Chapter 10

RSVP Overview

This chapter discusses the following topics:

RSVP Overview on page 236

RSVP Standards on page 237

JUNOS Software RSVP Protocol Implementation on page 238

RSVP Operation on page 238

RSVP Message Types on page 240

RSVP Reservation Styles on page 242

RSVP Refresh Reduction on page 243

MTU Signaling in RSVP on page 245

Link Protection on page 248

Node Protection on page 250

RSVP Graceful Restart on page 251

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 Internet Control Message Protocol (ICMP) and Internet Group Management Protocol (IGMP). RSVP runs as a separate software process in the JUNOS 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 RFCs and drafts.

The following 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* (the JUNOS software does not support the Null Service Object for maximum transmission unit [MTU] signaling in RSVP)

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 drafts also provides information about RSVP:

Fast Reroute Extensions to RSVP-TE for LSP Tunnels, Internet draft draft-ietf-mpls-rsvp-lsp-fastreroute-06.txt (expires November 2004)

Maximum Allocation Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering, Internet draft draft-ietf-tewg-diff-te-mam-00.txt

Russian Dolls Bandwidth Constraints Model for Diff-Serv-aware MPLS Traffic Engineering, Internet draft draft-ietf-tewg-diff-te-russian-01.txt

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

JUNOS Software 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 Multiprotocol Label Switching (MPLS) label-switched paths (LSPs). Supporting resource reservations over the Internet is only a secondary purpose of the JUNOS implementation. Since supporting resource reservations is secondary, the RSVP software does not support the following features:

- IP multicasting sessions.
- Traffic control—The software 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

The following sections describe RSVP operation:

- RSVP Operation Overview on page 239
- RSVP Authentication on page 239
- RSVP and IGP Hello Packets and Timers on page 240

RSVP Operation Overview

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, then 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 Authentication

JUNOS software supports both the RSVP authentication style described in RFC 2747 (allowing for multivendor compatibility) and the RSVP authentication style described in Internet draft draft-ietf-rsvp-md5-03.txt. The JUNOS software uses the authentication style described in Internet draft draft-ietf-rsvp-md5-08.txt by default. 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 and IGP Hello Packets and Timers

RSVP monitors the status of the IGP (ISIS or OSPF) neighbors and relies on the IGP protocols to detect when a node fails. If an IGP protocol declares a neighbor down (because hello packets are no longer being received), RSVP also brings down that neighbor. However, the IGP protocols and RSVP still act independently when bringing a neighbor up.

In standard RSVP, node failure detection occurs as a consequence of RSVP's soft-state timeout model. However, detection typically requires several minutes to time out the soft state. Hello packets allow the detection of the neighboring node's state changes more quickly.

In JUNOS software, RSVP hello packets are optional and are backward-compatible with RSVP implementations that do not support hello packets. For neighboring routers that do not support hello packets or on which RSVP hello is disabled, RSVP uses the soft-state timeout for loss detection and cannot benefit from fast IGP hello detection.

Configuring a short time for the ISIS or OSPF hello timers allows these protocols to detect node failures more quickly. RSVP also benefits from early detection by the IGP protocols. It is not necessary to explicitly configure a short RSVP hello timer. If you do configure the RSVP hello timer, you can configure a longer value and can still expect the failure of a neighboring router to be quickly detected by IGP.

Between hello-capable neighbors, hello packets are sent unicast toward each other. A loss of (2 x keep-multiplier + 1) consecutive hello packets causes the neighbor's state to go down, and all RSVP sessions to and from that neighbor are declared to be down.

By default, RSVP sends hello packets every 9 seconds. For information on how to configure the RSVP hello message timer, see "Configuring the RSVP Hello Interval" on page 262.

RSVP Message Types

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

Path Messages on page 241

Resv Messages on page 241

PathTear Messages on page 241

ResvTear Messages on page 241

PathErr Messages on page 241

ResvErr Messages on page 242

ResvConfirm Messages on page 242

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) x 1.5 x *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) x 1.5 x *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 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 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. PathErr messages are 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. ResvErr messages are 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 a 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, not a guarantee, of potential success.

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.

RSVP Refresh Reduction

RSVP relies on soft-state to maintain the path and reservation states on each router. If the corresponding refresh messages are not sent periodically, the states eventually time out and reservations are deleted. RSVP also sends its control messages as IP datagrams with no reliability guarantee. It relies on periodic refresh messages to handle the occasional loss of Path or Resv messages.

The RSVP refresh reduction extensions, based on RFC 2961, addresses the following problems that result from relying on periodic refresh messages to handle message loss:

Scalability—The scaling problem arises from the periodic transmission and processing overhead of refresh messages, which increases as the number of RSVP sessions increases.

Reliability and latency—The reliability and latency problem stems from the loss of nonrefresh RSVP messages or one-time RSVP messages such as Path Tear or Path Error. The time to recover from such a loss is usually tied to refresh interval and the keepalive timer.

The RSVP refresh reduction capability is advertised by enabling the refresh reduction (RR) capable bit in the RSVP common header. This bit is only significant between RSVP neighbors.

RSVP refresh reduction includes the following features:

- RSVP message bundling using the bundle message
- RSVP Message ID to reduce message processing overhead
- Reliable delivery of RSVP messages using Message ID, Message Ack, and Message Nack
- Summary refresh to reduce the amount of information transmitted every refresh interval

The RSVP refresh reduction specification (RFC 2961) allows you to enable some or all of the above capabilities on a router. It also describes various procedures that a router can use to detect the refresh reduction capabilities of its neighbor.

The JUNOS software supports all of the refresh reduction extensions, some of which can be selectively enabled or disabled. The JUNOS software supports Message ID and therefore can perform reliable message delivery only for Path and Resv messages.

For information on how to configure RSVP refresh reduction, see “Configuring RSVP Refresh Reduction” on page 259.

MTU Signaling in RSVP

The maximum transmission unit (MTU) is the largest size packet or frame, in bytes, that can be sent in a network. An MTU that is too large might cause retransmissions. Too small an MTU might cause the router to send and handle relatively more header overhead and acknowledgements. There are default values for MTUs associated with various protocols. You can also explicitly configure an MTU on an interface.

When an LSP is created across a set of links with different MTU sizes, the ingress router does not know what the smallest MTU is on the LSP path. By default, the maximum packet size for the LSP is based on the MTU for the outgoing interface for the LSP on the ingress router.

If this MTU is larger than the MTU of one of the intermediate links, traffic might be dropped, because MPLS packets cannot be fragmented. Also, the ingress router is not aware of this type of traffic loss, because the control plane for the LSP would still function normally.

To prevent this type of packet loss in MPLS LSPs, you can configure MTU signaling in RSVP. This feature is described in RFC 3209. Juniper supports the Integrated Services object for MTU signaling in RSVP. The Integrated Services object is described in RFCs 2210 and 2215. MTU signaling in RSVP is disabled by default.

To avoid packet loss due to MTU mismatches, the ingress router needs to do the following:

- Signal the MTU on the RSVP LSP—To prevent packet loss from an MTU mismatch, the ingress router needs to know what the smallest MTU value is along the path taken by the LSP. Once this MTU value is obtained, the ingress router can assign it to the LSP.
- Fragment packets—Using the assigned MTU value, packets that exceed the size of the MTU can be fragmented into smaller packets on the ingress router before they are sent over the RSVP LSP.

Once both MTU signaling and packet fragmentation have been enabled on an ingress router, any route resolving to an RSVP LSP on this router uses the signaled MTU value. For information on how to configure this feature, see “Configuring MTU Signaling in RSVP” on page 274.

The following sections describe how MTU signaling in RSVP works:

- How the Correct MTU Is Signaled in RSVP on page 246
- Determining an Outgoing MTU Value on page 246
- MTU Signaling in RSVP Limitations on page 247

How the Correct MTU Is Signaled in RSVP

How the correct MTU is signaled in RSVP varies depending on whether the network devices (for example, routers) explicitly support MTU signaling in RSVP or not.

If the network devices support MTU signaling in RSVP, the following occurs when you enable it:

The MTU is signaled from the ingress router to the egress router by means of the adspec object. Before forwarding this object, the ingress router enters the MTU value associated with the interface over which the path message is sent. At each hop in the path, the MTU value in the adspec object is updated to the minimum of the received value and the value of the outgoing interface.

The ingress router uses the traffic specification (Tspec) object to specify the parameters for the traffic it is going to send. The MTU value signaled for the Tspec object at the ingress router is the maximum MTU value (9192 bytes). This value does not change enroute to the egress router.

When the adspec object arrives at the egress router, the MTU value is correct for the path (meaning it is the smallest MTU value discovered). The egress router compares the MTU value in the adspec object to the MTU value in the Tspec object. It signals the smaller MTU using the flowspec object in the Resv message.

When the Resv object arrives at the ingress router, the MTU value in this object is used as the MTU for the next hops via the LSP.

In a network where there are devices that do not support MTU signaling in RSVP, you might have the following behaviors:

If the egress router does not support MTU signaling in RSVP, the MTU is set to the value of the outgoing interface on the ingress router. Setting the MTU to the value of the outgoing interface is the same as the default behavior when MTU signaling is not configured.

A Juniper Networks transit router that does not support MTU signaling in RSVP always propagates an MTU value of 1500 in the adspec object.

Determining an Outgoing MTU Value

The outgoing MTU value is the smaller of the value received in the adspec object compared to the the MTU value of the outgoing interface. The MTU value of the outgoing interface is determined as follows:

The outgoing MTU value is determined as follows:

If you configure an MTU value under the [family mpls] hierarchy level, this value is signaled.

If you do not configure an MTU, the inet MTU is signaled.

MTU Signaling in RSVP Limitations

The following are limitations to the MTU signaling in RSVP feature:

- Changes in the MTU value might cause a temporary loss of traffic in the following situations:

    - For link protection and node protection, the MTU of the bypass is only signaled at the time the bypass becomes active. During the time it takes for the new path MTU to be propagated, packet loss might occur because of an MTU mismatch.

    - For fast reroute, the MTU of the path is updated only after the detour becomes active, causing a delay in an update to the MTU at the ingress router. Until the MTU is updated, packet loss might occur if there is an MTU mismatch.

  In both cases, only packets that are larger than the detour or bypass MTU are lost.

- When an MTU is updated, it triggers a change in the next hop. Any change in the next hop causes the route statistics to be lost.

- The minimum MTU supported for MTU signaling in RSVP is 1488 bytes. This value prevents a false or incorrectly configured value from being used.

- For single-hop LSPs, the MTU value displayed by the `show` commands is the RSVP-signaled value. However, this MPLS value is ignored and the correct IP value is used.

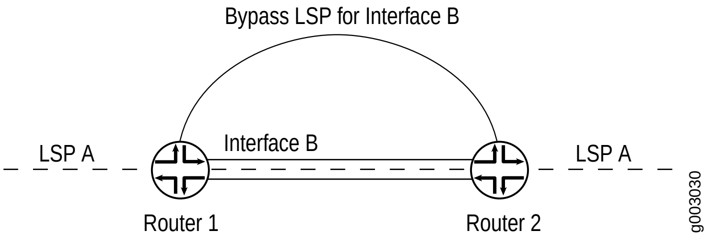
Link Protection

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 21, 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 21: Link Protection Creating 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.

NOTE: Link protection does not work on unnumbered interfaces.

To protect traffic over the entire route taken by an LSP, you should configure fast reroute. For more information, see “Configuring Fast Reroute” on page 77.

Fast Reroute, Node Protection, and Link Protection

The Internet draft draft-ietf-mpls-rsvp-lsp-fastreroute-00.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 the JUNOS software 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 the JUNOS software 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 4.

Table 4: One-to-One Backup Compared with Facility 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

Multiple Bypass LSPs

By default, link protection relies on a single bypass LSP to provide path protection for an interface. However, you can also specify multiple bypass LSPs to provide link protection for an interface. You can individually configure each of these bypass LSPs or create a single configuration for all of the bypass LSPs. If you do not configure the bypass LSPs individually, they all share the same path and bandwidth constraints.

The following algorithm describes how and when an additional bypass LSP is activated for an LSP:

1. If any currently active bypass can satisfy the requirements of the LSP (bandwidth, link protection, or node-link protection), the traffic is directed to that bypass.

2. If no active bypass LSP is available, scan through the manual bypass LSPs in first-in, first-out (FIFO) order, skipping those that are already active (each manual bypass can only be activated once). The first inactive manual bypass that can satisfy the requirements is activated and traffic is directed to that bypass.

3. If no manual bypass LSPs are available and if the max-bypasses statement activates multiple bypass LSPs for link protection, determine whether an automatically configured bypass LSP can satisfy the requirements. If an automatically configured bypass LSP is available and if the total number of active automatically configured bypass LSPs does not exceed the maximum bypass LSP limit (configured with the max-bypasses statement), activate another bypass LSP.

For information on how to configure multiple bypass LSPs for link protection, see "Configuring Bypass LSPs" on page 267.

Node Protection

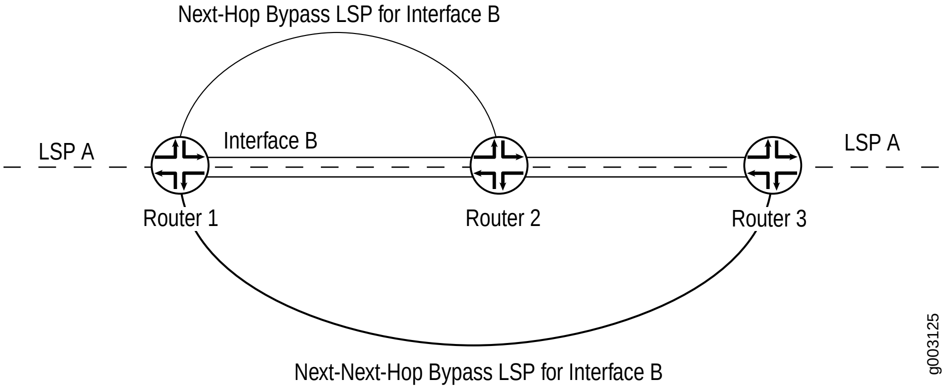
Node protection extends the capabilities of link protection. Link protection helps to ensure that traffic going over a specific interface to a neighboring router can continue to reach this router if that interface fails. Node protection ensures that traffic from an LSP traversing a neighboring router 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 22, both node protection and link protection are enabled on Interface B on Router 1. Both node protection 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 22: Node Protection Creating 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 can be enabled on both transit routers and ingress routers.

RSVP graceful restart is described in the following sections:

- RSVP Graceful Restart Standard on page 251
- RSVP Graceful Restart Terminology on page 251
- RSVP Graceful Restart Operation on page 252
- Processing the Restart Cap Object on page 253

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 (in milliseconds)—The default value is 60,000 milliseconds (1 minute). The restart time is advertised in the hello message. The time 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 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. RSVP graceful restart 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 Recover Label object to the restarting node to recover its forwarding state. This object 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, the JUNOS software 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 the 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.

Processing 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 object to this neighbor.