Command	Purpose
Device(config)# **ip rsvp sender**	Specifies the RSVP path parameters, including the destination and source addresses, the protocol, the destination and source ports, the previous hop address, the average bit rate, and the burst size.

Configuring a Reservation

Command	Purpose
Device(config)# **ip rsvp reservation**	Specifies the RSVP reservation parameters, including the destination and source addresses, the protocol, the destination and source ports, the next hop address, the next hop interface, the reservation style, the service type, the average bit rate, and the burst size.

Verifying RSVP Support for Frame Relay

- Multipoint Configuration, page 74
- Point-to-Point Configuration, page 76

Multipoint Configuration

To verify RSVP support for Frame Relay in a multipoint configuration, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has two reservations:
2. Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

DETAILED STEPS

Step 1 Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has two reservations:

Example:

```
Device# show ip rsvp installed
RSVP:Serial3/0
BPS    To              From            Protoc DPort  Sport  Weight Conversation
RSVP:Serial3/0.1
BPS    To              From            Protoc DPort  Sport  Weight Conversation
40K    145.20.22.212   145.10.10.211   UDP    10     10     0      24
50K    145.20.21.212   145.10.10.211   UDP    10     10     6      25
```

Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue.

Step 2 Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

Note In the following output, the first flow gets a reserved queue with a weight > 0, and the second flow gets the priority queue with a weight = 0.

Example:

```
Device# show ip rsvp installed detail
RSVP:Serial3/0 has the following installed reservations
RSVP:Serial3/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.20.21.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth:50K bits/sec, Maximum burst:1K bytes, Peak rate:50K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: RESERVED queue 25.  Weight:6
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 68 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.22.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth:40K bits/sec, Maximum burst:1K bytes, Peak rate:40K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: PRIORITY queue 24.  Weight:0
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 707 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
```

Point-to-Point Configuration

To verify RSVP support for Frame Relay in a point-to-point configuration, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has one reservation, and serial subinterface 3/0.2 has one reservation.
2. Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

DETAILED STEPS

Step 1 Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. The output in the following example shows that serial subinterface 3/0.1 has one reservation, and serial subinterface 3/0.2 has one reservation.

Example:

```
Device# show ip rsvp installed
RSVP:Serial3/0
BPS    To              From            Protoc DPort  Sport
RSVP:Serial3/0.1
BPS    To              From            Protoc DPort  Sport
50K    145.20.20.212   145.10.10.211   UDP    10     10
RSVP:Serial3/0.2
BPS    To              From            Protoc DPort  Sport
10K    145.20.21.212   145.10.10.211   UDP    11     11
```

Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue.

Step 2 Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, DLCI PVCs, and their current reservations.

Note In the following output, the first flow with a weight > 0 gets a reserved queue and the second flow with a weight = 0 gets the priority queue.

Example:

```
Device# show ip rsvp installed detail
RSVP:Serial3/0 has the following installed reservations
RSVP:Serial3/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth:50K bits/sec, Maximum burst:1K bytes, Peak rate:50K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: RESERVED queue 25.  Weight:6
  Data given reserved service:415 packets (509620 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 862 seconds
  Long-term average bitrate (bits/sec):4724 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 11, Source port is 11
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0: PRIORITY queue 24.  Weight:0
  Data given reserved service:85 packets (104380 bytes)
```

```
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 875 seconds
  Long-term average bitrate (bits/sec):954 reserved, 0M best-effort
RSVP:Serial3/0.2 has the following installedreservations
RSVP Reservation. Destination is 145.20.21.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 11, Source port is 11
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10Kbits/sec
QoS provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0:PRIORITY queue 24.  Weight:0
  Data given reserved service:85 packets (104380 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 875 seconds
  Long-term average bitrate (bits/sec):954 reserved, 0M best-effort
```

Monitoring and Maintaining RSVP Support for Frame Relay

Command	Purpose
`Device#` **show ip rsvp installed**	Displays information about interfaces and their admitted reservations.
`Device#` **show ip rsvp installed detail**	Displays additional information about interfaces, DLCIs, and their admitted reservations.
`Device#` **show queueing**	Displays all or selected configured queueing strategies.

Configuration Examples for Configuring RSVP Support for Frame Relay

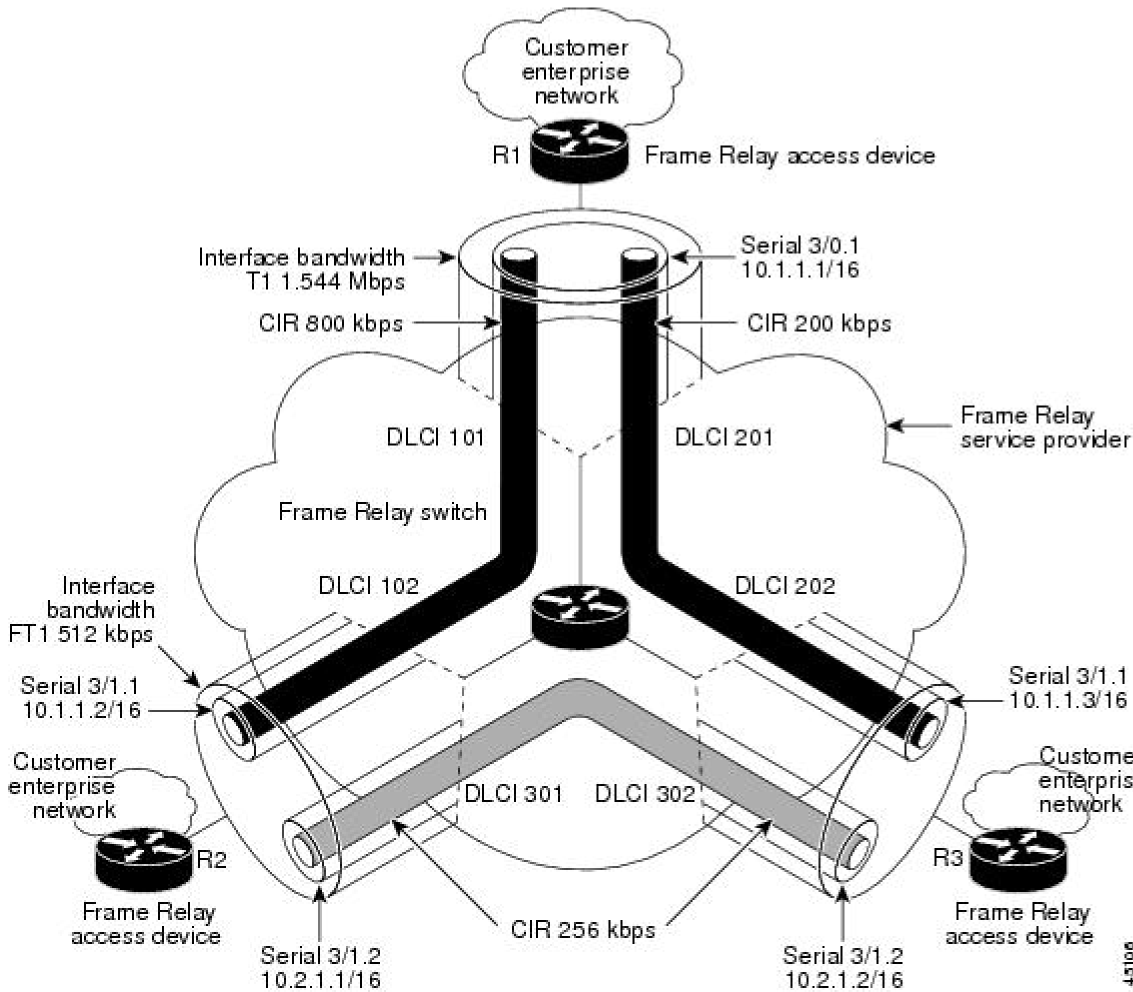
- Example Multipoint Configuration, page 78
- Example Point-to-Point Configuration, page 80

- Example Multipoint Configuration, page 78
- Example Point-to-Point Configuration, page 80

Example Multipoint Configuration

The figure below shows a multipoint interface configuration commonly used in Frame Relay environments in which multiple PVCs are configured on the same subinterface at device R1.

Figure 11 Multipoint Interface Configuration



RSVP performs admission control based on the minCIR of DLCI 101 and DLCI 201. The congestion point is not the 10.1.1.1/16 subinterface, but the CIR of DLCI 101 and DLCI 201.

The following example is a sample output for serial interface 3/0:

```
interface Serial3/0
 no ip address
 encapsulation frame-relay
 no fair-queue
 frame-relay traffic-shaping
 frame-relay lmi-type cisco
 ip rsvp bandwidth 350 350
!
interface Serial3/0.1 multipoint
 ip address 10.1.1.1 255.255.0.0
 frame-relay interface-dlci 101
```

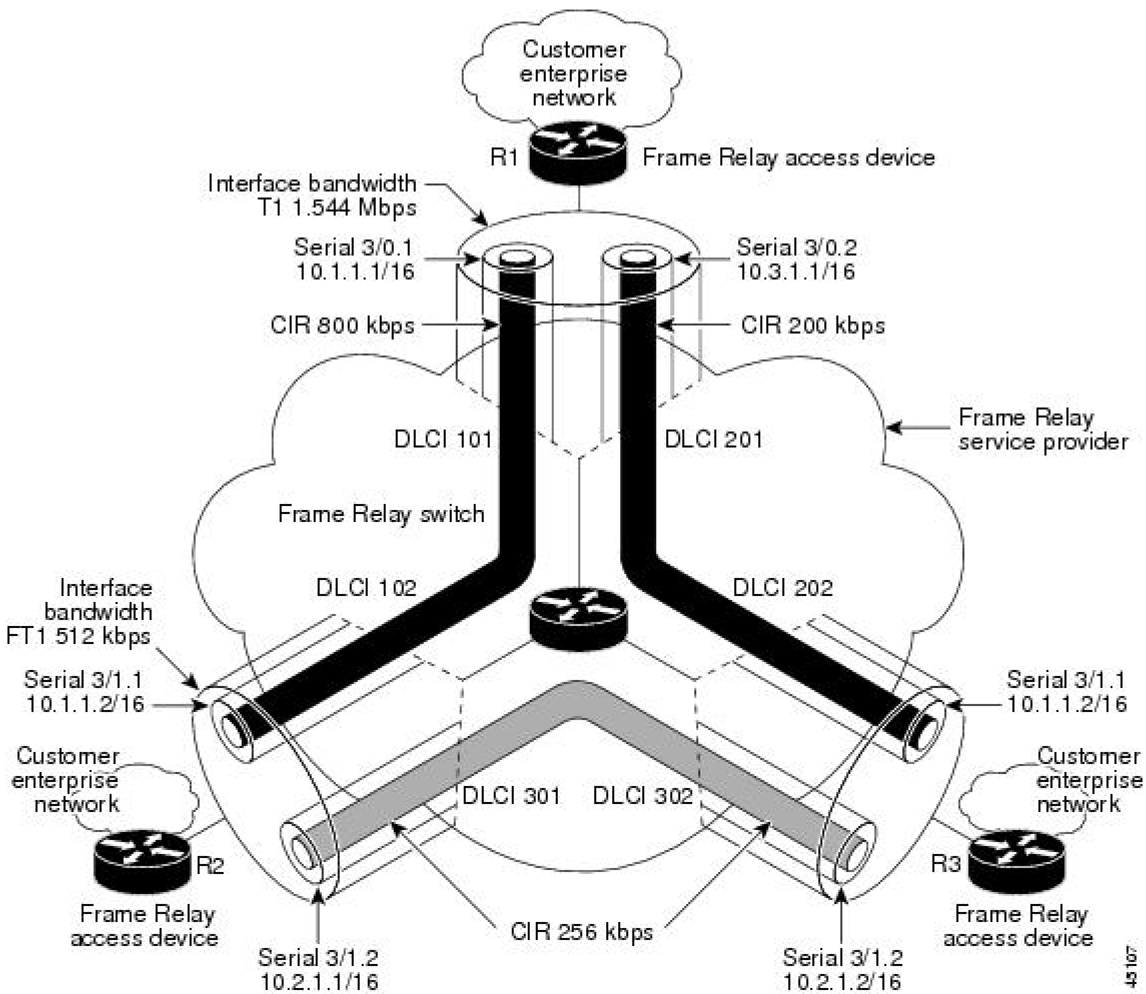
```
  class fr-voip
 frame-relay interface-dlci 201
  class fast-vcs
 ip rsvp bandwidth 350 350
ip rsvp pq-profile 6000 2000 ignore-peak-value
!
!
map-class frame-relay fr-voip
 frame-relay cir 800000
 frame-relay bc 8000
 frame-relay mincir 128000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
!
map-class frame-relay fast-vcs
 frame-relay cir 200000
 frame-relay bc 2000
 frame-relay mincir 60000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
!
```

Note When FRTS is enabled, the Frame Relay Committed Burst (Bc) value (in bits) should be configured to a maximum of 1/100th of the CIR value (in bits per second). This configuration ensures that the FRTS token bucket interval (Bc/CIR) does not exceed 10 Ms, and that voice packets are serviced promptly.

Example Point-to-Point Configuration

The figure below shows a point-to-point interface configuration commonly used in Frame Relay environments in which one PVC per subinterface is configured at device R1.

Figure 12 ***Sample Point-to-Point Interface Configuration***



Notice that the device interface bandwidth for R1 is T1 (1.544 Mbps), whereas the CIR value of DLCI 201 toward R3 is 256 kbps. For traffic flows from R1 to R3 over DLCI 201, the congestion point is the CIR for DLCI 201. As a result, RSVP performs admission control based on the minCIR and reserves resources, including queues and bandwidth, on the WFQ system that runs on each DLCI.

The following example is sample output for serial interface 3/0:

```
interface Serial3/0
 no ip address
 encapsulation frame-relay
no fair-queue
 frame-relay traffic-shaping
 frame-relay lmi-type cisco
 ip rsvp bandwidth 500 500
!
```

```
interface Serial3/0.1 point-to-point
 ip address 10.1.1.1 255.255.0.0
 frame-relay interface-dlci 101
  class fr-voip
 ip rsvp bandwidth 350 350
!
interface Serial3/0.2 point-to-point
 ip address 10.3.1.1 255.255.0.0
 frame-relay interface-dlci 201
  class fast-vcs
 ip rsvp bandwidth 150 150
ip rsvp pq-profile 6000 2000 ignore-peak-value
!
!
map-class frame-relay fr-voip
 frame-relay cir 800000
 frame-relay bc 8000
 frame-relay mincir 128000
 frame-relay fragment 280
 no frame-relay adaptive-shaping
 frame-relay fair-queue
```

Note When FRTS is enabled, the Frame Relay Committed Burst (Bc) value (in bits) should be configured to a maximum of 1/100th of the CIR value (in bits per second). This configuration ensures that the FRTS token bucket interval (Bc/CIR) does not exceed 10 Ms, and that voice packets are serviced promptly.

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

RSVP Scalability Enhancements

This document describes the Cisco Resource Reservation Protocol (RSVP) scalability enhancements. It identifies the supported platforms, provides configuration examples, and lists related IOS command line interface (CLI) commands.

This document includes the following major sections:

Feature Information For

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.

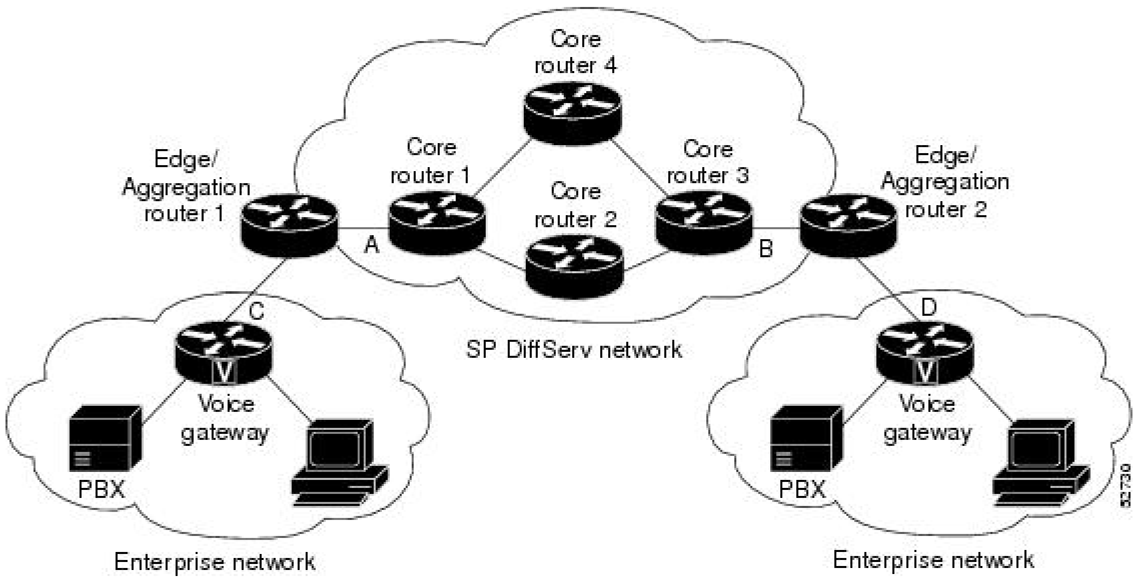
Feature Overview

RSVP typically performs admission control, classification, policing, and scheduling of data packets on a per-flow basis and keeps a database of information for each flow. RSVP scalability enhancements let you select a resource provider (formerly called a quality of service (QoS) provider) and disable data packet classification so that RSVP performs admission control only. This facilitates integration with service provider (differentiated services (DiffServ)) networks and enables scalability across enterprise networks.

Class-based weighted fair queueing (CBWFQ) provides the classification, policing, and scheduling functions. CBWFQ puts packets into classes based on the differentiated services code point (DSCP) value in the packet's Internet Protocol (IP) header, thereby eliminating the need for per-flow state and per-flow processing.

The figure below shows two enterprise networks interconnected through a service provider (SP) network. The SP network has an IP backbone configured as a DiffServ network. Each enterprise network has a voice gateway connected to an SP edge/aggregation device via a wide area network (WAN) link. The enterprise networks are connected to a private branch exchange (PBX).

Figure 13 RSVP/DiffServ Integration Topology



The voice gateways are running classic RSVP, which means RSVP is keeping a state per flow and also classifying, marking, and scheduling packets on a per flow basis. The edge/aggregation devices are running classic RSVP on the interfaces (labeled C and D) connected to the voice gateways and running RSVP for admission control only on the interfaces connected to core routers 1 and 3. The core devices in the DiffServ network are not running RSVP, but are forwarding the RSVP messages to the next hop. The core devices inside the DiffServ network implement a specific per hop behavior (PHB) for a collection of flows that have the same DSCP value.

The voice gateways identify voice data packets and set the appropriate DSCP in their IP headers such that the packets are classified into the priority class in the edge/aggregation devices and in core routers 1, 2, 3 or 1, 4, 3.

The interfaces or the edge/aggregation routers (labeled A and B) connected to core routers 1 and 3 are running RSVP, but are doing admission control only per flow against the RSVP bandwidth pool configured on the DiffServ interfaces of the edge/aggregation devices. CBWFQ is performing the classification, policing, and scheduling functions.

- Benefits, page 84
- Restrictions, page 85

Benefits

Enhanced Scalability

RSVP scalability enhancements handle similar flows on a per-class basis instead of a per-flow basis. Since fewer resources are required to maintain per-class QoS guarantees, faster processing results, thereby enhancing scalability.

Improved Device Performance

RSVP scalability enhancements improve device performance by reducing the cost for data packet classification and scheduling, which decrease central processing unit (CPU) resource consumption. The saved resources can then be used for other network management functions.

Restrictions

- Sources should not send marked packets without an installed reservation.
- Sources should not send marked packets that exceed the reserved bandwidth.
- Sources should not send marked packets to a destination other than the reserved path.

Supported Platforms

- Cisco 2600 series
- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series
- Cisco 7500 route/switch processor (RSP) only

Prerequisites

The network must support the following Cisco IOS features before the RSVP scalability enhancements are enabled:

- Resource Reservation Protocol (RSVP)
- Class-based weighted fair queueing (CBWFQ)

Configuration Tasks

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]	Enables RSVP on an interface.

Note The bandwidth that you configure on the interface must match the bandwidth that you configure for the CBWFQ priority queue. See the section on Configuration Examples, page 89.

Setting the Resource Provider

Note Resource provider was formerly called QoS provider.

To set the resource provider, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp resource-provider none**	Sets the resource provider to none.

Note Setting the resource provider to none instructs RSVP to *not* associate any resources, such as WFQ queues or bandwidth, with a reservation.

Disabling Data Packet Classification

To turn off (disable) data packet classification, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp data-packet classification none**	Disables data packet classification.

Note Disabling data packet classification instructs RSVP *not* to process every packet, but to perform admission control only.

Configuring Class and Policy Maps

To configure class and policy maps, use the following commands, beginning in global configuration mode:

SUMMARY STEPS

1. Device(config)# **class-map** *class-map-name*
2. Device(config)# **policy-map** *policy-map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Device(config)# **class-map** *class-map-name*	Specifies the name of the class for which you want to create or modify class map match criteria.
Step 2	Device(config)# **policy-map** *policy-map-name*	Specifies the name of the policy map to be created, added to, or modified before you can configure policies for classes whose match criteria are defined in a class map.

Attaching a Policy Map to an Interface

To attach a policy map to an interface, use the following command, beginning in interface configuration mode:

Command	Purpose
`Device(config-if)#` **service-policy** {**input** | **output**} *policy-map-name*	Attaches a single policy map to one or more interfaces to specify the service policy for those interfaces.

Note

If at the time you configure the RSVP scalability enhancements, there are existing reservations that use classic RSVP, no additional marking, classification, or scheduling is provided for these flows. You can also delete these reservations after you configure the RSVP scalability enhancements.

Verifying RSVP Scalability Enhancements Configuration

SUMMARY STEPS

1. Enter the **show ip rsvp interface detail**command to display information about interfaces, subinterfaces, resource providers, and data packet classification. The output in the following example shows that the ATM 6/0 interface has resource provider none configured and data packet classification is turned off:
2. Enter the **show ip rsvp installed detail**command to display information about interfaces, subinterfaces, their admitted reservations, bandwidth, resource providers, and data packet classification.
3. Wait for a while, then enter the **show ip rsvp installed detail**command again. In the following output, notice there is no increment in the number of packets classified:

DETAILED STEPS

Step 1 Enter the **show ip rsvp interface detail**command to display information about interfaces, subinterfaces, resource providers, and data packet classification. The output in the following example shows that the ATM 6/0 interface has resource provider none configured and data packet classification is turned off:

Example:

```
Device# show ip rsvp interface detail
 AT6/0:
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
   RSVP Data Packet Classification is OFF
   RSVP resource provider is: none
```

Note The last two lines in the preceding output verify that the RSVP scalability enhancements (disabled data packet classification and resource provider none) are present.

Step 2 Enter the **show ip rsvp installed detail**command to display information about interfaces, subinterfaces, their admitted reservations, bandwidth, resource providers, and data packet classification.

Example:

```
Device# show ip rsvp installed detail
RSVP: Ethernet3/3 has no installed reservations
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 54 seconds
  Long-term average bitrate (bits/sec): 0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 80 seconds
  Long-term average bitrate (bits/sec): 0M reserved, 0M best-effort
```

Step 3 Wait for a while, then enter the **show ip rsvp installed detail**command again. In the following output, notice there is no increment in the number of packets classified:

Example:

```
Device# show ip rsvp installed detail
```

```
RSVP: Ethernet3/3 has no installed reservations
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 60 seconds
  Long-term average bitrate (bits/sec): 0 reserved, OM best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 86 seconds
  Long-term average bitrate (bits/sec): OM reserved, 0M best-effort
```

Monitoring and Maintaining RSVP Scalability Enhancements

To monitor and maintain RSVP scalability enhancements, use the following commands in EXEC mode:

Command	Purpose
`Device#` **show ip rsvp installed**	Displays information about interfaces and their admitted reservations.
`Device#` **show ip rsvp installed detail**	Displays additional information about interfaces and their admitted reservations.
`Device#` **show ip rsvp interface**	Displays RSVP-related interface information.
`Device#` **show ip rsvp interface detail**	Displays additional RSVP-related interface information.
`Device#` **show queueing** [**custom** | **fair** | **priority** | **random-detect** [**interface** *serial-number*]]	Displays all or selected configured queueing strategies and available bandwidth for RSVP reservations.

Configuration Examples

- Example Configuring CBWFQ to Accommodate Reserved Traffic, page 90
- Example Configuring the Resource Provider as None with Data Classification Turned Off, page 90

- Example Configuring CBWFQ to Accommodate Reserved Traffic, page 90
- Example Configuring the Resource Provider as None with Data Classification Turned Off, page 90

Example Configuring CBWFQ to Accommodate Reserved Traffic

The following output shows a class map and a policy map being configured for voice:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# class-map match-all voice
Device(config-cmap)# match access-group 100
Device(config-cmap)# exit
Device(config)# policy-map wfq-voip
Device(config-pmap)# class voice
Device(config-pmap-c)# priority 24
Device(config-pmap-c)# end
Device#
```

Note The bandwidth that you configured for the CBWFQ priority queue (24 kbps) must match the bandwidth that you configured for the interface. See the section Enabling RSVP on an Interface, page 85.

The following output shows an access list being configured:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 100 permit udp any any range 16384 32500
```

The following output shows a class being applied to the outgoing interface:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# int atm6/0
Device(config-if)# service-policy output wfq-voip
```

The following output shows bandwidth being configured on an interface:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# int atm6/0
Device(config-if)# ip rsvp bandwidth 24
```

Note The bandwidth that you configure for the interface (24 kbps) must match the bandwidth that you configured for the CBWFQ priority queue.

Example Configuring the Resource Provider as None with Data Classification Turned Off

The **showrun** command displays the current configuration in the device:

```
Device# show run

int atm6/0
 class-map match-all voice
  match access-group 100
!
policy-map wfq-voip
  class voice
    priority 24
```

```
  class class-default
   fair-queue
!
interface ATM6/0
 ip address 20.20.22.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip route-cache cef
 atm uni-version 4.0
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111181.00
 no atm auto-configuration
 no atm ilmi-keepalive
 pvc blue 200/100
  abr 700 600
  inarp 1
  broadcast
  encapsulation aal5snap
  service-policy output wfq-voip
 !
 ip rsvp bandwidth 24 24
 ip rsvp signalling dscp 48
access-list 100 permit udp any any range 16384 32500
```

Here is output from the **showiprsvpinterfacedetail** command before resource provider none is configured and data-packet classification is turned off:

```
Device# show ip rsvp interface detail

 AT6/0:
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
```

Here is output from the **showqueueing**command before resource provider none is configured and data packet classification is turned off:

```
Device# s
how queueing int atm6/0
  Interface ATM6/0 VC 200/100
  Queueing strategy: weighted fair
  Output queue: 63/512/64/3950945 (size/max total/threshold/drops)
     Conversations  2/5/64 (active/max active/max total)
     Reserved Conversations 0/0 (allocated/max allocated)
     Available Bandwidth 450 kilobits/sec
```

Note

New reservations do not reduce the available bandwidth (450 kilobits/sec shown above). Instead RSVP performs admission control only using the bandwidth limit configured in the **iprsvpbandwidth** command. The bandwidth configured in this command should match the bandwidth configured in the CBWFQ class that you set up to handle the reserved traffic.

The following output shows resource provider none being configured:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# int atm6/0
Device(config-if)# ip rsvp resource-provider none

Device(config-if)# end
Device#
```

The following output shows data packet classification being turned off:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# int atm6/0
Device(config-if)# ip rsvp data-packet classification none
Device(config-if)# end
Device#
```

Here is output from the **showiprsvpinterfacedetail** command after resource provider none has been configured and data packet classification has been turned off:

```
Device# show ip rsvp interface detail
 AT6/0:
   Bandwidth:
     Curr allocated: 190K bits/sec
     Max. allowed (total): 112320K bits/sec
     Max. allowed (per flow): 112320K bits/sec
   Neighbors:
     Using IP encap: 1.  Using UDP encaps: 0
   DSCP value used in Path/Resv msgs: 0x30
   RSVP Data Packet Classification is OFF
   RSVP resource provider is: none
```

The following output from the **showiprsvpinstalleddetail** command verifies that resource provider none is configured and data packet classification is turned off:

```
Device# show ip rsvp installed detail
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3192 packets (1557696 bytes)
  Data given best-effort service: 42 packets (20496 bytes)
  Reserved traffic classified for 271 seconds
  Long-term average bitrate (bits/sec): 45880 reserved, 603 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1348 packets (657824 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 296 seconds
  Long-term average bitrate (bits/sec): 17755 reserved, 0M best-effort
```

The following output shows no increments in packet counts after the source sends data packets that match the reservation:

```
Device# show ip rsvp installed detail
RSVP: Ethernet3/3 has no installed reservations
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3192 packets (1557696 bytes)
  Data given best-effort service: 42 packets (20496 bytes)
  Reserved traffic classified for 282 seconds
  Long-term average bitrate (bits/sec): 44051 reserved, 579 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
```

```
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1348 packets (657824 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 307 seconds
  Long-term average bitrate (bits/sec): 17121 reserved, 0M best-effort
```

The following output shows that data packet classification is enabled again:

```
Device# configure terminal
Device(config)# int atm6/0
Device(config-if) no ip rsvp data-packet classification
Device(config-if)# end
```

The following output verifies that data packet classification is occurring:

```
Device# show ip rsvp installed detail
Enter configuration commands, one per line.  End with CNTL/Z.
RSVP: ATM6/0 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 14, Source port is 14
  Reserved bandwidth: 50K bits/sec, Maximum burst: 1K bytes, Peak rate: 50K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 3683 packets (1797304 bytes)
  Data given best-effort service: 47 packets (22936 bytes)
  Reserved traffic classified for 340 seconds
  Long-term average bitrate (bits/sec): 42201 reserved, 538 best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.10.10.211,
  Protocol is UDP, Destination port is 10, Source port is 10
  Reserved bandwidth: 20K bits/sec, Maximum burst: 1K bytes, Peak rate: 20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations
  Data given reserved service: 1556 packets (759328 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 364 seconds
  Long-term average bitrate (bits/sec): 16643 reserved, 0M best-effort
```

Here is output from the **showrun** command after you have performed all the previous configuration tasks:

```
Device# show run int atm6/0
 class-map match-all voice
  match access-group 100
!
policy-map wfq-voip
  class voice
    priority 24
  class class-default
   fair-queue
!
interface ATM6/0
 ip address 20.20.22.1 255.255.255.0
 no ip redirects
 no ip proxy-arp
 no ip route-cache cef
 atm uni-version 4.0
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111181.00
 no atm auto-configuration
 no atm ilmi-keepalive
 pvc blue 200/100
  abr 700 600
  inarp 1
  broadcast
```

```
  encapsulation aal5snap
  service-policy output wfq-voip
 !
 ip rsvp bandwidth 24 24
 ip rsvp signalling dscp 48
 ip rsvp data-packet classification none
 ip rsvp resource-provider none
access-list 100 permit udp any any range 16384 32500
```

Additional References

The following sections provide references related to the RSVP Scalability Enhancements feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS configuration tasks related to RSVP	"Configuring RSVP" module

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
RFC 2206 (RSVP Management Information Base using SMIv2)	To locate and download MIBs for selected platforms, software releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	Resource Reservation Protocol

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

Glossary

admission control --The process in which an RSVP reservation is accepted or rejected based on end-to-end available network resources.

aggregate --A collection of packets with the same DSCP.

bandwidth --The difference between the highest and lowest frequencies available for network signals. This term also describes the rated throughput capacity of a given network medium or protocol.

CBWFQ -- Class-based weighted fair queueing. A queueing mechanism that extends the standard WFQ functionality to provide support for user-defined traffic classes.

Class-based weighted fair queueing -- See CBWFQ .

differentiated services --See DiffServ.

differentiated services code point --See DSCP.

DiffServ --An architecture based on a simple model where traffic entering a network is classified and possibly conditioned at the boundaries of the network. The class of traffic is then identified with a DS code point or bit marking in the IP header. Within the core of the network, packets are forwarded according to the per-hop behavior associated with the DS code point.

DSCP --Differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

enterprise network --A large and diverse network connecting most major points in a company or other organization.

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

packet --A logical grouping of information that includes a header containing control information and (usually) user data. Packets most often refer to network layer units of data.

PBX --Private branch exchange. A digital or analog telephone switchboard located on the subscriber premises and used to connect private and public telephone networks.

PHB --Per hop behavior. A DiffServ concept that specifies how specifically marked packets are to be treated by each DiffServ device.

QoS --Quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

quality of service --See QoS.

Resource Reservation Protocol --See RSVP.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

Voice over IP --See VoIP.

VoIP --Voice over IP. The ability to carry normal telephony-style voice over an IP-based internet maintaining telephone-like functionality, reliability, and voice quality.

Weighted Fair Queueing --See WFQ.

WFQ --Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on relative bandwidth applied to each of the queues.

RSVP Support for ATM and PVCs

This document describes Cisco Resource Reservation Protocol (RSVP) support for the Asynchronous Transfer Mode/permanent virtual circuits (ATM/PVCs) feature. It identifies the supported platforms, provides configuration examples, and lists related IOS command line interface (CLI) commands.

This document includes the following major sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Feature Overview

Network administrators use queueing to manage congestion on a router interface or a permanent virtual circuit (PVC). In an ATM environment, the congestion point might not be the interface itself, but the PVC because of the traffic parameters, including the available bit rate (ABR), the constant bit rate (CBR), and the variable bit rate (VBR) associated with the PVC. For real-time traffic, such as voice flows, to be transmitted in a timely manner, the data rate must not exceed the traffic parameters, or packets might be dropped, thereby affecting voice quality. Fancy queueing such as class-based weighted fair queueing (CBWFQ), low latency queueing (LLQ), or weighted fair queueing (WFQ), can run on the PVC to provide the quality of service (QoS) guarantees for the traffic.

In previous releases, RSVP reservations were not constrained by the traffic parameters of the flow's outbound PVC. As a result, oversubscription could occur when the sum of the RSVP traffic and other traffic exceeded the PVC's capacity.

The RSVP support for ATM/PVCs feature allows RSVP to function with per-PVC queueing for voice-like flows. Specifically, RSVP can install reservations on PVCs defined at the interface and subinterface levels. There is no limit to the number of PVCs that can be configured per interface or subinterface.

RSVP Bandwidth Allocation and Modular QoS Command Line Interface (CLI)

RSVP can use an interface (or a PVC) queueing algorithm, such as WFQ, to ensure QoS for its data flows.

Admission Control

When WFQ is running, RSVP can co-exist with other QoS features on an interface (or PVC) that also reserve bandwidth and enforce QoS. When you configure multiple bandwidth-reserving features (such as RSVP, LLQ, CB-WFQ, and **ip rtp priority**), portions of the interface's (or PVC's) available bandwidth may be assigned to each of these features for use with flows that they classify.

An internal interface-based (or PVC-based) bandwidth manager prevents the amount of traffic reserved by these features from oversubscribing the interface (or PVC).

When you configure features such as LLQ and CB-WFQ, any classes that are assigned a bandwidth reserve their bandwidth at the time of configuration, and deduct this bandwidth from the bandwidth manager. If the configured bandwidth exceeds the interface's capacity, the configuration is rejected.

When RSVP is configured, no bandwidth is reserved. (The amount of bandwidth specified in the **ip rsvp bandwidth** command acts as a strict upper limit, and does **not** guarantee admission of any flows.) Only when an RSVP reservation arrives does RSVP attempt to reserve bandwidth out of the remaining pool of available bandwidth (that is, the bandwidth that has not been dedicated to traffic handled by other features.)

Data Packet Classification

By default, RSVP performs an efficient flow-based, datapacket classification to ensure QoS for its reserved traffic. This classification runs prior to queueing consideration by **ip rtp priority** or CB-WFQ. Thus, the use of a CB-WFQ class or **ip rtp priority** command is **not**required in order for RSVP data flows to be granted QoS. Any **ip rtp priority** or CB-WFQ configuration will not match RSVP flows, but they will reserve additional bandwidth for any non-RSVP flows that may match their classifiers.

If you do not want RSVP to perform per-flow classification, but prefer DiffServ classification instead, then you can configure RSVP to exclude itself from data packet classification, and configure LLQ for classification. For more information, see the "RSVP Scalability Enhancements" feature regarding DiffServ integration.

Benefits of RSVP Support for ATM PVCs

Accurate Admission Control

RSVP performs admission control based on the PVC's average cell rate, sustainable cell rate, or minimum cell rate, depending on the type of PVC that is configured, instead of the amount of bandwidth available on the interface.

Recognition of Layer 2 Overhead

RSVP automatically takes the Layer 2 overhead into account when admitting a flow. For each flow, RSVP determines the total amount of bandwidth required, including Layer 2 overhead, and uses this value for admission control with the WFQ bandwidth manager.

Improved QoS

RSVP provides QoS guarantees for high-priority traffic by reserving resources at the point of congestion (that is, the ATM PVC instead of the interface).

Flexible Configurations

RSVP provides support for point-to-point and multipoint interface configurations, thus enabling deployment of services such as voice over IP (VoIP) in ATM environments with QoS guarantees.

Prevention of Bandwidth Oversubscription

RSVP, CBWFQ, and ip rtp priority do not oversubscribe the amount of bandwidth available on the interface or the PVC even when they are running simultaneously. Prior to admitting a reservation, these features check an internal bandwidth manager to avoid oversubscription.

IP QoS Features Integration into ATM Environments

IP QoS features can now be integrated seamlessly from IP into ATM environments with RSVP providing admission control on a per PVC basis.

Restrictions

- Interface-level generic traffic shaping (GTS) is not supported.
- VC-level queueing and interface-level queueing on the same interface are not supported.
- Nonvoice RSVP flows are not supported.
- Multicast flows are not supported.
- ATM/PVCs must be preconfigured in the network.

Supported Platforms

- Cisco 3600 series (Cisco 3620, 3640, and 3660)
- Cisco 3810 multiservice access concentrator
- Cisco 7200 series

Prerequisites

The network must support the following Cisco IOS features before RSVP support for ATM/PVCs is enabled:

- Resource Reservation Protocol (RSVP)
- Weighted fair queueing (WFQ)

Configuration Tasks

See the following sections for configuration tasks for the RSVP support for ATM/PVCs feature. Each task in the list indicates whether the task is optional or required.

- Creating a PVC, page 100 (Required)
- Defining ATM QoS Traffic Parameters for a PVC, page 101 (Required)
- Defining a Policy Map for WFQ, page 101 (Required)
- Applying a Policy Map to a PVC, page 102 (Required)
- Enabling RSVP on an Interface, page 102 (Required)
- Configuring a Path, page 102 (Optional)
- Configuring a Reservation, page 102 (Optional)

Creating a PVC

To create a PVC, use the following command in interface configuration mode:

Command	Purpose
Router(config-if)# **pvc** [*name*] *vpi/vci* [**ilmi** | **qsaal** | **smds**]	Assigns a name and identifier to a PVC.

Defining ATM QoS Traffic Parameters for a PVC

Note In order for RSVP to reserve bandwidth, the ATM/PVC traffic parameters must be available bit rate (ABR), variable bit rate non real-time (VBR-NRT), or real-time variable bit rate (VBR). You can specify only one of these parameters per PVC connection; therefore, if you enter a new parameter, it will replace the existing one.

To configure ATM PVC traffic parameters, use *one* of the following commands beginning in interface-ATM-VC configuration mode:

Command	Purpose
`Router(config-if-atm-vc)#` **abr** *output-pcr output-mcr*	Configures the available bit rate (ABR). (ATM-CES port adapter and multiport T1/E1 ATM network module only.)
`Router(config-if-atm-vc)#` **vbr-nrt** *output-pcr output-scr output-mbs*	Configures the variable bit rate-non real time (VBR-NRT) QoS.
`Router(config-if-atm-vc)#` **vbr-rt** *peak-rate average-rate burst*	Configures the real-time variable bit rate (VBR). (Cisco MC3810 and multiport T1/E1 ATM network module only.)

The arguments used here are as follows:

- *-pcr*-- peak cell rate
- *-mcr*-- minimum cell rate
- *-scr*-- sustainable cell rate
- *-mbs*-- maximum burst size
- *output-mcr* , *output-scr*, and *average-rate*-- reservable bandwidth pool on the PVC

All features running on the PVC, including RSVP, CBWFQ, and LLQ, can use up to 75 percent of the reservable bandwidth pool.

Defining a Policy Map for WFQ

To define a policy map for WFQ, use the following commands, beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **policy-map** *policy-name*
2. Router(config-pmap)# **class** *class-name*
3. Router(config-pmap-c) **fair-queue***number-of-queues*

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **policy-map** *policy-name*	Specifies the policy map name; for example, wfq-voip.

	Command or Action	Purpose
Step 2	Router(config-pmap)# **class** *class-name*	Specifies the name of a previously defined class map, such as class-default.
Step 3	Router(config-pmap-c) **fair-queue***number-of-queues*	Specifies the number of queues to be reserved for the default class.

Applying a Policy Map to a PVC

To apply a policy map to a PVC, use the following command, beginning in interface-ATM-VC configuration mode:

Command	Purpose
`Router(config-if-atm-vc)#` **service-policy output** *policy-name*	Applies a policy map to the output direction of the interface.

Enabling RSVP on an Interface

To enable RSVP on an interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]	Enables RSVP on an interface.

Configuring a Path

To configure an RSVP path, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **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]*	Specifies the RSVP path parameters, including the destination and source addresses, the protocol, the destination and source ports, the previous hop address, the average bit rate, and the burst size.

Configuring a Reservation

To configure an RSVP reservation, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **ip rsvp reservation** *session-ip-address sender-ip-address* [**tcp** | **udp** | *ip-protocol*] *session-dport sender-sport next-hop-ip address nexthop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} [*bandwidth*] [*burst-size*]	Specifies the RSVP reservation parameters, including the destination and source addresses, the protocol, the destination and source ports, the next hop address, the next hop interface, the reservation style, the service type, the average bit rate, and the burst size.

Verifying RSVP Support for ATM PVCs Configuration

Multipoint Configuration

To verify RSVP support for ATM/PVCs multipoint configuration, use this procedure:

SUMMARY STEPS

1. Enter the **show ip rsvp installed**command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has four reservations:
2. Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.

DETAILED STEPS

Step 1 Enter the **show ip rsvp installed**command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has four reservations:

Example:

```
Router# show ip rsvp installed
RSVP:ATM6/0.1
BPS    To              From            Protoc DPort  Sport  Weight Conversation
10K    145.30.30.213   145.40.40.214   UDP    101    101    0      40
15K    145.20.20.212   145.40.40.214   UDP    100    100    6      41
15K    145.30.30.213   145.40.40.214   UDP    100    100    6      41
10K    145.20.20.212   145.40.40.214   UDP    101    101    0      40
```

Note Weight 0 is assigned to voice-like flows, which proceed to the priority queue (PQ).

Step 2 Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.

Note In the following output, the first flow has a weight = 0 and gets the PQ; the second flow has a weight > 0 and gets a reserved queue.

Example:

```
Router# show ip rsvp installed detail

RSVP:ATM6/0 has the following installed reservations
RSVP:ATM6/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514
  Resource provider for this flow:
    WFQ on ATM PVC 100/101 on AT6/0: PRIORITY queue 40.  Weight:0, BW 10 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 48 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514
  Resource provider for this flow:
    WFQ on ATM PVC 100/201 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 200 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514
  Resource provider for this flow:
    WFQ on ATM PVC 100/101 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 60 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
  Reserved bandwidth:10K bits/sec, Maximum burst:1K bytes, Peak rate:10K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514
  Resource provider for this flow:
    WFQ on ATM PVC 100/201 on AT6/0: PRIORITY queue 40.  Weight:0, BW 10 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 163 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
```

Point-to-Point Configuration

To verify RSVP support for ATM/PVCs point-to-point configuration, use this procedure:

SUMMARY STEPS

1. Enter the **show ip rsvp installed**command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has two reservations, and the ATM 6/0.2 subinterface has one reservation:
2. Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.

DETAILED STEPS

Step 1 Enter the **show ip rsvp installed**command to display information about interfaces, subinterfaces, PVCs, and their admitted reservations. The output in the following example shows that the ATM 6/0.1 subinterface has two reservations, and the ATM 6/0.2 subinterface has one reservation:

Example:

```
Router# show ip rsvp installed
RSVP:ATM6/0.1
BPS    To              From            Protoc DPort  Sport  Weight Conversation
15K    145.30.30.213   145.40.40.214   UDP    100    100    0      40
20K    145.30.30.213   145.40.40.214   UDP    101    101    6      41
RSVP:ATM6/0.2
BPS    To              From            Protoc DPort  Sport  Weight Conversation
150K   145.20.20.212   145.40.40.214   UDP    12     12     6      42
Router#
```

Note Weight 0 is assigned to voice-like flows, which proceed to the PQ.

Step 2 Enter the **show ip rsvp installed detail**command to display additional information about interfaces, subinterfaces, PVCs, and their current reservations.

Note In the following output, the first flow with a weight = 0 gets the PQ, and the second flow with a weight > 0 gets a reserved queue.

Example:

```
Router# show ip rsvp installed detail

RSVP:ATM6/0 has the following installed reservations
RSVP:ATM6/0.1 has the following installed reservations
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/101 on AT6/0: PRIORITY queue 40.  Weight:0, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 48 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  Reserved bandwidth:15K bits/sec, Maximum burst:1K bytes, Peak rate:15K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/201 on AT6/0: RESERVED queue 41.  Weight:6, BW 15 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 200 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP Reservation. Destination is 145.30.30.213, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 100, Source port is 100
  Reserved bandwidth:20K bits/sec, Maximum burst:1K bytes, Peak rate:20K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/101 on AT6/0: RESERVED queue 41.  Weight:6, BW 20 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
```

```
  Reserved traffic classified for 60 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
RSVP:ATM6/0.2 has the following installed reservations
RSVP Reservation. Destination is 145.20.20.212, Source is 145.40.40.214,
  Protocol is UDP, Destination port is 101, Source port is 101
  Reserved bandwidth:150K bits/sec, Maximum burst:1K bytes, Peak rate:150K bits/sec
  Min Policed Unit: 0 bytes, Max Pkt Size: 1514 bytes
  Resource provider for this flow:
    WFQ on ATM PVC 100/201 on AT6/0: PRIORITY queue 40.  Weight:0, BW 150 kbps
  Conversation supports 1 reservations
  Data given reserved service:0 packets (0M bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 163 seconds
  Long-term average bitrate (bits/sec):0M reserved, 0M best-effort
```

Monitoring and Maintaining RSVP Support for ATM and PVCs

To monitor and maintain RSVP support for ATM/PVCs, use the following commands in EXEC mode:

Command	Purpose
`Router# show ip rsvp installed`	Displays information about interfaces and their admitted reservations.
`Router#` **show ip rsvp installed detail**	Displays additional information about interfaces, PVCs, and their admitted reservations.
`Router#` **show queueing** [**custom** | **fair** | **priority** | **random-detect** [**interface** *serial-number*]]	Displays all or selected configured queueing strategies and available bandwidth for RSVP reservations.
`Router#` **show atm pvc** [*vpi/vci* | *name* | **interface atm** *interface-number*]	Displays all ATM PVCs and related traffic information.

Configuration Examples

This section provides point-to-point and multipoint configuration examples for the RSVP support for ATM/PVCs feature.

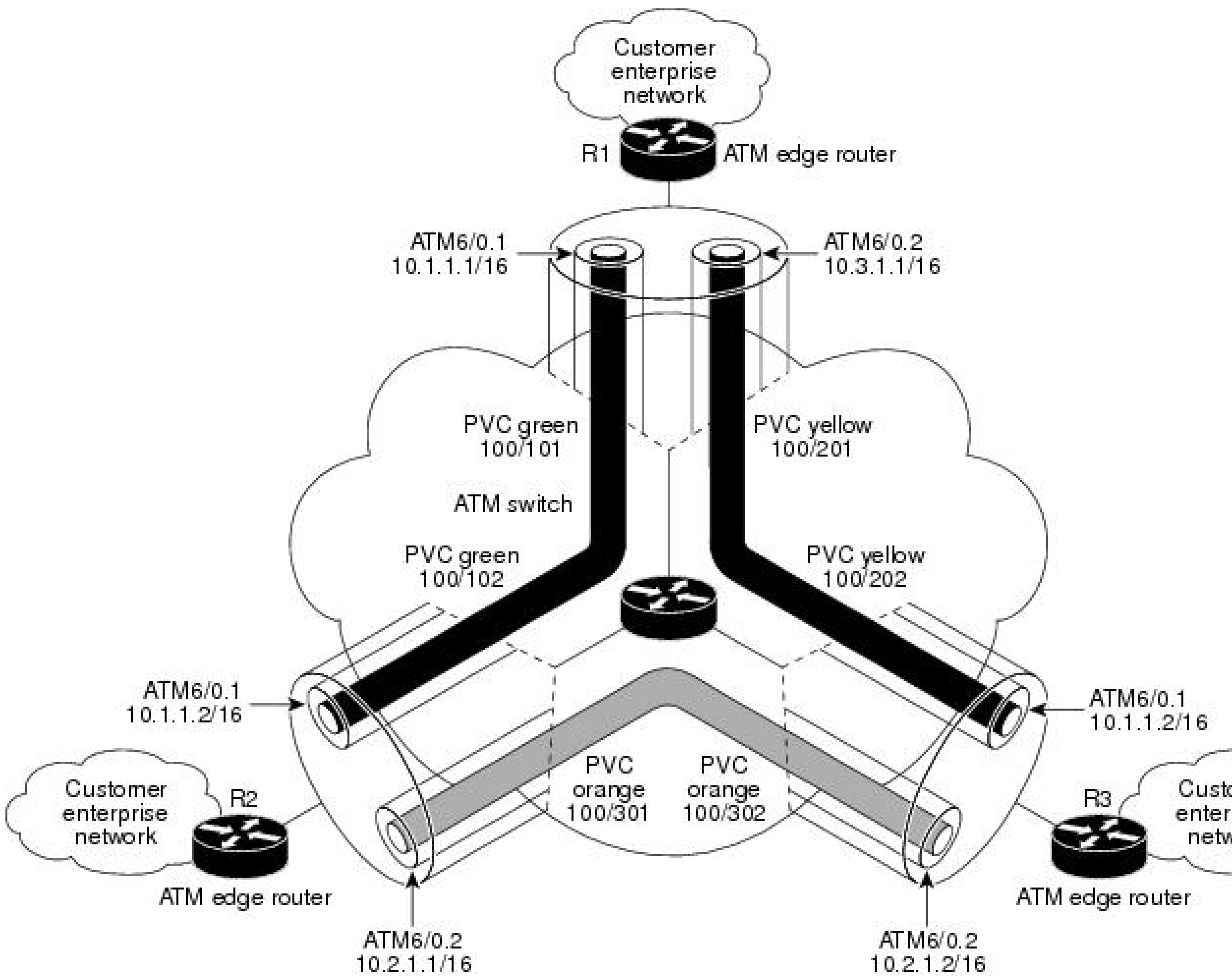
- Point-to-Point Configuration, page 106
- Multipoint Configuration, page 108

Point-to-Point Configuration

The figure below shows a sample point-to-point interface configuration commonly used in ATM environments in which one PVC per subinterface is configured at router R1.

Three small clouds represent office branches that are connected through PVCs over an ATM network.

Figure 14 Point-to-Point Interface Configuration



Here is sample output for a point-to-point configuration:

```
Router#
policy-map wfq-voip
  class class-default
   fair-queue

interface ATM6/0
 no ip address
 ip rsvp bandwidth 112320 112320
interface ATM6/0.1 point-to-point
 ip address 10.1.1.1 255.0.0.0
 pvc green 100/101
  vbr-rt 400 300 200
  inarp 1
  broadcast
  service-policy output wfq-voip
 ip rsvp bandwidth 1250 1250
 ip rsvp resource-provider wfq pvc

interface ATM6/0.2 point-to-point
 ip address 10.3.1.1 255.0.0.0
```

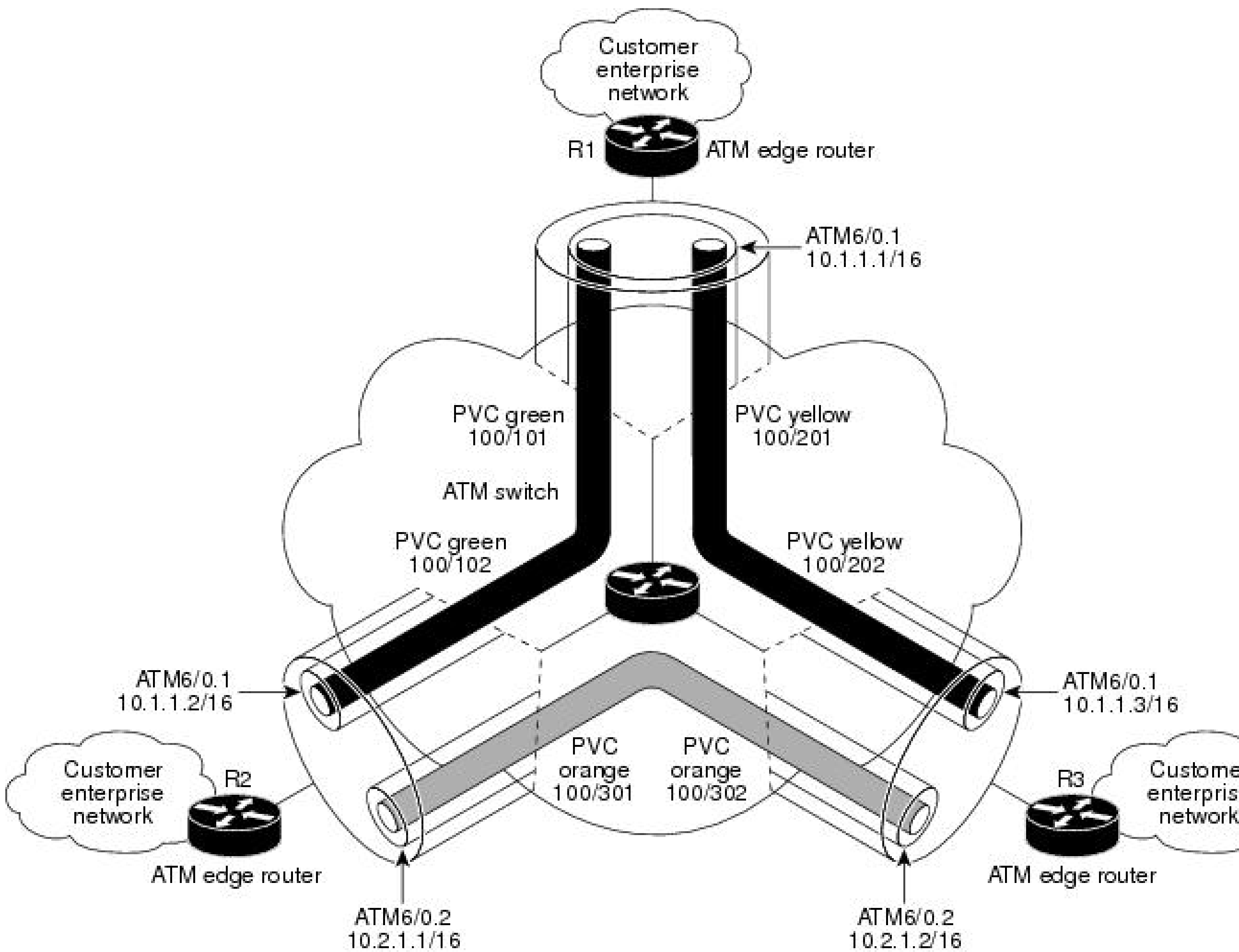
```
pvc yellow 100/201
 vbr-nrt 500 400 1000
 inarp 1
 broadcast
 service-policy output wfq-voip
ip rsvp bandwidth 1250 1250
ip rsvp resource-provider wfq pvc
```

Multipoint Configuration

The figure below shows a multipoint interface configuration commonly used in ATM environments in which multiple PVCs are configured on the same subinterface at router R1.

The customer enterprise network that includes R1 is the headquarters of a company with PVC connections to each remote office.

Figure 15 *Multipoint Interface Configuration*



Here is sample output for a multipoint configuration:

```
Router#
policy-map wfq-voip
  class class-default
   fair-queue
```

```
interface ATM6/0
 no ip address
 ip rsvp bandwidth 112320 112320
interface ATM6/0.1 multipoint
 ip address 10.1.1.1 255.0.0.0
 pvc green 100/101
  vbr-rt 400 300 200
  inarp 1
  broadcast
  service-policy output wfq-voip
 pvc yellow 100/201
  vbr-nrt 500 400 1000
  inarp 1
  broadcast
  service-policy output wfq-voip
 ip rsvp bandwidth 1250 1250
 ip rsvp resource-provider wfq pvc
```

Additional References

The following sections provide references related to the RSVP support for ATM/PVCs feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
Information about LLQ	"Configuring Weighted Fair Queueing" module
Information about traffic policing	"Policing and Shaping Overview" module

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
RFC 2206 (RSVP Management Information Base using SMIv2)	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification*

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

Glossary

AAL --ATM adaptation layer. AAL defines the conversion of user information into cells. AAL1 and AAL2 handle isochronous traffic, such as voice and video; AAL3/4 and AAL5 pertain to data communications through the segmentation and reassembly of packets.

ABR --Available bit rate. A QoS class defined by the ATM Forum for ATM networks. ABR is used for connections that do not require timing relationships between source and destination. ABR provides no guarantees in terms of cell loss or delay, providing only best-effort service. Traffic sources adjust their transmission rate in response to information they receive describing the status of the network and its capability to successfully deliver data.

admission control --The process in which an RSVP reservation is accepted or rejected based on end-to-end available network resources.

Asynchronous Transfer Mode --See ATM.

ATM --Asynchronous Transfer Mode. A cell-based data transfer technique in which channel demand determines packet allocation. This is an international standard for cell relay in which multiple service types (such as voice, video, or data) are conveyed in fixed-length (53-byte) cells. Fixed-length cells allow cell processing to occur in hardware, thereby reducing transit delays. ATM is designed to take advantage of high-speed transmission media such as E3, SONET, and T3.

available bit rate --See ABR.

bandwidth --The difference between the highest and lowest frequencies available for network signals. This term also describes the rated throughput capacity of a given network medium or protocol.

CBR --Constant bit rate. A QoS class defined by the ATM Forum for ATM networks. CBR is used for connections that depend on precise clocking to ensure undistorted delivery.

CBWFQ -- Class-based weighted fair queueing. A queueing mechanism that extends the standard WFQ functionality to provide support for user-defined traffic classes.

Class-based weighted fair queueing -- See CBWFQ .

constant bit rate --See CBR.

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

ILMI --Interim Local Management Interface. Described in the ATM Forum's UNI specification, ILMI allows end users to retrieve basic information, such as status and configuration about virtual connections and addresses, for a particular UNI.

Interim Local Management Interface --See ILMI.

latency --The delay between the time a device receives a packet and the time that the packet is forwarded out the destination port.

MUX --A multiplexing device that combines multiple signals for transmission over a single line. The signals are demultiplexed, or separated, at the receiving end.

payload --The portion of a cell, frame, or packet that contains upper-layer information (data).

permanent virtual circuit --See PVC.

point-to-multipoint connection --One of two fundamental connection types. It is a unidirectional connection in which a single source end system (known as a root node) connects to multiple destination end systems (known as leaves).

point-to-point connection --One of two fundamental connection types. It is a unidirectional or bidirectional connection between two end systems.

PQ --Priority queue. A routing feature in which frames in an output queue are assigned priority based on various characteristics such as packet size and interface type.

priority queue --See PQ.

PVC --Permanent virtual circuit or connection. A virtual circuit that is permanently established. PVCs save bandwidth associated with circuit establishment and teardown in situations where certain virtual circuits must exist all the time.

QoS --Quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

quality of service --See QoS.

reservable bandwidth pool --The amount of bandwidth on a link that features can set aside in order to provide QoS guarantees.

Resource Reservation Protocol --See RSVP.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

SNAP --Subnetwork Access Protocol. An Internet protocol that operates between a network entity in the subnetwork and a network entity in the end system. SNAP specifies a standard method of encapsulating IP datagrams and ARP messages on IEEE networks. The SNAP entity in the end system makes use of the services of the subnetwork and performs three key functions: data transfer, connection management, and QoS selection.

subnetwork access protocol --See SNAP.

SVC --Switched virtual circuit or connection. A virtual circuit that is dynamically established on demand and is torn down when transmission is complete. SVCs are used in situations where data transmission is sporadic.

switched virtual circuit --See SVC.

variable bit rate --See VBR.

VBR --Variable bit rate. A QoS class defined by the ATM Forum for ATM networks. VBR is subdivided into a real time (RT) class and a non-real time (NRT) class. VBR (RT) is used for connections in which there is a fixed timing relationship between samples. VBR (NRT) is used for connections where there is no fixed timing relationship between samples, but where a guaranteed QoS is still needed.

VC --Virtual circuit. A logical circuit created to ensure reliable communication between two network devices. A virtual circuit can be either permanent (PVC) or switched (SVC).

virtual circuit --See VC.

Voice over IP --See VoIP.

VoIP --Voice over IP. The ability to carry normal telephony-style voice over an IP-based internet maintaining telephone-like functionality, reliability, and voice quality.

weighted fair queueing --See WFQ.

WFQ --Weighted fair queueing. A queue management algorithm that provides a certain fraction of link bandwidth to each of several queues, based on relative bandwidth applied to each of the queues.

RSVP Local Policy Support

Feature History

Release	Modification
12.2(13)T	This feature was introduced.

This document describes the Resource Reservation Protocol (RSVP) Local Policy Support feature in Cisco IOS Release 12.2(13)T. It identifies the supported platforms, provides configuration examples, and lists related Cisco IOS command line interface (CLI) commands.

This document includes the following sections:

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

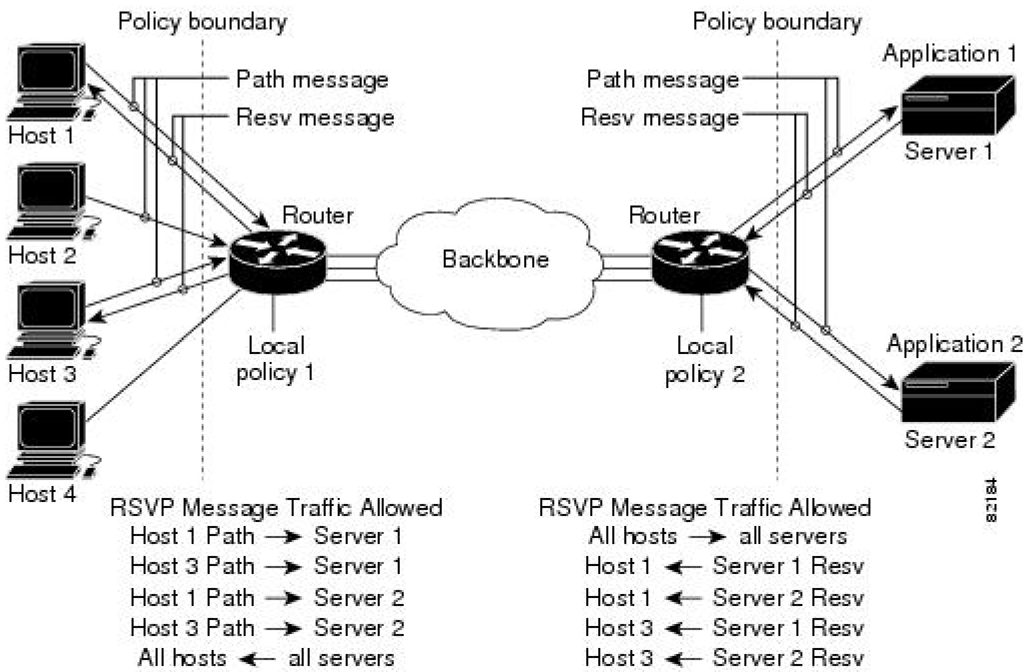
Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to www.cisco.com/go/cfn. An account on Cisco.com is not required.

Feature Overview

Network administrators need the ability to control the resources that RSVP reservations are allowed to use. For example, they may want to restrict RSVP reservations to certain subnets or from specific network servers.

The RSVP Local Policy Support feature allows network administrators to create default and access control list (ACL)-based policies. These policies, in turn, control how RSVP filters its signalling messages to allow or deny quality of service (QoS), as shown in the figure below, to networking applications based on the IP addresses of the requesting hosts.

Figure 16 ***RSVP Local Policy Configuration***



- Benefits of RSVP Local Policy Support, page 114

Benefits of RSVP Local Policy Support

RSVP Reservation Control

Network administrators can restrict the source of RSVP reservations to specific endpoints.

RSVP Reservation Preemption

High priority reservations can preempt existing reservations if there is otherwise no bandwidth available for the new, high priority reservation.

Supported Platforms

For supported platforms in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature

Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register http://www.cisco.com/register

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

http://www.cisco.com/go/fn

Availability of Cisco IOS Software Images

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

Prerequisites

RSVP must be configured on two or more routers or on one router and one host within the network before you can use the RSVP Local Policy Support feature.

Configuration Tasks

- Creating an RSVP Local Policy, page 115
- Specifying Command Line Interface Submodes, page 116
- Verifying RSVP Local Policy Configuration, page 116

Creating an RSVP Local Policy

To create an RSVP local policy, use the following command beginning in global configuration mode:

Command	Purpose
Router(config)# **ip rsvp policy local {default | acl** acl [acl1...acl8]**}**	Creates a local policy to determine how RSVP resources are used in a network.

Specifying Command Line Interface Submodes

To specify CLI submodes, use the following command beginning in local policy mode:

Command	Purpose
Router(config-rsvp-policy-local)# {**accept** | **forward** } {**all** | **path** | **path-error** | **resv** | **resv-error** }	Defines the properties of the default or ACL-based local policy that you are creating.

See the **ip rsvp policy local** command in the Cisco IOS Quality of Service Solutions Command Reference for more detailed information on submodes.

Verifying RSVP Local Policy Configuration

To verify RSVP local policy configuration, use this procedure:

SUMMARY STEPS

1. Enter the **show ip rsvp policy**command to display policy-related information including local and default policies configured, Common Open Policy Service (COPS) servers configured, and the preemption parameter configured--enabled or disabled.
2. Enter the **show ip rsvp policy local detail** command to display information about the (selected) local policies currently configured.

DETAILED STEPS

Step 1 Enter the **show ip rsvp policy**command to display policy-related information including local and default policies configured, Common Open Policy Service (COPS) servers configured, and the preemption parameter configured--enabled or disabled.

Note There are no COPS servers configured in the following output.

Example:

```
Router# show ip rsvp policy
Local policy:
    A=Accept    F=Forward
    Path:-- Resv:-- PathErr:-- ResvErr:-- ACL:104
```

```
    Path:-- Resv:-- PathErr:-- ResvErr:-- ACL:None [Default policy]
COPS:
Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Step 2 Enter the **show ip rsvp policy local detail** command to display information about the (selected) local policies currently configured.

Example:

```
Router# show ip rsvp policy local detail
Local policy for ACL(s): 104
    Preemption Priority: Start at 0, Hold at 0.
    Local Override: Disabled.
                Accept   Forward
    Path:       No       No
    Resv:       No       No
    PathError:  No       No
    ResvError:  No       No
Default local policy:
    Preemption Priority: Start at 0, Hold at 0.
    Local Override: Disabled.
                Accept   Forward
    Path:       No       No
    Resv:       No       No
    PathError:  No       No
    ResvError:  No       No
Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Monitoring and Maintaining RSVP Local Policy Support

To monitor and maintain the RSVP Local Policy Support feature, use the following commands in EXEC mode:

Command	Purpose
`Router#` **show ip rsvp policy**	Displays either the configured COPS servers or the local policies.
`Router#` **show ip rsvp policy local**	Displays selected local policies that have been configured.
`Router#` **show ip rsvp reservation detail**	Displays detailed RSVP-related receiver information currently in the database.
`Router#` **show ip rsvp sender detail**	Displays detailed RSVP-related sender information currently in the database.

Configuration Examples

Example RSVP Local Policy Support

In the following example, any RSVP nodes in the 192.168.101.0 subnet can initiate or respond to reservation requests, but all other nodes can respond only to reservation requests. This means that any 192.168.101.x node can send and receive Path, PathError, Resv, or ResvError messages. All other nodes can send only Resv or ResvError messages.

In the following example, ACL 104 is configured for a local policy:

```
Router# configure terminal
Router(config)# access-list 104 permit ip 192.168.101.0 0.0.0.255 any
Router(config)# ip rsvp policy local acl 104
Router(config-rsvp-policy-local)# forward
 all
Router(config-rsvp-policy-local)# end
```

In the following example, a default local policy is configured:

```
Router(config)# ip rsvp policy local default
Router(config-rsvp-policy-local)# forward resv
Router(config-rsvp-policy-local)# forward resverror
Router(config-rsvp-policy-local)# end
```

Additional References

The following sections provide references related to the RSVP Local Policy Support feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Signalling Overview	"Signalling Overview" module
QoS configuration tasks related to RSVP	"Configuring RSVP" module
Conceptual information and configuration tasks for classifying network traffic.	"Classifying Network Traffic" module
Congestion Management	"Congestion Management Overview" module
Cisco United Communications Manager (CallManager) and related features	"Overview of Cisco Unified Communications Manager and Cisco IOS Interoperability" module

Related Topic	Document Title
Regular expressions	"Using the Cisco IOS Command-Line Interface" module

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
None	--

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 RSVP Local Policy Support

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

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

Table 5 Feature Information for RSVP Local Policy Support

Feature Name	Releases	Feature Information
RSVP Local Policy Support	Cisco IOS XE Release 3.8S	In Cisco IOS XE Release 3.8S, support was added for the Cisco ASR 903 Router.

Glossary

access control list-- See ACL.

ACL-- access control list. An ACL consists of individual filtering rules grouped together in a single list. It is generally used to provide security filtering, though it may be used to provide a generic packet classification facility.

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

latency --The delay between the time a device receives a packet and the time that packet is forwarded out the destination port.

packet --A logical grouping of information that includes a header containing control information and (usually) user data. Packets most often refer to network layer units of data.

policy --Any defined rule that determines the use of resources within the network. A policy can be based on a user, a device, a subnetwork, a network, or an application.

port scanning --The act of systematically checking a computer's ports to find an access point.

Resource Reservation Protocol --See RSVP.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

tunnel --A secure communications path between two peers, such as routers.

Voice over IP --See VoIP.

VoIP --Voice over IP. The ability to carry normal telephony-style voice over an IP-based Internet maintaining telephone-like functionality, reliability, and voice quality.

RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature includes refresh reduction, which improves the scalability, latency, and reliability of Resource Reservation Protocol (RSVP) signaling to enhance network performance and message delivery.

History for the RSVP Refresh Reduction and Reliable Messaging Feature

Release	Modification
12.2(13)T	This feature was introduced.
12.0(24)S	This feature was integrated into Cisco IOS Release 12.0(24)S.
12.2(14)S	This feature was integrated into Cisco IOS Release 12.2(14)S.
12.0(26)S	Two commands, **ip rsvp signalling refresh misses**and **ip rsvp signalling refresh interval**, were added into Cisco IOS Release 12.0(26)S.
12.0(29)S	The *burst* and *max-size* argument defaults for the **ip rsvp signalling rate-limit** command were increased to 8 messages and 2000 bytes, respectively.
12.2(28)SB	This feature was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF5	This feature was integrated into Cisco IOS Release 12.2(18)SXF5.
12.2(33)SRB	This feature was integrated into Cisco IOS Release 12.2(33)SRB.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Refresh Reduction and Reliable Messaging

RSVP must be configured on two or more devices within the network before you can use the RSVP Refresh Reduction and Reliable Messaging feature.

Restrictions for RSVP Refresh Reduction and Reliable Messaging

Multicast flows are not supported for the reliable messages and summary refresh features.

Information About RSVP Refresh Reduction and Reliable Messaging

Feature Design of RSVP Refresh Reduction and Reliable Messaging

RSVP is a network-control, soft-state protocol that enables Internet applications to obtain special qualities of service (QoS) for their data flows. As a soft-state protocol, RSVP requires that state be periodically refreshed. If refresh messages are not transmitted during a specified interval, RSVP state automatically times out and is deleted.

In a network that uses RSVP signaling, reliability and latency problems occur when an RSVP message is lost in transmission. A lost RSVP setup message can cause a delayed or failed reservation; a lost RSVP refresh message can cause a delay in the modification of a reservation or in a reservation timeout. Intolerant applications can fail as a result.

Reliability problems can also occur when there is excessive RSVP refresh message traffic caused by a large number of reservations in the network. Using summary refresh messages can improve reliability by significantly reducing the amount of RSVP refresh traffic.

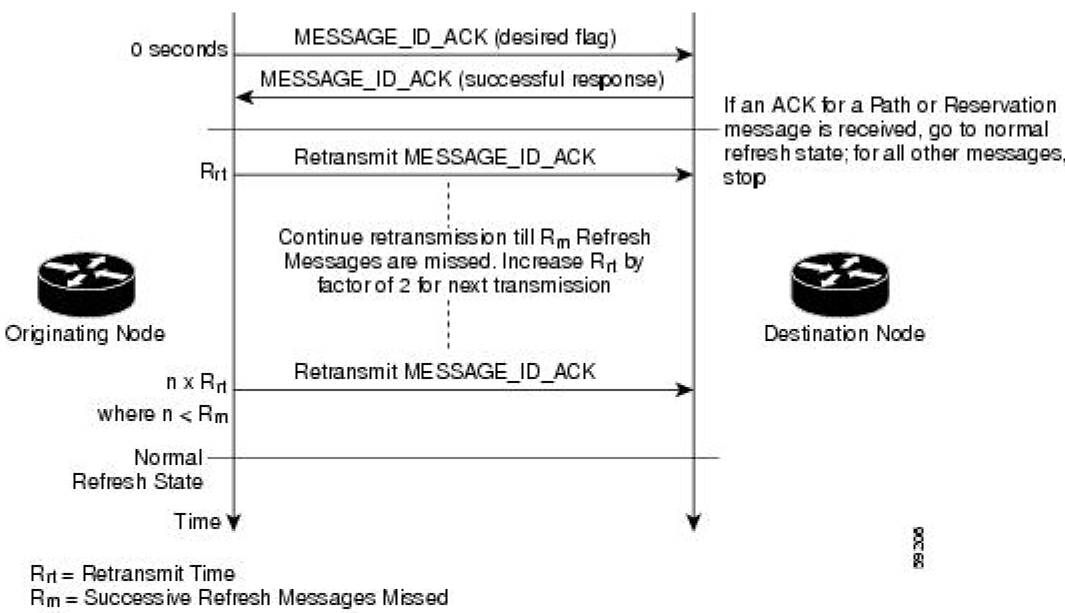
Note RSVP packets consist of headers that identify the types of messages, and object fields that contain attributes and properties describing how to interpret and act on the content.

Types of Messages in RSVP Refresh Reduction and Reliable Messaging

The RSVP Refresh Reduction and Reliable Messaging feature (see the figure below) includes refresh reduction, which improves the scalability, latency, and reliability of RSVP signaling by introducing the following extensions:

- Reliable messages (MESSAGE_ID, MESSAGE_ID_ACK objects, and ACK messages)
- Bundle messages (reception and processing only)
- Summary refresh messages (MESSAGE_ID_LIST and MESSAGE_ID_NACK objects)

Figure 17 ***RSVP Refresh Reduction and Reliable Messaging***



- Reliable Messages, page 126
- Bundle Messages, page 126
- Summary Refresh Messages, page 126

Reliable Messages

The reliable messages extension supports dependable message delivery among neighboring devices by implementing an acknowledgment mechanism that consists of a MESSAGE_ID object and a MESSAGE_ID_ACK object. The acknowledgments can be transmitted in an ACK message or piggybacked in other RSVP messages.

Each RSVP message contains one MESSAGE_ID object. If the ACK_Desired flag field is set within the MESSAGE_ID object, the receiver transmits a MESSAGE_ID_ACK object to the sender to confirm delivery.

Bundle Messages

A bundle message consists of several standard RSVP messages that are grouped into a single RSVP message.

A bundle message must contain at least one submessage. A submessage can be any RSVP message type other than another bundle message. Submessage types include Path, PathErr, Resv, ResvTear, ResvErr, ResvConf, and ACK.

Bundle messages are addressed directly to the RSVP neighbor. The bundle header immediately follows the IP header, and there is no intermediate transport header.

When a device receives a bundle message that is not addressed to one of its local IP addresses, it forwards the message.

Note Bundle messages can be received, but not sent.

Summary Refresh Messages

A summary refresh message supports the refreshing of RSVP state without the transmission of conventional Path and Resv messages. Therefore, the amount of information that must be transmitted and processed to maintain RSVP state synchronization is greatly reduced.

A summary refresh message carries a set of MESSAGE_ID objects that identify the Path and Resv states that should be refreshed. When an RSVP node receives a summary refresh message, the node matches each received MESSAGE_ID object with the locally installed Path or Resv state. If the MESSAGE_ID objects match the local state, the state is updated as if a standard RSVP refresh message were received. However, if a MESSAGE_ID object does not match the receiver’s local state, the receiver notifies the sender of the summary refresh message by transmitting a MESSAGE_ID_NACK object.

When a summary refresh message is used to refresh the state of an RSVP session, the transmission of conventional refresh messages is suppressed. The summary refresh extension cannot be used for a Path or Resv message that contains changes to a previously advertised state. Also, only a state that was previously advertised in Path or Resv messages containing MESSAGE_ID objects can be refreshed by using a summary refresh message.

Benefits of RSVP Refresh Reduction and Reliable Messaging

Enhanced Network Performance

Refresh reduction reduces the volume of steady-state network traffic generated, the amount of CPU resources used, and the response time, thereby enhancing network performance.

Improved Message Delivery

The MESSAGE_ID and the MESSAGE_ID_ACK objects ensure the reliable delivery of messages and support rapid state refresh when a network problem occurs. For example, MESSAGE_ID_ACK objects are used to detect link transmission losses.

How to Configure RSVP Refresh Reduction and Reliable Messaging

Enabling RSVP on an Interface

Perform the following task to enable RSVP on an interface.

SUMMARY STEPS

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

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet1`	Enters interface configuration mode. • The *type*and *number* arguments identify the interface to be configured.
Step 4	**ip rsvp bandwidth** [*interface-kbps* [*sub-pool*]] **Example:** `Device(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *sub-pool*arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000, and from 0 to 10000000, respectively.
Step 5	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Enabling RSVP Refresh Reduction

Perform the following task to enable RSVP refresh reduction.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling refresh reduction**
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp signalling refresh reduction** **Example:** `Device(config)# ip rsvp signalling refresh reduction`	Enables refresh reduction.
Step 4	**end** **Example:** `Device(config)#` **end**	Returns to privileged EXEC mode.

Verifying RSVP Refresh Reduction and Reliable Messaging

Perform the following task to verify that the RSVP Refresh Reduction and Reliable Messaging feature is functioning.

SUMMARY STEPS

1. **enable**
2. **clear ip rsvp counters** [**confirm**]
3. **show ip rsvp**
4. **show ip rsvp counters [interface** *interface-unit* **| summary | neighbor]**
5. **show ip rsvp interface [** *interface-type interface-number* **] [detail]**
6. **show ip rsvp neighbor [detail]**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**clear ip rsvp counters** [**confirm**] **Example:** `Device# clear ip rsvp counters`	(Optional) Clears (sets to zero) all IP RSVP counters that are being maintained by the device.
Step 3	**show ip rsvp** **Example:** `Device# show ip rsvp`	(Optional) Displays RSVP rate-limiting, refresh-reduction, and neighbor information.
Step 4	**show ip rsvp counters [interface** *interface-unit* | **summary** | **neighbor]** **Example:** `Device# show` **ip rsvp counters summary**	(Optional) Displays the number of RSVP messages that were sent and received on each interface. • The optional **summary** keyword displays the cumulative number of RSVP messages sent and received by the device over all interfaces.
Step 5	**show ip rsvp interface [** *interface-type interface-number* **] [detail]** **Example:** `Device# show ip rsvp interface detail`	(Optional) Displays information about interfaces on which RSVP is enabled including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth and signaling parameters.
Step 6	**show ip rsvp neighbor [detail]** **Example:** `Device# show ip rsvp neighbor detail`	(Optional) Displays RSVP-neighbor information including IP addresses. • The optional **detail** keyword displays the current RSVP neighbors and identifies if the neighbor is using IP, User Datagram Protocol (UDP), or RSVP encapsulation for a specified interface or all interfaces.

Configuration Examples for RSVP Refresh Reduction and Reliable Messaging

• Example RSVP Refresh Reduction and Reliable Messaging, page 130

Example RSVP Refresh Reduction and Reliable Messaging

In the following example, RSVP refresh reduction is enabled:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
```

```
Device(config)# interface Ethernet1
Device(config-if)# ip rsvp bandwidth 7500 7500
Device(config-if)# exit
Device(config)# ip rsvp signalling refresh reduction
Device(config)# end
```

The following example verifies that RSVP refresh reduction is enabled:

```
Device# show running-config
Building configuration...
Current configuration : 1503 bytes
!
version 12.2
no service single-slot-reload-enable
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
service internal
!
hostname Device
!
no logging buffered
logging rate-limit console 10 except errors
!
ip subnet-zero
ip cef
!
ip multicast-routing
no ip dhcp-client network-discovery
lcp max-session-starts 0
mpls traffic-eng tunnels
!
!
interface Loopback0
 ip address 192.168.1.1 255.255.255.0
 ip rsvp bandwidth 1705033 1705033
!
interface Tunnel777
 no ip address
 shutdown
!
interface Ethernet0
 ip address 192.168.0.195 255.0.0.0
 no ip mroute-cache
 media-type 10BaseT
!
interface Ethernet1
 ip address 192.168.5.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 ip rsvp bandwidth 7500 7500
!
interface Ethernet2
 ip address 192.168.1.2 255.255.255.0
 no ip redirects
 no ip proxy-arp
 ip pim dense-mode
 no ip mroute-cache
 media-type 10BaseT
 mpls traffic-eng tunnels
 ip rsvp bandwidth 7500 7500
!
interface Ethernet3
 ip address 192.168.2.2 255.255.255.0
 ip pim dense-mode
 media-type 10BaseT
 mpls traffic-eng tunnels
!
!
router eigrp 17
```

```
 network 192.168.0.0
 network 192.168.5.0
 network 192.168.12.0
 network 192.168.30.0
 auto-summary
 no eigrp log-neighbor-changes
!
ip classless
no ip http server
ip rsvp signalling refresh reduction
!
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line vty 0 4
 login
 transport input pad v120 telnet rlogin udptn
!
end
```

Additional References

The following sections provide references related to the RSVP Refresh Reduction and Reliable Messaging feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

Standards

Standard	Title
None	--

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource Reservation Protocol*
RFC 2206	*RSVP Management Information Base Using SMIv2*
RFC 2209	*RSVP--Version 1 Message Processing Rules*
RFC 2210	*The Use of RSVP with IETF Integrated Services*
RFC 2211/2212	*Specification of the Controlled-Load Network Element Service*
RFC 2702	*Requirements for Traffic Engineering over MPLS*
RFC 2749	*Common Open Policy Service (COPS) Usage for RSVP*
RFC 2750	*RSVP Extensions for Policy Control*
RFC 2814	*SBM Subnet Bandwidth Manager: A Protocol for RSVP-based Admission Control over IEEE 802-style Networks*
RFC 2961	*RSVP Refresh Overhead Reduction Extensions*
RFC 2996	*Format of the RSVP DCLASS Object*

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

RSVP Support for RTP Header Compression Phase 1

The Resource Reservation Protocol (RSVP) Support for Real-Time Transport Protocol (RTP) Header Compression, Phase 1 feature provides a method for decreasing a flow's reserved bandwidth requirements so that a physical link can accommodate more voice calls.

Feature Specifications for RSVP Support for RTP Header Compression, Phase 1

Feature History	
Release	Modification
12.2(15)T	This feature was introduced.
Supported Platforms	
For platforms supported in Cisco IOS Release 12.2(15)T, consult Cisco Feature Navigator.	

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Support for RTP Header Compression Phase 1

- Ensure that Real-Time Transport Protocol (RTP) or User Data Protocol (UDP) header compression is configured in the network.
- Ensure that RSVP is configured on two or more routers within the network before you can use this feature.

Restrictions for RSVP Support for RTP Header Compression Phase 1

- Routers do not generate compression hints, as described in RFC 3006, in this release.
- Signalled compression hints are not supported.
- Admission control with compression is limited to reservations with one sender per session.

Information About RSVP Support for RTP Header Compression Phase 1

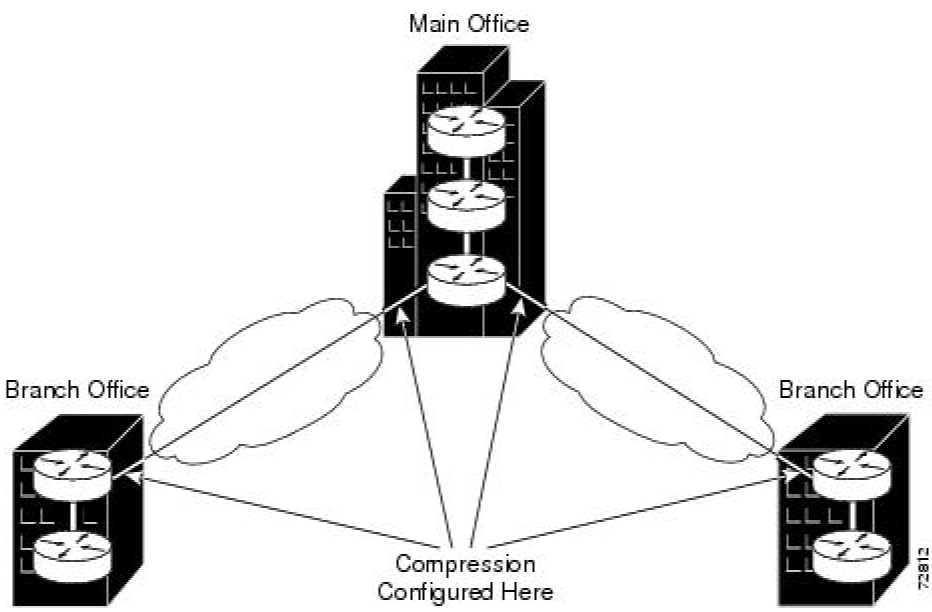
- Feature Design of RSVP Support for RTP Header Compression Phase 1, page 136
- Benefits of RSVP Support for RTP Header Compression Phase 1, page 138

Feature Design of RSVP Support for RTP Header Compression Phase 1

Network administrators use RSVP with Voice over IP (VoIP) to provide quality of service (QoS) for voice traffic in a network. Because VoIP is a real-time application, network administrators often configure compression within the network to decrease bandwidth requirements. Typically, compression is configured on slow serial lines (see the figure below), where the savings from reduced bandwidth requirements outweigh the additional costs associated with the compression and decompression processes.

Note RTP header compression is supported by Cisco routers.

Figure 18 *Configuring Compression*



Originating applications know if their traffic is considered compressible, but not whether the network can actually compress the data. Additionally, compression may be enabled on some links along the call's path, but not on others. Consequently, the originating applications must advertise their traffic's uncompressed bandwidth requirements, and receiving applications must request reservation of the full amount of bandwidth. This causes routers whose RSVP implementations do not take compression into consideration to admit the same number of flows on a link running compression as on one that is not.

- Predicting Compression within Admission Control, page 137

Predicting Compression within Admission Control

Network administrators, especially those whose networks have very low speed links, may want RSVP to use their links as fully as possible. Such links typically have minimum acceptable outgoing committed information rate (minCIR) values between 19 and 30 kbps. Without accounting for compression, RSVP can admit (at most) one G.723 voice call onto the link, despite the link's capacity for two compressed calls. Under these circumstances, network administrators may be willing to sacrifice a QoS guarantee for the last call, if the flow is less compressible than predicted, in exchange for the ability to admit it.

In order to account for compression during admission control, routers use signalled Tspec information, as well as their awareness of the compression schemes running on the flow's outbound interfaces, to make local decisions as to how much bandwidth should actually be reserved for a flow. By reserving fewer resources than signalled by the receiver, RSVP can allow links to be more fully used.

Benefits of RSVP Support for RTP Header Compression Phase 1

Additional Calls Accommodated on the Same Link

The RSVP Support for RTP Header Compression, Phase 1 feature performs admission control based on compressed bandwidth so that additional voice calls can be accommodated on the same physical link.

How to Configure RSVP Support for RTP Header Compression Phase 1

Configuring RSVP Admission-Control Compression

Note RSVP predicted compression is enabled by default.

Perform this task to configure RSVP admission-control compression.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface [*type number*]
4. **ip rsvp admission-control compression predict** [*method* {**rtp** | **udp**} [*bytes-saved N*]]
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.

	Command or Action	Purpose
Step 3	interface [*type number*] **Example:** `Router(config-if)# interface Serial3/0`	Enters interface configuration mode. • The *type number* argument identifies the interface to be configured.
Step 4	**ip rsvp admission-control compression predict** [*method* {**rtp** | **udp**} [*bytes-saved N*]] **Example:** `Router(config-if)#` **ip rsvp admission-control compression predict method udp bytes-saved 16**	Configures RSVP admission-control compression prediction. • The optional method keyword allows you to select Real-Time Transport Protocol (rtp) or User Data Protocol (udp) for your compression scheme. • The optional bytes-saved N keyword allows you to configure the predicted number of bytes saved per packet when RSVP predicts that compression will occur using the specified method.
Step 5	**end** **Example:** `Router(config-if)#` **end**	Exits to privileged EXEC mode.

Verifying RSVP Support for RTP Header Compression Phase 1 Configuration

Perform this task to verify that the RSVP Support for RTP Header Compression, Phase 1 feature is functioning.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp installed [detail]**
3. **show ip rsvp interface [interface-type interface-number] [detail]**

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	**show ip rsvp installed [detail]** **Example:** `Router# show ip rsvp installed detail`	Displays information about interfaces and their admitted reservations and the resources needed for a traffic control state block (TCSB) after taking compression into account. • The optional **detail** keyword displays the reservation's traffic parameters, downstream hop, compression, and resources used by RSVP to ensure QoS for this reservation.
Step 3	**show ip rsvp interface [interface-type interface-number] [detail]** **Example:** `Router# show ip rsvp interface detail`	Displays information about interfaces on which RSVP is enabled, including the current allocation budget and maximum available bandwidth and the RSVP bandwidth limit counter, taking compression into account. • The optional **detail** keyword displays RSVP parameters associated with an interface including bandwidth, admission control, and compression methods.

- Examples, page 140
- Troubleshooting Tips, page 141

Examples

- Sample Output for the show ip rsvp installed detail Command, page 140
- Sample Output for the show ip rsvp interface detail Command, page 140

Sample Output for the show ip rsvp installed detail Command

In this example, the show ip rsvp installed detail command displays information, including the predicted compression method, its reserved context ID, and the observed bytes saved per packet average, for the admitted flowspec.

```
Router# show ip rsvp installed detail
RSVP: Ethernet2/1 has no installed reservations
RSVP: Serial3/0 has the following installed reservations
RSVP Reservation. Destination is 10.1.1.2. Source is 10.1.1.1,
  Protocol is UDP, Destination port is 18054, Source port is 19156
  Compression: (method rtp, context ID = 1, 37.98 bytes-saved/pkt avg)
  Admitted flowspec:
    Reserved bandwidth: 65600 bits/sec, Maximum burst: 328 bytes, Peak rate: 80K bits/sec
    Min Policed Unit: 164 bytes, Max Pkt Size: 164 bytes
  Admitted flowspec (as required if compression were not applied):
    Reserved bandwidth: 80K bits/sec, Maximum burst: 400 bytes, Peak rate: 80K bits/sec
    Min Policed Unit: 200 bytes, Max Pkt Size: 200 bytes
  Resource provider for this flow:
    WFQ on FR PVC dlci 101 on Se3/0:  PRIORITY queue 24.  Weight: 0, BW 66 kbps
  Conversation supports 1 reservations [0x1000405]
  Data given reserved service: 3963 packets (642085 bytes)
Data given best-effort service: 0 packets (0 bytes)
Reserved traffic classified for 80 seconds
  Long-term average bitrate (bits/sec): 64901 reserved, 0 best-effort
  Policy: INSTALL. Policy source(s): Default
```

Sample Output for the show ip rsvp interface detail Command

In this example, the show ip rsvp interface detail command displays the current interfaces and their configured compression parameters.

```
Router# show ip rsvp interface detail
 Et2/1:
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 1158K bits/sec
     Max. allowed (per flow): 128K 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)
Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: disabled


Se3/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 1158K bits/sec
     Max. allowed (per flow): 128K 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)
   Neighbors:
     Using IP encap: 1.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: disabled
```

Troubleshooting Tips

The observed bytes-saved per packet value should not be less than the configured or default value. Otherwise, the flow may be experiencing degraded QoS. To avoid any QoS degradation for future flows, configure a lower bytes-saved per packet value.

Flows may achieve less compressibility than the default RSVP assumes for many reasons, including packets arriving out of order or having different differentiated services code point (DSCP) or precedence values, for example, due to policing upstream within the network.

If compression is enabled on a flow’s interface, but the compression prediction was unsuccessful, the reason appears in the output instead of the reserved compression ID and the observed bytes-saved per packet.

Configuration Examples for RSVP Support for RTP Header Compression Phase 1

• Example RSVP Support for RTP Header Compression Phase 1, page 142

Example RSVP Support for RTP Header Compression Phase 1

The following sample configuration shows the compression prediction enabled for flows using UDP and disabled for flows using RTP:

```
Router# configure terminal

Router(config)# interface Serial3/0
Router(config-if)# ip rsvp admission-control compression predict method udp bytes-saved 16
Router(config-if)# no
ip rsvp admission-control compression predict method rtp
```

Use the **show run** command to display all the RSVP configured parameters:

```
Router# show run

2d18h: %SYS-5-CONFIG_I: Configured from console by console

Router# show run int se3/0
Building configuration...


Current configuration : 339 bytes
!
interface Serial3/0
 ip address 10.2.1.1 255.255.0.0
 fair-queue 64 256 8
 serial restart_delay 0
 clock rate 128000
 ip rtp header-compression
 ip rsvp bandwidth
 no ip rsvp admission-control compression predict method rtp
 ip rsvp admission-control compression predict method udp bytes-saved 16
end
```

Additional References

For additional information related to RSVP Support for RTP Header Compression, Phase 1, refer to the following references:

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Signalling	"Signalling Overview" module
RSVP	"Configuring RSVP" module
Header compression concepts and topics	"Header Compression" module

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[1]	MIBs Link
• RFC 2206, RSVP Management Information Base using SMIv2	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

http://tools.cisco.com/ITDIT/MIBS/servlet/index

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to cco-locksmith@cisco.com. An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

http://www.cisco.com/register

RFCs

RFCs[2]	Title
RFC 2205	Resource Reservation Protocol (RSVP)
RFC 2508	Compressing IP/UDP/RTP Headers for Low-Speed Serial Links
RFC 3006	Integrated Services in the Presence of Compressible Flows

1 Not all supported MIBs are listed.
2 Not all supported RFCs are listed.

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 RSVP Support for RTP Header Compression

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

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

Table 6 Feature Information for RSVP Support for RTP Header Compression

Feature Name	Releases	Feature Information
RSVP Support for RTP Header Compression	Cisco IOS XE Release 3.8S	In Cisco IOS XE Release 3.8S, support was added for the Cisco ASR 903 Router.

Glossary

admission control --The process in which a Resource Reservation Protocol (RSVP) reservation is accepted or rejected based on end-to-end available network resources.

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

compression --The running of a data set through an algorithm that reduces the space required to store or the bandwidth required to transmit the data set.

DSCP --differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

flowspec --In IPv6, the traffic parameters of a stream of IP packets between two applications.

G.723 --A compression technique that can be used for compressing speech or audio signal components at a very low bit rate as part of the H.324 family of standards. This codec has two bit rates associated with it: 5.3 and 6.3 kbps. The higher bit rate is based on ML-MLQ technology and provides a somewhat higher quality of sound. The lower bit rate is based on code excited linear prediction (CELP) compression and provides system designers with additional flexibility. Described in the ITU-T standard in its G-series recommendations.

minCIR --The minimum acceptable incoming or outgoing committed information rate (CIR) for a Frame Relay virtual circuit.

packet --A logical grouping of information that includes a header containing control information and (usually) user data. Packets most often refer to network layer units of data.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

RTP --Real-Time Transport Protocol. A protocol that is designed to provide end-to-end network transport functions for applications transmitting real-time data, such as audio, video, or simulation data, over multicast or unicast network services. RTP provides such services as payload type identification, sequence numbering, timestamping, and delivery monitoring to real-time applications.

TCSB --traffic control state block. A Resource Reservatiion Protocol (RSVP) state that associates reservations with their reserved resources required for admission control.

Tspec --Traffic specification. The traffic characteristics of a data stream from a sender or receiver (included in a Path message).

UDP--User Datagram Protocol. A connectionless transport layer protocol in the TCP/IP protocol stack. UDP is a simple protocol that exchanges datagrams without acknowledgments or guaranteed delivery, requiring that error processing and retransmission be handled by other protocols. UDP is defined in RFC 768.

VoIP --Voice over IP. The ability to carry normal telephony-style voice over an IP-based Internet maintaining telephone-like functionality, reliability, and voice quality.

RSVP Message Authentication

The Resource Reservation Protocol (RSVP) Message Authentication feature provides a secure method to control quality of service (QoS) access to a network.

History for the RSVP Message Authentication Feature

Release	Modification
12.2(15)T	This feature was introduced.
12.0(26)S	Restrictions were added for interfaces that use Fast Reroute (FRR) node or link protection and for RSVP hellos for FRR for packet over SONET (POS) interfaces.
12.0(29)S	Support was added for per-neighbor keys.
12.2(33)SRA	This feature was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This feature was integrated into Cisco IOS Release 12.2(33)SXH.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Message Authentication

Ensure that RSVP is configured on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for RSVP Message Authentication

- The RSVP Message Authentication feature is only for authenticating RSVP neighbors.
- The RSVP Message Authentication feature cannot discriminate between various QoS applications or users, of which many may exist on an authenticated RSVP neighbor.
- Different send and accept lifetimes for the same key in a specific key chain are not supported; all RSVP key types are bidirectional.
- Authentication for graceful restart hello messages is supported for per-neighbor and per-access control list (ACL) keys, but not for per-interface keys.
- You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same device interface.
- For a Multiprotocol Label Switching/Traffic Engineering (MPLS/TE) configuration, use per-neighbor keys with physical addresses and device IDs.

Information About RSVP Message Authentication

Feature Design of RSVP Message Authentication

Network administrators need the ability to establish a security domain to control the set of systems that initiate RSVP requests.

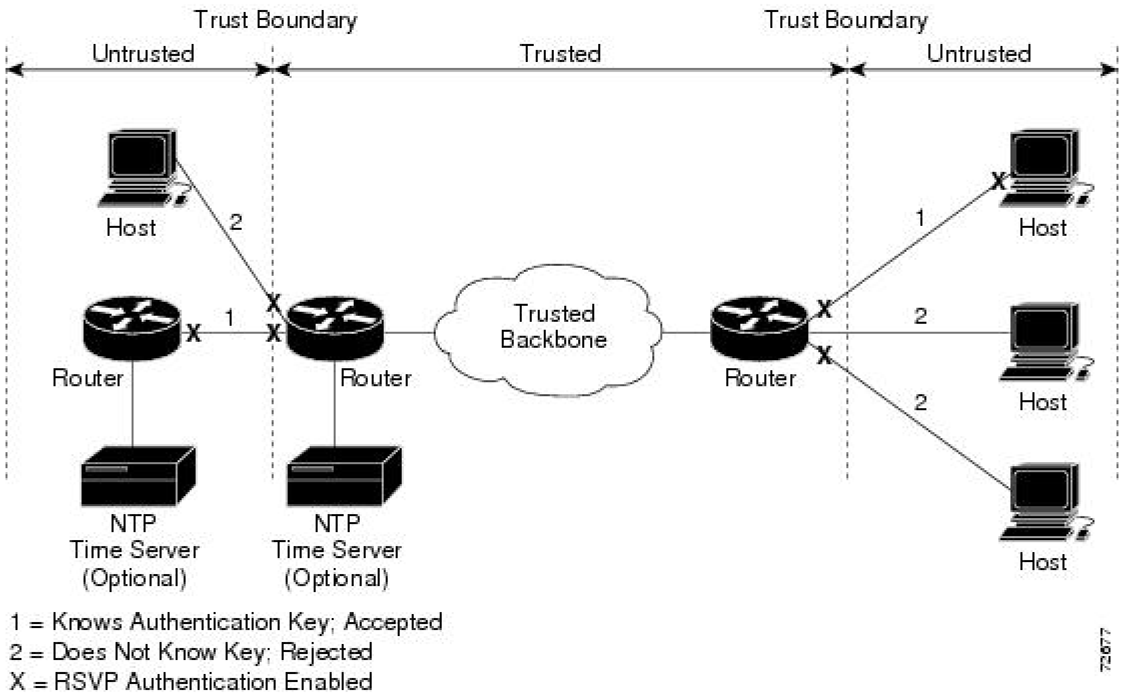
The RSVP Message Authentication feature permits neighbors in an RSVP network to use a secure hash to sign all RSVP signaling messages digitally, thus allowing the receiver of an RSVP message to verify the sender of the message without relying solely on the sender's IP address as is done by issuing the **ip rsvp neighbor** command with an ACL.

The signature is accomplished on a per-RSVP-hop basis with an RSVP integrity object in the RSVP message as defined in RFC 2747. This method provides protection against forgery or message modification. However, the receiver must know the security key used by the sender in order to validate the digital signature in the received RSVP message.

Network administrators manually configure a common key for each RSVP neighbor interface on the shared network. A sample configuration is shown in the figure below.

Figure 19

RSVP Message Authentication Configuration



Global Authentication and Parameter Inheritance

You can configure global defaults for all authentication parameters including key, type, window size, lifetime, and challenge. These defaults are inherited when you enable authentication for each neighbor or interface. However, you can also configure these parameters individually on a per-neighbor or per-interface basis in which case the inherited global defaults are ignored.

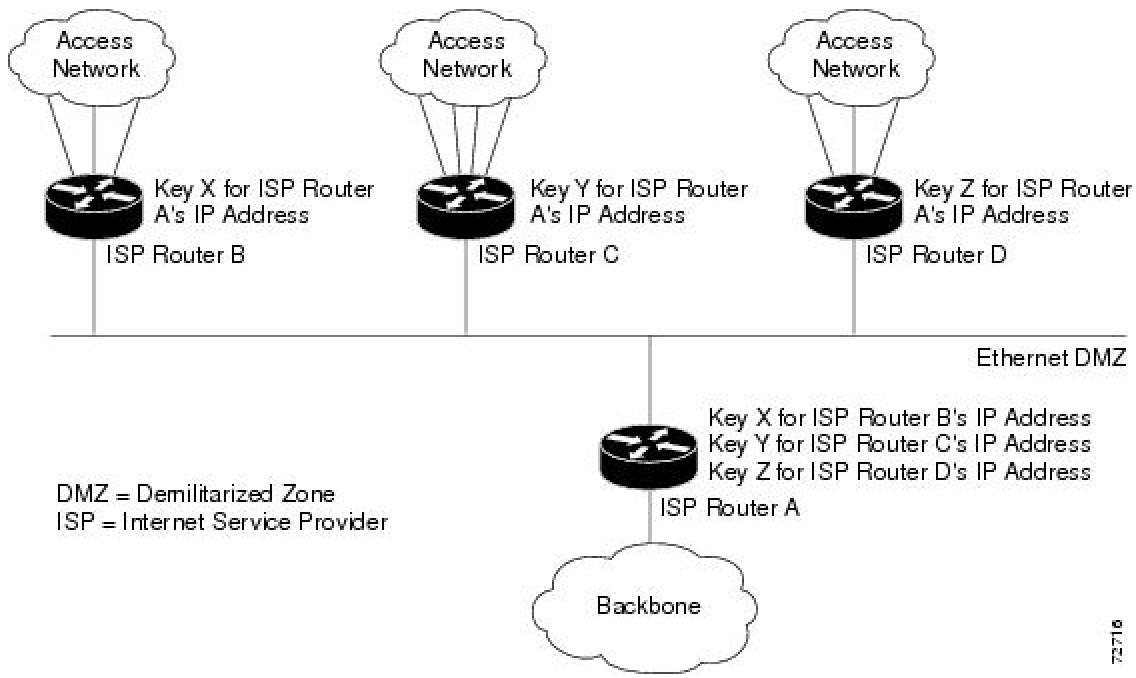
Using global authentication and parameter inheritance can simplify configuration because you can enable or disable authentication without having to change each per-neighbor or per-interface attribute. You can activate authentication for all neighbors by using two commands, one to define a global default key and one to enable authentication globally. However, using the same key for all neighbors does not provide the best network security.

Note RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (per-interface, per-neighbor, or global). RSVP goes from the most specific to the least specific; that is, per-neighbor, per-interface, and then global. The rules are slightly different when searching the configuration for the right key to authenticate an RSVP message-- per-neighbor, per-ACL, per-interface, and then global.

Per-Neighbor Keys

In the figure below, to enable authentication between Internet service provider (ISP) Routers A and B, A and C, and A and D, the ISPs must share a common key. However, sharing a common key also enables authentication between ISP Routers B and C, C and D, and B and D. You may not want authentication among all the ISPs because they might be different companies with unique security domains.

Figure 20 RSVP Message Authentication in an Ethernet Configuration



On ISP Router A, you create a different key for ISP Routers B, C, and D and assign them to their respective IP addresses using RSVP commands. On the other devices, create a key to communicate with ISP Router A’s IP address.

Key Chains

For each RSVP neighbor, you can configure a list of keys with specific IDs that are unique and have different lifetimes so that keys can be changed at predetermined intervals automatically without any disruption of service. Automatic key rotation enhances network security by minimizing the problems that could result if an untrusted source obtained, deduced, or guessed the current key.

Note If you use overlapping time windows for your key lifetimes, RSVP asks the Cisco software key manager component for the next live key starting at time T. The key manager walks the keys in the chain until it finds the first one with start time S and end time E such that S <= T <= E. Therefore, the key with the smallest value (E-T) may not be used next.

Benefits of RSVP Message Authentication

Improved Security

The RSVP Message Authentication feature greatly reduces the chance of an RSVP-based spoofing attack and provides a secure method to control QoS access to a network.

Multiple Environments

The RSVP Message Authentication feature can be used in traffic engineering (TE) and non-TE environments as well as with the subnetwork bandwidth manager (SBM).

Multiple Platforms and Interfaces

The RSVP Message Authentication feature can be used on any supported RSVP platform or interface.

How to Configure RSVP Message Authentication

The following configuration parameters instruct RSVP on how to generate and verify integrity objects in various RSVP messages.

Note There are two configuration procedures: full and minimal. There are also two types of authentication procedures: interface and neighbor.

Per-Interface Authentication--Full Configuration

Perform the following procedures for a full configuration for per-interface authentication:

Per-Interface Authentication--Minimal Configuration

Perform the following tasks for a minimal configuration for per-interface authentication:

Per-Neighbor Authentication--Full Configuration

Perform the following procedures for a full configuration for per-neighbor authentication:

Per-Neighbor Authentication--Minimal Configuration

Perform the following tasks for a minimal configuration for per-neighbor authentication:

- Enabling RSVP on an Interface, page 152
- Configuring an RSVP Authentication Type, page 153
- Configuring an RSVP Authentication Key, page 155
- Enabling RSVP Key Encryption, page 158
- Enabling RSVP Authentication Challenge, page 158
- Configuring RSVP Authentication Lifetime, page 161
- Configuring RSVP Authentication Window Size, page 164
- Activating RSVP Authentication, page 167

Enabling RSVP on an Interface

Perform this task to enable RSVP on an interface.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured.
Step 4	ip rsvp bandwidth [*interface-kbps* [*single-flow-kbps*]] **Example:** `Device(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10,000,000. **Note** Repeat this command for each interface that you want to enable.

	Command or Action	Purpose
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Type

Perform this task to configure an RSVP authentication type.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip rsvp authentication type** {**md5** | **sha-1**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication type for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ip rsvp authentication type** {**md5** | **sha-1**
	
	Example:
	`For interface authentication:`
	
	Example:
	`Device(config-if)#` **ip rsvp authentication type sha-1**
	
	Example:
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1 type sha-1**
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1 type sha-1**
	
	Example:
	
	Example:
	`For a global default:`
		Specifies the algorithm used to generate cryptographic signatures in RSVP messages on an interface or globally.
		• The algorithms are **md5**, the default, and **sha-1**, which is newer and more secure than **md5**.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

	Command or Action	Purpose
	Example: `Device(config)#` **ip rsvp authentication type sha-1**	
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Configuring an RSVP Authentication Key

Perform this task to configure an RSVP authentication key.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp authentication key passphrase**
5. **exit**
6. Do one of the following:
   - **ip rsvp authentication key-chain** *chain*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode. **Note** If you want to configure a key, proceed to Step 3; if you want to configure a key chain, proceed to Step 6.

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured. **Note** Omit this step and go to Step 6 if you want to configure only a key chain.
Step 4	**ip rsvp authentication key passphrase** **Example:** `Device(config-if)#` **ip rsvp authentication key** `11223344` **Example:**	Specifies the data string (key) for the authentication algorithm. • The key consists of 8 to 40 characters. It can include spaces and multiple words. It can also be encrypted or appear in clear text when displayed. **Note** Omit this step if you want to configure a key chain.
Step 5	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.

	Command or Action	Purpose

Step 6	Do one of the following:
	• **ip rsvp authentication key-chain** *chain*
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1 key-chain xzy**
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1 key-chain xzy**
	
	Example:
	
	Example:
	`For a global default:`
	
	Example:
	`Device(config)#` **ip rsvp authentication key-chain xzy**
		Specifies the data string (key chain) for the authentication algorithm.
		• The key chain must have at least one key, but can have up to 2,147,483647 keys.
		
		Note You cannot use the **ip rsvp authentication key** and the **ip rsvp authentication key-chain** commands on the same device interface. The commands supersede each other; however, no error message is generated.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

Step 7	**end** **Example:** `Device(config)#` **end**	Returns to privileged EXEC mode.

Enabling RSVP Key Encryption

Perform this task to enable RSVP key encryption when the key is stored in the configuration. (This prevents anyone from seeing the clear text key in the configuration file.)

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key config-key 1** *string*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key config-key 1** *string* **Example:** `Device(config)# key config-key 1 11223344`	Enables key encryption in the configuration file. **Note** The *string* argument can contain up to eight alphanumeric characters.
Step 4	**end** **Example:** `Device(config)#` **end**	Returns to privileged EXEC mode.

Enabling RSVP Authentication Challenge

Perform this task to enable RSVP authentication challenge.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip rsvp authentication challenge**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured. **Note** Omit this step if you are configuring an authentication challenge for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ip rsvp authentication challenge**
	
	Example:
	`For interface authentication:`
	
	Example:
	`Device(config-if)#` **ip rsvp authentication challenge**
	
	Example:
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1 challenge**
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1 challenge**
	
	Example:
	
	Example:
	`For a global default:`
		Makes RSVP perform a challenge-response handshake on an interface or globally when RSVP learns about any new challenge-capable neighbors on a network.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

	Command or Action	Purpose
	Example: `Device(config)#` **ip rsvp authentication challenge**	
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Configuring RSVP Authentication Lifetime

Perform this task to configure the lifetimes of security associations between RSVP neighbors.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip rsvp authentication lifetime** *hh* **:** *mm* **:** *ss*
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. **Note** Omit this step if you are configuring an authentication lifetime for a neighbor or setting a global default. • The *type number*argument identifies the interface to be configured.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ip rsvp authentication lifetime** *hh* **:** *mm* **:** *ss*
	
	Example:
	`For interface authentication:`
	
	Example:
	`Device(config-if)#` **ip rsvp authentication lifetime 00:05:00**
	
	Example:
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1 lifetime 00:05:00**
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1 lifetime 00:05:00**
	
	Example:
	
	Example:
	`For a global default:`
		Controls how long RSVP maintains security associations with RSVP neighbors on an interface or globally.
		• The default security association for **hh:mm:ss** is 30 minutes; the range is 1 second to 24 hours.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

	Command or Action	Purpose
	Example: `Device(config)#` **ip rsvp authentication 00:05:00**	
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Configuring RSVP Authentication Window Size

Perform this task to configure the RSVP authentication window size.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip rsvp authentication window-size n**
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured. **Note** Omit this step if you are configuring a window size for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ip rsvp authentication window-size n**
	
	Example:
	`For interface authentication:`
	
	Example:
	`Device(config-if)#` **ip rsvp authentication** `window-size 2`
	
	Example:
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1** `window-size 2`
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1** `window-size`
	
	Example:
	
	Example:
	`For a global default:`
		Specifies the maximum number of authenticated messages that can be received out of order on an interface or globally.
		• The default value is one message; the range is 1 to 64 messages.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

	Command or Action	Purpose
	Example: `Device(config)#` **ip rsvp authentication** `window-size 2`	
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Activating RSVP Authentication

Perform this task to activate RSVP authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Do one of the following:
   - **ip rsvp authentication**
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Enters interface configuration mode. • The *type number*argument identifies the interface to be configured. **Note** Omit this step if you are configuring authentication for a neighbor or setting a global default.

	Command or Action	Purpose

Step 4	Do one of the following:
	• **ip rsvp authentication**
	
	Example:
	`For interface authentication:`
	
	Example:
	`Device(config-if)#` **ip rsvp authentication**
	
	Example:
	
	Example:
	`For neighbor authentication:`
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor address 10.1.1.1**
	
	Example:
	
	Example:
	`Device(config)#` **ip rsvp authentication neighbor access-list 1**
	
	Example:
	
	Example:
	`For a global default:`
		Activates RSVP cryptographic authentication on an interface or globally.
		
		Note Omit the **neighbor address** *address*or the **neighbor access-list** *acl-name*or *acl-number*to set the global default.

	Command or Action	Purpose
	Example: `Device(config)#` **ip rsvp authentication**	
Step 5	**end** **Example:** `Device(config-if)#` **end**	Returns to privileged EXEC mode.

Verifying RSVP Message Authentication

Perform this task to verify that the RSVP Message Authentication feature is functioning.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp interface [detail] [interface-type interface-number]**
3. **show ip rsvp authentication** [**detail**] [**from**{*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}]
4. **show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp interface [detail] [interface-type interface-number]** **Example:** `Device# show ip rsvp interface detail`	Displays information about interfaces on which RSVP is enabled, including the current allocation budget and maximum available bandwidth. • The optional **detail** keyword displays the bandwidth, signaling, and authentication parameters.

	Command or Action	Purpose
Step 3	**show ip rsvp authentication** [**detail**] [**from**{*ip-address* | *hostname*}] [**to** {*ip-address* | *hostname*}] **Example:** `Device# show ip rsvp authentication detail`	Displays the security associations that RSVP has established with other RSVP neighbors. • The optional **detail** keyword displays state information that includes IP addresses, interfaces enabled, and configured cryptographic authentication parameters about security associations that RSVP has established with neighbors.

Step 4	**show ip rsvp counters** [**authentication** | **interface** *interface-unit* | **neighbor** | **summary**]
	
	Example:
	
	`Device# show ip rsvp counters summary`
	
	Example:
	
	Example:
	
	`Device# show ip rsvp counters authentication`
		Displays all RSVP counters.
		
		Note The errors counter increments whenever an authentication error occurs, but can also increment for errors not related to authentication.
		
		• The optional **authentication** keyword shows a list of RSVP authentication counters.
		• The optional **interface** *interface-unit* keyword argument combination shows the number of RSVP messages sent and received by the specific interface.
		• The optional **neighbor**keyword shows the number of RSVP messages sent and received by the specific neighbor.
		• The optional **summary**keyword shows the cumulative number of RSVP messages sent and received by the device. It does not print per-interface counters.

Configuring a Key Chain

Perform this task to configure a key chain for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **key chain** *name-of-chain*
4. {**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**key chain** *name-of-chain* **Example:** `Device(config)# key chain neighbor_V`	Enters key-chain mode.

Step 4	{**key** [*key-ID*] | **key-string** [*text*] | **accept-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}] | **send-lifetime** [*start-time* {**infinite** | *end-time* | **duration** *seconds*}]
	
	Example:
	
	`Device(config-keychain)# key 1`
	
	Example:
	
	Example:
	
	`Device(config-keychain)# key-string ABcXyz`
		Selects the parameters for the key chain. (These are submodes.)
		
		Note For details on these parameters, see the Cisco IOS IP Command Reference, Volume 2 of 4, Routing Protocols, Release 12.3T.
		
		Note **accept-lifetime** is ignored when a key chain is assigned to RSVP.

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

Binding a Key Chain to an RSVP Neighbor

Perform this task to bind a key chain to an RSVP neighbor for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - ip rsvp authentication neighbor address *address* **key-chain** *key-chain-name*
   -
   - **ip rsvp authentication neighbor access-list** *acl-name* **or** *acl-number* **key-chain** *key-chain-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	Do one of the following: • ip rsvp authentication neighbor address *address* **key-chain** *key-chain-name* • • **ip rsvp authentication neighbor access-list** *acl-name* **or** *acl-number* **key-chain** *key-chain-name* **Example:** `Device(config)# ip rsvp authentication neighbor access-list 1 key-chain neighbor_V`	Binds a key chain to an IP address or to an ACL and enters key-chain mode. **Note** If you are using an ACL, you must create it before you bind it to a key chain. See the ip rsvp authentication command in the Glossary, page 179 section for examples.
Step 4	**end** **Example:** `Device(config-keychain)# end`	Returns to privileged EXEC mode.

Troubleshooting Tips

After you enable RSVP authentication, RSVP logs system error events whenever an authentication check fails. These events are logged instead of just being displayed when debugging is enabled because they may indicate potential security attacks. The events are generated when:

- RSVP receives a message that does not contain the correct cryptographic signature. This could be due to misconfiguration of the authentication key or algorithm on one or more RSVP neighbors, but it may also indicate an (unsuccessful) attack.
- RSVP receives a message with the correct cryptographic signature, but with a duplicate authentication sequence number. This may indicate an (unsuccessful) message replay attack.
- RSVP receives a message with the correct cryptographic signature, but with an authentication sequence number that is outside the receive window. This could be due to a reordered burst of valid RSVP messages, but it may also indicate an (unsuccessful) message replay attack.
- Failed challenges result from timeouts or bad challenge responses.

To troubleshoot the RSVP Message Authentication feature, use the following commands in privileged EXEC mode.

Command	Purpose
`Device#` **`debug ip rsvp authentication`**	Displays output related to RSVP authentication.
`Device#` **debug ip rsvp dump signalling**	Displays brief information about signaling (Path and Resv) messages.
`Device#` **debug ip rsvp errors**	Displays error events including authentication errors.

Configuration Examples for RSVP Message Authentication

Example RSVP Message Authentication Per-Interface

In the following example, the cryptographic authentication parameters, including type, key, challenge, lifetime, and window size are configured; and authentication is activated:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface e0/0
Device(config-if)# ip rsvp bandwidth 7500 7500
Device(config-if)# ip rsvp authentication type sha-1
Device(config-if)# ip rsvp authentication key 11223344
Device(config-if)# ip rsvp authentication challenge
Device(config-if)# ip rsvp authentication lifetime 00:30:05
Device(config-if)# ip rsvp authentication window-size 2
Device(config-if)# ip rsvp authentication
```

In the following output from the **show ip rsvp interface detail** command, notice the cryptographic authentication parameters that you configured for the Ethernet0/0 interface:

```
Device# show ip rsvp interface detail
Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         11223344
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the preceding example, the authentication key appears in clear text. If you enter the **key-config-key 1** *string* command, the key appears encrypted, as in the following example:

```
Device# show ip rsvp interface detail
 Et0/0:
   Bandwidth:
     Curr allocated: 0 bits/sec
Max. allowed (total): 7500K bits/sec
     Max. allowed (per flow): 7500K bits/sec
     Max. allowed for LSP tunnels using sub-pools: 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Neighbors:
     Using IP encap: 0.  Using UDP encap: 0
   Signalling:
     Refresh reduction: disabled
   Authentication: enabled
     Key:         <encrypted>
     Type:        sha-1
     Window size: 2
     Challenge:   enabled
```

In the following output, notice that the authentication key changes from encrypted to clear text after the **no key config-key 1** command is issued:

```
Device# show running-config interface e0/0
Building configuration...
Current configuration :247 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 7>70>9:7<872>?74
 ip rsvp authentication
end
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# no key config-key 1

Device(config)# end


Device# show running-config
*Jan 30  08:02:09.559:%SYS-5-CONFIG_I:Configured from console by console
int e0/0
Building configuration...
```

```
Current configuration :239 bytes
!
interface Ethernet0/0
 ip address 192.168.101.2 255.255.255.0
 no ip directed-broadcast
 ip pim dense-mode
 no ip mroute-cache
 no cdp enable
 ip rsvp bandwidth 7500 7500
 ip rsvp authentication key 11223344
 ip rsvp authentication
end
```

Example RSVP Message Authentication Per-Neighbor

In the following example, a key chain with two keys for each neighbor is defined, then an access list and a key chain are created for neighbors V, Y, and Z and authentication is explicitly enabled for each neighbor and globally. However, only the neighbors specified will have their messages accepted; messages from other sources will be rejected. This enhances network security.

For security reasons, you should change keys on a regular basis. When the first key expires, the second key automatically takes over. At that point, you should change the first key's key-string to a new value and then set the send lifetimes to take over after the second key expires. The device will log an event when a key expires to remind you to update it.

The lifetimes of the first and second keys for each neighbor overlap. This allows for any clock synchronization problems that might cause the neighbors not to switch keys at the right time. You can avoid these overlaps by configuring the neighbors to use Network Time Protocol (NTP) to synchronize their clocks to a time server.

For an MPLS/TE configuration, physical addresses and device IDs are given.

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# key chain neighbor_V
Device(config-keychain)# key 1
Device(config-keychain-key)# key-string R72*UiAXy
Device(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# key 2
Device(config-keychain-key)# key-string Pl349&DaQ
Device(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# key chain neighbor_Y
Device(config-keychain)# key 3
Device(config-keychain-key)# key-string *ZXFwR!03
Device(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# key 4
Device(config-keychain-key)# key-string UnGR8f&lOmY
Device(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# key chain neighbor_Z
Device(config-keychain)# key 5
Device(config-keychain-key)# key-string P+T=77&/M
Device(config-keychain-key)# send-life 02:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# key 6
Device(config-keychain-key)# key-string payattention2me
Device(config-keychain-key)# send-life 01:00:00 1 jun 2003 02:00:00 1 aug 2003
Device(config-keychain-key)# exit
Device(config-keychain)# exit
Device(config)# end
```

> **Note** You can use the **key-config-key 1** *string* command to encrypt key chains for an interface, a neighbor, or globally.

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip access-list standard neighbor_V
Device(config-std-nacl)# permit 10.0.0.1
<------- physical address
Device(config-std-nacl)# permit 10.0.0.2
<------- physical address
Device(config-std-nacl)# permit 10.0.0.3
<------- router ID
Device(config-std-nacl)# exit
Device(config)# ip access-list standard neighbor_Y
Device(config-std-nacl)# permit 10.0.0.4
<------- physical address
Device(config-std-nacl)# permit 10.0.0.5
<------- physical address
Device(config-std-nacl)# permit 10.0.0.6
<------- router ID
Device(config-std-nacl)# exit
Device(config)# ip access-list standard neighbor_Z
Device(config-std-nacl)# permit 10.0.0.7
<------- physical address
Device(config-std-nacl)# permit 10.0.0.8
<------- physical address
Device(config-std-nacl)# permit 10.0.0.9
<------- router ID
Device(config-std-nacl)# exit
Device(config)# ip rsvp authentication neighbor access-list neighbor_V key-chain
neighbor_V
Device(config)# ip rsvp authentication neighbor access-list neighbor_Y key-chain
neighbor_Y
Device(config)# ip rsvp authentication neighbor access-list neighbor_Z key-chain
neighbor_Z
Device(config)# ip rsvp authentication
Device(config)# end
```

Additional References

The following sections provide references related to the RSVP Message Authentication feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Inter-AS features including local policy support and per-neighbor keys authentication	"MPLS Traffic Engineering--Inter-AS-TE" module

Standards

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

MIBs

MIBs	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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFCs	Title
RFC 1321	*The MD5 Message Digest Algorithm*
RFC 2104	*HMAC: Keyed-Hashing for Messaging Authentication*
RFC 2205	*Resource Reservation Protocol*
RFC 2209	*RSVP--Version 1 Message Processing Rules*
RFC 2401	*Security Architecture for the Internet Protocol*
RFC 2747	*RSVP Cryptographic Authentication*
RFC 3097	*RSVP Crytographic Authentication--Updated Message Type Value*
RFC 3174	*US Secure Hash Algorithm 1 (SHA1)*

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 RSVP Message Authentication

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

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

Table 7 Feature Information for RSVP Message Authentication

Feature Name	Releases	Feature Information
RSVP Message Authentication	Cisco IOS XE Release 3.8S	In Cisco IOS XE Release 3.8S, support was added for the Cisco ASR 903 Router.

Glossary

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

DMZ--demilitarized zone. The neutral zone between public and corporate networks.

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

key --A data string that is combined with source data according to an algorithm to produce output that is unreadable until decrypted.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another based on network layer information.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams they want to receive.

security association --A block of memory used to hold all the information RSVP needs to authenticate RSVP signaling messages from a specific RSVP neighbor.

spoofing --The act of a packet illegally claiming to be from an address from which it was not actually sent. Spoofing is designed to foil network security mechanisms, such as filters and access lists.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

trusted neighbor --A device with authorized access to information.

RSVP-Previous Hop Overwrite

The RSVP--Previous Hop Overwrite feature allows you to configure a Resource Reservation Protocol (RSVP) device, on a per interface basis, to populate an address other than the native interface address in the previous hop (PHOP) address field of the PHOP object when forwarding a PATH message onto that interface. You can configure the actual address for the device to use or an interface, including a loopback, from which to borrow the address.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP-Previous Hop Overwrite

You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for RSVP-Previous Hop Overwrite

- This feature is supported only on integrated services routers (ISRs).
- Unnumbered IP addresses are not allowed.

Information About RSVP-Previous Hop Overwrite

Feature Overview of RSVP-Previous Hop Overwrite

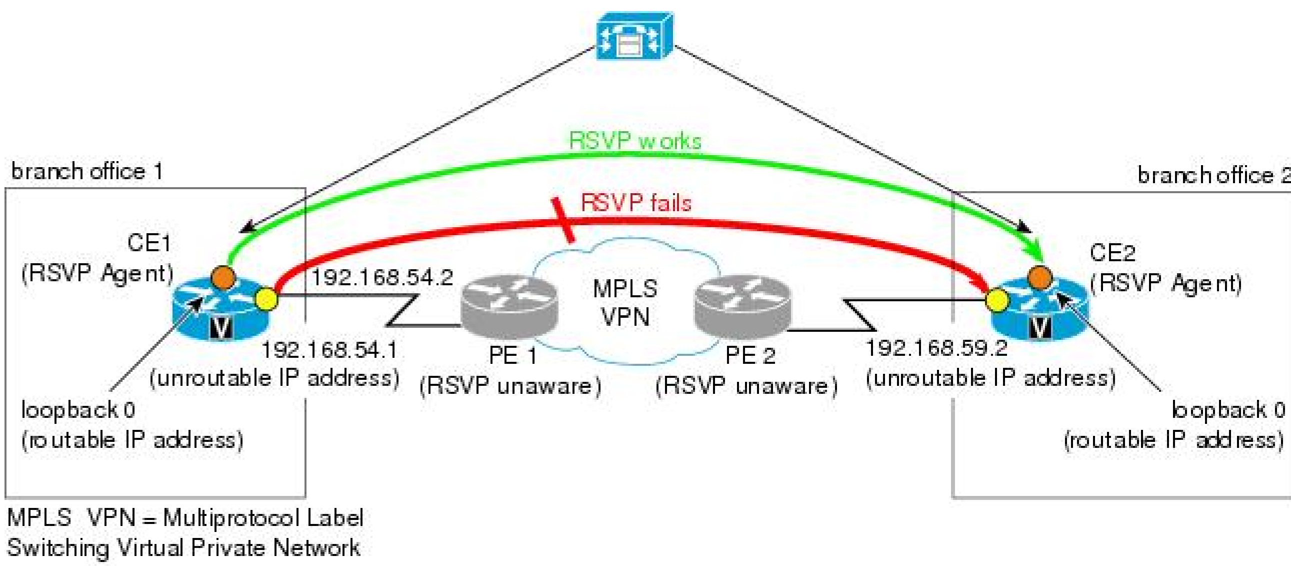
An RSVP PATH message contains a PHOP object that is rewritten at every RSVP hop. The object's purpose is to enable an RSVP device (R1) sending a PATH message to convey to the next RSVP device (R2) downstream that the previous RSVP hop is R1. R2 uses this information to forward the corresponding RESV message upstream hop-by-hop towards the sender.

The current behavior in Cisco software is that an RSVP device always sets the PHOP address to the IP address of the egress interface onto which the device transmits the PATH message.

There are situations where, although some IP addresses of R1 are reachable, the IP address of its egress interface is not reachable from a remote RSVP device R2. This results in the corresponding RESV message generated by R2 never reaching R1 and the reservation never being established.

The figure below shows a sample network in which the preceding scenario occurs and no reservation is established.

Figure 21 Sample PHOP Network with Unified Communcations Manager (CM)



In the figure above, when a call is made from branch office 1 to branch office 2, the RSVP Agent on customer edge (CE)1 tries to set up a session with CE2 and sends a PATH message. CE1 stamps its outgoing interface IP address (192.168.54.1), which is an unroutable IP address, in the PHOP object of the PATH message. This PATH message is tunneled across the service provider network and processed by CE2. CE2 records this IP address in the PHOP object of the received PATH message in the PSB (Path State Block).

CE2 has a receiver proxy configured for the destination address of the session. As a result, when CE2 replies back with a RESV message, CE2 tries to send the RESV message to the IP address that CE2 had recorded in its PSB. Because this IP address (192.168.54.1) is unroutable from CE2, the RESV message will fail.

Note Once you configure a source address on an interface, RSVP always uses the RSVP-overwritten address rather than the native interface address.

Benefits of RSVP-Previous Hop Overwrite

Flexibility and Customization

You can configure a CE to populate the PHOP object in a PATH message with an address that is reachable in the customer VPN. This enables the RESV message to find its way back towards the sender so that reservations can be established.

How to Configure RSVP-Previous Hop Overwrite

Configuring a Source Address or a Source Interface

Perform this task to configure a source address or a source interface.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
5. **ip rsvp source** {**address** *ip-address* | **interface** *type number*}
6. end

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**ip rsvp source** {**address** *ip-address* | **interface** *type number*} **Example:** `Device(config-if)# ip rsvp source` **address 10.1.3.13**	Configures an RSVP device to populate an address other than the native interface address in the PHOP address field of the hop object when forwarding a PATH message onto that interface. **Note** The source IP address that you configure should be a valid local IP address.
Step 6	end **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Verifying the PHOP Configuration

Note You can use the following **show** command in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp interface** [**detail**] [*interface-type interface-number*]
3. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** command in user EXEC mode.
Step 2	**show ip rsvp interface** [**detail**] [*interface-type interface-number*] **Example:** `Device# show ip rsvp interface detail ethernet0/1`	(Optional) Displays RSVP-related interface information. • The optional keywords and arguments display additional information.
Step 3	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

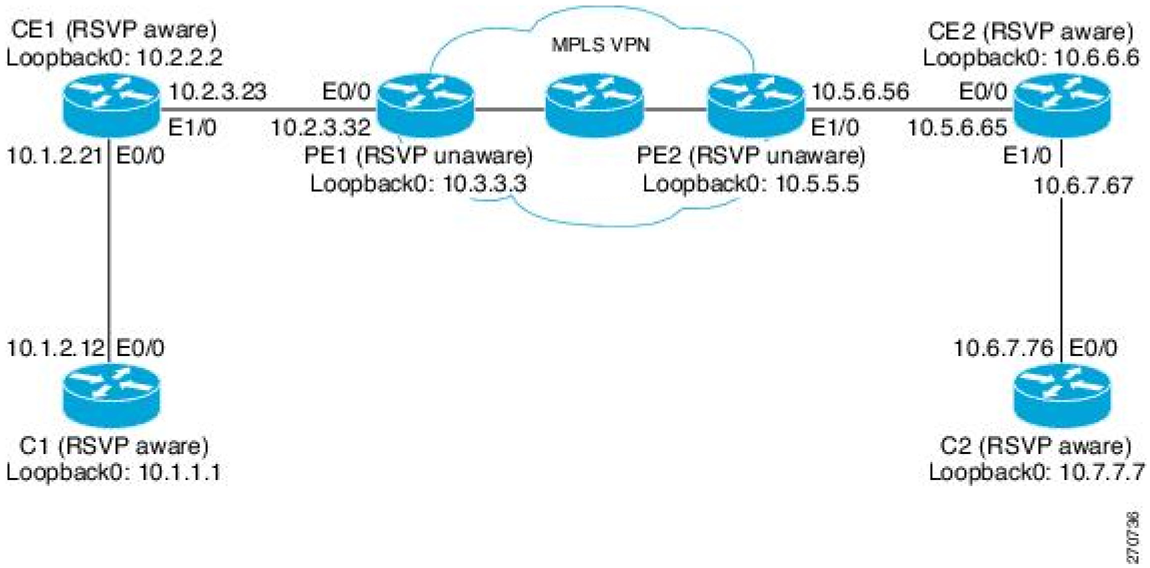
Configuration Examples for RSVP-Previous Hop Overwrite

- Examples Configuring RSVP-Previous Hop Overwrite, page 186
- Examples Verifying RSVP-Previous Hop Overwrite Configuration, page 187

Examples Configuring RSVP-Previous Hop Overwrite

The figure below shows a sample network in which PHOP is configured.

Figure 22 Sample PHOP Network



Configuring a Source Address on Device CE1 for the CE1-to-PE1 Interface

The following example configures a source address on the CE1-to-PE1 (Ethernet 1/0) interface in the figure above:

```
Device(CE1)# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Device(CE1)(config)# interface ethernet 1/0
Device(CE1)(config-if)# ip rsvp source address 10.2.2.2
 <---------------------
Device(CE1)(config-if)# end
```

Configuring a Source Address on Device CE2 for the CE2-to-PE2 Interface

The following example configures a source address on the CE2-to-PE2 (Ethernet 0/0) interface in the figure above:

```
Device(CE2)# configure terminal

Enter configuration commands, one per line.  End with CNTL/Z.
Device(CE2)(config)# interface ethernet 0/0
Device(CE2)(config-if)# ip rsvp source address 10.6.6.6
 <----------------------
Device(CE2)(config-if)# end
```

Creating a Listener Proxy on Device C2

The following example creates a listener proxy on Device C2 and requests that the receiver reply with a RESV message for the flow if the PATH message destination is 10.7.7.7 in the figure above:

```
Device(C2)# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Device(C2)(config)# ip rsvp listener 10.7.7.7 any any reply <--------------------
Device(C2)(config)# end
```

Creating a Session from Device C1 to Device C2

The following example creates an RSVP session from Device C1 to Device C2:

```
Device(C1)# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(C1)(config)# ip rsvp sender-host 10.7.7.7 10.1.1.1 UDP 100 200 1 1 <------------
Device(C1)(config)# end
```

Examples Verifying RSVP-Previous Hop Overwrite Configuration

Verifying the Source Address on Device CE1 for the CE1-to-PE1 Interface

The following example verifies the source address (10.2.2.2) configured on the CE1-to-PE1 (Ethernet 1/0) interface in the figure below:

```
Device(CE1)# show ip rsvp interface detail ethernet 1/0
 Et1/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 1K bits/sec
     Max. allowed (total): 100K bits/sec
     Max. allowed (per flow): 100K 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)
   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
     Ip address used in RSVP objects: 10.2.2.2 <-----------------------------------
   Authentication: disabled
     Key chain:   <none>
     Type:        md5
     Window size: 1
     Challenge:   disabled
   Hello Extension:
     State: Disabled
```

Verifying the Source Address on Device CE2 for the CE2-to-PE2 Interface

The following example verifies the source address configured on the CE2-to-PE2 (Ethernet 0/0) interface in the figure below:

```
Device(CE2)# show ip rsvp interface detail ethernet 0/0
 Et0/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 100K bits/sec
     Max. allowed (per flow): 100K 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)
```

```
  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
    Ip address used in RSVP objects: 10.6.6.6 <----------------------------------
  Authentication: disabled
    Key chain:   <none>
    Type:        md5
    Window size: 1
    Challenge:   disabled
  Hello Extension:
    State: Disabled
```

Verifying the Listener Proxy on Device C2

The following example verifies the listener proxy configured on Device C2 in the figure below:

```
Device(C2)# show ip rsvp listeners
To                Protocol   DPort   Description                  Action
10.7.7.7 <-------- any        any     RSVP Proxy                   reply
```

Verifying the Session from Device C1 to Device C2

The following example verifies that the session configured between Device C1 and Device C2 in the figure below is up:

```
Device(C1)# show ip rsvp reservation
To              From          Pro DPort Sport Next Hop      I/F      Fi Serv BPS
10.7.7.7       10.1.1.1       UDP 100   200   10.1.2.21     Et0/0    FF RATE 1K
```

Verifying the PHOP Address

The following example on Device CE2 verifies the source address configured on the CE1-to-PE1 interface in the figure below as the PHOP address:

```
Device(CE2)# show ip rsvp sender detail
PATH:
  Destination 10.7.7.7, Protocol_Id 17, Don't Police , DstPort 100
  Sender address: 10.1.1.1, port: 200
  Path refreshes:
    arriving: from PHOP 10.2.2.2 on Et0/0 every 30000 msecs <-------------------
  Traffic params - Rate: 1K bits/sec, Max. burst: 1K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  Path ID handle: CA000406.
  Incoming policy: Accepted. Policy source(s): Default
  Status:
  Output on Ethernet1/0. Policy status: Forwarding. Handle: 0E000402
    Policy source(s): Default
```

Verifying the Next-Hop Address

The following example on Device CE1 verifies the source address configured on the CE2-to-PE2 interface in the figure below as the next-hop address:

```
Device(CE1)# show ip rsvp reservation detail
RSVP Reservation. Destination is 10.7.7.7, Source is 10.1.1.1,
  Protocol is UDP, Destination port is 100, Source port is 200
  Next Hop: 10.6.6.6 on Ethernet1/0 <-------------------------------------------
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Resv ID handle: 03000400.
  Created: 07:01:40 IST Tue Mar 25 2008
  Average Bitrate is 1K bits/sec, Maximum Burst is 1K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
```

```
Status:
Policy: Forwarding. Policy source(s): Default
```

Additional References

The following sections provide references related to the RSVP--Previous Hop Overwrite feature.

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification*
RFC 2209	*Resource ReSerVation Protocol (RSVP)--Version 1 Message Processing Rules*
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*

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 RSVP-Previous Hop Overwrite

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

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

Table 8 Feature Information for RSVP--Previous Hop Overwrite

Feature Name	Releases	Feature Information
RSVP--Previous Hop Overwrite	12.4(20)T 15.0(1)SY	The RSVP--Previous Hop Overwrite feature allows you to configure a Resource Reservation Protocol (RSVP) device, on a per interface basis, to populate an address other than the native interface address in the previous hop (PHOP) address field of the PHOP object when forwarding a PATH message onto that interface. You can configure the actual address for the device to use, or an interface, including a loopback, from which to borrow the address. The following commands were introduced or modified: **debug ip rsvp, ip rsvp source, show ip rsvp interface.**

Glossary

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

RSVP Agent --Implements a Resource Reservation Protocol (RSVP) agent on Cisco IOS voice gateways that support Unified CM.

Unified Communcations Manager (CM)--The software-based, call-processing component of the Cisco IP telephony solution. The software extends enterprise telephony features and functions to packet telephony network devices such as IP phones, media processing devices, voice-over-IP (VoIP) gateways, and multimedia applications.

RSVP Application ID Support

The RSVP Application ID Support feature introduces application-specific reservations, which enhance the granularity for local policy match criteria so that you can manage quality of service (QoS) on the basis of application type.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Application ID Support

You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for RSVP Application ID Support

- RSVP policies apply only to PATH, RESV, PATHERROR, and RESVERROR messages.
- Merging of global and interface-based local policies is not supported; therefore, you cannot match on multiple policies.

Information About RSVP Application ID Support

Feature Overview of RSVP Application ID Support

How RSVP Functions

Multiple applications such as voice and video need RSVP support. RSVP admits requests until the bandwidth limit is reached. RSVP does not differentiate between the requests and is not aware of the type of application for which the bandwidth is requested.

As a result, RSVP can exhaust the allowed bandwidth by admitting requests that represent just one type of application, causing all subsequent requests to be rejected because of unavailable bandwidth. For example, a few video calls could prevent all or most of the voice calls from being admitted because the video calls require a large amount of bandwidth and not enough bandwidth remains to accommodate the voice calls. With this limitation, you would probably not deploy RSVP for multiple applications especially if voice happens to be one of the applications for which RSVP is required.

The solution is to allow configuration of separate bandwidth limits for individual applications or classes of traffic. Limiting bandwidth per application requires configuring a bandwidth limit per application and having each reservation flag the application to which the reservation belongs so that it can be admitted against the appropriate bandwidth limit.

Application and Sub Application Identity Policy Element for Use with RSVP (IETF RFC 2872) allows for creation of separate bandwidth reservation pools. For example, an RSVP reservation pool can be created for voice traffic, and a separate RSVP reservation pool can be created for video traffic. This prevents video traffic from overwhelming voice traffic.

Note Before this feature, you could create access control lists (ACLs) that match on the differentiated services code points (DSCPs) of the IP header in an RSVP message. However, multiple applications could use the same DSCP; therefore, you could not uniquely identify applications in order to define separate policies for them.

Sample Solution

The figure below shows a sample solution in which application ID is used. In this example, bandwidth is allocated between the voice and video sessions that are being created by Cisco CallManager (CCM). Video requires much more bandwidth than voice, and if you do not separate the reservations, the video traffic could overwhelm the voice traffic.

CCM has been enhanced to use the RSVP Application ID Support feature. In this example, when CCM makes the RSVP reservation, CCM has the ability to specify whether the reservation should be made against a video RSVP bandwidth pool or a voice RSVP bandwidth pool. If there is not enough bandwidth remaining in the requested pool, even though there is enough bandwidth in the total RSVP allocation, RSVP signals CCM that there is a problem with the reservation. The figure shows some of the signaling and data traffic that is sent during the session setup.

IMAGE MISSING; embedded not referenced

In this scenario, the IP phones and IP video devices do not directly support RSVP. In order to allow RSVP to reserve the bandwidth for these devices, the RSVP agent component in the Cisco device creates the reservation. During the setup of the voice or video session, CCM communicates with the RSVP agent and sends the parameters to reserve the necessary bandwidth.

When you want to make a voice or video call, the device signals CCM. CCM signals the RSVP agent, specifying the RSVP application ID that corresponds to the type of call, which is voice or video in this example. The RSVP agents establish the RSVP reservation across the network and tell CCM that the reservation has been made. CCM then completes the session establishment, and the Real-Time Transport Protocol (RTP) traffic streams flow between the phones (or video devices). If the RSVP agents are unable to create the bandwidth reservations for the requested application ID, they communicate that information back to CCM, which signals this information back to you.

Global and Per-Interface RSVP Policies

You can configure RSVP policies globally and on a per-interface basis. You can also configure multiple global policies and multiple policies per interface.

Global RSVP policies restrict how much RSVP bandwidth a device uses regardless of the number of interfaces. You should configure a global policy if your device has CPU restrictions, one interface, or multiple interfaces that do not require different bandwidth limits.

Per-interface RSVP policies allow you to configure separate bandwidth pools with varying limits so that no one application, such as video, can consume all the RSVP bandwidth on a specified interface at the expense of other applications, such as voice, which would be dropped. You should configure a per-interface policy when you need greater control of the available bandwidth.

How RSVP Policies Are Applied

RSVP searches for policies whenever an RSVP message is processed. The policy tells RSVP if any special handling is required for that message.

If your network configuration has global and per-interface RSVP policies, the per-interface policies are applied first meaning that RSVP looks for policy-match criteria in the order in which the policies were configured. RSVP searches for policy-match criteria in the following order:

- Nondefault interface policies
- Default interface policy
- Nondefault global policies
- Global default policy

If RSVP finds no policy-match criteria, it accepts all incoming messages. To change this decision from accept to reject, issue the **ip rsvp policy default-reject** command.

Preemption

Preemption happens when one reservation receives priority over another because there is insufficient bandwidth in an RSVP pool. There are two types of RSVP bandwidth pools: local policy pools and interface pools. Local policies can be global or interface-specific. RSVP performs admission control against these pools when a RESV message arrives.

If an incoming reservation request matches an RSVP local policy that has an RSVP bandwidth limit (as configured with the **maximum bandwidth group** submode command) and that limit has been reached, RSVP tries to preempt other lower-priority reservations admitted by that policy. When there are too few of these lower-priority reservations, RSVP rejects the incoming reservation request. Then RSVP looks at the interface bandwidth pool that you configured by using the **ip rsvp bandwidth** command. If that bandwidth limit has been reached, RSVP tries to preempt other lower-priority reservations on that interface to accommodate the new reservation request. At this point, RSVP does not consider which local policies admitted the reservations. When not enough bandwidth on that interface pool can be preempted, RSVP rejects the new reservation even though the new reservation was able to obtain bandwidth from the local policy pool.

Preemption can also happen when you manually reconfigure an RSVP bandwidth pool of any type to a lower value such that the existing reservations using that pool no longer fit in the pool.

- How Preemption Priorities Are Assigned and Signaled, page 196
- Controling Preemption, page 196

How Preemption Priorities Are Assigned and Signaled

If a received RSVP PATH or RESV message contains preemption priorities (signaled with an IETF RFC 3181 preemption priority policy element inside an IETF RFC 2750 POLICY_DATA object) and the priorities are higher than those contained in the matching local policy (if any), the offending message is rejected and a PATHERROR or RESVERROR message is sent in response. If the priorities are approved by the local policy, they are stored with the RSVP state in the device and forwarded to its neighbors.

If a received RSVP PATH or RESV message does not contain preemption priorities (as previously described) and you issued a global **ip rsvp policy preempt** command, and the message matches a local policy that contains a **preempt-priority** command, a POLICY_DATA object with a preemption priority element that contains the local policy's priorities is added to the message as part of the policy decision. These priorities are then stored with the RSVP state in the device and forwarded to neighbors.

Controling Preemption

The **ip rsvp policy preempt** command controls whether or not a device preempts any reservations when required. When you issue this command, a RESV message that subsequently arrives on an interface can preempt the bandwidth of one or more reservations on that interface if the assigned setup priority of the new reservation is higher than the assigned hold priorities of the installed reservations.

Benefits of RSVP Application ID Support

The RSVP Application ID Support feature provides the following benefits:

- Allows RSVP to identify applications uniquely and to separate bandwidth pools to be created for different applications so that one application cannot consume all the available bandwidth, thereby forcing others to be dropped.
- Integrates with the RSVP agent and CCM to provide a solution for call admission control (CAC) and QoS for Voice over IP (VoIP) and video conferencing applications in networks with multitiered, meshed topologies using signaling protocols such as SCCP to ensure that a single application does not overwhelm the available reserved bandwidth.
- Functions with any endpoint that complies with RFC 2872 or RFC 2205.

How to Configure RSVP Application ID Support

You can configure application IDs and local policies to use with RSVP-aware software programs such as CCM or to use with non-RSVP-aware applications such as static PATH and RESV messages.

- Configuring RSVP Application IDs and Local Policies for RSVP-Aware Software Programs, page 197
- Configuring RSVP Application IDs with Static Senders and Receivers for Non-RSVP-Aware Software Programs, page 202
- Verifying the RSVP Application ID Support Configuration, page 207

Configuring RSVP Application IDs and Local Policies for RSVP-Aware Software Programs

This section contains the following procedures:

Note The following two local policy configuration procedures are optional; however, you must choose one or both.

- Configuring an Application ID, page 197
- Configuring a Local Policy Globally, page 198
- Configuring a Local Policy on an Interface, page 200

Configuring an Application ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy identity** *alias* **policy-locator** *locator*
4. Repeat Step 3 as needed to configure additional application IDs.
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy identity** *alias* **policy-locator** *locator* **Example:** `Device(config)# ip rsvp policy identity rsvp-voice policy-locator APP=Voice`	Defines RSVP application IDs to use as match criteria for local policies. • Enter a value for the *alias*argument, which is a string used within the device to reference the identity in RSVP configuration commands and show displays. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the " " or ? characters as part of the alias or locator string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other devices. • Enter a value for the *locator* argument, which is a string that is signaled in RSVP messages and contains application IDs usually in X.500 Distinguished Name (DN) format. This can also be a regular expression. For more information on regular expressions, see the Configuring an Application ID, page 197 section.
Step 4	Repeat Step 3 as needed to configure additional application IDs.	Defines additional application IDs.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

- What to Do Next, page 198

What to Do Next

Configure a local policy globally, on an interface, or both.

Configuring a Local Policy Globally

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy local** {**ac l** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]}
4. Repeat Step 3 as needed to configure additional local policies.
5. {**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
6. Repeat Step 5 as needed to configure additional submode commands.
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy local** {**ac l** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config)# ip rsvp policy local` **identity rsvp-voice**	Creates a local policy to determine how RSVP resources are used in a network and enters local policy configuration mode. • Enter the **identity** *alias1* keyword and argument combination to specify an application ID alias.
Step 4	Repeat Step 3 as needed to configure additional local policies.	(Optional) Configures additional local policies.
Step 5	{**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Device(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local** command for more detailed information on submode commands.

	Command or Action	Purpose
Step 6	Repeat Step 5 as needed to configure additional submode commands.	(Optional) Configures additional submode commands.
Step 7	**end** **Example:** `Device(config-rsvp-policy-local)# end`	Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring a Local Policy on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. Repeat Step 3 as needed to configure additional interfaces.
5. ip rsvp bandwidth [*interface-kbps*] [*single-flow-kbps*]
6. Repeat Step 5 as needed to configure bandwidth for additional interfaces.
7. **ip rsvp policy local** {**ac l** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]}
8. Repeat Step 7 as needed to configure additional local policies.
9. {**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
10. Repeat Step 9 as needed to configure additional submode commands.
11. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and number and enters interface configuration mode.
Step 4	Repeat Step 3 as needed to configure additional interfaces.	(Optional) Configures additional interfaces.
Step 5	ip rsvp bandwidth [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 500 500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 1000,000.
Step 6	Repeat Step 5 as needed to configure bandwidth for additional interfaces.	(Optional) Configures bandwidth for additional interfaces.
Step 7	**ip rsvp policy local** {**ac l** *acl1* [*acl2...acl8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config-if)# ip rsvp policy local` **identity rsvp-voice**	Creates a local policy to determine how RSVP resources are used in a network. • Enter the **identity** *alias1* keyword argument combination to specify an application ID alias.
Step 8	Repeat Step 7 as needed to configure additional local policies.	(Optional) Configures additional local policies.
Step 9	{**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*] | **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Device(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating and enters local policy configuration mode. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local**command for more detailed information on submode commands.
Step 10	Repeat Step 9 as needed to configure additional submode commands.	(Optional) Configures additional submode commands.
Step 11	**end** **Example:** `Device(config-rsvp-policy-local)# end`	Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring RSVP Application IDs with Static Senders and Receivers for Non-RSVP-Aware Software Programs

Configuring an Application ID

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy identity** *alias* **policy-locator** *locator*
4. Repeat step 3 to configure additional application IDs.
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp policy identity** *alias* **policy-locator** *locator* **Example:** `Device(config)# ip rsvp policy identity rsvp-voice policy-locator "APP=Voice"`	Defines RSVP application IDs to use as match criteria for local policies. • Enter a value for the *alias*argument, which is a string used within the device to reference the identity in RSVP configuration commands and show displays. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the " " or ? characters as part of the alias or locator string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other devices. • Enter a value for the *locator* argument, which is a string that is signaled in RSVP messages and contains application IDs usually in X.500 Distinguished Name (DN) format. **Note** Repeat this step as needed to configure additional application IDs.
Step 4	Repeat step 3 to configure additional application IDs.	Configures additional application IDs.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Sender with an Application ID

Perform this task to configure a static RSVP sender with an application ID to make the device proxy an RSVP PATH message containing an application ID on behalf of an RSVP-unaware sender application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] **Example:** `Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 identity rsvp-voice`	Enables a device to simulate a host generating RSVP PATH messages. • The optional **identity** *alias* keyword and argument combination specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the " " or ? characters as part of the alias string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other devices.
Step 4	**end** **Example:** `Device(config)#` **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Receiver with an Application ID

Perform this task to configure a static RSVP receiver with an application ID to make the device proxy an RSVP RESV message containing an application ID on behalf of an RSVP-unaware receiver application.

Note You can also configure a static listener to use with an application ID. If an incoming PATH message contains an application ID and/or a preemption priority value, the listener includes them in the RESV message sent in reply. See the Feature Information for RSVP Application ID Support, page 214for more information.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port*{**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
   -
   - **ip rsvp reservation** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*[**identity** *alias*]
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose

Step 3	Do one of the following:
	• **ip rsvp reservation-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port*{**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
	•
	• **ip rsvp reservation** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port next-hop-ip-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*[**identity** *alias*]
	
	Example:
	`Device(config)# ip rsvp reservation-host 10.1.1.1 10.30.1.4 udp 20 30 se load 100 60 identity rsvp-voice`
	
	Example:
	
	Example:
	`Device(config)# ip rsvp reservation 10.1.1.1 0.0.0.0 udp 20 0 172.16.4.1 Ethernet1 wf rate 350 65 identity xyz`
		Enables a device to simulate a host generating RSVP RESV messages.
		• The optional **identity** *alias* keyword and argument combination specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E).
		Note If you use the " " or ? characters as part of the alias string itself, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other devices.
		Note Use the **ip rsvp reservation-host** command if the device is the destination or the **ip rsvp reservation** command to have the device proxy on behalf of a downstream host.

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

Verifying the RSVP Application ID Support Configuration

SUMMARY STEPS

1. **enable**
2. **show ip rsvp host** {**senders** | **receivers**} [*group-name* | *group-address*]
3. **show ip rsvp policy identity** [*regular-expression*]
4. **show ip rsvp policy local** [**detail**] [**interface** *name*] [**default**| **acl** *acl*| **origin-as** *as* | **identity** *alias*]
5. **show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]]
6. **show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]]
7. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the commands in user EXEC mode.
Step 2	**show ip rsvp host** {**senders** | **receivers**} [*group-name* | *group-address*] **Example:** `Device# show ip rsvp host senders`	Displays specific information for an RSVP host. **Note** Use this command only on devices from which PATH and RESV messages originate.
Step 3	**show ip rsvp policy identity** [*regular-expression*] **Example:** `Device# show ip rsvp policy identity voice100`	Displays selected RSVP identities in a device configuration. • The optional *regular-expression* argument allows pattern matching on the alias strings of the RSVP identities to be displayed. **Note** For more information on regular expressions, see the Verifying the RSVP Application ID Support Configuration, page 207.
Step 4	**show ip rsvp policy local** [**detail**] [**interface** *name*] [**default**| **acl** *acl*| **origin-as** *as* | **identity** *alias*] **Example:** `Device# show ip rsvp policy local identity voice100`	Displays the local policies currently configured. • The optional **detail** keyword and the optional **interface** *name* keyword and argument combination can be used with any of the match criteria.

	Command or Action	Purpose
Step 5	**show ip rsvp reservation** [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Device# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional **detail** keyword displays additional output with information about where the policy originated as well as which application ID was signaled in the RESV message. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**show ip rsvp sender** [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Device# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output with information that includes which application ID was signaled in the PATH message. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0 S and 12.2 S only.
Step 7	**exit** **Example:** `Device# exit`	Exits privileged EXEC mode and returns to user EXEC mode.

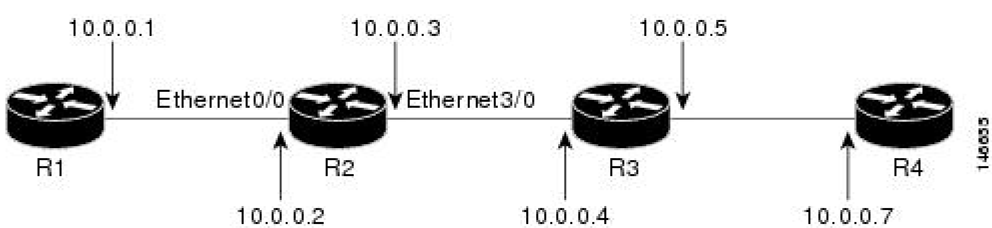
Configuration Examples for RSVP Application ID Support

- Example Configuring RSVP Application ID Support, page 208
- Example Verifying RSVP Application ID Support, page 210

Example Configuring RSVP Application ID Support

The four-device network in the figure below contains the following configurations:

Figure 23 ***Sample Network with Application Identities and Local Policies***



- Configuring a Proxy Receiver on R4, page 209
- Configuring an Application ID and a Global Local Policy on R3, page 209

- Configuring an Application ID and Separate Bandwidth Pools on R2 for Per-Interface Local Policies, page 209
- Configuring an Application ID and a Static Reservation from R1 to R4, page 210

Configuring a Proxy Receiver on R4

The following example configures R4 with a proxy receiver to create an RESV message to match the PATH message for the destination 10.0.0.7:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp listener 10.0.0.7 any any reply

Device(config)# end
```

Configuring an Application ID and a Global Local Policy on R3

The following example configures R3 with an application ID called video and a global local policy in which all RSVP messages are being accepted and forwarded:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp policy identity video policy-locator video
Device(config)# ip rsvp policy local identity video
Device(config-rsvp-policy-local)# forward all
Device(config-rsvp-policy-local)# end
```

Configuring an Application ID and Separate Bandwidth Pools on R2 for Per-Interface Local Policies

The following example configures R2 with an application ID called video, which is a wildcard regular expression to match any application ID that contains the substring video:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp policy identity video policy-locator .*Video.*
Device(config-rsvp-id)# end
```

The following example configures R2 with a local policy on ingress Ethernet interface 0/0:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface Ethernet0/0
Device(config-if)# ip address 10.0.0.2 255.0.0.0
Device(config-if)# no cdp enable
Device(config-if)# ip rsvp bandwidth 200
Device(config-if)# ip rsvp policy local identity video
Device(config-rsvp-policy-local)# maximum senders 10
Device(config-rsvp-policy-local)# maximum bandwidth group 100
Device(config-rsvp-policy-local)# maximum bandwidth single 10
Device(config-rsvp-policy-local)# forward all
Device(config-rsvp-policy-local)# end
```

The following example configures R2 with a local policy on egress Ethernet interface 3/0:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface Ethernet3/0
Device(config-if)# ip address 10.0.0.3 255.0.0.0
Device(config-if)# no cdp enable
Device(config-if)# ip rsvp bandwidth 200
```

```
Device(config-if)# ip rsvp policy local identity video
Device(config-rsvp-policy-local)# maximum senders 10
Device(config-rsvp-policy-local)# maximum bandwidth group 100
Device(config-rsvp-policy-local)# maximum bandwidth single 10
Device(config-rsvp-policy-local)# forward all
Device(config-rsvp-policy-local)# end
```

Note PATH messages arrive on ingress Ethernet interface 0/0 and RESV messages arrive on egress Ethernet interface 3/0.

Configuring an Application ID and a Static Reservation from R1 to R4

The following example configures R1 with an application ID called video and initiates a host generating a PATH message with that application ID:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp policy identity video policy-locator "GUID=www.cisco.com,
APP=Video, VER=1.0"
Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 identity video
Device(config)# end
```

Example Verifying RSVP Application ID Support

- Verifying the Application ID and the Global Local Policy on R3, page 210
- Verifying the Application ID and the Per-Interface Local Policies on R2, page 211
- Verifying the Application ID and the Reservation on R1, page 212

Verifying the Application ID and the Global Local Policy on R3

The following example verifies that a global local policy has been configured on R3 with an application ID called Video:

```
Device# show ip rsvp policy local detail
Global:
  Policy for ID(s): Video
    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           23000404.
                           Accept               Forward
    Path:                  Yes                  Yes
    Resv:                  Yes                  Yes
    PathError:             Yes                  Yes
    ResvError:             Yes                  Yes
                           Setup Priority       Hold Priority
    TE:                    N/A                  N/A
    Non-TE:                N/A                  N/A
                           Current              Limit
    Senders:               1                    N/A
    Receivers:             1                    N/A
    Conversations:         1                    N/A
    Group bandwidth (bps): 10K                  N/A
    Per-flow b/w (bps):    N/A                  N/A


Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Verifying the Application ID and the Per-Interface Local Policies on R2

The following example verifies that an application ID called Video has been created on R2:

```
Device# show ip rsvp policy identity
Alias: Video
  Type:    Application ID
  Locator: .*Video.*
```

The following example verifies that per-interface local policies have been created on Ethernet interface 0/0 and Ethernet interface 3/0 on R2:

```
Device# show ip rsvp policy local detail
Ethernet0/0:
  Policy for ID(s): Video
    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           26000404.
                           Accept                Forward
    Path:                  Yes                   Yes
    Resv:                  Yes                   Yes
    PathError:             Yes                   Yes
    ResvError:             Yes                   Yes
                           Setup Priority        Hold Priority
    TE:                    N/A                   N/A
    Non-TE:                N/A                   N/A
                           Current               Limit
    Senders:               1                     10
    Receivers:             0                     N/A
    Conversations:         0                     N/A
    Group bandwidth (bps): 0                     100K
    Per-flow b/w (bps):    N/A                   10K

Ethernet3/0:
  Policy for ID(s): Video
    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           5A00040A.
                           Accept                Forward
    Path:                  Yes                   Yes
    Resv:                  Yes                   Yes
    PathError:             Yes                   Yes
    ResvError:             Yes                   Yes
                           Setup Priority        Hold Priority
    TE:                    N/A                   N/A
    Non-TE:                N/A                   N/A
                           Current               Limit
    Senders:               0                     10
    Receivers:             1                     N/A
    Conversations:         1                     N/A
    Group bandwidth (bps): 10K                   100K
    Per-flow b/w (bps):    N/A                   10K
Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Note Notice in the above display that the ingress interface has only its senders counter incremented because the PATH message is checked there. However, the egress interface has its receivers, conversations, and group bandwidth counters incremented because the reservation is checked on the incoming interface, which is the egress interface on R2.

Verifying the Application ID and the Reservation on R1

The following example verifies that a PATH message containing the application ID called Video has been created on R1:

```
Device# show ip rsvp sender detail
PATH Session address: 10.0.0.7, port: 1. Protocol: UDP
  Sender address: 10.0.0.1, port: 1
     Inbound from: 10.0.0.1
 on interface:
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
                 Min Policed Unit: 0 bytes, Max Pkt Size 4294967295 bytes
  Path ID handle: 02000402.
  Incoming policy: Accepted. Policy source(s): Default
    Application ID: 'GUID=www.cisco.com, APP=Video, VER=1.0'
  Status: Proxied
  Output on Ethernet0/0. Policy status: Forwarding. Handle: 01000403
    Policy source(s): Default
```

Note You can issue the **debug ip rsvp dump path** and the **debug ip rsvp dump resv** commands to get more information about a sender and the application ID that it is using.

The following example verifies that a reservation with the application ID called Video has been created on R1:

```
Device# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.7, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 1, Source port is 1
  Next Hop is 10.0.0.2, Interface is Ethernet0/0
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Resv ID handle: 01000405.
  Created: 10:07:35 EST Thu Jan 12 2006
  Average Bitrate is 10K bits/sec, Maximum Burst is 10K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Status:
  Policy: Forwarding. Policy source(s): Default
    Application ID: 'GUID=www.cisco.com, APP=Video, VER=1.0'
```

Additional References

The following sections provide references related to the RSVP Application ID Support feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS configuration tasks related to RSVP	"Configuring RSVP" module
Cisco United Communications Manager (CallManager) and related features	"Overview of Cisco Unified Communications Manager and Cisco IOS Interoperability" module

Related Topic	Document Title
Regular expressions	"Using the Cisco IOS Command-Line Interface" module
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)
RFC 2872	Application and Sub Application Identity Policy Element for Use with RSVP
RFC 3181	Signaled Preemption Priority Policy Element
RFC 3182	Identity Representation for RSVP

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 RSVP Application ID Support

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 9 Feature Information for RSVP Application ID Support

Feature Name	Releases	Feature Information
RSVP Application ID Support	12.4(6)T, 12.2(33)SRB	The RSVP Application ID Support feature introduces application-specific reservations, which enhance the granularity for local policy-match criteria so that you can manage quality of service (QoS) on the basis of application type.

Glossary

ACL-- access control list. An ACL consists of individual filtering rules grouped together in a single list. It is generally used to provide security filtering, although it may be used to provide a generic packet classification facility.

admission control --The process in which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

application identity (ID) --A string that can be inserted in a policy element in a POLICY_DATA object of an RSVP message to identify the application and associate it with the RSVP reservation request, thus allowing devices along the path to make appropriate decisions based on the application information.

autonomous system --A collection of networks that share the same routing protocol and that are under the same system administration.

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term also is used to describe the rated throughput capacity of a given network medium or protocol.

CCM --Cisco CallManager. The software-based, call-processing component of the Cisco IP telephony solution. The software extends enterprise telephony features and functions to packet telephony network devices such as IP phones, media processing devices, Voice-over-IP (VoIP) gateways, and multimedia applications.

DSCP --differentiated services code point. The six most significant bits of the 1-byte IP type of service (ToS) field. The per-hop behavior represented by a particular DSCP value is configurable. DSCP values range between 0 and 63.

policy --Any defined rule that determines the use of resources within the network. A policy can be based on a user, a device, a subnetwork, a network, or an application.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

RSVP agent --Implements a Resource Reservation Protocol (RSVP) agent on Cisco IOS voice gateways that support Cisco CallManager 5.0.

RTP --Real-Time Transport Protocol. An Internet protocol for transmitting real-time data such as voice and video.

router --A network layer device that uses one or more metrics to determine the optimal path along which network traffic should be forwarded. Routers forward packets from one network to another on the basis of network layer information.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

RSVP Fast Local Repair

The RSVP Fast Local Repair feature provides quick adaptation to routing changes occurring in global as well as VRF routing domains, without the overhead of the refresh period to guarantee the quality of service (QoS) for data flows. With fast local repair (FLR), Resource Reservation Protocol (RSVP) speeds up its response to routing changes from 30 seconds to a few seconds.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP FLR

You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for RSVP FLR

- RSVP FLR applies only when RSVP is used to set up resource reservations for IPv4 unicast flows; IPv4 multicast flows are not supported.

- RSVP FLR does not apply to traffic engineering (TE) tunnels and, therefore, does not affect TE sessions.
- RSVP FLR does not support message bundling.

Information About RSVP FLR

Feature Overview of RSVP FLR

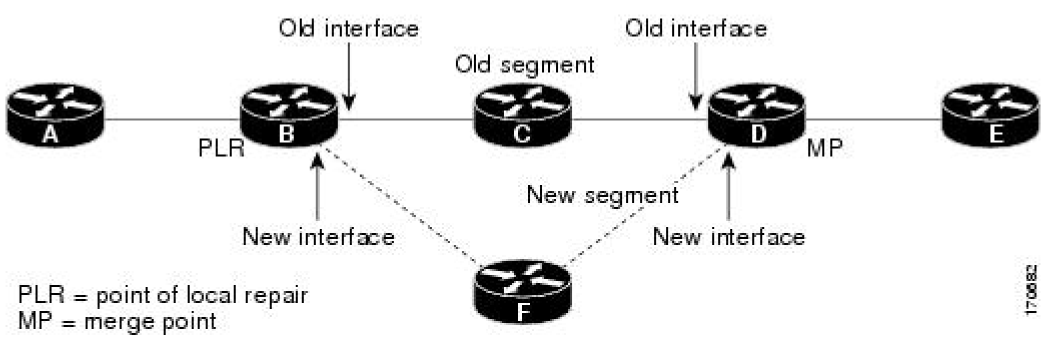
RSVP FLR provides for dynamic adaptation when routing changes occur in global or VRF routing domains. When a route changes, the next PATH and RESV message refreshes establish path and reservation states along the new route. Depending on the configured refresh interval, this reroute happens in tens of seconds. However, during this time, the QoS of flows is not guaranteed because congestion may occur while data packets travel over links where reservations are not yet in place.

In order to provide faster adaptation to routing changes, without the overhead of a refresh period, RSVP registers with the routing information base (RIB) and receives notifications when routes change, thereby triggering state refreshes for the affected destinations. These triggered refreshes use the new route information and, as a result, install reservations over the new path.

When routes change, RSVP has to reroute all affected paths and reservations. Without FLR, the reroute happens when refresh timers expire for the path states. With real time applications such as VoIP and VoD, the requirement changes and the reroute must happen quickly, within three seconds from the triggering event such as link down or link up.

The figure below illustrates the FLR process.

Figure 24 Overview of RSVP FLR



Initial RSVP states are installed for an IPv4 unicast flow over devices A, B, C, D, and E. Device A is the source or headend, while device E is the destination or tailend. The data packets are destined to an address of device E. Assume that a route change occurs, and the new path taken by the data packets is from device A to device B to device F to device D to device E; therefore, the old and new paths differ on the segments between device B and D. The device B to device C to device D segment is the old segment, while the device B to device F to device D segment is the new segment.

A route may change because of a link or node failure, or if a better path becomes available.

RSVP at device B detects that the route change affects the RSVP flow and initiates the FLR procedure. The node that initiates an FLR repair procedure, device B in the figure above, is the point of local repair (PLR).

The node where the new and old segments meet, device D in the figure above, is the merge point (MP). The interfaces at the PLR and the MP that are part of the old segment are the old interfaces, while the interfaces that are part of the new segment are the new interfaces.

If a route has changed because of a failure, the PLR may not be the node that detects the failure. For example, it is possible that the link from device C to device D fails, and although device C detects the failure, the route change at device B is the trigger for the FLR procedure. device C, in this case, is also referred to as the node that detects the failure.

The support for FLR in VRF domains means that RSVP can get a route change notification, even if there is a route change in any VRF domains, as RSVP FLR was previously supported only in the global routing domain.

Benefits of RSVP FLR

Faster Response Time to Routing Changes

FLR reduces the time that it takes for RSVP to determine that a physical link has gone down and that the data packets have been rerouted. Without FLR, RSVP may not recognize the link failure for 30 seconds when all of the sessions are impacted by having too much traffic for the available bandwidth. With FLR, this time can be significantly reduced to a few seconds.

After detecting the failure, RSVP recomputes the admission control across the new link. If the rerouted traffic fits on the new link, RSVP reserves the bandwidth and guarantees the QoS of the new traffic.

If admission control fails on the new route, RSVP does not explicate tear down the flow, but instead sends a RESVERROR message towards the receiver. If a proxy receiver is running, then RSVP sends a PATHERROR message towards the headend, in response to the RESVERROR message, indicating the admission failure. In both cases, with and without a proxy receiver, the application tears down the failed session either at the headend or at the final destination.

Until this happens, the data packets belonging to this session still flow over the rerouted segment although admission has failed and QoS is affected.

The support of FLR in VRF domains means that if there is a route change in any routing domain, RSVP can use FLR to adapt to the routing change, as RSVP FLR was previously supported only in the global routing domain.

How to Configure RSVP FLR

You can configure the RSVP FLR parameters in any order that you want.

- Configuring the RSVP FLR Wait Time, page 220
- Configuring the RSVP FLR Repair Rate, page 221
- Configuring the RSVP FLR Notifications, page 222
- Verifying the RSVP FLR Configuration, page 223

Configuring the RSVP FLR Wait Time

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]]
5. **ip rsvp signalling fast-local-repair wait-time** interval
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]] **Example:** `Device(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool**and *sub-pool-kbps*keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.

	Command or Action	Purpose
Step 5	**ip rsvp signalling fast-local-repair wait-time** interval **Example:** `Device(config-if)#` **ip rsvp signalling fast-local-repair wait-time 100**	Configures the delay that RSVP uses before starting an FLR procedure. • Values for the *interval* argument are 0 to 5000 milliseconds (ms); the default is 0.
Step 6	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring the RSVP FLR Repair Rate

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling fast-local-repair rate** rate
4. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp signalling fast-local-repair rate** rate **Example:** `Device(config)#` **ip rsvp signalling fast-local-repair rate 100**	Configures the repair rate that RSVP uses for an FLR procedure. • Values for the *rate*argument are 1 to 2500 messages per second; the default is 400. **Note** See the **ip rsvp signalling fast-local-repair rate**command for more information.
Step 4	**exit** **Example:** `Device(config)#` **exit**	(Optional) Returns to privileged EXEC mode.

Configuring the RSVP FLR Notifications

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp signalling fast-local-repair notifications** number
4. **exit**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ip rsvp signalling fast-local-repair notifications** number **Example:** `Device(config)#` **ip rsvp signalling fast-local-repair notifications 100**	Configures the number of path state blocks (PSBs) that RSVP processes before it suspends. • Values for the *number*argument are 10 to 10000; the default is 1000.
Step 4	**exit** **Example:** `Device(config)#` **exit**	(Optional) Returns to privileged EXEC mode.

Verifying the RSVP FLR Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp signalling fast-local-repair** [**statistics** [**detail**]]
3. **show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] **[interface-type interface-number**]
4. **show ip rsvp**
5. **show ip rsvp sender** [**vrf**{***** | *vrf-name*}][**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.

	Command or Action	Purpose
Step 2	**show ip rsvp signalling fast-local-repair** [**statistics** [**detail**]] **Example:** `Device# show ip rsvp signalling fast-local-repair statistics detail`	Displays FLR-specific information that RSVP maintains. • The optional **statistics** and **detail** keywords display additional information about the FLR parameters.
Step 3	**show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] **[interface-type interface-number**] **Example:** `Device# show ip rsvp interface` **ethernet 1/0**	Displays RSVP-related information. • The optional **detail** keyword displays additional information including FLR parameters.
Step 4	**show ip rsvp** **Example:** `Device# show ip rsvp`	Displays general RSVP related information.
Step 5	**show ip rsvp sender** [**vrf**{***** | *vrf-name*}][**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Device# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output including the FLR parameters. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for RSVP FLR

Example Configuring RSVP FLR

The configuration options for RSVP FLR are the following:

- Wait time
- Number of notifications
- Repair rate

Note You can configure these options in any order.

Configuring the Wait Time

The following example configures Ethernet interface 1/0 with a bandwidth of 200 kbps and a wait time of 1000 msec:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface ethernet1/0
Device(config-if)# ip rsvp bandwidth 200
Device(config-if)# ip rsvp signalling fast-local-repair wait-time 1000
```

Device(config-if)# **end**

Configuring the Number of Notifications

The following example configures the number of flows that are repaired before suspending to 100:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp signalling fast-local-repair notifications 100
Device(config)# end
```

Configuring the Repair Rate

The following example configures a repair rate of 100 messages per second:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# ip rsvp signalling fast-local-repair rate 100
Device(config)# end
```

Example Verifying the RSVP FLR Configuration

This section contains the following examples:

- Example Verifying the RSVP FLR Configuration, page 225
- Example Verifying the RSVP FLR Configuration, page 225
- Example Verifying the RSVP FLR Configuration, page 225

Verifying the Details for FLR Procedures

The following example displays detailed information about FLR procedures:

```
Device# show ip rsvp signalling fast-local-repair statistics detail
Fast Local Repair: enabled
  Max repair rate (paths/sec): 10
  Max processed   (paths/run): 10
FLR Statistics:
  FLR 1: DONE
    Start Time: 05:18:54 IST Mon Nov 5 2007
    Number of PSBs repaired:          2
```

```
    Used Repair Rate (msgs/sec):      10
    RIB notification processing time: 0(us).
    Time of last PSB refresh:         5025(ms).
    Time of last Resv received:       6086(ms).
    Time of last Perr received:       0(us).
    Suspend count: 0
    FLR Pacing Unit: 100 msec.
    Affected neighbors:
      Nbr Address   Interface    Relative Delay Values (msec)      VRF
      10.1.2.12         Et0/3          [5000  ,..., 5000  ]        vrfRed
      10.1.2.12         Et1/3          [5000  ,..., 5000  ]        vrfBlue
```

Verifying Configuration Details for a Specific Interface

The following example from the **show ip rsvp interface detail** command displays detailed information, including FLR, for the Ethernet 1/0 interface:

```
Device# show ip rsvp interface detail ethernet1/0
  Et1/0:
    RSVP: Enabled
    Interface State: Up
    Bandwidth:
      Curr allocated: 9K bits/sec
      Max. allowed (total): 300K bits/sec
      Max. allowed (per flow): 300K bits/sec
      Max. allowed for LSP tunnels using sub-pools (pool 1): 0 bits/sec
      Set aside by policy (total): 0 bits/sec
    Traffic Control:
      RSVP Data Packet Classification is ON via CEF callbacks
    Signalling:
      DSCP value used in RSVP msgs: 0x30
      Number of refresh intervals to enforce blockade state: 4
    FLR Wait Time (IPv4 flows):
      Repair is delayed by 1000 msec.
    Authentication: disabled
      Key chain:   <none>
      Type:        md5
      Window size: 1
      Challenge:   disabled
    Hello Extension:
      State: Disabled
```

Verifying Configuration Details Before, During, and After an FLR Procedure

The following is sample output from the **show ip rsvp sender detail** command before an FLR procedure has occurred:

```
Device# show ip rsvp sender detail
PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.3.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 01000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Output on Ethernet1/0. Policy status: Forwarding. Handle: 02000400
     Policy source(s): Default
   Path FLR: Never repaired
```

The following is sample output from the **show ip rsvp sender detail** command at the PLR during an FLR procedure:

```
Device# show ip rsvp sender detail
PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
```

```
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 01000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Path FLR: PSB is currently being repaired...try later
   PLR - Old Segments: 1
    Output on Ethernet1/0, nhop 172.5.36.34
    Time before expiry: 2 refreshes
    Policy status: Forwarding. Handle: 02000400
       Policy source(s): Default
```

The following is sample output from the **show ip rsvp sender detail** command at the MP during an FLR procedure:

```
Device# show ip rsvp sender detail
PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.37.35 on Et1/0 every 30000 msecs
Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 09000406.
   Incoming policy: Accepted. Policy source(s): Default
   Status: Proxy-terminated
   Path FLR: Never repaired
   MP - Old Segments: 1
    Input on Serial2/0, phop 172.16.36.35
    Time before expiry: 9 refreshes
```

The following is sample output from the **show ip rsvp sender detail** command at the PLR after an FLR procedure:

```
Device# show ip rsvp sender detail
PATH:
   Destination 192.168.101.21, Protocol_Id 17, Don't Police , DstPort 1
   Sender address: 10.10.10.10, port: 1
   Path refreshes:
     arriving: from PHOP 172.16.31.34 on Et0/0 every 30000 msecs
   Traffic params - Rate: 9K bits/sec, Max. burst: 9K bytes
     Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
   Path ID handle: 05000401.
   Incoming policy: Accepted. Policy source(s): Default
   Status:
   Output on Serial3/0. Policy status: Forwarding. Handle: 3B000406
     Policy source(s): Default
   Path FLR: Started 12:56:16 EST Thu Nov 16 2006, PSB repaired 532(ms) after.
```

Resv/Perr: Received 992(ms) after.

Additional References

The following sections provide references related to the RSVP FLR feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification
RFC 2209	Resource ReSerVation Protocol (RSVP)--Version 1 Messaging Processing Rules

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 RSVP FLR

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 10 Feature Information for RSVP FLR

Feature Name	Releases	Feature Information
RSVP Fast Local Repair	12.2(33)SRB, 15.0(1)M	The RSVP Fast Local Repair feature provides quick adaptation to routing changes without the overhead of the refresh period to guarantee QoS for data flows. With FLR, RSVP speeds up its response to routing changes from 30 seconds to a few seconds. This feature was integrated into Cisco IOS Release 15.0(1)M. Support for FLR in VRF domains was added. The following commands were introduced or modified: **clear ip rsvp signalling fast-local-repair statistics**, **ip rsvp signalling fast-local-repair notifications**, **ip rsvp signalling fast-local-repair rate**, **ip rsvp signalling fast-local-repair wait-time**, **show ip rsvp**, **show ip rsvp interface**, **show ip rsvp sender**, **show ip rsvp signalling fast-local-repair**.

Glossary

admission control --The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

message pacing-- A system for managing volume and timing that permits messages from multiple sources to be spaced apart over time. RSVP message pacing maintains, on an outgoing basis, a count of the messages that it has been forced to drop because the output queue for the interface used for the message pacing was full.

MP --merge point. The node where the new and old FLR segments meet.

PLR --point of local repair. The node that initiates an FLR procedure.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

VRF --Virtual Routing and Forwarding. VRF is A VPN routing and forwarding instance. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a PE device.

RSVP Interface-Based Receiver Proxy

The RSVP Interface-Based Receiver Proxy feature lets you configure a proxy device by outbound interface instead of configuring a destination address for each flow going through the same interface.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Interface-Based Receiver Proxy

You must configure an IP address and enable Resource Reservation Protocol (RSVP) on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for RSVP Interface-Based Receiver Proxy

- Filtering using access control lists (ACLs), application IDs, or other mechanisms is not supported.
- A provider edge (PE) router cannot switch from being a proxy node to a transit node for a given flow during the lifetime of the flow.

Information About RSVP Interface-Based Receiver Proxy

Feature Overview of RSVP Interface-Based Receiver Proxy

The RSVP Interface-Based Receiver Proxy feature allows you to use RSVP to signal reservations and guarantee bandwidth on behalf of a receiver that does not support RSVP, by terminating the PATH message and generating a RESV message in the upstream direction on an RSVP-capable router on the path to the endpoint. An example is a video-on-demand flow from a video server to a set-top box, which is a computer that acts as a receiver and decodes the incoming video signal from the video server.

Because set-top boxes may not support RSVP natively, you cannot configure end-to-end RSVP reservations between a video server and a set-top box. Instead, you can enable the RSVP interface-based receiver proxy on the router that is closest to that set-top box.

The router terminates the end-to-end sessions for many set-top boxes and performs admission control on the outbound (or egress) interface of the PATH message, where the receiver proxy is configured, as a proxy for Call Admission Control (CAC) on the router-to-set-top link. The RSVP interface-based receiver proxy determines which PATH messages to terminate by looking at the outbound interface to be used by the traffic flow.

You can configure an RSVP interface-based receiver proxy to terminate PATH messages going out a specified interface with a specific action (reply with RESV, or reject). The most common application is to configure the receiver proxy on the edge of an administrative domain on interdomain interfaces. The router then terminates PATH messages going out the administrative domain while still permitting PATH messages transitioning through the router within the same administrative domain to continue downstream.

In the video-on-demand example described above, the last-hop Layer 3 router supporting RSVP implements the receiver proxy, which is then configured on the interfaces facing the Layer 2 distribution network (for example, Digital Subscriber Line access [DSLAM] or cable distribution). Also, since RSVP is running and performing CAC on the router with the receiver proxy, you can configure RSVP enhancements such as local policy and Common Open Policy Service (COPS) for more fine-grained control on video flow CAC.

The router terminates the end-to-end sessions for many set-top boxes, with the assumption that the links further downstream (for example, from the DSLAM to the set-top box) never become congested or, more likely, in the case of congestion, that the voice and video traffic from the router gets the highest priority and access to the bandwidth.

Benefits of RSVP Interface-Based Receiver Proxy

Ease of Use and Scalability Improvement

Previously, you had to configure a receiver proxy for every separate RSVP stream or set-top box. Now you can configure the proxy by outbound interface. For example, if there were 100 set-top boxes downstream from the proxy router, you had to configure 100 proxies. With this enhancement, you configure only the outbound interface(s). In addition, the receiver proxy is guaranteed to terminate the reservation only on the last hop within the core network. Nodes that may function as transit nodes for some PATH messages but should proxy others depending on their placement in the network can perform the correct functions on a flow-by-flow basis.

In the video-on-demand example described above, a PATH message that transits through an edge router to another edge router (around the edge) is not terminated, whereas an otherwise identical PATH message that actually exits the aggregation network and transitions to the access network is terminated. This allows for more accurate CAC in the network and also simplifies and reduces configuration requirements.

How to Configure RSVP Interface-Based Receiver Proxy

Enabling RSVP on an Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]]
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** *interface number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] [**sub-pool** [*sub-pool-kbps*]] **Example:** `Router(config-if)# ip rsvp bandwidth 7500 7500`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. • The optional **sub-pool**and *sub-pool-kbps*keyword and argument specify subpool traffic and the amount of bandwidth that can be allocated by RSVP flows. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**end** **Example:** `Router(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring a Receiver Proxy on an Outbound Interface

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *interface number*
4. **ip rsvp listener outbound** {**reply** | **reject**}
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.

	Command or Action	Purpose
Step 3	**interface** *interface number* **Example:** `Router(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp listener outbound** {**reply** | **reject**} **Example:** `Router(config-if)# ip rsvp listener outbound reject`	Configures an RSVP router to listen for PATH messages sent through a specified interface. • Enter the **reply** keyword or the **reject** keyword to specify the response that you want to PATH messages.
Step 5	**end** **Example:** `Router(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Verifying the RSVP Interface-Based Receiver Proxy Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp listeners** [*dst* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**]
3. **show ip rsvp sender** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]]
4. **show ip rsvp reservation** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]]
5. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Router> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.

	Command or Action	Purpose
Step 2	**show ip rsvp listeners** [*dst* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**] **Example:** `Router# show ip rsvp listeners`	Displays RSVP listeners for a specified port or protocol.
Step 3	**show ip rsvp sender** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional **detail** keyword displays additional output. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 4	**show ip rsvp reservation** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-addr*| *hostname*] [**source** *ip-addr*| *hostname*] [**dst-port** *port*] [**src-port** *port*]] **Example:** `Router# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional **detail** keyword displays additional output. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 5	**exit** **Example:** `Router# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

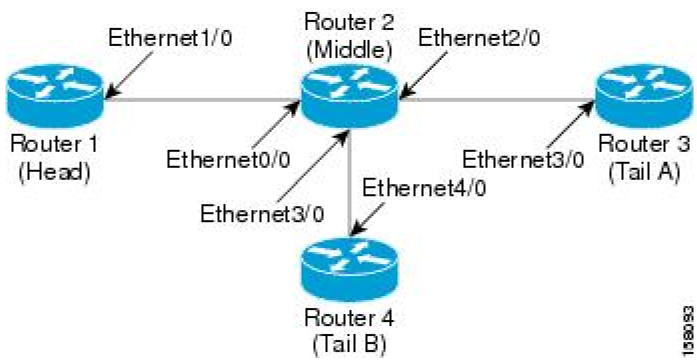
Configuration Examples for RSVP Interface-Based Receiver Proxy

- Examples Configuring RSVP Interface-Based Receiver Proxy, page 237
- Examples Verifying RSVP Interface-Based Receiver Proxy, page 237

Examples Configuring RSVP Interface-Based Receiver Proxy

The four-router network in the figure below contains the following configurations:

Figure 25 Sample Network with an Interface-Based Receiver Proxy Configure



Configuring a Receiver Proxy on a Middle Router on Behalf of Tailend Routers

The following example configures a receiver proxy, also called a listener, on the middle router (Router 2) on behalf of the two tailend routers (Routers 3 and 4):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface ethernet 2/0
Router(config-if)# ip rsvp listener outbound reply
Router(config-if)# exit
Router(config)# interface ethernet 3/0
Router(config-if)# ip rsvp listener outbound reject
Router(config-if)# end
```

Configuring PATH Messages from a Headend Router to Tailend Routers to Test the Receiver Proxy

Note If you do not have another headend router generating RSVP PATH messages available, configure one in the network for the specific purpose of testing RSVP features such as the receiver proxy. Note that these commands are not expected (or supported) in a final deployment.

The following example configures four PATH messages from the headend router (Router 1) to the tailend routers (Routers 3 and 4):

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp sender-host 10.0.0.5 10.0.0.1 TCP 2 2 100 10
Router(config)# ip rsvp sender-host 10.0.0.5 10.0.0.1 UDP 1 1 100 10
Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 TCP 4 4 100 10
Router(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 UDP 3 3 100 10
Router(config)# end
```

Examples Verifying RSVP Interface-Based Receiver Proxy

Verifying the PATH Messages in the Database

The following example verifies that the PATH messages you configured are in the database:

```
Router# show ip rsvp sender
To              From            Pro DPort Sport Prev Hop        I/F      BPS
10.0.0.5        10.0.0.1        TCP 2     2     none            none     100K
10.0.0.5        10.0.0.1        UDP 1     1     none            none     100K
10.0.0.7        10.0.0.1        TCP 4     4     none            none     100K
10.0.0.7        10.0.0.1        UDP 3     3     none            none     100K
```

The following example verifies that a PATH message has been terminated by a receiver proxy configured to reply.

Note A receiver proxy that is configured to reject does not cause any state to be stored in the RSVP database; therefore, this **show** command does not display these PATHS. Only one PATH message is shown.

```
Router# show ip rsvp sender detail
PATH:
  Destination 10.0.0.5, Protocol_Id 17, Don't Police , DstPort 1
  Sender address: 10.0.0.1, port: 1
  Path refreshes:
    arriving: from PHOP 10.1.2.1 on Et0/0 every 30000 msecs
  Traffic params - Rate: 100K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  Path ID handle: 01000402.
  Incoming policy: Accepted. Policy source(s): Default
  Status: Proxy-terminated
  Output on Ethernet2/0. Policy status: NOT Forwarding. Handle: 02000401
    Policy source(s):
  Path FLR: Never repaired
```

Verifying the Running Configuration

The following example verifies the configuration for Ethernet interface 2/0:

```
Router# show running-config interface Ethernet2/0
Building configuration...
Current configuration : 132 bytes
!
interface Ethernet2/0
 ip address 172.16.0.1 255.0.0.0
 no cdp enable
 ip rsvp bandwidth 2000
 ip rsvp listener outbound reply
end
```

The following example verifies the configuration for Ethernet interface 3/0:

```
Router# show running-config interface Ethernet3/0
Building configuration...
Current configuration : 133 bytes
!
interface Ethernet3/0
 ip address 172.16.0.2 255.0.0.0
 no cdp enable
 ip rsvp bandwidth 2000
 ip rsvp listener outbound reject
end
```

Verifying the Listeners

The following example verifies the listeners (proxies) that you configured on the middle router (Router 2) on behalf of the two tailend routers (Routers 3 and 4):

```
To                 Protocol   DPort   Description                 Action    OutIf
10.0.0.0           0          0       RSVP Proxy                  reply     Et2/0
10.0.0.0           0          0       RSVP Proxy                  reject    Et3/0
```

Verifying the Reservations

The following example displays reservations established by the middle router (Router 2) on behalf of the tailend routers (Routers 3 and 4) as seen from the headend router (Router 1):

```
Router# show ip rsvp reservation
To            From          Pro DPort Sport Next Hop      I/F      Fi Serv BPS
10.0.0.7      10.0.0.1      TCP 4     4     10.0.0.2      Et1/0    FF RATE 100K
10.0.0.7      10.0.0.1      UDP 3     3     10.0.0.2      Et1/0    FF RATE 100K
```

The following example verifies that a reservation is locally generated (proxied). Only one reservation is shown:

```
Router# show ip rsvp reservation detail
RSVP Reservation. Destination is 10.0.0.7, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 1, Source port is 1
  Next Hop: 10.2.3.3 on Ethernet2/0
  Reservation Style is Fixed-Filter, QoS Service is Guaranteed-Rate
  Resv ID handle: 01000405.
  Created: 09:24:24 EST Fri Jun 2 2006
  Average Bitrate is 100K bits/sec, Maximum Burst is 10K bytes
  Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Status: Proxied
  Policy: Forwarding. Policy source(s): Default
```

Verifying CAC on an Outbound Interface

The following example verifies that the proxied reservation performed CAC on the local outbound interface:

```
Router# show ip rsvp installed
RSVP: Ethernet3/0 has no installed reservations
RSVP: Ethernet2/0
BPS    To              From            Protoc DPort  Sport
100K   10.0.0.7        10.0.0.1        UDP    1       1
```

Additional References

The following sections provide references related to the RSVP Interface-Based Receiver Proxy feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS configuration tasks related to RSVP	"Configuring RSVP" module

Related Topic	Document Title
Internet draft	*RSVP Proxy Approaches* , Internet draft, October 2006 [draft-lefaucheur-tsvwg-rsvp-proxy-00.txt]
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	Resource ReSerVation Protocol (RSVP)

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 RSVP Interface-Based Receiver Proxy

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

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

Table 11 Feature Information for RSVP Interface-Based Receiver Proxy

Feature Name	Releases	Feature Information
RSVP Interface-Based Receiver Proxy	12.2(28)SXF5 12.2(33)SRB, 15.0(1)M	The RSVP Interface-Based Receiver Proxy feature lets you configure a proxy router by outbound interface instead of configuring a destination address for each flow going through the same interface. In Cisco IOS Release 12.2(33)SRB, support was added for the Cisco 7600 series routers. This feature was integrated into Cisco IOS Release 15.0(1)M. The following commands were introduced or modified: **ip rsvp bandwidth**, **ip rsvp listener outbound**, **show ip rsvp listeners**, **show ip rsvp reservation**, **show ip rsvp sender**.

Glossary

flow --A stream of data traveling between two endpoints across a network (for example, from one LAN station to another). Multiple flows can be transmitted on a single circuit.

PE router --provider edge router. A router that is part of a service provider’s network and is connected to a customer edge (CE) router.

proxy --A component of RSVP that manages all locally originated and terminated state.

receiver proxy --A configurable feature that allows a router to proxy RSVP RESV messages for local or remote destinations.

RSVP --Resource Reservation Protocol. A protocol for reserving network resources to provide quality of service guarantees to application flows.

set-top box --A computer that acts as a receiver and decodes the incoming signal from a satellite dish, a cable network, or a telephone line.

RSVP-VRF Lite Admission Control

The RSVP--VRF Lite Admission Control feature introduces support for Resource Reservation Protocol (RSVP) call admission control (CAC) in an IP session within the context of a virtual routing and forwarding (VRF) instance.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP-VRF Lite Admission Control

You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for RSVP-VRF Lite Admission Control

- Multi-topology routing (MTR) is not supported.
- Multiprotocol Label Switching (MPLS) virtual private network (VPN) VRFs are not supported.
- RSVP authentication is not supported.

Information About RSVP-VRF Lite Admission Control

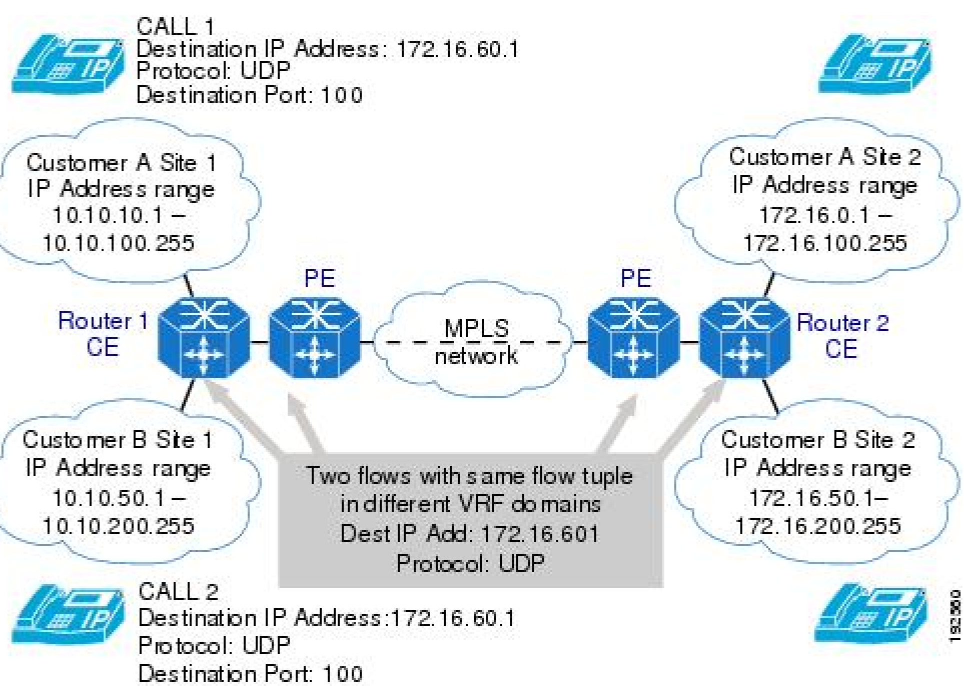
Overview of RSVP-VRF Lite Admission Control

An RSVP flow is identified by its tuple, which includes its destination IP address, its destination port, and its protocol. This tuple should be unique on all the nodes along the path from the sender to the receiver. In the context of the global routing domain, each flow can be uniquely identified through its tuple.

However, with the implementation of virtual routing and forwarding (VRF), a separate instance of the routing and forwarding table for each VRF routing domain can exist. Each of the VRF instances has its own address pool range, which could overlap between VRF routing domains. This poses a problem to the existing implementation of RSVP, where sessions are identified by the tuple. Sessions with the same tuple can exist in the context of different VRF domains. To solve the problem, the tuple has to be extended to take into account the VRF instance. The new tuple has a *VRF ID, a destination IP address* , a *destination port, and a protocol* . The VRF ID is derived based on the interface on which an RSVP packet has been received and is not signaled using RSVP. Therefore, each node needs to infer the VRF ID based on the RSVP control packet's incoming interface.

The figure below shows a VRF-lite deployment scenario.

Figure 26 RSVP VRF Deployment in VRF-Lite Network



The figure above shows VRF lite configured on Router 1 customer edge (CE) and Router 2 CE, and MPLS-VPN configured between the provider edge (PE) devices. In such a deployment scenario, the RSVP implementation needs to be VRF aware in the CE devices; that is, the flows must be recognized in the context of the VRF domain in which the sender and receiver of the flow reside. However, RSVP QoS is not enabled on the PE devices.

On the CE devices, with VRF lite functionality, VRF is identified based on the VRF configured on the incoming interface; that is, on the interface facing the customer site and the interface facing the PE.

Benefits of RSVP-VRF Lite Admission Control

The RSVP--VRF Lite Admission Control feature provides the benefits of RSVP in a VRF-lite environment to include the following:

- Guaranteed QoS through explicit admission control
- Virtualization
- Security
- Separation of routing contexts
- Overlapping of IP addresses

How to Configure RSVP-VRF Lite Admission Control

Note The tasks described in this section explain configuring a receiver proxy and a static sender, for you to quickly initiate and terminate an RSVP session, and verify your setup. In these tasks, the IOS RSVP implementation behaves as an RSVP endpoint and an RSVP initiator.

Enabling RSVP on an Interface

Perform this task to enable RSVP on all the interfaces along the path from the sender to the receiver.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip routing**
4. **ip vrf** *vrf-name*
5. **exit**
6. **interface** *type number*
7. **ip vrf forwarding** *vrf-name*
8. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
9. Repeat the previous step for each interface that you want to enable.
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip routing** **Example:** `Device(config)# ip routing`	Enables IP routing.
Step 4	**ip vrf** *vrf-name* **Example:** `Device(config)# ip vrf vrf1`	Defines a VRF instance and enters VRF configuration mode.
Step 5	**exit** **Example:** `Device(config-vrf)# exit`	Exits VRF configuration mode and enters global configuration mode.

	Command or Action	Purpose
Step 6	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 7	**ip vrf forwarding** *vrf-name* **Example:** `Device(config-if)# ip vrf forwarding vrf1`	Associates a VRF instance with an interface or subinterface.
Step 8	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 1158 100`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface that you want to enable.
Step 9	Repeat the previous step for each interface that you want to enable.	--
Step 10	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring a Receiver Proxy on a Tailend Device

Perform this task to configure a receiver proxy with a VRF on a tailend device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**} **Example:** `Device(config)# ip rsvp listener` *vrf myvrf* `192.168.2.1 any any reply`	Configures an RSVP device to listen for PATH messages. • Enter the appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring a Static Sender on a Headend Device

Perform this task to configure a static sender with a VRF on a headend device to make the device proxy an RSVP PATH message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp sender-host** *session-ip-address sender-ip-address* {**tcp** | **udp** | *ip-protocol*} *session-d-port sender-s-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*] **Example:** `Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 vrf myvrf`	Enables a device to simulate a host generating RSVP PATH messages. • Enter the appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config)#` **end**	Exits global configuration mode and returns to privileged EXEC mode.

Configuring an RSVP Application Identity That Is VRF Aware

Perform the following task to configure an RSVP application identity that is VRF aware.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config)# ip rsvp policy vrf myvrf identity voice policy-locator voiceStream`	Creates a local policy for a VRF and enters local policy configuration mode. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 4	**end** **Example:** `Device(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Configuring an RSVP Local Policy That Is VRF Aware

Perform the following task to configure an RSVP local policy that is VRF aware.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]}
4. {**accept** | **forward**[**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy vrf** *vrf-name* {**identity**{*alias* **policy-locator** *regular-expression*| **local**}}{**acl** *acl1*[*acl2...acl8*] | **default** | **identity** *alias1*[*alias2...alias4*]| **origin-as** *as1* [*as2...as8*]} **Example:** `Device(config)# ip rsvp policy vrf myvrf local default`	Creates a local policy for a VRF and enters local policy configuration mode. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 4	{**accept** | **forward**[**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** [**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*]| **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Device(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. **Note** See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 5	**end** **Example:** `Device(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Verifying the RSVP-VRF Lite Admission Control Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode and in any order.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp counters** [**authentication**][**interface** *type number*| **neighbor**[**vrf**{*****| *vrf-name*}] | **state teardown**| **summary**]
3. **show ip rsvp host vrf** {***** | *vrf-name*} {**receivers** | **senders**} [*group-name* | *group-address*]
4. **show ip rsvp installed** [**vrf**{***** | *vrf-name*}] [*interface-type interface-number*] [**detail**]
5. **show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] [*interface-type interface-number*]
6. **show ip rsvp listeners** [*ip-address* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**]
7. **show ip rsvp neighbor** [**detail** | **inactive**[**detail**]] | **vrf**{***** | *vrf-name*}]
8. **show ip rsvp policy vrf** {***** | *vrf-name*} [**identity**[*alias*]] | **local**[**acl** *acl* | **default** | **detail**[**acl** *acl* | **default** | **identity** *alias* | **interface** *interface-type* | **origin-as** *as-number*]]
9. **show ip rsvp request** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]]
10. **show ip rsvp reservation** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}]
11. **show ip rsvp sender** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}]
12. **show ip rsvp signalling fast-local-repair** [**statistics**[**detail**]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**show ip rsvp counters** [**authentication**][**interface** *type number*| **neighbor**[**vrf**{*****| *vrf-name*}] | **state teardown**| **summary**] **Example:** `Device#` **show ip rsvp counters neighbor vrf myvrf**	(Optional) Displays the number of RSVP messages that were sent and received on each interface. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 3	**show ip rsvp host vrf** {***** | *vrf-name*} {**receivers** | **senders**} [*group-name* | *group-address*] **Example:** `Device(config)# show ip rsvp vrf * senders`	(Optional) Displays specific information for an RSVP host configured with a VRF instance. • Enter the *vrf-name*name and any other appropriate keywords and arguments.

	Command or Action	Purpose
Step 4	**show ip rsvp installed** [**vrf**{***** | *vrf-name*}] [*interface-type interface-number*] [**detail**] **Example:** `Device#` **show ip rsvp installed vrf myvrf detail**	(Optional) Displays RSVP-related installed filters and corresponding bandwidth information. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 5	**show ip rsvp interface** [**vrf**{***** | *vrf-name*}] [**detail**] [*interface-type interface-number*] **Example:** `Device#` **show ip rsvp interface vrf myvrf detail**	(Optional) Displays information related to RSVP. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 6	**show ip rsvp listeners** [*ip-address* | **any** | **vrf**{***** | *vrf-name*}] [**udp** | **tcp** | **any** | *protocol*] [*dst-port* | **any**] **Example:** `Device#` **show ip rsvp listeners vrf myvrf1**	(Optional) Displays the RSVP listeners for a specified port or protocol. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 7	**show ip rsvp neighbor** [**detail** | **inactive**[**detail**]] | **vrf**{***** | *vrf-name*}] **Example:** `Device#` **show ip rsvp neighbor vrf myvrf1**	(Optional) Displays current RSVP neighbors. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 8	**show ip rsvp policy vrf** {***** | *vrf-name*} [**identity**[*alias*]] | **local**[**acl** *acl* | **default** | **detail**[**acl** *acl* | **default** | **identity** *alias* | **interface** *interface-type* | **origin-as** *as-number*]] **Example:** `Device#` **show ip rsvp policy vrf myvrf1**	(Optional) Displays information for an RSVP policy configured with a VRF instance. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 9	**show ip rsvp request** [**vrf**{***** | *vrf-name*}] [**detail**] [**filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] **Example:** `Device#` **show ip rsvp request vrf myvrf1**	(Optional) Displays RSVP-related request information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.

	Command or Action	Purpose
Step 10	**show ip rsvp reservation** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}] **Example:** `Device#` **show ip rsvp reservation vrf myvrf1**	(Optional) Displays RSVP-related receiver information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 11	**show ip rsvp sender** [**detail** | **filter** [**destination** *ip-address*| *hostname*] [**dst-port** *port-number*] [**source** *ip-address*| *hostname*] [**src-port** *port-number*]] [**vrf**{***** | *vrf-name*}] **Example:** `Device#` **show ip rsvp sender vrf myvrf1**	(Optional) Displays RSVP PATH-related sender information currently in the database. • Enter the *vrf-name*name and any other appropriate keywords and arguments.
Step 12	**show ip rsvp signalling fast-local-repair** [**statistics**[**detail**]] **Example:** `Device#` **show ip rsvp signalling fast-local repair statistics detail**	(Optional) Displays fast-local-repair (FLR)-specific information, including VRF, maintained by RSVP.

Configuration Examples for RSVP-VRF Lite Admission Control

Examples Configuring RSVP-VRF Lite Admission Control

The following example enables RSVP on a device interface along the path from the sender to the receiver.

Note If the interface lies in a VRF domain, use the **ip rsvp bandwidth** command to enable RSVP for that VRF.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface Ethernet0/0
Device(config-if)# ip rsvp bandwidth 1158 100
Device(config-if)# end
```

The following example configures a receiver proxy with a specified VRF on a tailend device:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# ip rsvp listener vrf myvrf 192.168.2.1 any any reply

Device(config)# end
```

The following example configures a static sender with a specified VRF on a headend device:

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# ip rsvp sender-host 10.0.0.7 10.0.0.1 udp 1 1 10 10 vrf myvrf
Device(config)# end
```

Examples Verifying RSVP-VRF Lite Admission Control

In the following example, all the interfaces associated with the VRF named myvrf display in detail:

```
Device# show ip rsvp interface vrf myvrf detail
Se1/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 300K bits/sec
     Max. allowed (total): 400K bits/sec
     Max. allowed (per flow): 400K bits/sec
     Max. allowed for LSP tunnels using sub-pools (pool 1): 0 bits/sec
     Set aside by policy (total): 0 bits/sec
   Traffic Control:
     RSVP Data Packet Classification is OFF
     RSVP resource provider is: none
   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
   FRR Extension:
     Backup Path: Not Configured
   BFD Extension:
     State: Disabled
     Interval: Not Configured
   RSVP Hello Extension:
     State: Disabled
   RFC 3175 Aggregation: Enabled
     Role: interior
   VRF: myvrf
```

The following example displays details of the RSVP reservations installed for RSVP session that belong to the VRF named myvrf:

```
Device# show ip rsvp installed vrf myvrf detail
RSVP: FastEthernet2/0 has the following installed reservations
RSVP Reservation. Destination is 10.10.10.10. Source is 10.10.10.12,
  Protocol is UDP, Destination port is 10, Source port is 10
  Traffic Control ID handle: C8000407
  Created: 22:51:26 UTC Sun Feb 17 2008
  Admitted flowspec:
    Reserved bandwidth: 10K bits/sec, Maximum burst: 10K bytes, Peak rate: 10K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Resource provider for this flow: None
  Conversation supports 1 reservations [0xBF000406]
  Data given reserved service: 0 packets (0 bytes)
  Data given best-effort service: 0 packets (0 bytes)
  Reserved traffic classified for 12783 seconds
```

```
  Long-term average bitrate (bits/sec): 0 reserved, 0 best-effort
  Policy: INSTALL. Policy source(s): Default
VRF : myvrf
```

The following example shows the listeners configured for the VRF named myvrf:

```
Device# show ip rsvp listeners vrf myvrf
VRF : myvrf1
```

To Protocol DPort Description Action OutIf

10.0.2.1 any any RSVP Proxy reply

The following example shows the neighbors created for the VRF named myvrf:

```
Device# show ip rsvp neighbor vrf myvrf
VRF: myvrf
Neighbor        Encapsulation  Time since msg rcvd/sent
10.10.15.3         Raw IP         00:00:14   00:00:06
10.10.16.2         Raw IP         00:00:29   00:00:15
```

The following example displays all the locally created RSVP senders for the configured VRFs:

```
Device# show ip rsvp host vrf * senders
VRF: vrf2
To                  From                 Pro DPort Sport Prev Hop      I/F      BPS
192.168.104.4       198.168.104.12       UDP 10    10    none          none     10K
  Mode(s): Host CLI
VRF: vrf1
To                  From                 Pro DPort Sport Prev Hop      I/F      BPS
192.168.105.4       198.168.105.12       UDP 10    10    none          none     10K
  Mode(s): Host CLI
```

Additional References

The following sections provide references related to the RSVP--VRF Lite Admission Control feature.

Related Documents

Related Topic	Document Title
QoS commands: complete command syntax, command modes, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
VRF-related internet draft	*Support for RSVP in Layer 3 VPNs*, Internet draft, November 19, 2007 [draft-davie-tsvwg-rsvp-l3vpn-01.txt]
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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 IOS 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 by this feature, and support for existing RFCs has not been modified by this feature.	--

Technical Assistance

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

Feature Information for RSVP-VRF Lite Admission Control

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

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

Table 12 Feature Information for RSVP--VRF Lite Admission Control

Feature Name	Releases	Feature Information
RSVP--VRF Lite Admission Control	15.0(1)SY	The RSVP--VRF Lite Admission Control feature introduces support for RSVP CAC in an IP session within the context of a VRF instance. The following commands were introduced or modified by this feature: **debug ip rsvp** , **ip rsvp listener, ip rsvp policy vrf, ip rsvp reservation-host, ip rsvp sender-host, show ip rsvp counters, show ip rsvp host vrf, show ip rsvp installed, show ip rsvp interface, show ip rsvp listeners, show ip rsvp neighbor, show ip rsvp policy vrf, show ip rsvp request, show ip rsvp reservation, show ip rsvp sender, show ip rsvp signalling fast-local-repair.**

Glossary

admission control --The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability. Quality of service focuses on achieving appropriate network performance for networked applications; it is superior to best effort performance.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications that run on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

VRF --virtual routing and forwarding. An extension of IP routing that provides multiple routing instances. A VRF consists of an IP routing table, a derived forwarding table, a set of interfaces that use the forwarding table, and a set of rules and routing protocols that determine what goes into the forwarding table. In general, a VRF includes the routing information that defines a customer VPN site that is attached to a provider edge (PE) device.

Configuring RSVP Support for LLQ

This chapter describes the tasks for configuring the RSVP Support for Low Latency Queueing (LLQ) feature.

For complete conceptual information, see the chapter "Signalling Overview" in this book.

For a complete description of the RSVP Support for LLQ commands in this chapter, see the Cisco IOS Quality of Service Solutions Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

RSVP Support for LLQ Configuration Task List

To configure RSVP support for LLQ, perform the tasks described in the following sections. The tasks in the first two sections are required; the tasks in the remaining sections are optional.

Configuring Flow Classification

To configure flow classification, use the following command in global configuration mode:

Command	Purpose
`Router#(config)#` **ip rsvp pq-profile**	Specifies the criteria for determining which flows go into the priority queue.

Enabling RSVP and WFQ

To enable RSVP and weighted fair queueing (WFQ), use the following commands beginning in global configuration mode:

SUMMARY STEPS

1. Router(config)# **interface s2/0**
2. Router(config-if)# **ip rsvp bandwidth**
3. Router(config-if)# **fair-queue**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# **interface s2/0**	Enables an interface; for example, serial interface 2/0.
Step 2	Router(config-if)# **ip rsvp bandwidth**	Enables RSVP on an interface.
Step 3	Router(config-if)# **fair-queue**	Enables WFQ on an interface with priority queueing (PQ) support.

Configuring a Burst Factor

To configure a burst factor, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **ip rsvp burst policing**	Specifies a burst factor on a per-interface basis.

Configuring a Path

To configure a path, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ip rsvp sender**	Specifies the RSVP path parameters, including the destination and source addresses, the protocol, the destination and source ports, the previous hop address, the average bit rate, and the burst size.

Configuring a Reservation

To configure a reservation, use the following command in global configuration mode:

Command	Purpose
Router(config)# **ip rsvp reservation**	Specifies the RSVP reservation parameters, including the destination and source addresses, the protocol, the destination and source ports, the next hop address, the input interface, the service type, the average bit rate, and the burst size.

Verifying RSVP Support for LLQ Configuration

To verify RSVP support for LLQ configuration, perform the following steps:

SUMMARY STEPS

1. Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. A sample output is shown.
2. Enter the **show ip rsvp installed detail**command to display additional information about interfaces and their current reservations. A sample output is shown.

DETAILED STEPS

Step 1 Enter the **show ip rsvp installed**command to display information about interfaces and their admitted reservations. A sample output is shown.
This output shows that Ethernet interface 2/1 has four reservations and serial interface 3/0 has none.

Example:

```
Router# show ip rsvp installed
RSVP:Ethernet2/1
BPS    To              From            Protoc DPort  Sport  Weight Conversation
44K    145.20.0.202    145.10.0.201    UDP    1000   1000   0      264
44K    145.20.0.202    145.10.0.201    UDP    1001   1001   13     266
98K    145.20.0.202    145.10.0.201    UDP    1002   1002   6      265
1K     145.20.0.202    145.10.0.201    UDP    10     10     0      264
RSVP:Serial3/0 has no installed reservations
Router#
```

Note In the sample output, weight 0 is assigned to voice-like flows, which proceed to the priority queue.

Step 2 Enter the **show ip rsvp installed detail**command to display additional information about interfaces and their current reservations. A sample output is shown.

Example:

```
Router# show ip rsvp installed detail
RSVP:Ethernet2/1 has the following installed reservations
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1000, Source port is 1000
  Reserved bandwidth:44K bits/sec, Maximum burst:1K bytes, Peak rate:44K bits/sec
  Resource provider for this flow:
    WFQ on hw idb Se3/0: PRIORITY queue 264.  Weight:0, BW 44 kbps
  Conversation supports 1 reservations
  Data given reserved service:316 packets (15800 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 104 seconds
  Long-term average bitrate (bits/sec):1212 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1001, Source port is 1001
  Reserved bandwidth:44K bits/sec, Maximum burst:3K bytes, Peak rate:44K bits/sec
  Resource provider for this flow:
    WFQ on hw idb Se3/0: RESERVED queue 266.  Weight:13, BW 44 kbps
  Conversation supports 1 reservations
  Data given reserved service:9 packets (450 bytes)
  Data given best-effort service:0 packets (0 bytes)
  Reserved traffic classified for 107 seconds
  Long-term average bitrate (bits/sec):33 reserved, 0M best-effort
RSVP Reservation. Destination is 145.20.0.202, Source is 145.10.0.201,
  Protocol is UDP, Destination port is 1002, Source port is 1002
Router#
```

Note In the sample output, the first flow gets the priority queue (weight = 0) while the second flow does not.

Example:

Monitoring and Maintaining RSVP Support for LLQ

To monitor and maintain the RSVP Support for LLQ feature, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router#` **show ip rsvp installed**	Displays information about interfaces and their admitted reservations.
`Router#` **show ip rsvp installed detail**	Displays additional information about interfaces and their admitted reservations.
`Router#` **show queue** *interface-type interface-number*	Displays queueing configuration and statistics for a particular interface.

Example RSVP Support for LLQ Configuration

This section provides a configuration example for the RSVP Support for LLQ feature.

In the following example, PQ parameters, including flow rate and burst factor, are defined:

```
Router(config)# ip rsvp pq-profile ?
  <1-1048576>  Max Flow Rate (bytes/second)
  voice-like   Voice-like flows
  <cr>
Router(config)# ip rsvp pq-profile 11000 1500 ?
  <100-4000>         Max Peak to Average Ratio (in %)
  ignore-peak-value  Ignore the flow's p/r ratio
  <cr>
Router(config)# ip rsvp pq-profile 11000 1500 ignore-peak-value
Router(config)# end
Router# sh run | include pq-profile
ip rsvp pq-profile 11000 1500 ignore-peak-value
```

In the following example, RSVP is enabled:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface loopback 40
Router(config-if)# ip rsvp bandwidth ?
  <1-10000000>  Reservable Bandwidth(KBPS)
  <cr>
Router(config-if)# ip rsvp bandwidth 300 ?
  <1-10000000>  Largest Reservable Flow(KBPS)
  <cr>
Router(config-if)# ip rsvp bandwidth 300 30 ?
  <cr>
Router(config-if)# ip rsvp bandwidth 300 30
Router(config-if)# end
```

In the following example, WFQ is enabled:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# interface e0/1
Router(config-if)# fair-queue
Router(config-if)# fair-queue 64
```

In the following example, a burst factor is configured:

```
Router(config)# interface e3/0
Router(config-if)# ip rsvp burst policing 200
```

In the following example, a path is defined:

```
Router(config)# ip rsvp sender 145.20.20.202 145.10.10.201 udp 10 20
145.10.10.201 loopback 10  80 10
```

In the following example, a reservation is defined:

```
Router(config)# ip rsvp reservation 145.20.20.202 145.10.10.201 udp
10 20 145.20.20.202 lo20 ff load 80 10
```

Configuring RSVP-ATM QoS Interworking

This chapter describes the tasks for configuring the RSVP-ATM QoS Interworking feature, which provides support for Controlled Load Service using RSVP over an ATM core network.

For complete conceptual information, see the "Signalling Overview" module.

For a complete description of the RSVP-ATM QoS Interworking commands in this module, see the Cisco IOS Quality of Service Solutions Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

RSVP-ATM QoS Interworking Configuration Task List

To configure RSVP-ATM QoS Interworking, perform the tasks described in the following sections. Each task is identified as either optional or required.

- Enabling RSVP and Limiting Reservable Bandwidth, page 268 (Required)
- Enabling Creation of SVCs for Reserved Flows, page 268 (Required)
- Limiting the Peak Rate Applied to the PCR for SVCs, page 271 (Optional)
- Configuring per-VC DWRED, page 271 (Required)
- Monitoring RSVP-ATM Configuration for an Interface, page 271 (Optional)

Before you configure RSVP-ATM QoS Interworking, you must enable and configure the following features:

- Cisco Express Forwarding (CEF) switching (required for RSVP-ATM)
- Distributed CEF (dCEF) (required for per-switched virtual circuit (SVC) DWRED)
- NetFlow services

For information about CEF and dCEF, refer to the Cisco IOS Switching Services Command Reference.

The RSVP-ATM QoS Interworking feature does not support Resource Reservation Protocol (RSVP) with multicast.

See the end of this module for the section "RSVP-ATM QoS Interworking Configuration Examples, page 272."

Enabling RSVP and Limiting Reservable Bandwidth

RSVP 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. By default, RSVP is disabled so that it is backward compatible with systems that do not implement RSVP.

To enable RSVP on an interface and restrict the total amount of bandwidth that can be reserved for RSVP and the amount that can be reserved for a single RSVP reservation or flow, use the following command in global configuration mode:

Command	Purpose
`Router(config)#` **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]	Enables RSVP for IP on an interface.

For RSVP over ATM, reservations are needed primarily between routers across the ATM backbone. To limit the number of locations where reservations are made, enable RSVP selectively only at subinterfaces corresponding to router-to-router connections across the backbone network. Preventing reservations being made between the host and the router both limits VC usage and reduces load on the router.

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.

On subinterfaces, the more restrictive of the available bandwidths of the physical interface and the subinterface is applied.

Enabling Creation of SVCs for Reserved Flows

Normally, reservations are serviced when RSVP classifies packets and a queueing mechanism polices the packet. To enable establishment of an SVC to service each new RSVP reservation on the interface, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **ip rsvp svc-required**	Enables creation of an SVC for each new reservation made on the interface or subinterface.

To ensure defined QoS, SVCs created in response to RSVP reservation requests are established having QoS profiles consistent with the mapped RSVP flow specifications.

The sustainable cell rate (SCR) of an ATM SVC is equal to the RSVP reservation rate; the maximum burst size (MBS) of an ATM SVC is equal to the RSVP burst size. RSVP attempts to compensate for the cell tax when establishing the reservation so that the requested bandwidth is actually available for IP data traffic.

The sustained cell rate formula is given as follows:

```
r
atm =
r
rsvp * (53/48) * (MPS + DLE + (MPS + DLE) % 48)/MPS
```

The formula terms used in the equation (and subsequent equations) are described in the table below, followed by an explanation of how the formula was derived.

Table 13 SCR Formula Terms

Term	Definition
ratm	ATM rate (SCR).
rrsvp	RSVP rate.
MPS	Minimum IP packet size, including the IP headers (300 bytes minimum).
DLE	Data-link encapsulation overhead. For RSVP ATM SVCs, ATM adaptation layer 5 (AAL5), Subnetwork Access Protocol (SNAP) encapsulation is used, which imposes a 5-byte encapsulation header on each protocol data unit (PDU).
%	Modulus operator. It yields the integer remainder from an integer division operation. For example, 57 % 53 results in 4.
CPS	Cell payload size. The total number of bytes in all the payloads of all the cells required to send a single packet with encapsulation.
UCO	Unused cell overhead (0 to 47).
COMP	Compensation factor. CPS divided by MPS.

There are two reasons for converting from RSVP rate to the ATM cell rate, as follows:

- To account for the ATM encapsulation header overhead and cell header overhead
- To account for the fact that ATM cell sizes are fixed

Because a portion of the last cell is unused, it is possible that a certain IP packet size requires more ATM cell layer bytes.

MPS + DLE is the length of the data packet that needs to be segmented into a number of fixed-length (48-byte payload) pieces that would then be put into a cell and sent.

Because the CPS needs to be greater than or equal to MPS + DLE, CPS must be larger than MPS.

CPS can be calculated as follows:

```
CPS = ceil((MPS + DLE)/48) * 48
```

where ceil(*x*) is the ceiling operator that returns the smallest integer greater than or equal to the real number *x* . Upon expanding the implementation of the ceil(*x*) operator, the expression can be arithmetically transformed into the following equation:

```
CPS = MPS + DLE + (MPS + DLE) % 48
```

where (MPS + DLE) % 48 yields the integer remainder when MPS + DLE is divided by 48. Because (MPS + DLE) % 48 is equal to the UCO, the equation for CPS can be rewritten as follows:

```
CPS = MPS + DLE + UCO
```

Because the IP bit rate was calculated by considering only the IP data and header (that is, packets of length MPS or larger), the IP bit rate (r rsvp) needs to be multiplied by COMP. According to the table above, COMP = CPS/MPS. Thus:

```
ATM cell payload bit rate =
r
rsvp * COMP =
r
rsvp * CPS/MPS
```

When expanded, the ATM cell payload bit rate is as follows:

```
ATM cell payload bit rate =
r
rsvp * (MPS + DLE + UCO)/MPS
```

Each ATM cell has a 5-byte header and a 48-byte payload, resulting in a 53-byte cell. Because the entire cell needs to be accounted for (not just the payload), we need to multiply the equation by a compensation factor of 53/48, which yields the desired equation:

```
r
atm =
r
rsvp * (53/48) * (MPS + DLE + UCO)/MPS
```

Thus, the SCR of the SVC created to carry the RSVP flow is calculated by the following formula:

```
r
atm =
r
rsvp * (53/48) * (MPS + DLE + (MPS + DLE) % 48)/MPS
```

The ATM peak cell rate (PCR) is derived using the same formula as the cell rate formula. It is either based on the maximum line rate of the ATM interface or on a configured maximum.

The maximum burst size of the SVC is derived by the following formula:

```
r
atm =
r
rsvp * (MPS + DLE + UCO)/(MPS * 48)
```

Note that the actual PCR, SCR, and MBS will be slightly larger than these formulas indicate.

See the task "Limiting the Peak Rate Applied to the PCR for SVCs, page 271" for information on setting the PCR of the ATM SVC.

Each new RSVP reservation causes establishment of a new SVC. If an existing reservation is refreshed, no new signalling is needed. If the reservation is not refreshed and it times out, the SVC is torn down. If the reservation is refreshed but the RSVP flowspec has changed, the existing SVC is torn down and a new one with the correct QoS parameters is established.

Limiting the Peak Rate Applied to the PCR for SVCs

To set a limit on the PCR of reservations for all new RSVP SVCs established on the current interface or any of its subinterfaces, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **ip rsvp atm-peak-rate-limit** *limit*	Configures the peak rate limit for new RSVP SVCs on an interface or subinterface.

For Controlled Load Service, the nominal peak rate is not defined and is taken as infinity. Consequently, the PCR is set to the available line rate. However, you can use the **ip rsvp atm-peak-rate-limit** command to further limit the PCR to a specific value on a per-interface basis.

Configuring per-VC DWRED

To configure Distributed Weighted Random Early Detection (DWRED) with per-VC DWRED enabled as a drop policy at the interface level for a specific DWRED group, use the following command in interface configuration mode:

Command	Purpose
`Router(config-if)#` **random-detect** [**attach** *group-name*]	Configures interface-level per-VC DWRED for a specific DWRED group.

The per SVC-DWRED drop policy ensures that packets that match reservations and conform to the appropriate token bucket have the highest priority. Attaching DWRED group definitions to the interface to support per-VC DWRED drop policy ensures that if packets must be dropped, then best-effort packets are dropped first and not those that conform to the appropriate QoS determined by the token bucket of the RSVP. This drop policy meets the loss requirements of controlled load called for by the Controlled Load Service class.

To meet the loss goals of controlled load, it is necessary to ensure that if packets must be dropped, best-effort packets are dropped first. Given that packets matching reservations and conforming to the appropriate token bucket will have the highest precedence, per-SVC DWRED is used as the drop policy.

Note In order to use per-SVC DWRED, dCEF must be configured on the router. For information on how to configure dCEF, refer to the *Cisco IOS Switching Services Configuration Guide* and the *Cisco IOS Switching Services Command Reference* .

Monitoring RSVP-ATM Configuration for an Interface

To display the peak rate limit for the interface, the IP Precedence and ToS bit values configured for packets that conform to and exceed the flowspec, and other RSVP-related information for the interface, such as whether the interface has been configured to establish SVCs to service reservation request messages and whether RSVP is enabled to attach itself to NetFlow, use the following commands in EXEC mode, as needed:

Command	Purpose
Router# **show ip rsvp atm-peak-rate-limit** [*interface*]	Displays the current peak rate limit set for an interface, if any.
Router# **show ip rsvp interface** [*interface-type interface-number*]	Displays RSVP-related interface information.
Router# **show ip rsvp** {**precedence** | **tos**} [*interface*]	Displays the IP Precedence bit values and type of service (ToS) bit values to be used to mark the ToS byte of the IP headers of all packets in an RSVP reserved path that conform to or exceed the RSVP flowspec for a given interface.

RSVP-ATM QoS Interworking Configuration Examples

This section provides RSPV-ATM QoS Internetworking configuration examples.

For information about configuring RSVP-ATM QoS Internetworking, see the" RSVP-ATM QoS Interworking Configuration Task List" section in this module.

The following example configures two Cisco 7500 series routers that connect over an ATM core network through a permanent virtual circuit (PVC) and multiple SVCs. As depicted in the figure below, Router A is connected to the ATM core network downstream; upstream it is connected across an Ethernet connection to the RSVP sender host system. Router B is connected upstream to the ATM core network and downstream across an Ethernet connection to the RSVP receiver host.

The example configuration shows three PVCs, two of which are required by ATM. One of the PVCs is used for RSVP-ATM QoS Interworking. It is used for transmission of best-effort traffic and to control traffic such as routing and RSVP messages. The ATM SVCs are established in response to reservation request messages in order to service those requests.

Figure 27 ***Example RSVP-ATM QoS Interworking Configuration***

Router A Configuration

The following portion of the example configures Router A in global configuration mode. It enables CEF, which must be turned on before the RSVP-ATM QoS Interworking feature can be enabled at the interface configuration level.

```
RouterA# config terminal
RouterA(config)# ip routing
RouterA(config)# ip cef
```

The following segment of the configuration for Router A configures ATM interface 2/1/0. The **ip route-cache flow**command enables NetFlow on the interface. If you do not enter the **ip rsvp**

bandwidthcommand before the **ip rsvp svc-required**command, a warning is issued requesting that you change the order of the commands.

The **ip rsvp bandwidth** command enables RSVP on the interface with default values for bandwidth allocation to RSVP. The **ip rsvp svc-required**command enables establishment of an SVC to service each new RSVP reservation on the interface. The **ip rsvp tos** and **ip rsvp precedence** commands configure conform and exceed values to be used for setting the ToS and IP Precedence bits of packets that either conform to or exceed the RSVP flowspec. (Note that once set, the ToS and IP Precedence bit values remain for the duration of the packet.)

You should configure the **ip route-cache flow** command only on the input interfaces of a router on whose output interfaces you configured the **ip rsvp svc-required** command.

```
RouterA(config)# interface ATM2/1/0
RouterA(config-if)# no shut
RouterA(config-if)# ip address 145.5.5.1 255.255.255.0
RouterA(config-if)# no ip proxy
RouterA(config-if)# no ip redirects
RouterA(config-if)# ip route-cache
RouterA(config-if)# ip mroute-cache
RouterA(config-if)# ip route-cache flow
RouterA(config-if)# no ip mroute-cache
RouterA(config-if)# ip route-cache cef
RouterA(config-if)# atm pvc 1 0 5 qsaal
RouterA(config-if)# atm pvc 2 0 16 ilmi
RouterA(config-if)# atm esi-address 111111111151.00
RouterA(config-if)# pvc pvc12 0/51
RouterA(config-if-atm-vc)# inarp 5
RouterA(config-if-atm-vc)# broadcast
RouterA(config-if-atm-vc)# exit
RouterA(config-if)# ip rsvp bandwidth
RouterA(config-if)# ip rsvp svc-required
RouterA(config-if)# ip rsvp tos conform 4
RouterA(config-if)# ip rsvp precedence conform 3 exceed 2
```

The following portion of the configuration configures Ethernet interface 0/1 on Router A that is used for the connection between the sender host and Router A. RSVP is enabled on the interface with default bandwidth allocations.

```
RouterA(config)# interface Ethernet0/1
RouterA(config-if)# ip address 145.1.1.1 255.255.255.0
RouterA(config-if)# no ip proxy
RouterA(config-if)# no ip redirects
RouterA(config-if)# no shut
RouterA(config-if)# ip route-cache
RouterA(config-if)# ip mroute-cache
RouterA(config-if)# ip route-cache flow
RouterA(config-if)# no ip mroute-cache
RouterA(config-if)# ip route-cache cef
RouterA(config-if)# fair
RouterA(config-if)# ip rsvp bandwidth
```

The following section displays configuration for Router A after the preceding commands were used to configure it:

```
RouterA# write terminal
 Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterA
boot system tftp rsp-jv-mz 171.69.209.28
enable password
!
```

```
ip subnet-zero
ip cef
interface Ethernet0/1
 ip address 145.1.1.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 7500 7500
 no ip route-cache cef
 no ip mroute-cache
 fair-queue 64 256 1000
!
interface ATM2/1/0
 ip address 145.5.5.1 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 112320 112320
 ip rsvp svc-required
 ip route-cache flow
 ip rsvp tos conform 4
 ip rsvp precedence conform 3 exceed 2
 no ip route-cache cef
 no ip route-cache distributed
 no ip mroute-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111151.00
 pvc pvc12 0/51
  inarp 5
  broadcast
!
```

Router B Configuration

Router B is configured similarly to Router A. In the following global configuration portion of the example, Router B is configured so that CEF is e enabled before the RSVP-ATM QoS Interworking feature can be enabled.

```
RouterB# config terminal
RouterB(config)# ip routing
RouterB(config)# ip cef
```

The following segment of the configuration for Router B configures ATM interface 3/0/0. The **ip rsvp bandwidth**command enables RSVP and the **ip route-cache flow**command enables NetFlow on the interface. The **ip rsvp svc-required**command enables the RSVP-ATM QoS Interworking feature, allowing for the establishment of an SVC to service each new RSVP reservation on the interface.

```
RouterB(config)# interface ATM3/0/0
RouterB(config-if)# atm pvc 1 0 5 qsaal
RouterB(config-if)# atm pvc 2 0 16 ilmi
RouterB(config-if)# atm esi-address 111111111152.00
RouterB(config-if)# pvc pvc12 0/52
RouterB(config-if-atm-vc)# inarp 5
RouterB(config-if-atm-vc)# broadcast
RouterB(config-if-atm-vc)# exit
RouterB(config-if)# ip rsvp bandwidth
RouterB(config-if)# ip route-cache flow
RouterB(config-if)# ip rsvp svc-required
```

The following portion of the configuration configures the Ethernet interface on Router B. This interface is used for the connection between the receiver host and Router B. RSVP is enabled on the interface.

```
RouterB(config)# interface Ethernet0/2
RouterB(config-if)# no shut
RouterB(config-if)# ip address 145.4.4.2 255.255.255.0
RouterB(config-if)# no ip proxy
```

```
RouterB(config-if)# no ip redirects
RouterB(config-if)# ip route-cache
RouterB(config-if)# ip mroute-cache
RouterB(config-if)# ip route-cache flow
RouterB(config-if)# no ip mroute-cache
RouterB(config-if)# ip route-cache cef
RouterB(config-if)# fair
RouterB(config-if)# ip rsvp bandwidth
RouterB(config-if)# end
RouterB(config)# ip routing
RouterB(config)# router eigrp 17
RouterB(config-router)# network 145.5.5.0
RouterB(config-router)# network 145.4.4.0
```

The following section displays configuration for Router B after the preceding commands were used to configure it:

```
RouterB# write terminal
Current configuration:
!
version 12.0
service timestamps debug uptime
service timestamps log uptime
no service password-encryption
!
hostname RouterB
!
!
boot system tftp rsp-jv-mz 171.69.209.28
enable password
!
ip subnet-zero
ip cef distributed
interface Ethernet0/2
 ip address 145.4.4.2 255.255.255.0
 no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 7500 7500
 ip route-cache flow
 no ip mroute-cache
 fair-queue 64 256 1000
!
interface ATM3/0/0
 ip address 145.5.5.2 255.255.255.0
no ip redirects
 no ip directed-broadcast
 no ip proxy-arp
 ip rsvp bandwidth 112320 112320
 ip rsvp svc-required
 ip route-cache flow
 no ip route-cache cef
 no ip route-cache distributed
 no ip mroute-cache
 atm pvc 1 0 5 qsaal
 atm pvc 2 0 16 ilmi
 atm esi-address 111111111152.00
 pvc pvc12 0/52
  inarp 5
  broadcast
 !
```

Configuring COPS for RSVP

This chapter describes the tasks for configuring the COPS for RSVP feature. Common Open Policy Service (COPS) is a protocol for communicating network traffic policy information to network devices. Resource Reservation Protocol (RSVP) is a means for reserving network resources--primarily bandwidth--to guarantee that applications sending end-to-end across the Internet will perform at the desired speed and quality.

For complete conceptual information, see the "Signalling Overview" in this book.

For a complete description of the COPS for RSVP commands in this chapter, refer to the Cisco IOS Quality of Service Solutions Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index, or search online.

Finding Support Information for Platforms and Cisco IOS and Catalyst OS Software Images

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

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

COPS for RSVP Configuration Task List

To configure COPS for RSVP, perform the tasks described in the following sections.

Specifying COPS Servers and Enabling COPS for RSVP

To specify COPS servers and enable COPS for RSVP, use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy cops servers 161.44.130.168 161.44.129.6**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy cops servers 161.44.130.168 161.44.129.6**	Tells the router to request RSVP policy decisions from the first server listed, and if that fails to connect, from the next server listed. Also enables a COPS-RSVP client on the router.

Restricting RSVP Policy to Specific Access Control Lists

To restrict RSVP policy to specific access control lists (ACLs), use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy cops 40 160 servers 161.44.130.164 161.44.129.2**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy cops 40 160 servers 161.44.130.164 161.44.129.2**	Tells the router to apply RSVP policy to messages that match ACLs 40 and 160, and specifies the servers for those sessions.

Rejecting Unmatched RSVP Messages

To reject unmatched RSVP messages, use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy default-reject**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy default-reject**	Tells the router to reject unmatched PATH and RESV messages, instead of just letting them pass through unadjudicated.

Confining Policy to PATH and RESV Messages

To confine policy to PATH and RESV messages, use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy cops minimal**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy cops minimal**	Tells the router to adjudicate only PATH and RESV messages, and to accept and pass onward PATH ERROR, RESV ERROR, and RESV CONFIRM messages.

Retaining RSVP Information After Losing Connection with the COPS Server

To retain RSVP information after losing connection with the COPS server, use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy cops timeout 600**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy cops timeout 600**	Tells the router to hold policy information for 10 minutes (600 seconds) while attempting to reconnect to a COPS server.

Reporting the Results of Outsourcing and Configuration Decisions

To report the results of outsourcing and configuration decisions, use the following commands beginning in interface configuration mode:

SUMMARY STEPS

1. Router(config-if)# **configure terminal**
2. Router(config)# **ip rsvp policy cops report-all**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config-if)# **configure terminal**	Enters global configuration mode.
Step 2	Router(config)# **ip rsvp policy cops report-all**	Tells the router to report to the Policy Decision Point (PDP) the success or failure of outsourcing and configuration decisions.

Verifying the Configuration

To verify the COPS for RSVP configuration, use the following commands in EXEC mode, as needed:

Command	Purpose
`Router#` **show cops servers**	Displays server addresses, port, state, keepalives, and policy client information.
`Router#` **show ip rsvp policy cops**	Displays policy server addresses, ACL IDs, and client/server connection status.
`Router#` **show ip rsvp policy**	Displays ACL IDs and their connection status.

COPS for RSVP Configuration Examples

Examples COPS Server Specified

The following example specifies the COPS server and enables COPS for RSVP on the server. Both of these functions are accomplished by using the **ip rsvp policy cops** command. By implication, the default settings for all remaining COPS for RSVP commands are accepted.

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy cops servers 161.44.130.168 161.44.129.6
Router(config)# exit
```

Example RSVP Behavior Customized

Once the COPS server has been specified and COPS for RSVP has been enabled, the remaining COPS for RSVP commands can be used to customize the COPS for RSVP behavior of the router. The following example uses the remaining COPS for RSVP commands to customize the RSVP behavior of the router:

```
Router# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)# ip rsvp policy cops 40 160 servers 161.44.130.168 161.44.129.6
Router(config)# ip rsvp policy default-reject
Router(config)# ip rsvp policy cops minimal
Router(config)# ip rsvp policy cops timeout 600
Router(config)# ip rsvp policy cops report-all
Router(config)# exit
```

Example Verification of the COPS for RSVP Configuration

The following examples display three views of the COPS for RSVP configuration on the router, which can be used to verify the COPS for RSVP configuration.

This example displays the policy server address, state, keepalives, and policy client information:

```
Router# show cops servers
COPS SERVER: Address: 161.44.135.172. Port: 3288. State: 0. Keepalive: 120 sec
Number of clients: 1. Number of sessions: 1.
    COPS CLIENT: Client type: 1.  State: 0.
```

This example displays the policy server address, the ACL ID, and the client/server connection status:

```
Router# show ip rsvp policy cops
COPS/RSVP entry. ACLs: 40 60
PDPs: 161.44.135.172
Current state: Connected
Currently connected to PDP 161.44.135.172, port 0
```

This example displays the ACL ID numbers and the status for each ACL ID:

```
Router# show ip rsvp policy
Local policy: Currently unsupported
COPS:
ACLs: 40 60 . State: CONNECTED.
ACLs: 40 160 . State: CONNECTING.
```

RSVP Aggregation

The RSVP Aggregation feature allows the Resource Reservation Protocol (RSVP) state to be reduced within an RSVP/DiffServ network by aggregating many smaller reservations into a single, larger reservation at the edge.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 RSVP Aggregation

You must configure at least two aggregating nodes (provider edge [PE] devices), one interior node (provider [P] device) and two end user nodes (customer edge [CE] devices) within your network.

You must configure your network to support the following Cisco IOS features:

- RSVP
- Class Based Weighted Fair Queuing (CBWFQ)
- RSVP Scalability Enhancements

Note You configure these features because Cisco IOS Release 12.2(33)SRC supports control plane aggregation only. Dataplane aggregation must be achieved by using the RSVP Scalability Enhancements.

Restrictions for RSVP Aggregation

Functionality Restrictions

The following functionality is not supported:

- Multilevel aggregation
- Multiple, adjacent aggregation regions
- Dynamic resizing of aggregate reservations
- Policing of end-to-end (E2E) reservations by the aggregator
- Policing of aggregate reservations by interior devices
- Differentiated Services Code Point (DSCP) marking by the aggregator
- Equal Cost Multiple Paths (ECMP) load-balancing within the aggregation region
- RSVP Fast Local Repair in case of a routing change resulting in a different aggregator or deaggregator, admission control is performed on E2E PATH refresh
- Multicast RSVP reservations
- RSVP policy servers including Common Open Policy Server (COPS)
- Dataplane aggregation

The following functionality is supported:

- Multiple, non-adjacent aggregation regions
- Control plane aggregation

Note RSVP/DiffServ using CBWFQ provides the dataplane aggregation.

Configuration Restrictions

- Sources should not send marked packets without an installed reservation.
- Sources should not send marked packets that exceed the reserved bandwidth.
- Sources should not send marked packets to a destination other than the reserved path.
- All RSVP capable devices within an aggregation region regardless of role must support the aggregation feature to recognize the RFC 3175 RSVP message formats properly.
- E2E reservations must be present to establish dynamic aggregates; aggregates cannot be established manually.
- Aggregates are established at a fixed bandwidth regardless of the number of current E2E reservations being aggregated.
- Aggregators and deaggregators must be paired to avoid blackholing of E2E reservations because of dynamic aggregate establishment.

Note

Blackholing means that the reservation is never established. If an E2E reservation crosses from an exterior to an interior interface, the E2E reservation turns into an RSVP-E2E-IGNORE protocol packet. If there is no corresponding deaggregator, a device where this RSVP-E2E-IGNORE reservation crosses an interior to an exterior interface, then the RSVP-E2E-IGNORE reservation is never restored to an E2E reservation. The RSVP-E2E-IGNORE reservation eventually reaches its destination, which is the RSVP receiver; however, the RSVP receiver does not know what to do with the RSVP-E2E-IGNORE reservation and discards the packet.

Information About RSVP Aggregation

Feature Overview of RSVP Aggregation

High Level Overview

The establishment of a single RSVP reservation requires a large amount of resources including memory allocated for the associated data structures, CPU for handling signaling messages, I/O operations for datapath programming, interprocess communication, and signaling message transmission.

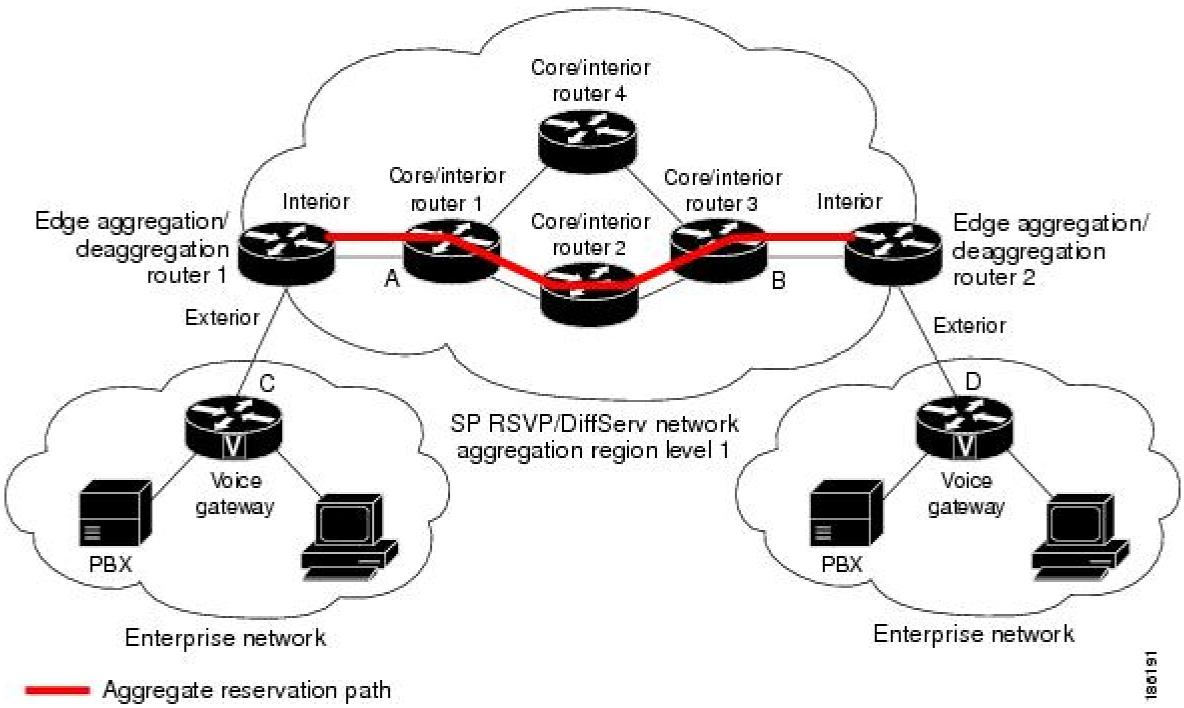
When a large number of small reservations are established, the resources required for setting and maintaining these reservations may exceed a node’s capacity to the point where the node’s performance is significantly degraded or it becomes unusable. The RSVP Aggregation feature addresses this scalability issue by introducing flow aggregation.

Flow aggregation is a mechanism wherein RSVP state can be reduced within a core device by aggregating many smaller reservations into a single, larger reservation at the network edge. This preserves the ability to perform connection admission control on core device links within the RSVP/DiffServ network while reducing signaling resource overhead.

How Aggregation Functions

Common segments of multiple end-to-end (E2E) reservations are aggregated over an aggregation region into a larger reservation that is called an aggregate reservation. An aggregation region is a connected set of nodes that are capable of performing RSVP aggregation as shown in the figure below.

Figure 28 RSVP Aggregation Network Overview



There are three types of nodes within an aggregation region:

- Aggregator--Aggregates multiple E2E reservations.
- Deaggregator--Deaggregates E2E reservations; provides mapping of E2E reservations onto aggregates.
- Interior--Neither aggregates or deaggregates, but is an RSVP core device that understands RFC 3175 formatted RSVP messages. Core/interior devices 1 through 4 are examples shown in the figure above.

There are two types of interfaces on the aggregator/deaggregator nodes:

- Exterior interface--The interface is not part of the aggregate region.
- Interior interface--The interface is part of the aggregate region.

Any device that is part of the aggregate region must have at least one interior interface and may have one or more exterior interfaces. Depending on the types of interfaces spanned by an IPv4 flow, a node can be an aggregator, a deaggregator, or an interior device with respect to that flow.

- Aggregate RSVP DiffServ Integration Topology, page 286

Aggregate RSVP DiffServ Integration Topology

RSVP aggregation further enhances RSVP scalability within an RSVP/DiffServ network as shown in the figure above by allowing the establishment of aggregate reservations across an aggregation region. This allows for aggregated connection admission control on core/interior device interfaces. Running RSVP on the core/interior devices allows for more predictable bandwidth use during normal and failure scenarios.

The voice gateways are running classic RSVP, which means RSVP is keeping a state per flow and also classifying, marking, and scheduling packets on a per-flow basis. The edge/aggregation devices are running

RSVP with scalability enhancements for admission control on the exterior interfaces connected to the voice gateways and running RSVP aggregation on the interfaces connected to core/interior devices 1 and 3. The core/interior devices in the RSVP/DiffServ network are running RSVP for the establishment of the aggregate reservations. The edge and core/interior devices inside the RSVP/DiffServ network also implement a specific per hop behavior (PHB) for a collection of flows that have the same DSCP.

The voice gateways identify voice data packets and set the appropriate DSCP in their IP headers so that the packets are classified into the priority class in the edge/aggregation devices and in core/interior devices 1, 2, 3 or 1, 4, 3.

The interior interfaces on the edge/aggregation/deaggregation devices (labeled A and B) connected to core/ interior devices 1 and 3 are running RSVP aggregation. They are performing admission control only per flow against the RSVP bandwidth of the aggregate reservation for the corresponding DSCP.

Admission control is performed at the deaggregator because it is the first edge node to receive the returning E2E RSVP RESV message. CBWFQ is performing the classification, policing, and scheduling functions on all nodes within the RSVP/DiffServ network including the edge devices.

Aggregate reservations are dynamically established over an aggregation region when an E2E reservation enters an aggregation region by crossing from an exterior to an interior interface; for example, when voice gateway C initiates an E2E reservation to voice gateway D. The aggregation is accomplished by "hiding" the E2E RSVP messages from the RSVP nodes inside the aggregation region. This is achieved with a new IP protocol, RSVP-E2E-IGNORE, that replaces the standard RSVP protocol in E2E PATH, PATHTEAR, and RESVCONF messages. This protocol change to RSVP-E2E-IGNORE is performed by the aggregator when the message enters the aggregation region and later restored back to RSVP by the deaggregator when the message exits the aggregation region. Thus, the aggregator and deaggregator pairs for a given flow are dynamically discovered during the E2E PATH establishment.

The deaggregator device 2 is responsible for mapping the E2E PATH onto an aggregate reservation per the configured policy. If an aggregate reservation with the corresponding aggregator device 1 and a DSCP is established, the E2E PATH is forwarded. Otherwise a new aggregate at the requisite DSCP is established, and then the E2E PATH is forwarded. The establishment of this new aggregate is for the fixed bandwidth parameters configured at the deaggregator device 2. Aggregate PATH messages are sent from the aggregator to the deaggregator using RSVP’s normal IP protocol. Aggregate RESV messages are sent back from the deaggregator to the aggregator, thus establishing an aggregate reservation on behalf of the set of E2E flows that use this aggregator and deaggregator. All RSVP capable interior nodes process the aggregate reservation request following normal RSVP processing including any configured local policy.

The RSVP-E2E-IGNORE messages are ignored by the core/interior devices, no E2E reservation states are created, and the message is forwarded as IP. As a consequence, the previous hop/next hop (PHOP/ NHOP) for each RSVP-E2E-IGNORE message received at the deaggregator or aggregator is the aggregator or deaggregator node. Therefore, all messages destined to the next or previous hop (RSVP error messages, for example) do not require the protocol to be changed when they traverse the aggregation region.

By setting up a small number of aggregate reservations on behalf of a large number of E2E flows, the number of states stored at core/interior devices and the amount of signal processing within the aggregation region is reduced.

In addition, by using differentiated services mechanisms for classification and scheduling of traffic supported by aggregate reservations rather than performing per aggregate reservation classification and scheduling, the amount of classification and scheduling state in the aggregation region is further reduced. This reduction is independent of the number of E2E reservations and the number of aggregate reservations in the aggregation region. One or more RSVP/DiffServ DSCPs are used to identify the traffic covered by aggregate reservations, and one or more RSVP/DiffServ per hop behaviors (PHBs) are used to offer the required forwarding treatment to this traffic. There may be more than one aggregate reservation between the same pair of devices, each representing different classes of traffic and each using a different DSCP and a different PHB.

Integration with RSVP Features

RSVP aggregation has been integrated with many RSVP features, including the following:

- RSVP Fast Local Repair
- RSVP Local Policy Support
- RSVP Refresh Reduction and Reliable Messaging

Benefits of RSVP Aggregation

Enhanced Scalability

Aggregating a large number of small reservations into one reservation requires fewer resources for signaling, setting, and maintaining the reservation thereby increasing scalability.

Enhanced Bandwidth Usage within RSVP/DiffServ Core Network

Aggregate reservations across an RSVP/DiffServ network allow for more predictable bandwidth use of core links across RSVP/DiffServ PHBs. Aggregate reservations can use RSVP fast local repair and local policy preemption features for determining bandwidth use during failure scenarios.

How to Configure RSVP Aggregation

Configuring RSVP Scalability Enhancements

Note All interfaces on nodes running Cisco IOS Release 12.2(33)SRC software must be configured with RSVP Scalability Enhancements.

Note Interior nodes only require RSVP Scalability Enhancements (RSVP/DiffServ) configuration. Interior nodes simply need to have RSVP/DiffServ configured and be running Cisco IOS Release 12.2(33)SRC with RSVP aggregation support to enable the nodes to process per normal RSVP processing rules RFC 3175 formatted messages properly. This is because Cisco IOS Release 12.2(33)SRC supports control plane aggregation only. Dataplane aggregation must be achieved by using the RSVP Scalability Enhancements.

Perform these tasks on all nodes within the aggregation region including aggregators, deaggregators, and interior nodes.

This section includes the following procedures:

Enabling RSVP on an Interface

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.

	Command or Action	Purpose
Step 4	**ip rsvp bandwidth** [*interface-kbps*][*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth 7500`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface that you want to enable.
Step 5	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Setting the Resource Provider

Note Resource provider was formerly called QoS provider.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface *type number*
4. ip rsvp resource-provider [none | wfq-interface | wfq-pvc]
5. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	interface *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	ip rsvp resource-provider [none | wfq-interface | wfq-pvc] **Example:** `Device(config-if)# ip rsvp resource-provider none`	Sets the resource provider. • Enter the optional **none** keyword to set the resource provider to none regardless of whether one is configured on the interface. **Note** Setting the resource provider to **none** instructs RSVP to *not* associate any resources, such as weighted fair queueing (WFQ) queues or bandwidth, with a reservation. • Enter the optional **wfq-interface** keyword to specify WFQ as the resource provider on the interface. • Enter the optional **wfq-pvc** keyword to specify WFQ as the resource provider on the permanent virtual circuit (PVC) or connection.
Step 5	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Disabling Data Packet Classification

Note Disabling data packet classification instructs RSVP not to process every packet, but to perform admission control only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface *type number*
4. ip rsvp data-packet classification none
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	ip rsvp data-packet classification none **Example:** `Device(config-if)# ip rsvp data-packet classification none`	Disables data packet classification.
Step 5	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring Class and Policy Maps

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. class-map [type {stack | access-control | port-filter | queue-threshold}] [match-all | match-any] class-map-name
4. **match access-group** {access-group | **name** access-group-name}
5. **exit**
6. policy-map [type access-control] policy-map-name
7. class {class-name | class-default}
8. priority {bandwidth-kbps | percent percentage} [burst]
9. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	class-map [type {stack | access-control | port-filter | queue-threshold}] [match-all | match-any] class-map-name **Example:** `Device(config)# class-map match-all voice`	Creates a class map to be used for matching packets to a specified class and enters class-map configuration mode. • The optional **type stack** keywords enable the flexible packet matching (FPM) functionality to determine the correct protocol stack in which to examine. **Note** If the appropriate protocol header description files (PHDFs) have been loaded onto the device (via the **load protocol**command), a stack of protocol headers can be defined so the filter can determine which headers are present and in what order. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest. **Note** You must specify a stack class map (via the **type stack** keywords) before you can specify an access-control class map (via the type **access-control** keywords). • The optional **type port-filter** keywords create a port-filter class-map that enables the TCP/UDP port policing of control plane packets. **Note** When enabled, these keywords provide filtering of traffic destined to specific ports on the control plane host subinterface. • The optional **type queue-threshold** keywords enable queue thresholding that limits the total number of packets for a specified protocol that is allowed in the control plane IP input queue. This feature applies only to control plane host subinterface. • The optional **match-all** | **match-any** keywords determine how packets are evaluated when multiple match criteria exist. Packets must either meet all of the match criteria (**match-all**) or one of the match criteria (**match-any**) in order to be considered a member of the class.
Step 4	**match access-group** {access-group | **name** access-group-name} **Example:** `Device(config-cmap)# match access-group 100`	Specifies the numbered access list against whose contents packets are checked to determine if they match the criteria specified in the class map. **Note** After you create the class map, you configure its match criteria. Here are some of the commands that you can use: • ◦ **match access-group** ◦ **match input-interface** ◦ **match mpls experimental** ◦ **match protocol**

	Command or Action	Purpose
Step 5	**exit** **Example:** `Device(config-cmap)# exit`	Exits to global configuration mode.
Step 6	policy-map [type access-control] policy-map-name **Example:** `Device(config)# policy-map wfq-voip`	Creates or modifies a policy map that can be attached to one or more interfaces to specify a service policy and enters policy-map configuration mode. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest.
Step 7	class {class-name | class-default} **Example:** `Device(config-pmap-c)# class voice`	Specifies the class so that you can configure or modify its policy. Enters policy-map class configuration mode. • Enter the *class name* or use the **class-default**keyword.
Step 8	priority {bandwidth-kbps | percent percentage} [burst] **Example:** `Device(config-pmap-c)# priority 24`	(Optional) Prioritizes a class of traffic belonging to a policy map. • The optional *burst* argument specifies the burst size in bytes. The burst size configures the network to accommodate temporary bursts of traffic. The default burst value, which is computed as 200 milliseconds of traffic at the configured bandwidth rate, is used when the *burst* argument is not specified. The range of the burst is from 32 to 2000000 bytes.
Step 9	**end** **Example:** `Device(config--pmap-c)#` **end**	(Optional) Returns to privileged EXEC mode.

Attaching a Policy Map to an Interface

Note If at the time you configure the RSVP scalability enhancements, there are existing reservations that use classic RSVP, no additional marking, classification, or scheduling is provided for these flows. You can also delete these reservations after you configure the RSVP scalability enhancements.

SUMMARY STEPS

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

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**service-policy** [**type access-control**] {**input** | **output**} policy-map-name **Example:** `Device(config-if)# service-policy output POLICY-ATM`	Specifies the name of the policy map to be attached to the input or output direction of the interface. **Note** Policy maps can be attached in the input or output direction of an interface. The direction and the device to which the policy map should be attached vary according to the network configuration. When using the **service-policy** command to attach the policy map to an interface, be sure to choose the device and the interface direction that are appropriate for the network configuration. • The optional **type access-control** keywords determine the exact pattern to look for in the protocol stack of interest. • Enter the *policy-map name*.
Step 5	**end** **Example:** `Device(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Configuring Interfaces with Aggregation Role

Perform this task on aggregator and deaggregators to specify which interfaces are facing the aggregation region.

Note You do not need to perform this task on interior devices; that is, nodes having interior interfaces only.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. interface *type number*
4. **ip rsvp aggregation role interior**
5. Repeat Step 4 as needed to configure additional aggregator and deaggregator interfaces.
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	interface *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp aggregation role interior** **Example:** `Device(config-if)# ip rsvp aggregation role interior`	Enables RSVP aggregation on an aggregator or deaggregator's interface.
Step 5	Repeat Step 4 as needed to configure additional aggregator and deaggregator interfaces.	Configures additional aggregator and deaggregator interfaces.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Aggregation Mapping on a Deaggregator

Note Typically, an edge device acts as both an aggregator and deaggregator because of the unidirectional nature of RSVP reservations. Most applications require bidirectional reservations. Therefore, these parameters are used by a deaggregator when mapping E2E reservations onto aggregates during the dynamic aggregate reservation process.

You should configure an access control list (ACL) to define a group of RSVP endpoints whose reservations will be aggregated onto a single aggregate reservation session identified by the specified DSCP. Then for each ACL, define a map configuration.

Note In classic (unaggregated) RSVP, a session is identified in the reservation message session object by the destination IP address and protocol information. In RSVP aggregation, a session is identified by the destination IP address and DSCP within the session object of the aggregate RSVP message. E2E reservations are mapped onto a particular aggregate RSVP session identified by the E2E reservation session object alone or a combination of the session object and sender template or filter spec.

Extended ACLs

The ACLs used within the **ip rsvp aggregation ip map** command match the RSVP message objects as follows for an extended ACL:

- Source IP address and port match the RSVP PATH message sender template or RSVP RESV message filter spec; this is the IP source or the RSVP sender.
- Destination IP address and port match the RSVP PATH/RESV message session object IP address; this is the IP destination address or the RSVP receiver.
- Protocol matches the RSVP PATH/RESV message session object protocol; if protocol = IP, then it matches the source or destination address as above.

Standard ACLs

The ACLs used within the **ip rsvp aggregation ip map** command match the RSVP message objects as follows for a standard ACL:

- IP address matches the RSVP PATH message sender template or RSVP RESV message filter spec; this is the IP source address or the RSVP sender.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip map** {**access-list** {*acl-number*} | **any**} **dscp** *value*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp aggregation ip map** {**access-list** {*acl-number*} | **any**} **dscp** *value* **Example:** `Device(config)#` **ip rsvp aggregation ip map any dscp af41**	Configures RSVP aggregation rules that tell a device how to map E2E reservations onto aggregate reservations. • The keywords and arguments specify additional information such as DSCP values.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring Aggregate Reservation Attributes on a Deaggregator

Perform this task on a deaggregator to configure the aggregate reservation attributes (also called token bucket parameters) on a per-DSCP basis.

Note Typically, an edge device acts as both an aggregator and deaggregator because of the unidirectional nature of RSVP reservations. Most applications require bidirectional reservations. Therefore, these parameters are used by a deaggregator when mapping E2E reservations onto aggregates during the dynamic aggregate reservation process.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip reservation dscp** *value* [**aggregator** *agg-ip-address*] **traffic-params static rate** *data-rate* [**burst** *burst-size*] [**peak** *peak-rate*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp aggregation ip reservation dscp** *value* [**aggregator** *agg-ip-address*] **traffic-params static rate** *data-rate* [**burst** *burst-size*] [**peak** *peak-rate*] **Example:** `Device(config)#` **ip rsvp aggregation ip reservation dscp af11 aggregator 10.10.10.10 traffic-params static rate 10 burst 8 peak 10**	Configures RSVP aggregate reservation attributes (also called token bucket parameters) on a per-DSCP basis. • The keywords and arguments specify additional information.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring an RSVP Aggregation Device ID

Perform this task on aggregators and deaggregators to configure an RSVP aggregation device ID.

Note Both aggregators and deaggregators need to be identified with a stable and routable IP address. This is the RFC 3175 device ID, which is also the IP address of the loopback interface with the lowest number. If there is no loopback interface configured or all those configured are down, then there will be no device ID assigned for the aggregating/deaggregating function and aggregate reservations will not be established.

Note The device ID may change if the associated loopback interface goes down or its IP address is removed. In this case, the E2E and aggregate sessions are torn down. If a new device ID is determined, new E2E and aggregate sessions will use the new device ID.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface loopback** *number*
4. **ip address** *ip-address subnet-mask/prefix*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface loopback** *number* **Example:** `Device(config)# interface loopback 1`	Creates a loopback interface and enters interface configuration mode. • Enter a value for the *number* argument. The range is 0 to 2147483647.

	Command or Action	Purpose
Step 4	**ip address** *ip-address subnet-mask/prefix* **Example:** `Device(config-if)# ip address 192.168.50.1 255.255.255.0`	Configures an IP address and subnet mask or prefix on the loopback interface.
Step 5	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Enabling RSVP Aggregation

Perform this task on aggregators and deaggregators to enable RSVP aggregation globally after you have completed all the previous aggregator and deaggregator configurations.

Note This task registers a device to receive RSVP-E2E-IGNORE messages. It is not necessary to perform this task on interior devices because they are only processing RSVP aggregate reservations. If you do so, you may decrease performance because the interior device will then unnecessarily process all the RSVP-E2E-IGNORE messages.

Note If you enable RSVP aggregation globally on an interior device, then you should configure all interfaces as interior.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp aggregation ip**
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp aggregation ip** **Example:** `Device(config)# ip rsvp aggregation ip`	Enables RSVP aggregation globally on an aggregator or deaggregator.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring RSVP Local Policy

Perform this task to apply a local policy to an RSVP aggregate reservation.

Note In classic (unaggregated) RSVP, a session is identified in the reservation message session object by the destination IP address and protocol information. In RSVP aggregation, a session is identified by the destination IP address and DSCP within the session object of the aggregate RSVP message. The **dscp-ip** keyword matches the DSCP within the session object.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp policy local {acl** *acl1 acl2...acl8*] | **dscp-ip** *value1* [*value2 ... value8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1 as2...as8* **}**
4. {**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** {**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*} **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]}
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp policy local {acl** *acl1 acl2...acl8*] | **dscp-ip** *value1* [*value2 ... value8*] | **default** | **identity** *alias1* [*alias2...alias4*] | **origin-as** *as1 as2...as8* **}** **Example:** `Device(config)# ip rsvp policy local` **dscp-ip 46**	Creates a local policy to determine how RSVP resources are used in a network and enters local policy configuration mode. • Enter the **dscp-ip** *value*keyword and argument combination to specify a DSCP for matching the session object DCSP within the aggregate reservations. Values can be the following: ◦ 0 to 63--Numerical. The default value is 0. ◦ af11 to af43--Assured forwarding (AF). ◦ cs1 to cs7--Type of service (ToS) precedence. ◦ default--Default DSCP. ◦ ef--Expedited Forwarding (EF). **Note** You must associate at least one DSCP with a DSCP-based policy. However, you can associate as many as eight.
Step 4	{**accept** | forward [**all** | **path**| **path-error** | **resv**| **resv-error**] | **default** | **exit** | **fast-reroute** | **local-override** | **maximum** {**bandwidth** [**group** *x*] [**single** *y*] | **senders** *n*} **preempt-priority** [**traffic-eng** *x*] *setup-priority* [*hold-priority*]} **Example:** `Device(config-rsvp-policy-local)# forward all`	(Optional) Defines the properties of the dscp-ip local policy that you are creating. (These are the submode commands.) **Note** This is an optional step. An empty policy rejects everything, which may be desired in some cases. See the **ip rsvp policy local** command for more detailed information on submode commands.
Step 5	**end** **Example:** `Device(config-rsvp-policy-local)# end`	(Optional) Exits local policy configuration mode and returns to privileged EXEC mode.

Verifying the RSVP Aggregation Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp aggregation ip** [**endpoints** | **interface** [*if-name*] | **map** [**dscp** *value*]| **reservation** [**dscp** *value*[**aggregator** *ip-address*]]
3. **show ip rsvp aggregation ip endpoints** [**role**{**aggregator**| **deaggregator**}] [*ip-address*] [**dscp** *value*] [**detail**]
4. **show ip rsvp** [**atm-peak-rate-limit**| **counters**| **host**| **installed**| **interface**| **listeners**| **neighbor**| **policy**| **precedence**| **request**| **reservation**| **sbm**| **sender**| **signalling**| **tos**]
5. **show ip rsvp reservation** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]]
6. **show ip rsvp sender** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]]
7. **show ip rsvp installed** [*interface-type interface-number*] [**detail**]
8. **show ip rsvp interface** [**detail**] [*interface-type interface-number*]
9. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp aggregation ip** [**endpoints** | **interface** [*if-name*] | **map** [**dscp** *value*]| **reservation** [**dscp** *value*[**aggregator** *ip-address*]] **Example:** `Device# show ip rsvp aggregation ip`	(Optional) Displays RSVP summary aggregation information. • The optional keywords and arguments display additional information.
Step 3	**show ip rsvp aggregation ip endpoints** [**role**{**aggregator**| **deaggregator**}] [*ip-address*] [**dscp** *value*] [**detail**] **Example:** `Device# show ip rsvp aggregation ip endpooints`	(Optional) Displays RSVP information about aggregator and deaggregator devices for currently established aggregate reservations. • The optional keywords and arguments display additional information.

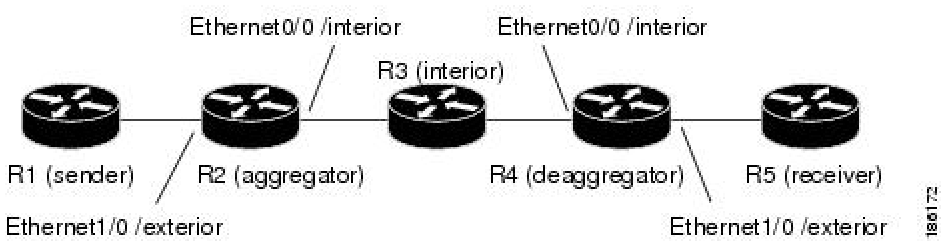
	Command or Action	Purpose
Step 4	**show ip rsvp** [**atm-peak-rate-limit**| **counters**| **host**| **installed**| **interface**| **listeners**| **neighbor**| **policy**| **precedence**| **request**| **reservation**| **sbm**| **sender**| **signalling**| **tos**] **Example:** `Device# show ip rsvp`	(Optional) Displays specific information for RSVP categories. • The optional keywords display additional information.
Step 5	**show ip rsvp reservation** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]] **Example:** `Device# show ip rsvp reservation detail`	(Optional) Displays RSVP-related receiver information currently in the database. • The optional keywords and arguments display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 6	**show ip rsvp sender** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]] **Example:** `Device# show ip rsvp sender detail`	(Optional) Displays RSVP PATH-related sender information currently in the database. • The optional keywords and arguments display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 7	**show ip rsvp installed** [*interface-type interface-number*] [**detail**] **Example:** `Device# show ip rsvp installed detail`	(Optional) Displays RSVP-related installed filters and corresponding bandwidth information. • The optional keywords and arguments display additional information.
Step 8	**show ip rsvp interface** [**detail**] [*interface-type interface-number*] **Example:** `Device# show ip rsvp interface detail`	(Optional) Displays RSVP-related interface information. • The optional keywords and arguments display additional information.
Step 9	**end** **Example:** `Device# end`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for RSVP Aggregation

Examples Configuring RSVP Aggregation

The figure below shows a five-device network in which RSVP aggregation is configured.

Figure 29 Sample RSVP Aggregation Network



Configuring RSVP/ DiffServ Attributes on an Interior Device

The following example configures RSVP/DiffServ attributes on an interior device (R3 in the figure above).

- Ethernet interface 0/0 is enabled for RSVP and the amount of bandwidth available for reservations is configured.
- A resource provider is configured and data packet classification is disabled because RSVP aggregation supports control plane aggregation only.

```
Device# configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface Ethernet0/0

Device(config-if)# ip rsvp bandwidth 400

Device(config-if)# ip rsvp resource-provider none

Device(config-if)# ip rsvp data-packet classification none

Device(config-if)# end
```

Configuring RSVP Aggregation on an Aggregator or Deaggregator

The following example configures RSVP aggregation attributes on an aggregator or deaggregator (R2 and R4 in Figure 2

- Loopback 1 is configured to establish an RSVP aggregation device ID.
- Ethernet interface 0/0 is enabled for RSVP and the amount of bandwidth available for reservations is configured.
- Ethernet interface 0/0 on an aggregator or deaggregator is configured to face an aggregation region.
- A resource provider is configured and data packet classification is disabled because RSVP aggregation supports control plane aggregation only.

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# interface Loopback 1
Device(config)# ip address 192.168.50.1 255.255.255.0
Device(config)# interface Ethernet0/0
Device(config-if)# ip rsvp bandwidth 400
Device(config-if)# ip rsvp aggregation role interior
Device(config-if)# ip rsvp resource-provider none
Device(config-if)# ip rsvp data-packet classification none
Device(config-if)# end
```

Configuring RSVP Aggregation Attributes and Parameters

The following example configures additional RSVP aggregation attributes, including a global rule for mapping all E2E reservations onto a single aggregate with DSCP AF41 and the token bucket parameters for aggregate reservations, because dynamic resizing is not supported. This configuration is only required on nodes performing the deaggregation function (R4 in the figure above).

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# ip rsvp aggregation ip map any dscp af41

Device(config)# ip rsvp aggregation ip reservation dscp af41 aggregator 10.10.10.10 traffic-
params static rate 10 burst 8 peak 10

Device(config)# end
```

Configuring an Access List for a Deaggregator

In the following example, access list 1 is defined for all RSVP messages whose RSVP PATH message sender template source address is in the 10.1.0.0 subnet so that the deaggregator (R4 in the figure above) maps those reservations onto an aggregate reservation for the DSCP associated with the AF41 PHB:

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# access-list 1 permit 10.1.0.0 0.0.255.255

Device(config)# ip rsvp aggregation ip map access-list 1 dscp af41

Device(config)# end
```

Configuring RSVP Aggregation

After you configure your RSVP aggregation attributes, you are ready to enable aggregation globally.

When you enable aggregation on a device, the device can act as an aggregator or a deaggregator. To perform aggregator and deaggregator functions, the RSVP process must see messages with the RSVP-E2E-IGNORE protocol type (134) on a device; otherwise, the messages are forwarded as data by the device's data plane. The **ip rsvp aggregation ip** command enables RSVP to identify messages with the RSVP-E2E-IGNORE protocol.

Note This registers a device to receive RSVP-E2E-IGNORE messages. It is not necessary to configure this command on interior nodes that are only processing RSVP aggregate reservations and forwarding RSVP-E2E-IGNORE messages as IP datagrams). Since the device is loaded with an image that supports aggregation, the device will process aggregate (RFC 3175 formatted) messages correctly. Enabling aggregation on an interior mode may decrease performance because the interior node will then unnecessarily process all RSVP-E2E-IGNORE messages.

Note If you enable aggregation on an interior node, you must configure all its interfaces as interior. Otherwise, all the interfaces have the exterior role, and any E2E PATH (E2E-IGNORE) messages arriving at the device are discarded.

In summary, there are two options for an interior device (R3 in the figure above):

- No RSVP aggregation configuration commands are entered.
- RSVP aggregation is enabled and all interfaces are configured as interior.

Configuring RSVP Local Policy

You can configure a local policy optionally on any RSVP capable node. In this example, a local policy is configured on a deaggregator to set the preemption priority values within the RSVP RESV aggregate messages based upon matching the DSCP within the aggregate RSVP messages session object. This allows the bandwidth available for RSVP reservations to be used first by reservations of DSCP EF over DSCP AF41 on interior or aggregation nodes. Any aggregate reservation for another DSCP will have a preemption priority of 0, the default.

Note Within the RSVP RESV aggregate message at the deaggregator, this local policy sets an RFC 3181 "Signaled Preemption Priority Policy Element" that can be used by interior nodes or the aggregator that has **ip rsvp preemption** enabled.

The following example sets the preemption priority locally for RSVP aggregate reservations during establishment on an interior device (R3 in the figure above):

```
Device# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Device(config)# ip rsvp policy local dscp-ip ef

Device(config-rsvp-local-policy)# 5 5

Device(config-rsvp-local-policy)# exit

Device(config)# ip rsvp policy local dscp-ip af41

Device(config-rsvp-local-policy)# 2 2

Device(config-rsvp-local-policy)# end
```

Example Verifying the RSVP Aggregation Configuration

This section contains the following verification examples:

Verifying RSVP Aggregation and Configured Reservations

The following example verifies that RSVP aggregation is enabled and displays information about the reservations currently established and configured map and reservation policies:

```
Device# show ip rsvp aggregation ip
RFC 3175 Aggregation:  Enabled
  Level: 1
  Default QoS service: Controlled-Load
  Number of signaled aggregate reservations:  2
  Number of signaled E2E reservations:        8
  Number of configured map commands:          4
  Number of configured reservation commands:  1
```

Verifying Configured Interfaces and Their Roles

The following example displays the configured interfaces and whether they are interior or exterior in regard to the aggregation region:

```
Device# show ip rsvp aggregation ip interface
Interface Name       Role
-------------------- --------
Ethernet0/0          interior
Serial2/0            exterior
Serial3/0            exterior
```

Verifying Aggregator and Deaggregator Reservations

The following example displays information about the aggregators and deaggregators when established reservations are present:

```
Device# show ip rsvp aggregation ip endpoints detail
Role  DSCP Aggregator      Deaggregator    State  Rate    Used    QBM PoolID
----- ---- --------------- --------------- ------ ------- ------- ----------
Agg   46   10.3.3.3         10.4.4.4         ESTABL 100K    100K    0x00000003
   Aggregate Reservation for the following E2E Flows (PSBs):
To             From           Pro DPort Sport  Prev Hop       I/F      BPS
10.4.4.4      10.1.1.1       UDP 1     1     10.23.20.3     Et1/0    100K
   Aggregate Reservation for the following E2E Flows (RSBs):
To             From           Pro DPort Sport  Next Hop       I/F      Fi Serv BPS
10.4.4.4      10.1.1.1       UDP 1     1     10.4.4.4       Se2/0    FF RATE 100K
   Aggregate Reservation for the following E2E Flows (Reqs):
To             From           Pro DPort Sport  Next Hop       I/F      Fi Serv BPS
10.4.4.4      10.1.1.1       UDP 1     1     10.23.20.3     Et1/0    FF RATE 100K
```

Additional References

The following sections provide references related to the RSVP Aggregation feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
Cisco IOS commands	Cisco IOS Master Commands List, All Releases
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Information on RSVP local policies	"RSVP Local Policy Support" module
Information on RSVP scalability enhancements	"RSVP Scalability Enhancements" module

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification*
RFC 2209	*Resource ReSerVation Protocol (RSVP)--Version 1 Message Processing Rules*
RFC 3175	*Aggregation of RSVP for IPv4 and IPv6 Reservations*
RFC 3181	*Signaled Preemption Priority Policy Element*
RFC 4804	Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels

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 RSVP Aggregation

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

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

Table 14 Feature Information for RSVP Aggregation

Feature Name	Releases	Feature Information
RSVP Aggregation	12.2(33)SRC	The RSVP Aggregation feature allows the Resource Reservation Protocol (RSVP) state to be reduced within an RSVP/DiffServ network by aggregating many smaller reservations into a single, larger reservation at the edge.

Glossary

admission control --The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

aggregate --AnRSVP flow that represents multiple end-to-end (E2E) flows; for example, a Multiprotocol Label Switching Traffic Engineering (MPLS-TE) tunnel may be an aggregate for many E2E flows.

aggregation region --An area where E2E flows are represented by aggregate flows, with aggregators and deaggregators at the edge; for example, an MPLS-TE core, where TE tunnels are aggregates for E2E flows. An aggregation region contains a connected set of nodes that are capable of performing RSVP aggregation.

aggregator --The device that processes the E2E PATH message as it enters the aggregation region. This device is also called the TE tunnel head-end device; it forwards the message from an exterior interface to an interior interface.

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

deaggregator --The device that processes the E2E PATH message as it leaves the aggregation region. This device is also called the TE tunnel tail-end device; it forwards the message from an interior interface to an exterior interface.

E2E --end-to-end. An RSVP flow that crosses an aggregation region, and whose state is represented in aggregate within this region, such as a classic RSVP unicast flow crossing an MPLS-TE core.

LSP --label-switched path. A configured connection between two devices, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications running on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

state --Information that a device must maintain about each LSP. The information is used for rerouting tunnels.

TE --traffic engineering. The techniques and processes used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel --Secure communications path between two peers, such as two devices.

MPLS TE-Tunnel-Based Admission Control

The MPLS TE--Tunnel-Based Admission Control (TBAC) feature enables classic Resource Reservation Protocol (RSVP) unicast reservations that are traveling across a Multiprotocol Label Switching Traffic Engineering (MPLS TE) core to be aggregated over an MPLS TE tunnel.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 MPLS TE-Tunnel-Based Admission Control

- You must configure an MPLS TE tunnel in the network.
- You must configure RSVP on one or more interfaces on at least two neighboring devices that share a link within the network.

Restrictions for MPLS TE-Tunnel-Based Admission Control

- Only IPv4 unicast RSVP flows are supported.
- Primary, one-hop tunnels are not supported. The TE tunnel cannot be a member of a class-based tunnel selection (CBTS) bundle.

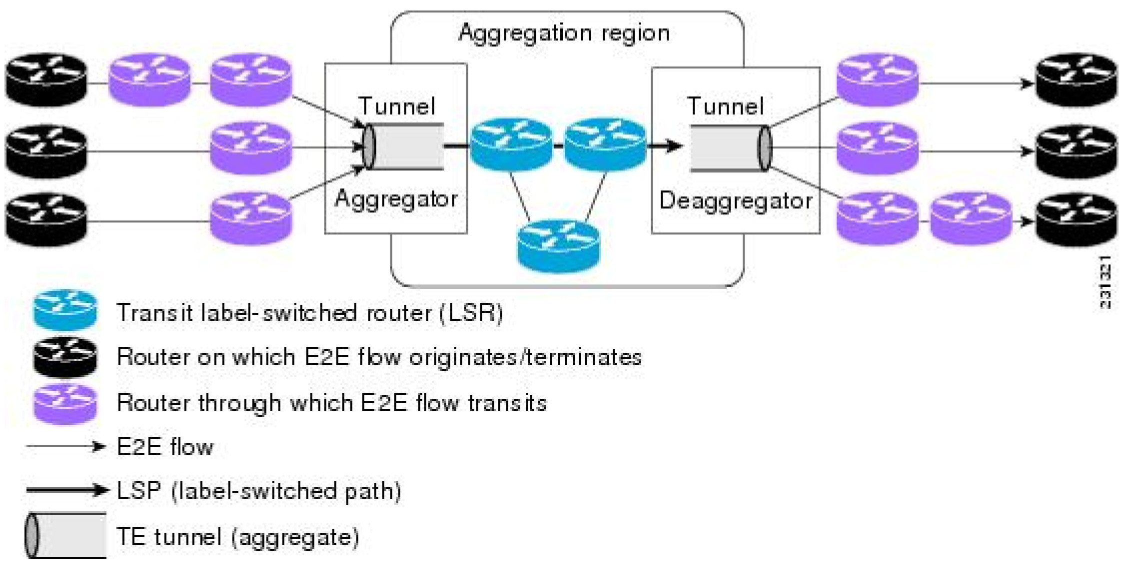
- Multi-Topology Routing (MTR) is not supported.
- Only preestablished aggregates are supported. They can be configured statically or dynamically using command-line interface (CLI) commands.
- This feature is supported on Cisco 7600 series devices only.

Information About MPLS TE-Tunnel-Based Admission Control

Feature Overview of MPLS TE-Tunnel-Based Admission Control

TBAC aggregates reservations from multiple, classic RSVP sessions over different forms of tunneling technologies that include MPLS TE tunnels, which act as aggregate reservations in the core. The figure below gives an overview of TBAC.

Figure 30 TBAC Overview



The figure above shows three RSVP end-to-end (E2E) flows that originate at devices on the far left, and terminate on devices at the far right. These flows are classic RSVP unicast flows, meaning that RSVP is maintaining a state for each flow. There is nothing special about these flows, except that along their path, these flows encounter an MPLS-TE core, where there is a desire to avoid creating a per-flow RSVP state.

When the E2E flows reach the edge of the MPLS-TE core, they are aggregated onto a TE tunnel. This means that when transiting through the MPLS-TE core, their state is represented by a single state; the TE tunnel is within the aggregation region, and their packets are forwarded (label-switched) by the TE tunnel. For example, if 100 E2E flows traverse the same aggregator and deaggregator, rather than creating 100 RSVP states (PATH and RESV messages) within the aggregation region, a single RSVP-TE state is created, that of the aggregate, which is the TE tunnel, to allocate and maintain the resources used by the 100 E2E flows. In particular, the bandwidth consumed by E2E flows within the core is allocated and maintained in aggregate by the TE tunnel. The bandwidth of each E2E flow is normally admitted into the

TE tunnel at the headend, just as any E2E flow's bandwidth is admitted onto an outbound link in the absence of aggregation.

Benefits of MPLS TE-Tunnel-Based Admission Control

To understand the benefits of TBAC, you should be familiar with how Call Admission Control (CAC) works for RSVP and QoS.

Cost Effective

Real-time traffic is very sensitive to loss and delay. CAC avoids QoS degradation for real-time traffic because CAC ensures that the accepted load always matches the current network capacity. As a result, you do not have to overprovision the network to compensate for absolute worst peak traffic or for reduced capacity in case of failure.

Improved Accuracy

CAC uses RSVP signaling, which follows the exact same path as the real-time flow, and devices make a CAC decision at every hop. This ensures that the CAC decision is very accurate and dynamically adjusts to the current conditions such as a reroute or an additional link. Also, RSVP provides an explicit CAC response (admitted or rejected) to the application, so that the application can react appropriately and fast; for example, sending a busy signal for a voice call, rerouting the voice call on an alternate VoIP route, or displaying a message for video on demand.

RSVP and MPLS TE Combined

TBAC allows you to combine the benefits of RSVP with those of MPLS TE. Specifically, you can use MPLS TE inside the network to ensure that the transported traffic can take advantage of Fast Reroute protection (50-millisecond restoration), Constraint Based Routing (CBR), and aggregate bandwidth reservation.

Seamless Deployment

TBAC allows you to deploy IPv4 RSVP without any impact on the MPLS part of the network because IPv4 RSVP is effectively tunneled inside MPLS TE tunnels that operate unchanged as per regular RSVP TE. No upgrade or additional protocol is needed in the MPLS core.

Enhanced Scaling Capability

TBAC aggregates multiple IPv4 RSVP reservations ingressing from the same MPLS TE headend device into a single MPLS TE tunnel and egressing from the same MPLS TE tailend device.

How to Configure MPLS TE-Tunnel-Based Admission Control

Enabling RSVP QoS

Perform this task to enable RSVP QoS globally on a device.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip rsvp qos**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ip rsvp qos** **Example:** `Device(config)# ip rsvp qos`	Enables RSVP QoS globally on a device.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Enabling MPLS TE

Perform this task to enable MPLS TE globally on a device that is running RSVP QoS.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **mpls traffic-eng tunnels**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**mpls traffic-eng tunnels** **Example:** `Device(config)# mpls traffic-eng tunnels`	Enables MPLS TE globally on a device.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring an MPLS TE Tunnel Interface

Perform this task to configure MPLS-TE tunneling on an interface.

You must configure an MPLS-TE tunnel in your network before you can proceed. For detailed information, see the "MPLS Traffic Engineering (TE)--Automatic Bandwidth Adjustment for TE Tunnels" module.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel number**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel number** **Example:** `Device(config)# interface tunnel1`	Specifies a tunnel interface and enters interface configuration mode.
Step 4	**end** **Example:** `Device(config-if)# end`	(Optional) Returns to privileged EXEC mode.

Configuring RSVP Bandwidth on an MPLS TE Tunnel Interface

Perform this task to configure RSVP bandwidth on the MPLS TE tunnel interface that you are using for the aggregation.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel number**
4. ip rsvp bandwidth [*interface-kbps*] [*single-flow-kbps*]
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 tunnel number** **Example:** `Router(config)# interface tunnel1`	Specifies a tunnel interface and enters interface configuration mode.
Step 4	ip rsvp bandwidth [*interface-kbps*] [*single-flow-kbps*] **Example:** `Router(config-if)# ip rsvp bandwidth 7500`	Enables RSVP bandwidth on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** You must enter a value for the *interface-kbps*argument on a tunnel interface.
Step 5	**end** **Example:** `Router(config-if)#` **end**	(Optional) Returns to privileged EXEC mode.

Verifying the TBAC Configuration

Note You can use the following **show** commands in user EXEC or privileged EXEC mode.

SUMMARY STEPS

1. **enable**
2. **show ip rsvp** [**atm-peak-rate-limit**| **counters**| **host**| **installed**| **interface**| **listeners**| **neighbor**| **policy**| **precedence**| **request**| **reservation**| **sbm**| **sender**| **signalling**| **tos**]
3. **show ip rsvp reservation** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]]
4. **show ip rsvp sender** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*][**src-port** *port-number*]]
5. **show mpls traffic-eng link-management bandwidth-allocation** [*interface-name* | **summary** [*interface-name*]]
6. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	(Optional) Enables privileged EXEC mode. • Enter your password if prompted. **Note** Skip this step if you are using the **show** commands in user EXEC mode.
Step 2	**show ip rsvp** [**atm-peak-rate-limit**| **counters**| **host**| **installed**| **interface**| **listeners**| **neighbor**| **policy**| **precedence**| **request**| **reservation**| **sbm**| **sender**| **signalling**| **tos**] **Example:** `Device# show ip rsvp`	Displays specific information for RSVP categories. • The optional keywords display additional information.
Step 3	**show ip rsvp reservation** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Device# show ip rsvp reservation detail`	Displays RSVP-related receiver information currently in the database. • The optional keywords display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.
Step 4	**show ip rsvp sender** [**detail**] [**filter**[**destination** *ip-address* | *hostname*] [**dst-port** *port-number*] [**source** *ip-address* | *hostname*] [**src-port** *port-number*]] **Example:** `Device# show ip rsvp sender detail`	Displays RSVP PATH-related sender information currently in the database. • The optional keywords display additional information. **Note** The optional **filter** keyword is supported in Cisco IOS Releases 12.0S and 12.2S only.

	Command or Action	Purpose
Step 5	**show mpls traffic-eng link-management bandwidth-allocation** [*interface-name* | **summary** [*interface-name*]] **Example:** `Device# show mpls traffic-eng link-management bandwidth-allocation`	Displays current local link information. • The optional keywords display additional information.
Step 6	**exit** **Example:** `Device# exit`	(Optional) Exits privileged EXEC mode and returns to user EXEC mode.

Configuration Examples for MPLS TE-Tunnel-Based Admission Control

Example Configuring TBAC

Note

You must have an MPLS TE tunnel already configured in your network. For detailed information, see the "MPLS Traffic Engineering (TE)--Automatic Bandwidth Adjustment for TE Tunnels" module.

The following example enables RSVP and MPLS TE globally on a device and then configures a tunnel interface and bandwidth of 7500 kbps on the tunnel interface traversed by the RSVP flows:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Device(config)# ip rsvp qos

Device(config)# mpls traffic-eng tunnels

Device(config)# interface tunnel1

Device(config-if)# ip rsvp bandwidth 7500

Device(config-if)# end
```

Example Configuring RSVP Local Policy on a Tunnel Interface

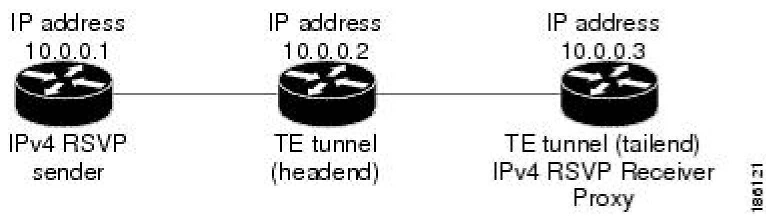
The following example configures an RSVP default local policy on a tunnel interface:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.

Device(config)# interface tunnel1

Device(config-if)# ip rsvp policy local default

Device(config-rsvp-local-if-policy)# max bandwidth single 10

Device(config-rsvp-local-if-policy)# forward all

Device(config-rsvp-local-if-policy)# end
```

Example Verifying the TBAC Configuration

The figure below shows a network in which TBAC is configured.

Figure 31 *Sample TBAC Network*



The following example verifies that RSVP and MPLS TE are enabled and coexist on the headend device (10.0.0.2 in the figure above):

```
Device# show ip rsvp
RSVP: enabled (on 3 interface(s))
  RSVP QoS enabled <-------
  MPLS/TE signalling enabled <------
Signalling:
   Refresh interval (msec): 30000
   Refresh misses: 4
...
```

The following example verifies that RSVP and MPLS TE are enabled and coexist on the tailend device (10.0.0.3 in the figure above):

```
Device# show ip rsvp
RSVP: enabled (on 3 interface(s))
  RSVP QoS enabled <-------
  MPLS/TE signalling enabled <------
Signalling:
   Refresh interval (msec): 30000
   Refresh misses: 4
...
```

The following examples verify that an IPv4 flow is traveling through a TE tunnel (a label-switched path [LSP]) on the headend device (10.0.0.2 in the figure above):

```
Device# show ip rsvp sender
To              From          Pro DPort Sport Prev Hop       I/F      BPS
10.0.0.3        10.0.0.1      UDP 2     2     10.0.0.1       Et0/0    10K <-- IPv4 flow
10.0.0.3        10.0.0.2      0   1     11    none           none     100K <-- TE tunnel


Device# show ip rsvp reservation
To            From        Pro DPort Sport Next Hop     I/F    Fi Serv BPS
10.0.0.3      10.0.0.1    UDP 2     2     10.0.0.3     Tu1    SE RATE 10K <-- IPv4 flow
10.0.0.3      10.0.0.2    0   1     11    10.1.0.2     Et1/0  SE LOAD 100K <-- TE tunnel
```

The following examples verify that an IPv4 flow is traveling through a TE tunnel (LSP) on the tailend device (10.0.0.3 in the figure above):

```
Device# show ip rsvp sender
To              From          Pro DPort Sport Prev Hop       I/F      BPS
10.0.0.3        10.0.0.1      UDP 2     2     10.0.0.2       Et1/0    10K <-- IPv4 flow
10.0.0.3        10.0.0.2      0   1     11    10.1.0.1       Et1/0    100K <-- TE tunnel


Device# show ip rsvp reservation
To            From          Pro DPort Sport Next Hop   I/F    Fi Serv BPS
10.0.0.3      10.0.0.1      UDP 2     2     none       none   SE RATE 10K <-- IPv4 flow
10.0.0.3      10.0.0.2      0   1     11    none       none   SE LOAD 100K <-- TE tunnel
```

The following examples display detailed information about the IPv4 flow and the TE tunnel (LSP) on the headend device (10.0.0.2 in the figure above):

```
Device# show ip rsvp sender detail
PATH: <---------------------------------------------- IPv4 flow information begins here.
  Destination 10.0.0.3, Protocol_Id 17, Don't Police , DstPort 2
  Sender address: 10.0.0.1, port: 2
  Path refreshes:
    arriving: from PHOP 10.0.0.10 on Et0/0 every 30000 msecs. Timeout in 189 sec
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  Path ID handle: 02000412.
  Incoming policy: Accepted. Policy source(s): Default
  Status:
  Output on Tunnel1, out of band. Policy status: Forwarding. Handle: 0800040E <--- TE
tunnel verified
    Policy source(s): Default
  Path FLR: Never repaired
PATH: <----------------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Path refreshes:
    sent:     to   NHOP 10.1.0.2 on Ethernet1/0
  ...


Device# show ip rsvp reservation detail
RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1,<--- IPv4 flow information
begins here.
  Protocol is UDP, Destination port is 2, Source port is 2
  Next Hop: 10.0.0.3 on Tunnel1, out of band <-------------------- TE tunnel verified
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  ...
Reservation: <---------------------------------------- TE Tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Next Hop: 10.1.0.2 on Ethernet1/0
  Label: 0 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...
```

```
Device# show ip rsvp installed detail

RSVP: Ethernet0/0 has no installed reservations

RSVP: Ethernet1/0 has the following installed reservations
RSVP Reservation. Destination is 10.0.0.3. Source is 10.0.0.2,
  Protocol is 0  , Destination port is 1, Source port is 11
  Traffic Control ID handle: 03000405
  Created: 04:46:55 EST Fri Oct 26 2007 <------
 IPv4 flow information
  Admitted flowspec:
    Reserved bandwidth: 100K bits/sec, Maximum burst: 1K bytes, Peak rate: 100K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 1500 bytes
  Resource provider for this flow: None
  ...
RSVP: Tunnel1 has the following installed reservations <------ TE tunnel verified
RSVP Reservation. Destination is 10.0.0.3. Source is 10.0.0.1,
  Protocol is UDP, Destination port is 2, Source port is 2
  Traffic Control ID handle: 01000415
  Created: 04:57:07 EST Fri Oct 26 2007 <-----
 IPv4 flow information
  Admitted flowspec:
    Reserved bandwidth: 10K bits/sec, Maximum burst: 10K bytes, Peak rate: 10K bits/sec
    Min Policed Unit: 0 bytes, Max Pkt Size: 0 bytes
  Resource provider for this flow: None
  ...
```

```
Device# show ip rsvp interface detail


 Et0/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
     ...
 Et1/0:
   RSVP: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 0 bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
    ...
Tu1: <---------------------------------- TE tunnel information begins here.
   RSVP: Enabled
   RSVP aggregation over MPLS TE: Enabled
   Interface State: Up
   Bandwidth:
     Curr allocated: 20K bits/sec
     Max. allowed (total): 3M bits/sec
     Max. allowed (per flow): 3M bits/sec
     ...
```

The following examples display detailed information about the IPv4 flow and the TE tunnel (LSP) on the tailend device (10.0.0.3 in the figure above):

```
Device# show ip rsvp sender detail
PATH: <----------------------------------------------- IPv4 flow information begins here.
  Destination 10.0.0.3, Protocol_Id 17, Don't Police , DstPort 2
  Sender address: 10.0.0.1, port: 2
  Path refreshes:
    arriving: from PHOP 10.0.0.2 on Et1/0 every 30000 msecs, out of band. Timeout in 188
sec
  Traffic params - Rate: 10K bits/sec, Max. burst: 10K bytes
    Min Policed Unit: 0 bytes, Max Pkt Size 2147483647 bytes
  ...
```

```
PATH: <---------------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Path refreshes:
    arriving: from PHOP 10.1.0.1 on Et1/0 every 30000 msecs. Timeout in 202 sec
  ...


Device# show ip rsvp reservation detail

RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1, <--- IPv4 flow information
begins here.
  Protocol is UDP, Destination port is 2, Source port is 2
  Next Hop: none
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  ...

Reservation: <-------------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Next Hop: none
  Label: 1 (outgoing)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...


Device# show ip rsvp request detail

RSVP Reservation. Destination is 10.0.0.3, Source is 10.0.0.1,
  Protocol is UDP, Destination port is 2, Source port is 2
  Prev Hop: 10.0.0.2 on Ethernet1/0, out of band <-------------- TE tunnel verified
  Reservation Style is Shared-Explicit, QoS Service is Guaranteed-Rate
  Average Bitrate is 10K bits/sec, Maximum Burst is 10K bytes
  ...

Request: <----------------------------------- TE tunnel information begins here.
  Tun Dest:   10.0.0.3  Tun ID: 1  Ext Tun ID: 10.0.0.2
  Tun Sender: 10.0.0.2  LSP ID: 11
  Prev Hop: 10.1.0.1 on Ethernet1/0
  Label: 0 (incoming)
  Reservation Style is Shared-Explicit, QoS Service is Controlled-Load
  ...
```

Example Verifying the RSVP Local Policy Configuration

The following example verifies that a default local policy has been configured on tunnel interface 1:

```
Device# show run interface tunnnel 1
Building configuration...

Current configuration : 419 bytes
!
interface Tunnel1
 bandwidth 3000
 ip unnumbered Loopback0
 tunnel destination 10.0.0.3
 tunnel mode mpls traffic-eng
 tunnel mpls traffic-eng autoroute announce
 tunnel mpls traffic-eng priority 1 1
 tunnel mpls traffic-eng bandwidth 100
 tunnel mpls traffic-eng path-option 1 dynamic
 tunnel mpls traffic-eng fast-reroute
 ip rsvp policy local default <---------------- Local policy information begins here.
  max bandwidth single 10
  forward all
 ip rsvp bandwidth 3000
end


The following example provides additional information about the default local policy
configured on tunnel interface 1:
```

```
Device# show ip rsvp policy local detail
Tunnel1:
  Default policy:

    Preemption Scope: Unrestricted.
    Local Override:   Disabled.
    Fast ReRoute:     Accept.
    Handle:           BC000413.

                           Accept                Forward
    Path:                  Yes                   Yes
    Resv:                  Yes                   Yes
    PathError:             Yes                   Yes
    ResvError:             Yes                   Yes

                           Setup Priority        Hold Priority
    TE:                    N/A                   N/A
    Non-TE:                N/A                   N/A

                           Current               Limit
    Senders:               0                     N/A
    Receivers:             1                     N/A
    Conversations:         1                     N/A
    Group bandwidth (bps): 10K                   N/A
    Per-flow b/w (bps):    N/A                   10K

Generic policy settings:
    Default policy: Accept all
    Preemption:     Disabled
```

Additional References

The following sections provide references related to the MPLS TE Tunnel-Based Admission Control (TBAC) feature.

Related Documents

Related Topic	Document Title
RSVP commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	*Cisco IOS Quality of Service Solutions Command Reference*
QoS features including signaling, classification, and congestion management	"Quality of Service Overview" module
Cisco IOS commands	Cisco IOS Master Commands List, All Releases

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 IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: http://www.cisco.com/go/mibs

RFCs

RFC	Title
RFC 2205	*Resource ReSerVation Protocol (RSVP)--Version 1 Functional Specification*
RFC 2209	*Resource ReSerVation Protocol (RSVP)--Version 1 Message Processing Rules*
RFC 3175	*Aggregation of RSVP for IPv4 and IPv6 Reservations*
RFC 3209	*RSVP-TE: Extensions to RSVP for LSP Tunnels*
RFC 4804	*Aggregation of Resource ReSerVation Protocol (RSVP) Reservations over MPLS TE/DS-TE Tunnels*

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 MPLS TE-Tunnel-Based Admission Control

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 15 Feature Information for MPLS TE--Tunnel-Based Admission Control (TBAC)

Feature Name	Releases	Feature Information
MPLS TE Tunnel-Based Admission Control (TBAC)	12.2(33)SRC	The MPLS TE--Tunnel-Based Admission Control (TBAC) feature enables classic Resource Reservation Protocol (RSVP) unicast reservations that are traveling across a Multiprotocol Label Switching Traffic Engineering (MPLS TE) core to be aggregated over an MPLS TE tunnel.

Glossary

admission control --The process by which an RSVP reservation is accepted or rejected on the basis of end-to-end available network resources.

aggregate--An RSVP flow that represents multiple E2E flows; for example, an MPLS-TE tunnel may be an aggregate for many E2E flows.

aggregation region --A area where E2E flows are represented by aggregate flows, with aggregators and deaggregators at the edge; for example, an MPLS-TE core, where TE tunnels are aggregates for E2E flows. An aggregation region contains a connected set of nodes that are capable of performing RSVP aggregation.

aggregator --The device that processes the E2E PATH message as it enters the aggregation region. This device is also called the TE tunnel headend device; it forwards the message from an exterior interface to an interior interface.

bandwidth --The difference between the highest and lowest frequencies available for network signals. The term is also used to describe the rated throughput capacity of a given network medium or protocol.

deaggregator --The device that processes the E2E PATH message as it leaves the aggregation region. This device is also called the TE tunnel tailend device; it forwards the message from an interior interface to an exterior interface.

E2E --end-to-end. An RSVP flow that crosses an aggregation region and whose state is represented in aggregate within this region; for example, a classic RSVP unicast flow that crosses an MPLS-TE core.

LSP --label-switched path. A configured connection between two devices, in which label switching is used to carry the packets. The purpose of an LSP is to carry data packets.

MPLS --Multiprotocol Label Switching. Packet-forwarding technology, used in the network core, that applies data link layer labels to tell switching nodes how to forward data, resulting in faster and more scalable forwarding than network layer routing normally can do.

QoS --quality of service. A measure of performance for a transmission system that reflects its transmission quality and service availability.

RSVP --Resource Reservation Protocol. A protocol that supports the reservation of resources across an IP network. Applications that run on IP end systems can use RSVP to indicate to other nodes the nature (bandwidth, jitter, maximum burst, and so on) of the packet streams that they want to receive.

state --Information that a device must maintain about each LSP. The information is used for rerouting tunnels.

TE --traffic engineering. The techniques and processes that are used to cause routed traffic to travel through the network on a path other than the one that would have been chosen if standard routing methods had been used.

tunnel --Secure communications path between two peers, such as two devices.

Configuring Subnetwork Bandwidth Manager

This chapter describes the tasks for configuring the Subnetwork Bandwidth Manager (SBM) feature, which is a signalling feature that enables Resource Reservation Protocol (RSVP)-based admission control over IEEE 802-styled networks.

For complete conceptual information, see "Signalling Overview" module.

For a complete description of the SBM commands in this chapter, see the Cisco IOS Quality of Service Solutions Command Reference. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

Use Cisco Feature Navigator to find information about platform support and software image support. Cisco Feature Navigator enables you to determine which Cisco IOS and Catalyst OS software images support a specific software release, feature set, or platform. To access Cisco Feature Navigator, go to http://www.cisco.com/go/cfn . An account on Cisco.com is not required.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

Subnetwork Bandwidth Manager Configuration Task List

To configure SBM, perform the tasks described in the following sections. The task in the first section is required; the tasks in the remaining sections are optional.

- Configuring an Interface as a Designated SBM Candidate, page 334 (Required)
- Configuring the NonResvSendLimit Object, page 334 (Optional)
- Verifying Configuration of SBM State, page 335 (Optional)

- Verifying Configuration of SBM State, page 335

Configuring an Interface as a Designated SBM Candidate

SBM is used in conjunction with RSVP. Therefore, before you configure an interface as a Designated SBM (DSBM) contender, ensure that RSVP is enabled on that interface.

To configure the interface as a DSBM candidate, use the following command in interface configuration mode:

Command	Purpose
`Device(config-if)#` **ip rsvp dsbm candidate** [*priority*]	Configures the interface to participate as a contender in the DSBM dynamic election process, whose winner is based on the highest priority.

Configuring the NonResvSendLimit Object

The NonResvSendLimit object specifies how much traffic can be sent onto a managed segment without a valid RSVP reservation.

To configure the NonResvSendLimit object parameters, use the following commands in interface configuration mode, as needed:

Command	Purpose
`Device(config-if)#` **ip rsvp dsbm non-resv-send-limit rate** *kBps*	Configures the average rate, in kbps, for the DSBM candidate.
`Device(config-if)#` **ip rsvp dsbm non-resv-send-limit burst** *kilobytes*	Configures the maximum burst size, in KB, for the DSBM candidate.
`Device(config-if)#` **ip rsvp dsbm non-resv-send-limit peak** *kBps*	Configures the peak rate, in kbps, for the DSBM candidate.
`Device(config-if)#` **ip rsvp dsbm non-resv-send-limit min-unit** *bytes*	Configures the minimum policed unit, in bytes, for the DSBM candidate.
`Device(config-if)#` **ip rsvp dsbm non-resv-send-limit max-unit** *bytes*	Configures the maximum packet size, in bytes, for the DSBM candidate.

To configure the per-flow limit on the amount of traffic that can be sent without a valid RSVP reservation, configure the **rate**, **burst**, **peak**, **min-unit**, and **max-unit** keywords for finite values from 0 to infinity.

To allow all traffic to be sent without a valid RSVP reservation, configure the **rate**, **burst**, **peak**, **min-unit**, and **max-unit** keywords for unlimited. To configure the parameters for unlimited, you can either omit the command or enter the **no** version of the command (for example, **no ip rsvp dsbm non-resv-send-limit rate**). Unlimited is the default value.

The absence of the NonResvSendLimit object allows any amount of traffic to be sent without a valid RSVP reservation.

Verifying Configuration of SBM State

To display information that enables you to determine if an interface has been configured as a DSBM candidate and which of the contenders has been elected the DSBM, use the following command in EXEC mode:

Command	Purpose
Device# **show ip rsvp sbm** [**detail**] [*interface*]	Displays information about an SBM configured for a specific RSVP-enabled interface or for all RSVP-enabled interfaces on the device. Using the **detail** keyword allows you to view the values for the NonResvSendLimit object.

The displayed output from the **show ip rsvp sbm** command identifies the interface by name and IP address, and it shows whether the interface has been configured as a DSBM contender. If the interface is a contender, the DSBM Priority field displays its priority. The DSBM election process is dynamic, addressing any new contenders configured as participants. Consequently, at any given time, an incumbent DSBM might be replaced by one configured with a higher priority. The following example shows sample output from the **show ip rsvp sbm** command:

```
Device# show ip rsvp sbm
Interface DSBM Addr      DSBM Priority    DSBM Candidate   My Priority
Et1       1.1.1.1        70               yes              70
Et2       145.2.2.150    100              yes              100
```

If you use the **detail** keyword, the output is shown in a different format. In the left column, the local DSBM candidate configuration is shown; in the right column, the corresponding information for the current DSBM is shown. In the following example, the local DSBM candidate won election and is the current DSBM:

```
Device# show ip rsvp sbm detail
Interface:Ethernet2
Local Configuration               Current DSBM
  IP Address:10.2.2.150           IP Address:10.2.2.150
  DSBM candidate:yes               I Am DSBM:yes
  Priority:100                     Priority:100
  Non Resv Send Limit               Non Resv Send Limit
    Rate:500 Kbytes/sec              Rate:500 Kbytes/sec
    Burst:1000 Kbytes                Burst:1000 Kbytes
    Peak:500 Kbytes/sec              Peak:500 Kbytes/sec
    Min Unit:unlimited               Min Unit:unlimited
    Max Unit:unlimited               Max Unit:unlimited
```

Example Subnetwork Bandwidth Manager Candidate Configuration

In the following example, RSVP and SBM are enabled on Ethernet interface 2. After RSVP is enabled, the interface is configured as a DSBM and SBM candidate with a priority of 100. The configured priority is high, making this interface a good contender for DSBM status. However, the maximum configurable priority value is 128, so another interface configured with a higher priority could win the election and become the DSBM.

```
interface Ethernet2
 ip address 145.2.2.150 255.255.255.0
```

```
 no ip directed-broadcast
 ip pim sparse-dense-mode
 no ip mroute-cache
 media-type 10BaseT
 ip rsvp bandwidth 7500 7500
 ip rsvp dsbm candidate 100
 ip rsvp dsbm non-resv-send-limit rate 500
 ip rsvp dsbm non-resv-send-limit burst 1000
 ip rsvp dsbm non-resv-send-limit peak 500
end
```

PfR RSVP Control

The PfR RSVP Control feature introduces the ability to perform application-aware path selection for traffic that is controlled by Resource Reservation Protocol (RSVP). This feature allows RSVP flows to be learned by Performance Routing (PfR) and protocol Path messages to be redirected after the PfR master controller determines the best exit using PfR policies.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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

Information About PfR RSVP Control

PfR and RSVP Control

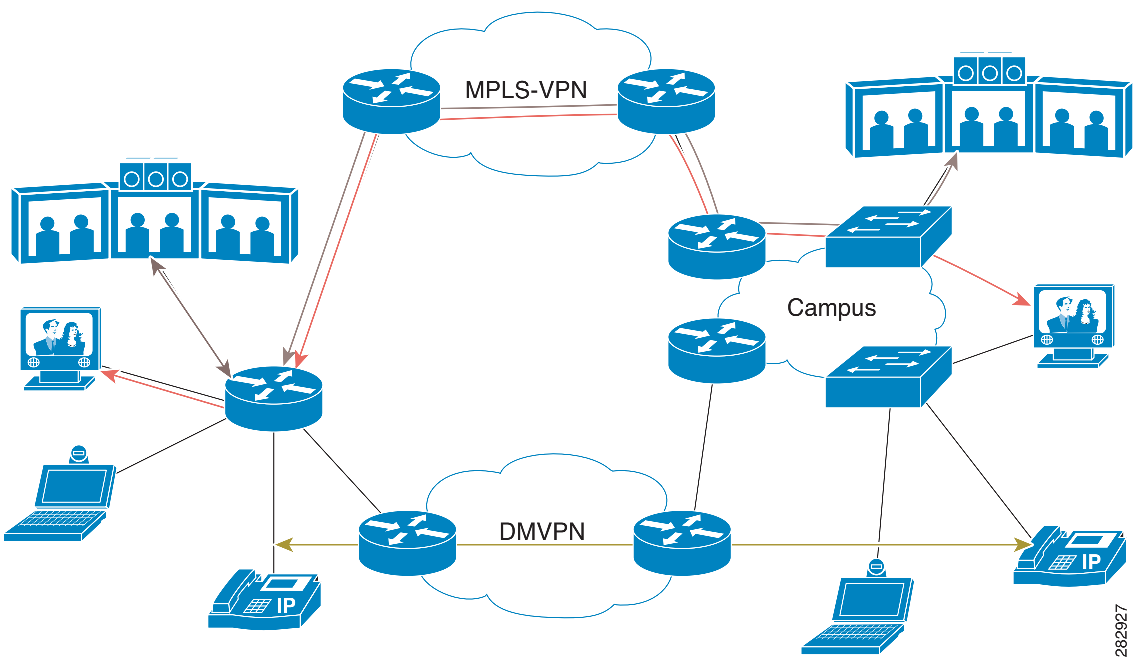
The PfR RSVP Control feature introduces the ability for Performance Routing (PfR) to learn, monitor, and optimize Resource Reservation Protocol (RSVP) flows. PfR is an integrated Cisco IOS solution that allows you to monitor IP traffic flows and then define policies and rules based on traffic class performance, link load distribution, link bandwidth monetary cost, and traffic type. PfR provides active and passive monitoring systems, dynamic failure detection, and automatic path correction. Deploying PfR enables intelligent load distribution and optimal route selection in an enterprise network that uses multiple ISP or WAN connections at the network edge.

PfR can monitor and control applications and prefixes that are configured or learned by observing traffic that is flowing on the network. The master controller (MC) is a centralized policy decision point at which policies are defined and applied to various traffic classes that traverse the border routers (BRs). The MC can be configured to learn and control traffic classes on the network. The MC makes exit selections and instructs the BRs to enforce the exit selection. While the current PfR implementation can be used to optimize voice/video traffic, the control exercised by PfR is not aware of technologies such as RSVP. The PfR RSVP integration will help RSVP leverage the application-specific control of routes that PfR can provide.

RSVP is a standards-based control protocol that allows for resources to be reserved to allow for better reliability for voice/video traffic. RSVP achieves this by signaling the traffic profile before the actual data flow to reserve resources for the data flow. Establishing end-to-end resource reservations along a media path allows RSVP to guarantee that resources are available when they are needed. RSVP consults the forwarding plane database (or CEF) in order to achieve path congruency with the media flow. The routes in the CEF database are mostly dictated by the routing protocols where the only metric for determining the best route is the cumulative cost of the links on that path.

In the diagram shown below, there are two paths for the network on the left to reach the campus network on the right. One path uses the DMVPN cloud, and the other path uses the MPLS-VPN cloud. Depending on the speed and bandwidth required, it might make sense to route video applications over the MPLS-VPN network while routing voice applications over the DMVPN network. Such kind of application-aware path selection is not possible in CEF, but PfR can determine the best path for specific application traffic based on performance criteria.

Figure 32 Application-Aware Path Selection



With the RSVP integration, PfR will learn, monitor, and optimize RSVP flows. RSVP is included as a new learn source. PfR will learn RSVP flows that traverse internal and external interfaces. Each RSVP flow is learned as a PfR traffic class and is controlled independently of the other RSVP flows. While filtering of the learned flows is supported with prefix lists and route maps, aggregating RSVP flows is not advised. The PfR master controller (MC) chooses a best exit based on the configured PfR policies and installs route maps to redirect traffic. If any of the RSVP flows enters an Out-of-Policy (OOP) condition, PfR will find and switch the RSVP flow to a new exit. RSVP will reinstall the reservation on the new path at the time of refresh (usually within a span of 30 seconds) or as a Fast Local Repair (FLR) case in less than 5 seconds.

The intent of the PfR RSVP Control feature is to identify and install route maps at the time the router receives an RSVP Path message. The route map captures the data traffic, while RSVP uses this path for the Path message.

RSVP flows are learned as PfR traffic classes defined as a single application flow that can be identified by the source address, source port, destination address, destination port and IP protocol. This microflow is optimized as an application by PfR, and a dynamic policy route is created by PfR to forward this traffic class over the selected exit.

All RSVP flows are optimized only after PfR checks that there is enough bandwidth on the exit that is being considered. This information is pushed periodically from the BRs to the MC. On the BR itself, RSVP notifies PfR every time the bandwidth pool on an interface changes.

Equivalent-Path Round-Robin Resolver

PfR introduced a new resolver with the PfR RSVP Control feature. PfR, by default, uses a random resolver to decide between equivalent paths, exits with the same cost determined by the PfR policies. When the round-robin resolver is configured using the **equivalent-path-round-robin** command, the next exit (next-hop interface) is selected and compared to the running PfR policy. The round-robin resolver is handed an array of equivalent exits from which it chooses in a round-robin fashion. Exits are pruned in the same fashion they are today by each resolver. If the exit matches the policy, the exit becomes the best exit. The round-robin resolver does not do any specific RSVP checking. To return to using the random resolver, enter the no form of the **equivalent-path-round-robin** command.

Any PfR traffiic class can use the round-robin resolver, and it provides a load-balancing scheme for multiple equivalent paths as determined by PfR policy.

RSVP Post Dial Delay Timer for Best Path Selection

In the PfR RSVP Control feature, the **rsvp post-dial-delay** command was introduced to set a value for the RSVP post dial delay timer that runs on the border routers when RSVP flow learning is enabled on a PfR master controller. The timer is updated on the border routers at the start of every PfR learn cycle, and the timer determines the delay, in milliseconds, before the routing path is returned to RSVP. When the PfR and RSVP integration is enabled, PfR tries to locate a best path for any RSVP flows that are learned before the delay timer expires. If the current path is not the best path, PfR attempts to install the new path. RSVP reacts to this policy route injection as a case of Fast Local Repair (FLR) and resignals a new reservation path.

RSVP Signaling Retries for Alternative Reservation Path

The PfR RSVP Control feature introduced a new command, **rsvp signaling-retries**, which is configured on a master controller and is used to instruct PfR to provide an alternate reservation path when an RSVP reservation returns an error condition. If an alternate path is provided by PfR, RSVP can resend the reservation signal. The default number of retries is set to 0; no signaling retries are to be permitted, and a reservation error message is sent when a reservation failure occurs.

Performance Statistics from PfR Commands

The PfR master controller learns and monitors IP traffic that flows through the border routers, and the master controller selects the best exit for a traffic flow based on configured policies and the performance information received from the border routers. To view some of the performance data collected by the master controller, you can use the following commands:

- **show pfr master active-probes**
- **show pfr master border**
- **show pfr master exits**
- **show pfr master statistics**
- **show pfr master traffic-class**
- **show pfr master traffic-class performance**

All these commands are entered at the master controller, and some of the commands have keywords and arguments to filter the output. For detailed information about these commands, see the *Cisco IOS Performance Routing Command Reference*.

How to Configure PfR RSVP Control

- Configuring PfR RSVP Control Using a Learn List, page 340
- Displaying PfR RSVP Control Information, page 344
- Displaying PfR Performance and Statistics Information, page 348

Configuring PfR RSVP Control Using a Learn List

Perform this task on the master controller to define a learn list that contains traffic classes that are automatically learned based on RSVP flows and filtered by a prefix list. In this task, the goal is to optimize all video traffic that is learned from RSVP flows.

The VIDEO traffic class is defined as any prefix that matches 10.100.0.0/16 or 10.200.0.0/16 and a PfR policy, named POLICY_RSVP_VIDEO, is created.

The learn lists are referenced in a PfR policy using a PfR map and are activated using the **policy-rules** (PfR) command.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*}
4. **pfr master**
5. **policy-rules** *map-name*
6. **rsvp signaling-retries** *number*
7. **rsvp post-dial-delay** *msecs*
8. **learn**
9. **list seq** *number* **refname** *refname*
10. **traffic-class prefix-list** *prefix-list-name* [**inside**]
11. **rsvp**
12. **exit**
13. Repeat Step 9 to Step 12 to configure additional learn lists.
14. **exit**
15. Use the **exit** command as necessary to return to global configuration mode.
16. **pfr-map** *map-name sequence-number*
17. **match pfr learn list** *refname*
18. **set mode route control**
19. **set resolve equivalent-path-round-robin**
20. **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	**ip prefix-list** *list-name* [**seq** *seq-value*] {**deny** *network/length* | **permit** *network/length*} **Example:** `Router(config)# ip prefix-list RSVP_VIDEO seq 10 permit 10.100.0.0/16`	Creates an IP prefix list to filter prefixes for learning. • An IP prefix list is used under learn list configuration mode to filter IP addresses that are learned. • The example creates an IP prefix list named RSVP_VIDEO for PfR to profile the prefix, 10.100.0.0/16.
Step 4	**pfr master** **Example:** `Router(config)# pfr master`	Enters PfR master controller configuration mode to configure a Cisco router as a master controller and to configure master controller policy and timer settings.
Step 5	**policy-rules** *map-name* **Example:** `Router(config-pfr-mc)# policy-rules POLICY_RSVP_VIDEO`	Selects a PfR map and applies the configuration under PfR master controller configuration mode. • Use the *map-name* argument to specify the PfR map name to be activated. • The example applies the PfR map named POLICY_RSVP_VIDEO which includes the learn list configured in this task.
Step 6	**rsvp signaling-retries** *number* **Example:** `Router(config-pfr-mc)# rsvp signaling-retries 1`	Specifies the number of alternate paths that PfR provides for an RSVP reservation when a reservation error condition is detected. • Use the *number* argument to specify the number of alternate paths. • The example configured in this task shows how to configure PfR to set the number of alternate paths for RSVP signaling retries to 1.
Step 7	**rsvp post-dial-delay** *msecs* **Example:** `Router(config-pfr-mc)# rsvp post-dial-delay 100`	Configures the RSVP post dial delay timer to set the delay before PfR returns the routing path to RSVP. • Use the *msecs* argument to specify the delay, in milliseconds. • The example configured in this task shows how to configure PfR to set the RSVP post dial delay to 100 milliseconds.
Step 8	**learn** **Example:** `Router(config-pfr-mc)# learn`	Enters PfR Top Talker and Top Delay learning configuration mode to automatically learn traffic classes.

	Command or Action	Purpose
Step 9	**list seq** *number* **refname** *refname* **Example:** `Router(config-pfr-mc-learn)# list seq 10 refname LEARN_RSVP_VIDEO`	Creates a PfR learn list and enters learn list configuration mode. • Use the **seq** keyword and *number* argument to specify a sequence number used to determine the order in which learn list criteria are applied. • Use the **refname** keyword and *refname* argument to specify a reference name for the learn list. • The example creates a learn list named LEARN_RSVP_VIDEO.
Step 10	**traffic-class prefix-list** *prefix-list-name* [**inside**] **Example:** `Router(config-pfr-mc-learn-list)# traffic-class prefix-list RSVP_VIDEO`	Configures the master controller to automatically learn traffic based only on destination prefixes. • Use the *prefix-list-name* argument to specify a prefix list. • The example defines a traffic class using the prefix list named RSVP_VIDEO.
Step 11	**rsvp** **Example:** `Router(config-pfr-mc-learn-list)# rsvp`	Configures the master controller to learn the top prefixes based on RSVP flows. • When this command is enabled, the master controller will learn the top prefixes across all border routers according to the highest outbound throughput. • The example configures a master controller to learn the top prefixes based on RSVP flows for the LEARN_RSVP_VIDEO learn list.
Step 12	**exit** **Example:** `Router(config-pfr-mc-learn-list)# exit`	Exits learn list configuration mode, and returns to PfR Top Talker and Top Delay learning configuration mode.
Step 13	Repeat Step 9 to Step 12 to configure additional learn lists.	--
Step 14	**exit** **Example:** `Router(config-pfr-mc-learn)# exit`	Exits PfR Top Talker and Top Delay learn configuration mode, and returns to PfR master controller configuration mode.
Step 15	Use the **exit** command as necessary to return to global configuration mode.	--

	Command or Action	Purpose
Step 16	**pfr-map** *map-name sequence-number* **Example:** `Router(config)# pfr-map POLICY_RSVP_VIDEO 10`	Enters PfR map configuration mode to configure a PfR map. • The example creates a PfR map named POLICY_RSVP_VIDEO.
Step 17	**match pfr learn list** *refname* **Example:** `Router(config-pfr-map)# match pfr learn list LEARN_RSVP_VIDEO`	Creates a match clause entry in a PfR map to match PfR-learned prefixes. • Only one match clause can be configured for each PfR map sequence. • The example defines a traffic class using the criteria defined in the PfR learn list named LEARN_RSVP_VIDEO. **Note** Only the syntax relevant to this task is used here.
Step 18	**set mode route control** **Example:** `Router(config-pfr-map)# set mode route control`	Creates a set clause entry to configure route control for matched traffic. • In control mode, the master controller analyzes monitored prefixes and implements changes based on policy parameters.
Step 19	**set resolve equivalent-path-round-robin** **Example:** `Router(config-pfr-map)# set resolve equivalent-path-round-robin`	Creates a set clause entry to specify the use of the equivalent-path round-robin resolver. • In this task, the equivalent-path round-robin resolver is used to choose between equivalent paths instead of the random resolver.
Step 20	**end** **Example:** `Router(config-pfr-map)# end`	(Optional) Exits PfR map configuration mode and returns to privileged EXEC mode.

Displaying PfR RSVP Control Information

Although the PfR RSVP Control feature is configured on a master controller, the border routers actually collect the performance information, and there are **show** and **debug** commands available to display the RSVP information for both the master controller and border routers. The first few commands in this task are entered on a master controller and, for the rest of the commands, there is a step to move to a border router through which the application traffic is flowing. These **show** and **debug** commands can be entered in any order.

SUMMARY STEPS

1. **enable**
2. **show pfr master traffic-class** [**rsvp**] [**active** | **passive** | **status**] [**detail**]
3. **show pfr master policy** [*sequence-number* | *policy-name* | **default** | **dynamic**]
4. **debug pfr master rsvp**
5. Move to a border router through which the RSVP traffic is flowing.
6. **enable**
7. **show pfr border rsvp**
8. **show pfr border routes rsvp-cache**
9. **debug pfr border rsvp**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show pfr master traffic-class** [**rsvp**] [**active** | **passive** | **status**] [**detail**]

This command is used to display information about PfR traffic classes that are learned as RSVP traffic classes.

Example:

```
Router# show pfr master traffic-class rsvp

OER Prefix Statistics:
 Pas - Passive, Act - Active, S - Short term, L - Long term, Dly - Delay (ms),
 P - Percentage below threshold, Jit - Jitter (ms),
 MOS - Mean Opinion Score
 Los - Packet Loss (packets-per-million), Un - Unreachable (flows-per-million),
 E - Egress, I - Ingress, Bw - Bandwidth (kbps), N - Not applicable
 U - unknown, * - uncontrolled, + - control more specific, @ - active probe all
 # - Prefix monitor mode is Special, & - Blackholed Prefix
 % - Force Next-Hop, ^ - Prefix is denied

DstPrefix           Appl_ID Dscp Prot     SrcPort     DstPort SrcPrefix
           Flags             State     Time            CurrBR  CurrI/F Protocol
         PasSDly  PasLDly   PasSUn   PasLUn  PasSLos  PasLLos      EBw      IBw
         ActSDly  ActLDly   ActSUn   ActLUn  ActSJit  ActPMOS  ActSLos  ActLLos
--------------------------------------------------------------------------------
10.1.0.10/32              N    N  tcp       75-75       75-75 10.1.0.12/32
                          INPOLICY       @0        10.1.0.24 Tu24            PBR
               U        U        0        0        0        0        0        0
               1        1        0        0        N        N        N        N
```

Step 3 **show pfr master policy** [*sequence-number* | *policy-name* | **default** | **dynamic**]

This command is used to display policy information. The following example uses the **dynamic** keyword to display the policies dynamically created by provider applications. Note the RSVP configuration commands.

Example:

```
Router# show pfr master policy dynamic

Dynamic Policies:

  proxy id 10.3.3.3
  sequence no. 18446744069421203465, provider id 1001, provider priority 65535
    host priority 65535, policy priority 101, Session id 9
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
  proxy id 10.3.3.3
  sequence no. 18446744069421269001, provider id 1001, provider priority 65535
    host priority 65535, policy priority 102, Session id 9
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
  proxy id 10.3.3.4
  sequence no. 18446744069421334538, provider id 1001, provider priority 65535
    host priority 65535, policy priority 103, Session id 10
  backoff 90 90 90
  delay relative 50
  holddown 90
  periodic 0
  probe frequency 56
  mode route control
  mode monitor both
  mode select-exit good
  loss relative 10
  jitter threshold 20
  mos threshold 3.60 percent 30
  unreachable relative 50
  next-hop not set
  forwarding interface not set
  resolve delay priority 11 variance 20
  resolve utilization priority 12 variance 20
```

Step 4 **debug pfr master rsvp**

Displays debugging information about PfR RSVP events on a PfR master controller.

Example:

```
Router# debug pfr master rsvp

Jan 23 21:18:19.439 PST: PFR_MC_RSVP: recvd a RSVP flow
Jan 23 21:18:19.439 PST: PFR_MC_RSVP: Processing 1 rsvp flows
Jan 23 21:18:19.439 PST: PFR_MC_RSVP: Resolve: src: 10.1.0.12 dst: 10.1.25.19 pr
oto: 17 sport min: 1 sport max: 1 dport min: 1 dport max: 1 from BR 10.1.0.23
Jan 23 21:18:19.439 PST: PFR_MC_RSVP: Marking: 10.1.0.23, FastEthernet1/0
Jan 23 21:18:19.439 PST: %OER_MC-5-NOTICE: Uncontrol Prefix 10.1.25.19/32, Probe frequency changed
Jan 23 21:18:19.439 PST: PFR_MC_RSVP: Marked: 10.1.0.23, FastEthernet1/0 as current
Jan 23 21:18:19.467 PST: PFR_MC_RSVP: recv new pool size
Jan 23 21:18:19.467 PST: PFR_MC_RSVP: Update from 10.1.0.23, Fa1/0: pool 8999
Jan 23 21:18:20.943 PST: %OER_MC-5-NOTICE: Prefix Learning WRITING DATA
Jan 23 21:18:21.003 PST: %OER_MC-5-NOTICE: Prefix Learning STARTED
Jan 23 21:18:22.475 PST: PFR_MC_RSVP: RSVP resolver invoked
Jan 23 21:18:22.475 PST: PFR RSVP MC:  10.1.25.19/32 Appl 17 [1, 1][1, 1] 0:
        BR 10.1.0.23, Exit Fa1/0, is current exit
Jan 23 21:18:22.475 PST: PFR RSVP MC:  10.1.25.19/32 Appl 17 [1, 1][1, 1] 0:
        BR 10.1.0.23, Exit Fa1/0, is current exit
Jan 23 21:18:22.475 PST: PFR_MC_RSVP: BR:10.1.0.23 Exit:Fa1/0pool size : 8999
est : 8999 tc->tspec: 1, fit: 8999
Jan 23 21:18:22.475 PST: PFR_MC_RSVP: BR:10.1.0.24 Exit:Tu24pool size : 9000
est : 9000 tc->tspec: 1, fit: 8999
Jan 23 21:18:22.475 PST: PFR_MC_RSVP: BR:10.1.0.23 Exit:Fa1/1pool size : 9000
est : 9000 tc->tspec: 1, fit: 8999
```

Step 5 Move to a border router through which the RSVP traffic is flowing.

Step 6 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 7 **show pfr border rsvp**

The following example shows information about the current values for the RSVP post dial timeout timer and signaling retries on a PfR border router:

Example:

```
Router# show pfr border rsvp

PfR BR RSVP parameters:
       RSVP Signaling retries:         1
       Post-dial-timeout(msec):        0
```

Step 8 **show pfr border routes rsvp-cache**

This command is used to show all the RSVP paths that PfR is aware of.

Note Only syntax appropriate to this example is shown.

Example:

```
Router# show pfr border routes rsvp-cache

SrcIP       DstIP       Protocol Src_port Dst_port Nexthop       Egress I/F PfR/RIB
----------- ----------- -------- -------- -------- ------------- ---------- --------
10.1.25.19  10.1.35.5   UDP      1027     1027     10.1.248.5    Gi1/0       RIB*
10.1.0.12   10.1.24.10  UDP      48       48       10.1.248.24   Gi1/0       PfR*
```

```
10.1.0.12   10.1.42.19  UDP      23       23       10.1.248.24   Gi1/0       PfR*
10.1.0.12   10.1.18.10  UDP      12       12       172.16.43.2   Fa1/1       PfR*
```

Step 9 **debug pfr border rsvp**

Displays debugging information about PfR RSVP events on a PfR border router.

Example:

```
Router# debug pfr border rsvp

Jan 23 21:18:19.434 PST: PfR RSVP:RESOLVE called for src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1; tspec 1
Jan 23 21:18:19.434 PST: PfR RSVP:hash index = 618
Jan 23 21:18:19.434 PST: PfR RSVP:Searching flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1
Jan 23 21:18:19.434 PST: PfR RSVP:Add flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1
Jan 23 21:18:19.434 PST: PfR RSVP:hash index = 618
Jan 23 21:18:19.434 PST: PfR RSVP:Searching flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1
Jan 23 21:18:19.434 PST: PfR RSVP:hash index = 618
Jan 23 21:18:19.434 PST: PfR RSVP:successfully added the flow to the db
Jan 23 21:18:19.434 PST: PfR RSVP:flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1 lookup; topoid: 0
Jan 23 21:18:19.434 PST: PfR RSVP(det):ret nh: 10.185.252.1, idb: 35
Jan 23 21:18:19.434 PST: PfR RSVP:Adding new context
Jan 23 21:18:19.434 PST: PfR RSVP(det):Num contexts: 0
Jan 23 21:18:19.434 PST: PfR RSVP(det):Num contexts: 1
Jan 23 21:18:19.434 PST: PfR RSVP:flow src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1 now pending notify
Jan 23 21:18:19.434 PST: PfR RSVP:Resolve on flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1
Jan 23 21:18:19.434 PST: PfR RSVP:Filtering flow: src: 10.1.0.12 dst: 10.1.25.19
 proto: 17 sport: 1 dport: 1
```

Displaying PfR Performance and Statistics Information

Enter the commands in this task to view more detailed performance or statistical information about PfR traffic classes or exits. The commands can be entered in any order within each section.

SUMMARY STEPS

1. **enable**
2. **show pfr master traffic-class** [**policy** *policy-seq-number* | *rc-protocol* | **state** {**hold**| **in** | **out** | **uncontrolled**}] [**detail**]
3. **show pfr master traffic-class performance** [**application** *application-name* [*prefix*] | **history** [**active** | **passive**] | **inside** | **learn** [**delay** | **inside** | **list** *list-name* | **rsvp** | **throughput**] | **policy** *policy-seq-number* | *rc-protocol* | **state** {**hold** | **in** | **out** | **uncontrolled**} | **static**] [**detail**]
4. **show pfr master exits**
5. **show pfr master active-probes** [**assignment** | **running**] [**forced** *policy-sequence-number* | **longest-match**]
6. **show pfr master border** [*ip-address*] [**detail** | **report** | **statistics** | **topology**]
7. **show pfr master statistics** [**active-probe** | **border** | **cc** | **exit** | **netflow** | **prefix** | **process** | **system** | **timers**]

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode. Enter your password if prompted.

Example:

```
Router> enable
```

Step 2 **show pfr master traffic-class** [**policy** *policy-seq-number* | *rc-protocol* | **state** {**hold**| **in** | **out** | **uncontrolled**}] [**detail**]

This command is used to display information about traffic classes that are monitored and controlled by a PfR master controller. In this example, the **state in** keywords are used to filter the output to show only traffic classes that are in an in-policy state.

Example:

```
Router# show pfr master traffic-class state in

OER Prefix Statistics:
 Pas - Passive, Act - Active, S - Short term, L - Long term, Dly - Delay (ms),
 P - Percentage below threshold, Jit - Jitter (ms),
 MOS - Mean Opinion Score
 Los - Packet Loss (packets-per-million), Un - Unreachable (flows-per-million),
 E - Egress, I - Ingress, Bw - Bandwidth (kbps), N - Not applicable
 U - unknown, * - uncontrolled, + - control more specific, @ - active probe all
 # - Prefix monitor mode is Special, & - Blackholed Prefix
 % - Force Next-Hop, ^ - Prefix is denied

DstPrefix           Appl_ID Dscp Prot     SrcPort     DstPort SrcPrefix
           Flags             State     Time            CurrBR  CurrI/F Protocol
         PasSDly  PasLDly   PasSUn   PasLUn  PasSLos  PasLLos      EBw      IBw
         ActSDly  ActLDly   ActSUn   ActLUn  ActSJit  ActPMOS  ActSLos  ActLLos
--------------------------------------------------------------------------------
10.1.0.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.1 Et0/0           BGP
              14       14        0        0        0        0       78        9
               N        N        N        N        N        N

10.2.0.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.2 Et0/0           BGP
              14       14        0        0        0        0       75        9
               N        N        N        N        N        N

10.3.0.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.3 Et0/0           BGP
              14       14        0        0        0        0       77        9
               N        N        N        N        N        N

10.4.0.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.4 Et0/0           BGP
              14       14        0        0        0        0       77        9
               N        N        N        N        N        N

10.1.8.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.3 Et0/0           BGP
              14       14    62500    73359        0        0        5        1
               N        N        N        N        N        N

10.1.1.0/24                N    N    N           N           N N
                           INPOLICY        0          10.1.1.2 Et0/0           BGP
```

```
              14       14     9635     9386     1605     1547       34        4
               N        N        N        N        N        N
```

Step 3 **show pfr master traffic-class performance** [**application** *application-name* [*prefix*] | **history** [**active** | **passive**] | **inside** | **learn** [**delay** | **inside** | **list** *list-name* | **rsvp** | **throughput**] | **policy** *policy-seq-number* | *rc-protocol* | **state** {**hold** | **in** | **out** | **uncontrolled**} | **static**] [**detail**]

This command displays performance information about traffic classes that are monitored and controlled by a PfR master controller.

Note Only the syntax applicable to this example is shown.

Example:

The following output shows traffic-class performance history on current exits during the last 60 minutes.

```
Router# show pfr master traffic-class performance history

Prefix: 10.70.0.0/16
efix performance history records
 Current index 1, S_avg interval(min) 5, L_avg interval(min) 60

Age       Border          Interface       OOP/RteChg Reasons
Pas: DSum  Samples  DAvg  PktLoss  Unreach   Ebytes   Ibytes     Pkts    Flows
Act: Dsum Attempts  DAvg    Comps  Unreach   Jitter LoMOSCnt   MOSCnt
00:00:33  10.1.1.4        Et0/0
Pas: 6466      517    12        2       58  3400299   336921    10499     2117
Act:    0        0     0        0        0        N        N        N
00:01:35  10.1.1.4        Et0/0
Pas:15661     1334    11        4      157  4908315   884578    20927     3765
Act:    0        0     0        0        0        N        N        N
00:02:37  10.1.1.4        Et0/0
Pas:13756     1164    11        9      126  6181747   756877    21232     4079
Act:    0        0     0        0        0        N        N        N
00:03:43  10.1.1.1        Et0/0
Pas:14350     1217    11        6      153  6839987   794944    22919     4434
Act:    0        0     0        0        0        N        N        N
00:04:39  10.1.1.3        Et0/0
Pas:13431     1129    11       10      122  6603568   730905    21491     4160
Act:    0        0     0        0        0        N        N        N
00:05:42  10.1.1.2        Et0/0
Pas:14200     1186    11        9      125  4566305   765525    18718     3461
Act:    0        0     0        0        0        N        N        N
00:06:39  10.1.1.3        Et0/0
Pas:14108     1207    11        5      150  3171450   795278    16671     2903
Act:    0        0     0        0        0        N        N        N
00:07:39  10.1.1.4        Et0/0
Pas:11554      983    11       15      133  8386375   642790    23238     4793
Act:    0        0     0        0        0        N        N        N
```

Step 4 **show pfr master exits**

This command is used to display information about the exits that are used for PfR traffic classes, including the IP address and interface of the border router, the exit policy, and exit performance data. The example below shows RSVP pool information.

Example:

```
Router# show pfr master exits

PfR Master Controller Exits:

General Info:
=============
  E - External
  I - Internal
  N/A - Not Applicable
```

```
                                                                                    Up/
  ID Name         Border          Interface   ifIdx IP Address      Mask Policy      Type Down
 --- ------------ --------------- ----------- ----- --------------- ---- ----------- ---- ----
   6              10.1.0.23       Fa1/0           9 10.185.252.23     27 Util          E  UP
   5              10.1.0.23       Fa1/1          10 172.16.43.23      27 Util          E  UP
   4              10.1.0.24       Tu24           33 10.20.20.24       24 Util          E  UP

Global Exit Policy:
===================
    Range Egress:      In Policy - No difference between exits - Policy 10%
   Range Ingress:      In Policy - No difference between entrances - Policy 0%
     Util Egress:      In Policy
    Util Ingress:      In Policy
            Cost:      In Policy

Exits Performance:
==================
                   Egress                                                Ingress
    ---------------------------------------------------- ----------------------------------
 ID Capacity  MaxUtil    Usage   %      RSVP POOL    OOP Capacity  MaxUtil    Usage   %  OOP
 --- -------- -------- -------- --- -------------- ----- -------- -------- -------- --- -----
  6   100000    90000       66   0           9000    N/A   100000   100000       40   0  N/A
  5   100000    90000       34   0           8452    N/A   100000   100000       26   0  N/A
  4   100000    90000      128   0           5669    N/A   100000   100000      104   0  N/A

TC and BW Distribution:
=======================
                       # of TCs                  BW (kbps)           Probe   Active
     Name/ID   Current Controlled InPolicy    Controlled       Total  Failed  Unreach
                                                                     (count) (fpm)
        ----   ---------------------------   ----------------------  ------  --------
           6        0          0        0             0          66       0         0
           5      548        548      548             0          34       0         0
           4     3202       3202     3202             0         128       0         0

Exit Related TC Stats:
======================
                                          Priority
                                     highest       nth
                                    ------------------
  Number of TCs with range:            0         0
   Number of TCs with util:            0         0
   Number of TCs with cost:            0         0

       Total number of TCs:      3800
```

Step 5 **show pfr master active-probes** [**assignment** | **running**] [**forced** *policy-sequence-number* | **longest-match**]

The following example shows the status of all created or in-progress probes.

Example:

```
Router# show pfr master active-probes running

PfR Master Controller running probes:

Border         Interface   Type     Target         TPort Codec    Freq Forced Pkts DSCP
                                                                       (Pol
                                                                        Seq)
-------------- ----------- -------- -------------- ----- -------- ---- ------ ---- ----
10.100.100.200 Ethernet1/0 tcp-conn 10.100.200.100 65535 g711alaw 10   20     100  ef
10.2.2.3       Ethernet1/0 tcp-conn 10.1.5.1       23    N        56   10     1    defa
10.1.1.1       Ethernet1/0 tcp-conn 10.1.5.1       23    N        30   N      1    defa
10.1.1.2       Ethernet1/0 tcp-conn 10.1.2.1       23    N        56   N      1    defa
10.2.2.3       Ethernet1/0 tcp-conn 10.1.2.1       23    N        56   N      1    defa
10.1.1.1       Ethernet1/0 tcp-conn 10.1.2.1       23    N        56   N      1    defa
```

Step 6 **show pfr master border** [*ip-address*] [**detail** | **report** | **statistics** | **topology**]

Entered on a master controller, this command displays statistics about all the border routers.

Example:

```
Router# show pfr master border statistics

PFR Master Controller Border
 MC Version: 2.3
 Keepalive : 5 second
 Keepalive : DISABLED
                                                           Last
Border           Status Up/Down UpTime   AuthFail  Receive  Version
---------------- ------ ------- -------- --------  -------- -------
10.200.200.200   ACTIVE UP      03:12:12        0  00:00:04 2.2
10.1.1.2         ACTIVE UP      03:10:53        0  00:00:10 2.2
10.1.1.1         ACTIVE UP      03:12:12        0  00:01:00 2.2

Border Connection Statistics
==============================

                          Bytes         Bytes    Msg    Msg    Sec Buf
Border                     Sent         Recvd   Sent  Recvd Bytes Used
---------------- -------------- ------------- ------ ------ ----------
10.200.200.200           345899        373749      5     10          0
10.1.1.2                 345899        373749      5     10          0
10.1.1.1                 345899        373749      5     10          0

                 Socket Invalid   Context
Border           Closed Message Not Found
---------------- ------ ------- ---------
10.200.200.200        5      10       100
10.1.1.2              5      10       100
10.1.1.1              5      10       100
```

Step 7 **show pfr master statistics** [**active-probe** | **border** | **cc** | **exit** | **netflow** | **prefix** | **process** | **system** | **timers**]

This command displays statistics from the master controller. Use the keywords to filter the display information. In the example below, the **system** keyword displays PfR system statistics.

Example:

```
Router# show pfr master statistics system

   Active Timers: 14
    Total Traffic Classes = 65, Prefixes = 65, Appls =0
   TC state:
    DEFAULT = 0, HOLDDOWN = 11, INPOLICY = 54, OOP = 0, CHOOSE = 0,
    Inside = 1, Probe all = 0, Non-op = 0, Denied = 0
    Controlled 60, Uncontrolled 5, Alloced 65, Freed 0, No memory 0
   Errors:
    Invalid state = 0, Ctrl timeout = 0, Ctrl rej = 0, No ctx = 7616,
    Martians = 0
    Total Policies = 0
    Total Active Probe Targets = 325
    Total Active Probes Running = 0
   Cumulative Route Changes:
    Total  : 3246
    Delay  : 0
    Loss   : 0
    Jitter : 0
    MOS    : 0
    Range  : 0
    Cost   : 0
    Util   : 0
   Cumulative Out-of-Policy Events:
    Total  : 0
    Delay  : 0
    Loss   : 0
    Jitter : 0
    MOS    : 0
    Range  : 0
```

```
Cost    : 0
Util    :
```

Configuration Examples for PfR RSVP Control

Example Defining Traffic Classes Using RSVP Flows

The following example, configured on the master controller, defines a learn list that will contain traffic classes that are automatically learned based on RSVP flows and filtered by a prefix list. In this example, the goal is to optimize all video traffic using the policy named POLICY_RSVP_VIDEO. The RSVP_VIDEO traffic class is defined as any prefix that matches 10.100.0.0/16 or 10.200.0.0/16 and is learned from RSVP flows.

This example configures prefix learning based on RSVP traffic flows.

```
ip prefix-list RSVP_VIDEO permit seq 10 10.100.0.0/16
ip prefix-list RSVP_VIDEO permit seq 20 10.200.0.0/16
pfr master
 policy-rules POLICY_RSVP_VIDEO
 rsvp signaling-retries 1
 rsvp post-dial-delay 100
 learn
 list seq 10 refname LEARN_RSVP_VIDEO
 traffic-class prefix-list RSVP_VIDEO
 rsvp
 exit
 exit
pfr-map POLICY_RSVP_VIDEO 10
 match learn list LEARN_RSVP_VIDEO
 set mode route control
 set resolve equivalent-path-round-robin
 end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Cisco PfR commands: complete command syntax, command mode, command history, defaults, usage guidelines and examples	*Cisco IOS Performance Routing Command Reference*
Basic PfR configuration	"Configuring Basic Performance Routing" module

Related Topic	Document Title
Concepts required to understand the Performance Routing operational phases	"Understanding Performance Routing" module
Advanced PfR configuration	"Configuring Advanced Performance Routing" module
IP SLAs overview	*IP SLAs Configuration Guide*

Technical Assistance

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

Feature Information for PfR RSVP Control

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 16 Feature Information for PfR RSVP Control

Feature Name	Releases	Feature Information
PfR RSVP Control	15.2(1)T	PfR RSVP Control provides support for optimizing RSVP flows using application-aware PfR techniques. The following commands were introduced or modified by this feature: **debug pfr master rsvp**, **debug pfr border rsvp**, **rsvp (PfR)**, **rsvp post-dial-delay**, **rsvp signaling-retries**, **resolve (PfR)**, **set resolve (PfR)**, **show pfr border rsvp**, **show pfr border routes**, **show pfr master active-probes**, **show pfr master border**, **show pfr master exits**, **show pfr master policy**, **show pfr master statistics show pfr master traffic-class** and **show pfr master traffic-class performance**.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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.

CAC for IPv6 Flows

The CAC for IPv6 Flows feature provides IPv6 support for Resource Reservation Protocol (RSVP). By enabling this feature, the network is made to support the complete RSVP IPv6 functionality for Call Admission Control (CAC) and Medianet.

Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see Bug Search Tool and 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 module.

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 CAC for IPv6 Flows

You must configure RSVP on one or more interfaces on at least two neighboring routers that share a link within the network.

Restrictions for CAC for IPv6 Flows

- The RSVP functionality is not High Availability (HA) compliant; hence, the CAC for IPv6 Flows feature is noncompliant with HA.
- Multiprotocol Label Switching (MPLS) virtual private network (VPN) Virtual Routing and Forwarding (VRF) instances are not supported.

Information About CAC for IPv6 Flows

The CAC for IPv6 Flows feature provides IPv6 support for RSVP, which allows, services that run RSVP as a transport protocol such as CAC, TE, mediatrace, and medianet, to be IPv6 compliant.

RSVP signaling can be initiated and terminated by the following entities:

- RSVP at the endpoint
- RSVP source or receiver proxy
- RSVP agent or application server
- RSVP proxy from the network device (router or switch)

To enable the CAC for IPv6 Flows feature, the endpoints and application servers are designed to be IPv6 systems that signal RSVP to the network.

Differences Between IPv4 and IPv6 Flows

Following are the differences between IPv4 and IPv6 flows in an RSVP network:

- For general routing purposes, global IPv6 addresses are not required on all intermediate devices. Link local adddresses are used instead. However, global IPv6 addresses are required on ingress and egress interfaces.
- Link local addresses are used for neighbor authentication in an IPv6 network as opposed to how global IP addresses are used in an IPv4 network.

IPv6 Support for RSVP Features

The CAC for IPv6 Flows feature extends IPv6 support to the following RSVP features:

- CAC
- Transport Protocol
- RSVP policy support for global and interface configuration modes, except access control list (ACL) support
- RSVP authentication, except ACL support
- Previous hop (PHOP) overwrite in interface configuration mode
- Fast Local Repair (FLR)
- Ingress CAC
- Flexible bandwidth
- Virtual Routing and Forwarding (VRF)
- RSVP reliable messaging
- Flow Metadata

For more information about each of these features, see *QoS: RSVP Configuration Guide*.

How to Configure CAC for IPv6 Flows

Adding Senders or Receivers for IPv6 Flows to the RSVP Database

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 rsvp sender** *session-ipv6-address sender-ipv6-address* [**tcp** | **udp** | *ip-protocol*] *session-dest-port sender-source-port previous-hop-ipv6-address previous-hop-interface bandwidth burst-size*
4. **ipv6 rsvp reservation** *session-ipv6-address sender-ipv6-address* [**tcp** | **udp** | *ip-protocol*] *session-dest-port sender-source-port next-hop-ipv6-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size*
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 rsvp sender** *session-ipv6-address sender-ipv6-address* [**tcp** | **udp** | *ip-protocol*] *session-dest-port sender-source-port previous-hop-ipv6-address previous-hop-interface bandwidth burst-size* **Example:** `Device(config)# ipv6 rsvp sender 2001:DB8:1:: 2001:DB8:2:: tcp 2 3 2001:DB8::1 fastEthernet 0/1 2 3`	Adds senders to the RSVP database. • Enables a networking device to behave like it is receiving and processing IPv6 RSVP PATH messages from the sender or previous hop routes containing the indicated attributes. • The related **ipv6 rsvp sender-host** command enables a device to simulate a host generating IPv6 RSVP PATH messages. It is used mostly for debugging and testing purposes.

	Command or Action	Purpose
Step 4	**ipv6 rsvp reservation** *session-ipv6-address sender-ipv6-address* [**tcp** | **udp** | *ip-protocol*] *session-dest-port sender-source-port next-hop-ipv6-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* **Example:** `Device(config)# ipv6 rsvp reservation 2001:DB8:1:: 2001:DB8::2 tcp 2 3 2001:DB8::3 fastEthernet 0/1 ff load 2 4`	Adds receivers to the RSVP database and enables a device to behave like it is receiving and processing IPv6 RSVP RESV messages. • The related **ipv6 rsvp reservation-host** command enables a device to simulate a host generating RSVP RESV messages. It is used mostly for debugging and testing purposes.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Sender for IPv6 Flows

Perform this task to configure a static sender for IPv6 flows with a VRF on a headend device, to make the device proxy an IPv6 RSVP PATH message.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 rsvp sender-host** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-destination-port sender-source-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*]
4. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 3	**ipv6 rsvp sender-host** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-destination-port sender-source-port bandwidth burst-size* [**identity** *alias*] [**vrf** *vrf-name*] **Example:** `Device(config)# ipv6 rsvp sender-host 2001:DB8:1:: 2001:DB8:2:: udp 1 1 10 10 vrf myvrf`	Enables a networking device to simulate a host generating IPv6 RSVP PATH messages. • The optional **identity** *alias*identity alias keyword and argument pair specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E). **Note** If you use the quotation marks (" ") or a question mark (?) as part of the alias string, you must type the CTRL/V key sequence before entering the embedded " " or ? characters. The alias is never transmitted to other networking devices.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring a Static Receiver for IPv6 Flows

Perform this task to configure a static RSVP receiver with an application ID to make the device proxy an IPv6 RSVP RESV message containing an application ID on behalf of an RSVP-unaware receiver application.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. Do one of the following:
   - **ipv6 rsvp reservation-host** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-dest-port sender-source-port* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
   - **ipv6 rsvp reservation** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-dest-port sender-source-port next-hop-ipv6-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
4. **end**

DETAILED STEPS

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

Step 3	Do one of the following:
	• **ipv6 rsvp reservation-host** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-dest-port sender-source-port* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
	• **ipv6 rsvp reservation** *session-ipv6-address sender-ipv6-address* {**tcp** | **udp** | *ip-protocol*} *session-dest-port sender-source-port next-hop-ipv6-address next-hop-interface* {**ff** | **se** | **wf**} {**rate** | **load**} *bandwidth burst-size* [**identity** *alias*]
	
	Example:
	
	`Device(config)# ipv6 rsvp reservation-host 2001:DB8::1 2001:DB8::2 udp 20 30 se load 100 60 identity rsvp-voice`
	
	Example:
	
	`Device(config)# ipv6 rsvp reservation 2001:DB8:1:: FFFF:FFFF:: udp 20 0 172.16.4.1 Ethernet1 wf rate 350 65 identity xyz`
		Enables a device to simulate a host generating IPv6 RSVP RESV messages.
		• The optional **identity** *alias* keyword and argument pair specifies an application ID alias. The string can have as many as 64 printable characters (in the range 0x20 to 0x7E).
		
		Note If you use the quotation marks (" ") or a question mark (?) as part of the alias string, you must type the CTRL/V key sequence before entering the embedded " " or ? character. The alias is never transmitted to other devices.
		
		Note Use the **ipv6 rsvp reservation-host** command if the device is the destination or the **ipv6 rsvp reservation** command to have the device proxy on behalf of a downstream host.

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

Configuring a Receiver Proxy for IPv6 Flows on a Tailend Device

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**}
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 rsvp listener** [**vrf** *vrf-name*] *dst* {**udp** | **tcp** | **any** | *number*} {**any** | *dst-port*} {**announce** | **reply** | **reject**} **Example:** `Device(config)# ipv6 rsvp listener vrf myvrf 2001:DB8:1:: any any reply`	Configures an RSVP device to listen for IPv6 PATH messages.
Step 4	**end** **Example:** `Device(config)# end`	(Optional) Returns to privileged EXEC mode.

Configuring RSVP as a Transport Protocol for IPv6 Flows

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 rsvp transport client** *client-id*
4. **ipv6 rsvp transport sender-host** [**tcp** | **udp**] *destination-ipv6-address source-ipv6-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:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 rsvp transport client** *client-id* **Example:** `Device(config)# ipv6 rsvp transport client 2`	Registers an RSVP transport client ID with RSVP for IPv6 flows. • This command is used for debugging and testing.
Step 4	**ipv6 rsvp transport sender-host** [**tcp** | **udp**] *destination-ipv6-address source-ipv6-address ip-protocol dest-port source-port client-id init-id instance-id* [**vrf** *vrf-name*] [**data** *data-value*] **Example:** `Device(config)# ipv6 rsvp transport sender-host tcp 2001:DB8:10:: 2001:DB:11:: 3 4 5 2 3 4 vrf vr1`	Creates an RSVP transport session, which enables a networking device to simulate a host generating IPv6 RSVP PATH message. • This command is used for debugging and testing purposes.
Step 5	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Binding a Key Chain to an RSVP IPv6 Neighbor

Perform this task to bind a key chain to an RSVP IPv6 neighbor for neighbor authentication.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 rsvp authentication neighbor address** *ipv6-address* **key-chain** *key-chain-name*
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**ipv6 rsvp authentication neighbor address** *ipv6-address* **key-chain** *key-chain-name* **Example:** `Device(config)# ipv6 rsvp authentication neighbor address 2001:db8:1:: 1 key-chain neighbor_V`	Binds a key chain to an IPv6 address or to an ACL and enters key-chain mode.
Step 4	**end** **Example:** `Device(config-keychain)# end`	Returns to privileged EXEC mode.

Configuring PHOP for IPv6 Flows

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*]
5. **ipv6 rsvp source** {**address** *ipv6-address* | **interface** *type number*}
6. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface Ethernet0/0`	Configures the interface type and enters interface configuration mode.
Step 4	**ip rsvp bandwidth** [*interface-kbps*] [*single-flow-kbps*] **Example:** `Device(config-if)# ip rsvp bandwidth`	Enables RSVP on an interface. • The optional *interface-kbps* and *single-flow-kbps* arguments specify the amount of bandwidth that can be allocated by RSVP flows or to a single flow, respectively. Values are from 1 to 10000000. **Note** Repeat this command for each interface on which you want to enable RSVP.
Step 5	**ipv6 rsvp source** {**address** *ipv6-address* | **interface** *type number*} **Example:** `Device(config-if)# ipv6 rsvp source` **address** `2001:DB8::1`	Configures an RSVP device to populate an address other than the native interface address in the PHOP address field of the hop object when forwarding a PATH message onto that interface. **Note** The source IPv6 address that you configure should be a valid local IP address.

	Command or Action	Purpose
Step 6	**end** **Example:** `Device(config-if)# end`	Returns to privileged EXEC mode.

Configuration Examples for CAC for IPv6 Flows

Example: Entering Senders or Receivers for IPv6 Flows to the RSVP Database

The following example shows how to add senders or receivers for IPv6 flows to the RSVP database:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp sender 2001:DB8:1:: 2001:DB8:2:: tcp 2 3 2001:DB8::1
fastEthernet 0/1 2 3
Device(config)# ipv6 rsvp reservation 2001:DB8:1:: 2001:DB8::2 tcp 2 3 2001:DB8::3
fastEthernet 0/1 ff load 2 4
Device(config)# end
```

Example: Configuring a Static Sender for IPv6 Flows

The following example shows how to configure a static sender for IPv6 flows:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp sender-host 2001:DB8:1:: 2001:DB8:2:: udp 1 1 10 10 vrf myvrf
Device(config)# end
```

Example: Configuring a Static Receiver for IPv6 Flows

The following example shows how to configure a static receiver for IPv6 flows:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp reservation-host 2001:DB8::1 2001:DB8::2 udp 20 30 se load 100
```

```
60 identity rsvp-voice
Device(config)# end
```

Example: Configuring a Receiver Proxy for IPv6 Flows on a Tailend Device

The following example shows how to configure a receiver proxy for IPv6 flows on a tailend device:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp listener vrf myvrf 2001:DB8:1:: any any reply
Device(config)# end
```

Example: Configuring RSVP as a Transport Protocol for IPv6 Flows

The following example shows how to configure RSVP as transport protoocol for IPv6 flows:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp transport client 2
Device(config)# ipv6 rsvp transport sender-host tcp 2001:DB8:10:: 2001:DB:11:: 3 4 5 2 3
4 vrf vrf1
Device(config)# end
```

Example: Binding a Key Chain to an RSVP IPv6 Neighbor

The following example shows how to bind a key chain to an RSVP IPv6 neighbor:

```
Device> enable
Device# configure terminal
Device(config)# ipv6 rsvp authentication neighbor access-list 1 key-chain neighbor_V
Device(config)# end
```

Example: Configuring PHOP for IPv6 Flows

The following example shows how to configure PHOP for IPv6 flows:

```
Device# configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Device(config)# interface ethernet 1/0
Device(config-if)# ipv6 rsvp source address 2001:DB8::1
Device(config-if)# end
```

Additional References

Related Documents

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

Related Topic	Document Title
RSVP commands	*Quality of Service Command Reference*
Configuring RSVP, RSVP-PHOP, RSVP Fast Local Repair, RSVP Message Authentication and other related RSVP features	*QoS: RSVP Configuration Guide*

MIBs

MIB	MIBs Link
None	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

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 CAC for IPv6 Flows

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 17 Feature Information for CAC for IPv6 Flows

Feature Name	Releases	Feature Information
CAC for IPv6 Flows	15.2(3)T	The CAC for IPv6 Flows feature provides IPv6 support for RSVP. By enabling this feature, the network is made to support the complete RSVP IPv6 functionality for CAC and Medianet. The following commands were introduced or modified: **clear ipv6 rsvp authentication**, **clear ipv6 rsvp reservation**, **clear ipv6 rsvp sender**, **ip rsvp bandwidth**, **ip rsvp listener outbound**, **ip rsvp signaling rate-limit**, **ipv6 rsvp sender**, **ipv6 rsvp sender-host**, **ipv6 rsvp source**, **ipv6 rsvp listener**, **ipv6 rsvp reservation**, **ipv6 rsvp reservation-host**, **ipv6 rsvp transport sender-host**, **show ipv6 rsvp authentication**, **show ipv6 rsvp host**, **show ipv6 rsvp installed**, **show ipv6 rsvp listeners**, **show ipv6 rsvp neighbor**, **show ipv6 rsvp request**, **show ipv6 signaling blockade**, **show ipv6 rsvp transport sender**.