Chapter 10
RSVP Overview

This chapter discusses the following topics:

RSVP Overview

RSVP is a resource reservation setup protocol that is used by both network hosts and routers. Hosts use RSVP to request a specific quality of service (QoS) from the network for particular application flows. Routers use RSVP to deliver QoS requests to all routers along the data path. RSVP also can maintain and refresh states for a requested QoS application flow.

RSVP treats an application flow as a simplex connection. That is, the QoS request travels only in one direction—from the sender to the receiver. RSVP is a transport layer protocol that uses IP as its network layer. However, RSVP does not transport application flows. Rather, it is more of an Internet control protocol, similar to ICMP, IGMP, IS-IS, or OSPF. RSVP runs as a separate software process in the JUNOS Internet software and is not in the packet forwarding path.

RSVP is not a routing protocol, but rather is designed to operate with current and future unicast and multicast routing protocols. The routing protocols are responsible for choosing the routes to use to forward packets, and RSVP consults local routing tables to obtain routes. RSVP is responsible only for ensuring the QoS of packets traveling along a data path.

The receiver in an application flow is responsible for requesting the desired QoS from the sender. To do this, the receiver issues an RSVP QoS request on behalf of the local application. The request propagates to all routers in reverse direction of the data paths toward the sender. In this process, RSVP requests might be merged, resulting in a protocol that scales well when there are a large number of receivers.

Because the number of receivers in an application flow is likely to change, and the flow of delivery paths might change during the life of an application flow, RSVP takes a soft-state approach in its design, creating and removing the protocol states in routers and hosts incrementally over time. RSVP sends periodic refresh messages to maintain its state and to recover from occasional lost messages. In the absence of refresh messages, the RSVP states automatically time out and are deleted.

RSVP Standards

RSVP is described in several RFC draft documents. The following documents provide a good overview of RSVP:

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

RFC 2209, *Resource ReSerVation Protocol (RSVP), Version 1, Message Processing Rules*

RFC 2210, *The Use of RSVP with IETF Integrated Services*

RFC 2211, *Specification of the Controlled-Load Network Element Service*

RFC 2215, *General Characterization Parameters for Integrated Service Network Elements*

RFC 2216, *Network Element Service Specification Template*

Extensions to RSVP for LSP Tunnels, Internet draft draft-ietf-mpls-rsvp-lsp-tunnel-05.txt

RFC 2747, RSVP Cryptographic Authentication

RSVP Refresh Reduction Extensions, Internet draft draft-ietf-rsvp-refresh-reduct-05.txt

To access Internet RFCs and drafts, go to the IETF Web site at http://www.ietf.org.

JUNOS RSVP Protocol Implementation

The JUNOS implementation of RSVP supports RSVP Version 1. The software includes support for all mandatory objects and RSVP message types, and supports message integrity and node authentications through the Integrity Object.

The primary purpose of the JUNOS RSVP software is to support dynamic signaling within MPLS label-switched paths. Supporting resource reservations over the Internet is only a secondary purpose of the JUNOS implementation. Because of this, the RSVP software does not support the following features:

IP multicasting sessions.

Traffic control—It cannot make resource reservations for real-time video or audio sessions.

With regards to the protocol mechanism, packet processing, and RSVP objects supported, the JUNOS implementation of the software is inter-operable with other RSVP implementations.

RSVP Operation

RSVP creates independent sessions to handle each data flow. A session is identified by a combination of the destination address, an optional destination port, and a protocol. Within a session, there can be one or more senders. Each sender is identified by a combination of its source address and source port. An out-of-band mechanism, such as a session announcement protocol or human communication, is used to communicate the session identifier to all senders and receivers.

A typical RSVP session involves the following sequence of events:

1. A potential sender starts sending RSVP Path messages to the session address.

2. A receiver, wanting to join the session, registers itself if necessary. For example, a receiver in a multicast application would register itself with IGMP.

3. The receiver receives the Path messages.

4. The receiver sends appropriate Resv messages toward the sender. These messages carry a flow descriptor, which is used by routers along the path to make reservations in their link-layer media.

5. The sender receives the Resv message, and then it starts sending application data.

This sequence of events is not necessarily strictly synchronized. For example, receivers can register themselves before receiving Path messages from the sender, and application data can flow before the sender receives Resv messages. Application data that is delivered before the actual reservation contained in the Resv message typically is treated as best effort, non-real-time traffic with no QoS guarantee.

RSVP Message Types

RSVP uses several types of messages to establish and remove paths for data flows, to establish and remove reservation information, to confirm the establishment of reservations, and to report errors.

Path Messages

Each sender host transmits Path messages downstream along the routes provided by the unicast and multicast routing protocols. Path messages follow the exact paths of application data, creating path states in the routers along the way, thus enabling routers to learn the previous hop and next-hop node for the session. Path messages are sent periodically to refresh path states.

The refresh interval is controlled by a variable called the *refresh time*, which is the periodical refresh timer expressed in seconds. A path state times out if a router does not receive a specified number of consecutive Path messages. This number is specified by a variable called *keep-multiplier*. Path states are kept for ((*keep-multiplier* + 0.5) * 1.5 * *refresh-time*) seconds.

Resv Messages

Each receiver host sends reservation request (Resv) messages upstream toward senders and sender applications. Resv messages must follow exactly the reverse path of Path messages. Resv messages create and maintain a reservation state in each router along the way.

Resv messages are sent periodically to refresh reservation states. The refresh interval is controlled by the same refresh time variable, and reservation states are kept for ((*keep-multiplier* + 0.5) * 1.5 * *refresh-time*) seconds.

PathTear Messages

PathTear messages remove (tear down) path states as well as dependent reservation states in any routers along a path. PathTear messages follow the same path as Path messages. A PathTear typically is initiated by a sender application or by a router when its path state times out.

PathTear messages are not required, but they enhance network performance because they release network resources quickly. If PathTear messages are lost or not generated, path states eventually time out when they are not refreshed, and then the resources associated with the path are released.

ResvTear Messages

ResvTear messages remove reservation states along a path. These messages travel upstream toward senders of the session. In a sense, ResvTear messages are the reverse of Resv messages. ResvTear messages typically are initiated by a receiver application or by a router when its reservation state times out.

ResvTear messages are not required, but they enhance network performance because they release network resources quickly. If ResvTear messages are lost or not generated, reservation states eventually time out when they are not refreshed, and then the resources associated with the reservation are released.

PathErr Messages

When path errors occur (usually because of parameter problems in a Path message), the router sends a unicast PathErr message to the sender that issued the Path message. Using PathErr messages is advisory; these messages do not alter any path state along the way.

ResvErr Messages

When a reservation request fails, a ResvErr error message is delivered to all the receivers involved. Using ResvErr messages is advisory; these messages do not alter any reservation state along the way.

ResvConfirm Messages

Receivers can request confirmation of a reservation request, and this confirmation is sent with ResvConfirm message. Because of the complex RSVP flow-merging rules, a confirmation message does not necessary provide end-to-end confirmation of the entire path. Therefore, ResvConfirm messages are an indication of potential success only, with no guarantees.

The Resource Reservation Protocol (RSVP) is a resource reservation setup protocol that is designed to interact with integrated services on the Internet.

RSVP Reservation Styles

A reservation request includes options for specifying the reservation style. The reservation styles define how reservations for different senders within the same session are treated and how senders are selected.

Two options specify how reservations for different senders within the same session are treated:

Distinct reservation—Each receiver establishes its own reservation with each upstream sender.

Shared reservation—All receivers makes a single reservation that is shared among many senders.

Two options specify how senders are selected:

Explicit sender—List all selected senders.

Wildcard sender—Select all senders, which then participate in the session.

The following reservation styles, formed by a combination of these four options, currently are defined:

Fixed filter (FF)—This reservation style consists of distinct reservations among explicit senders. Examples of applications that use fixed-filter style reservations are video applications and unicast applications, which both require flows that have a separate reservation for each sender.

Wildcard filter (WF)—This reservation style consists of shared reservations among wildcard senders. This type of reservation reserves bandwidth for any and all senders, and propagates upstream toward all senders, automatically extending to new senders as they appear. An example application for wildcard filter reservations is an audio application in which each sender transmits a distinct data stream. Typically, only a few senders are transmitting at any one time. Such a flow does not require a separate reservation for each sender; a single reservation is sufficient.

Shared explicit (SE)—This reservation style consists of shared reservations among explicit senders. This type of reservation reserves bandwidth for a limited group of senders. An example application is an audio application similar to that described for wildcard filter reservations.