

Chapter 10

RSVP Overview

This chapter discusses the following topics:

- RSVP Overview on page 149
- RSVP Standards on page 150
- JUNOS RSVP Protocol Implementation on page 151
- RSVP Operation on page 151
- RSVP Authentication on page 152
- RSVP Message Types on page 152
- RSVP Reservation Styles on page 154
- Link Protection on page 155
- Node Protection on page 156
- RSVP Graceful Restart on page 157

RSVP Overview

The Resource Reservation Protocol (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 and IGMP. 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 preferred 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 Internet RFCs and drafts.

The following Internet RFCs provide an overview of RSVP and RSVP features:

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

RFC 2747, RSVP Cryptographic Authentication

RFC 2961, RSVP Refresh Overhead Reduction Extensions

RFC 3209, RSVP-TE: Extensions to RSVP for LSP Tunnels

RFC 3473, Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions (only Section 9, Fault Handling)

The following Internet draft also provides information about RSVP:

Fast Reroute Extensions to RSVP-TE for LSP Tunnels, Internet draft
draft-ietf-mpls-rsvp-lsp-fastreroute-01.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 regard to the protocol mechanism, packet processing, and RSVP objects supported, the JUNOS implementation of the software is interoperable 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, nonreal-time traffic with no QoS guarantee.

RSVP Authentication

JUNOS 5.3 and later releases support the RSVP authentication style described in RFC 2747, *RSVP Cryptographic Authentication*. However, by default JUNOS uses the RSVP authentication style described in the IETF Internet draft draft-ietf-rsvp-md5-03.txt. If the router receives an RFC 2747 style RSVP authentication from a neighbor, it switches to this style of authentication for that neighbor. The RSVP authentication style for each neighboring router is determined separately.

RSVP Message Types

RSVP uses the following 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 on page 152

Resv Messages on page 152

PathTear Messages on page 153

ResvTear Messages on page 153

PathErr Messages on page 153

ResvErr Messages on page 153

ResvConfirm Messages on page 153

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 $(\textit{keep-multiplier} + 0.5) * 1.5 * \textit{refresh-time}$ 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 $(\textit{keep-multiplier} + 0.5) * 1.5 * \textit{refresh-time}$ 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 necessarily 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 make 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. A sample 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. A sample application is an audio application similar to that described for wildcard filter reservations.

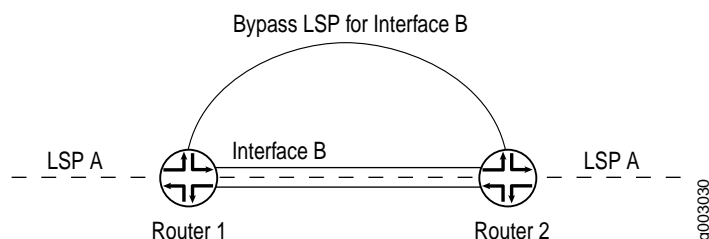
Link Protection

Link protection helps to ensure that traffic going over a specific interface to a neighboring router can continue to reach this router if that interface fails. When link protection is configured for an interface and an LSP that traverses this interface, a bypass LSP is created that will handle this traffic if the interface fails. The bypass LSP uses a different interface and path to reach the same destination. The path used can be configured explicitly or you can rely on CSPF. The RSVP metric for the bypass LSP is set in the range of 20,000 through 29,999 (this value is not user configurable).

If a link-protected interface fails, traffic is quickly switched to the bypass LSP. Note that a bypass LSP cannot share the same egress interface with the LSPs it monitors.

In Figure 20, link protection is enabled on Interface B between Router 1 and Router 2. It is also enabled on LSP A, an LSP that traverses the link between Router 1 and Router 2. If the link between Router 1 and Router 2 fails, traffic from LSP A is quickly switched to the bypass LSP generated by link protection.

Figure 20: Link Protection Creates a Bypass LSP for the Protected Interface



Although LSPs traversing an interface can be configured to take advantage of link protection, it is important to note that it is specifically the interface that benefits from link protection. If link protection is enabled on an interface, but not on a particular LSP traversing that interface, then if the interface fails that LSP will also fail.

To protect traffic over the entire route taken by an LSP, you should configure fast reroute. For more information, see “Configure Fast Reroute” on page 54.

Fast Reroute, Node Protection, and Link Protection

The Internet draft draft-ietf-mpls-rsvp-lsp-fastreroute-02.txt, *Fast Reroute Extensions to RSVP-TE for LSP Tunnels*, describes two different types of traffic protection for RSVP-signaled LSPs:

One-to-one backup—In JUNOS this type of traffic protection is provided by fast reroute. Each LSP requires a protecting LSP to be signaled at each hop except the egress router. This protecting LSP cannot be shared.

Facility backup—This is sometimes called many-to-one backup. In JUNOS this type of traffic protection is provided by node and link protection. Each LSP requires a protecting LSP to be signaled at each hop except the egress router. Unlike fast reroute, this protecting LSP can be shared by other LSPs.

The information above is summarized in Table 3.

Table 3: One-to-One Backup as Compared to Many-to-One Backup

| | One-to-One Backup | Facility Backup |
|---------------------------------------|-------------------|--|
| Name of the Protecting LSP | Detour LSP | Bypass LSP |
| Sharing of the Protecting LSP | Cannot be shared | Can be shared by multiple LSPs |
| JUNOS Configuration Statements | fast-reroute | node-link-protection and link-protection |

Node Protection

Node protection extends the capabilities of link protection. Link protection helps to ensure that traffic going over a specific interface to a neighboring router can continue to reach this router if that interface fails. Node protection ensures that traffic from an LSP traversing a neighboring router can continue to reach its destination even if the neighboring router fails.

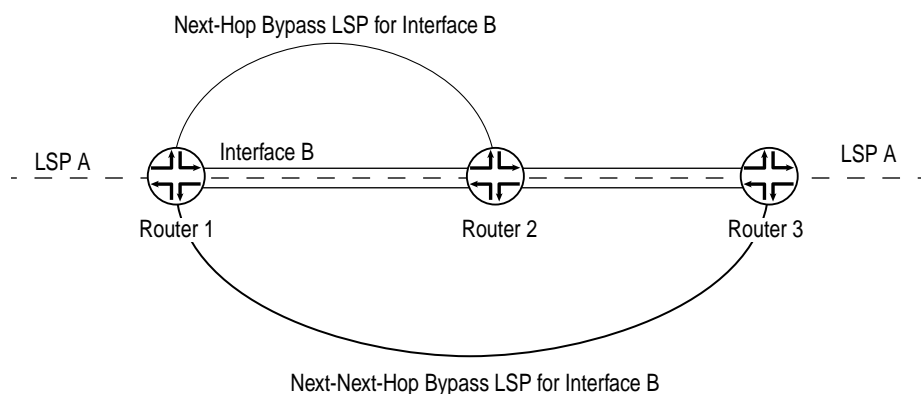
When you enable node protection for an LSP, you must also enable link protection. Once enabled, node protection and link protection establish the following types of bypass LSPs:

Next-hop bypass LSP—Provides an alternate route for an LSP to reach a neighboring router. This type of bypass LSP is established when you enable either node protection or link protection.

Next-next-hop bypass LSP—Provides an alternate route for an LSP to get around a neighboring router enroute to the destination router. This type of bypass LSP is established exclusively when node protection is configured.

In Figure 21, both node and link protection are enabled on Interface B on Router 1. Both node and link protection are also enabled on LSP A, an LSP that traverses the link transiting Router 1, Router 2, and Router 3. If the link between Router 1 and Router 2 fails, traffic from LSP A is quickly switched to the next-hop bypass LSP generated by link protection. If Router 2 suffers a hardware or software failure, traffic from LSP A is switched to the next-next-hop bypass LSP generated by node protection.

Figure 21: Node Protection Creates a Next-Next-Hop Bypass LSP



The time needed by node protection to switch traffic to a next-next-hop bypass LSP can be significantly longer than the time needed by link protection to switch traffic to a next-hop bypass LSP. Link protection relies on a hardware mechanism to detect a link failure, allowing it to quickly switch traffic to a next-hop bypass LSP.

Node failures are often due to software problems on the node router. Node protection relies on the receipt of hello messages from a neighboring router to determine whether it is still functioning. The time it takes node protection to divert traffic partly depends on how often the node router sends hello messages and how long it takes the node-protected router to react to having not received a hello message. However, once the failure is detected, traffic can be quickly diverted to the next-next-hop bypass LSP.

RSVP Graceful Restart

RSVP graceful restart allows a router undergoing a restart to inform its adjacent neighbors of its condition. The restarting router requests a grace period from the neighbor or peer, which can then cooperate with the restarting router. The restarting router can still forward MPLS traffic during the restart period; convergence in the network is not disrupted. The restart is not visible to the rest of the network, and the restarting router is not removed from the network topology.

RSVP graceful restart is described in the following sections:

RSVP Graceful Restart Standard on page 158

RSVP Graceful Restart Terminology on page 158

RSVP Graceful Restart Operation on page 158

Process the Restart_Cap Object on page 159

RSVP Graceful Restart Standard

RSVP graceful restart is described in RFC 3473, *Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource Reservation Protocol-Traffic Engineering (RSVP-TE) Extensions* (only Section 9, *Fault Handling*).

RSVP Graceful Restart Terminology

The following terminology is specific to RSVP graceful restart:

Restart time—The default value is 60,000 milliseconds (1 minute). The restart time is advertised in the hello message. It indicates how long a neighbor should wait to receive a hello message from a restarting router before declaring that router dead and purging states.

The JUNOS software can override a neighbor's advertised restart time if the time is greater than one-third the local restart time. For example, given the default restart time of 60 seconds, a router would wait 20 seconds or less to receive a hello message from a restarting neighbor. If the restart time is zero, the restarting neighbor can immediately be declared dead.

Recovery time (in milliseconds)—Applies only when the control channel is up (the hello exchange is complete) before the restart time. Applies only to nodal faults.

When a graceful restart is in progress, the time left to complete a recovery is advertised. At other times, this value is zero. The maximum advertised recovery time is 2 minutes (120,000 milliseconds).

During the recovery time, a restarting node attempts to recover its lost states with assistance from its neighbors. The neighbor of the restarting node must send the Path messages with the recovery labels to the restarting node within a period of one half of the recovery time. The restarting node considers its graceful restart complete after its advertised recovery time.

RSVP Graceful Restart Operation

For RSVP graceful restart to function, the feature must be enabled on the global routing instance. RSVP graceful restart can be disabled at the protocol level (for RSVP alone) or at the global level for all protocols.

RSVP graceful restart requires the following of a restarting router and the router's neighbors:

For the restarting router, RSVP graceful restart attempts to maintain the routes installed by RSVP and the allocated labels, so that traffic continues to be forwarded without disruption. This is done quickly enough to reduce or eliminate the impact on neighboring nodes.

The neighboring routers must have RSVP graceful restart helper mode enabled, thus allowing them to assist a router attempting to restart RSVP.

An object called `Restart_Cap` that is sent in RSVP hello messages advertises a node's restart capability. The neighboring node sends a `RecoverLabel` object to the restarting node to recover its forwarding state. This is essentially the old label that the restarting node advertised before the node went down.

The following lists the RSVP graceful restart behaviors, which vary depending on the configuration and on which features are enabled:

If you disable helper mode, JUNOS does not attempt to help a neighbor restart RSVP. Any information that arrives with a Restart_Cap object from a neighbor is ignored.

When you enable graceful restart under the routing instance configuration, the router can restart gracefully with the help of its neighbors. RSVP advertises a Restart_Cap object (RSVP-RESTART) in hello messages in which restart and recovery times are specified (neither value is 0).

If you explicitly disable RSVP graceful restart under the [protocols rsvp] hierarchy level, the Restart_Cap object is advertised with restart and recovery times specified as 0. The restart of neighboring routers is supported (unless helper mode is disabled), but the router itself does not preserve RSVP forwarding state and cannot recover its control state.

If after a restart RSVP realizes that no forwarding state has been preserved, the Restart_Cap object is advertised with restart and recovery times specified as 0.

If graceful-restart and helper mode are disabled, RSVP graceful restart is completely disabled. The router neither recognizes nor advertises the RSVP graceful restart objects.

You cannot explicitly configure values for the restart and recovery times.

Unlike other protocols, there is no way for RSVP to determine that it has completed a restart procedure, other than a fixed timeout. All RSVP graceful restart procedures are timer-based. A show rsvp version command might indicate that the restart is still in progress even if all RSVP sessions are back up and the routes are restored.

For ingress routers, no attempt is made to preserve labels across a restart.

Process the Restart_Cap Object

The following assumptions are made about a neighbor based on the Restart_Cap object (assuming that a control channel failure can be distinguished unambiguously from a node restart):

A neighbor that does not advertise the Restart_Cap object in its hello messages cannot assist a router with state or label recovery, nor can it perform an RSVP graceful restart.

After a restart, a neighbor advertising a Restart_Cap object with a restart time equal to any value and a recovery time equal to 0 has not preserved its forwarding state. When a recovery time equals 0, the neighbor is considered dead and any states related to this neighbor are purged, regardless of the value of the restart time.

After a restart, a neighbor advertising recovery-time with a value other than 0 can keep or has kept the forwarding state. If the local router is helping its neighbor with restart or recovery procedures, it sends a Recover_Label to this neighbor.

