Implementing RSVP for MPLS-TE and MPLS O-UNI

The Multiprotocol Label Switching (MPLS) is a standards-based solution, driven by the Internet Engineering Task Force (IETF), devised to convert the Internet and IP backbones from best-effort networks into business-class transport media.

Resource Reservation Protocol (RSVP) is a signaling protocol that enables systems to request resource reservations from the network. RSVP processes protocol messages from other systems, processes resource requests from local clients, and generates protocol messages. As a result, resources are reserved for data flows on behalf of local and remote clients. RSVP creates, maintains, and deletes these resource reservations.

RSVP provides a secure method to control quality-of-service (QoS) access to a network.

MPLS Traffic Engineering (MPLS-TE) and MPLS Optical User Network Interface (MPLS O-UNI) use RSVP to signal label switched paths (LSPs).

Feature History for Implementing RSVP for MPLS-TE and MPLS O-UNI

Release	Modification
Release 2.0	This feature was introduced.
Release 3.0	No modification.
Release 3.2	Support was added for ACL-based prefix filtering.
Release 3.4.1	Support was added for RSVP authentication.
Release 3.9.0	The RSVP MIB feature was added.

Prerequisites for Implementing RSVP for MPLS-TE and MPLS O-UNI

These prerequisites are required to implement RSVP for MPLS-TE and MPLS O-UNI:

- You must be in a user group associated with a task group that includes the proper task IDs. The command reference guides include the task IDs required for each command. If you suspect user group assignment is preventing you from using a command, contact your AAA administrator for assistance.

- Either a composite mini-image plus an MPLS package, or a full image, must be installed.

Information About Implementing RSVP for MPLS-TE and MPLS O-UNI

To implement MPLS RSVP, you must understand the these concepts:

Related Topics

How to Implement RSVP Authentication, on page 20

Overview of RSVP for MPLS-TE and MPLS O-UNI

RSVP is a network control protocol that enables Internet applications to signal LSPs for MPLS-TE, and LSPs for O-UNI. The RSVP implementation is compliant with the IETF RFC 2205, RFC 3209, and OIF2000.125.7.

When configuring an O-UNI LSP, the RSVP session is bidirectional. The exchange of data between a pair of machines actually constitutes a single RSVP session. The RSVP session is established using an Out-Of-Band (OOB) IP Control Channel (IPCC) with the neighbor. The RSVP messages are transported over an interface other than the data interface.

RSVP supports extensions according to OIF2000.125.7 requirements, including:

- Generalized Label Request

- Generalized UNI Attribute

- UNI Session

- New Error Spec sub-codes

RSVP is automatically enabled on interfaces on which MPLS-TE is configured. For MPLS-TE LSPs with nonzero bandwidth, the RSVP bandwidth has to be configured on the interfaces. There is no need to configure RSVP, if all MPLS-TE LSPs have zero bandwidth . For O-UNI, there is no need for any RSVP configuration .

RSVP Refresh Reduction, defined in RFC 2961, includes support for reliable messages and summary refresh messages. Reliable messages are retransmitted rapidly if the message is lost. Because each summary refresh message contains information to refresh multiple states, this greatly reduces the amount of messaging needed to refresh states. For refresh reduction to be used between two routers, it must be enabled on both routers. Refresh Reduction is enabled by default.

Message rate limiting for RSVP allows you to set a maximum threshold on the rate at which RSVP messages are sent on an interface. Message rate limiting is disabled by default.

The process that implements RSVP is restartable. A software upgrade, process placement or process failure of RSVP or any of its collaborators, has been designed to ensure Nonstop Forwarding (NSF) of the data plane.

RSVP supports graceful restart, which is compliant with RFC 3473. It follows the procedures that apply when the node reestablishes communication with the neighbor's control plane within a configured restart time.

It is important to note that RSVP is not a routing protocol. RSVP works in conjunction with routing protocols and installs the equivalent of dynamic access lists along the routes that routing protocols calculate. Because of this, implementing RSVP in an existing network does not require migration to a new routing protocol.

Related Topics

LSP Setup

LSP setup is initiated when the LSP head node sends path messages to the tail node (see the RSVP Operation figure).

Figure 1: RSVP Operation

This figure illustrates an LSP setup for non-O-UNI applications. In the case of an O-UNI application, the RSVP signaling messages are exchanged on a control channel, and the corresponding data channel to be used is acquired from the LMP Manager module based on the control channel. Also the O-UNI LSP's are by default bidirectional while the MPLS-TE LSP's are uni-directional.
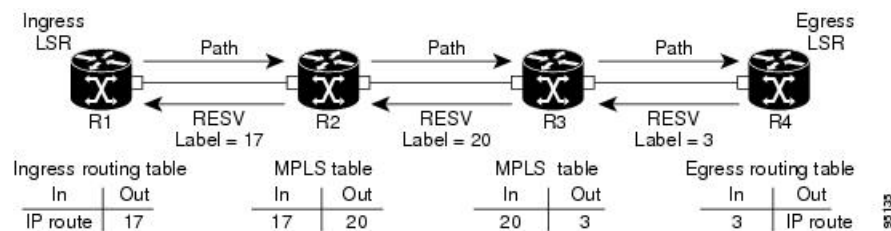
The Path messages reserve resources along the path to each node, creating Path soft states on each node. When the tail node receives a path message, it sends a reservation (RESV) message with a label back to the previous node. When the reservation message arrives at the previous node, it causes the reserved resources to be locked and forwarding entries are programmed with the MPLS label sent from the tail-end node. A new MPLS label is allocated and sent to the next node upstream.

When the reservation message reaches the head node, the label is programmed and the MPLS data starts to flow along the path.

High Availability

RSVP is designed to ensure nonstop forwarding under the following constraints:

• Ability to tolerate the failure of one or more MPLS/O-UNI processes.

• Ability to tolerate the failure of one RP of a 1:1 redundant pair.

• Hitless software upgrade.

The RSVP high availability (HA) design follows the constraints of the underlying architecture where processes can fail without affecting the operation of other processes. A process failure of RSVP or any of its collaborators does not cause any traffic loss or cause established LSPs to go down. When RSVP restarts, it recovers its signaling states from its neighbors. No special configuration or manual intervention are required. You may configure RSVP graceful restart, which offers a standard mechanism to recover RSVP state information from neighbors after a failure.

Graceful Restart

RSVP graceful restart provides a control plane mechanism to ensure high availability (HA), which allows detection and recovery from failure conditions while preserving nonstop forwarding services on the systems running Cisco IOS XR software.

RSVP graceful restart provides a mechanism that minimizes the negative effects on MPLS traffic caused by these types of faults:

• Disruption of communication channels between two nodes when the communication channels are separate from the data channels. This is called *control channel failure*.

• Control plane of a node fails but the node preserves its data forwarding states. This is called *node failure*.

The procedure for RSVP graceful restart is described in the "Fault Handling" section of RFC 3473, *Generalized MPLS Signaling, RSVP-TE Extensions*. One of the main advantages of using RSVP graceful restart is recovery of the control plane while preserving nonstop forwarding and existing labels.

Note   RSVP graceful restart feature is not supported when TE is running over multiple IGP instances which have different TE router-ids. This causes the TE tunnels to constantly flap.

Graceful Restart: Standard and Interface-Based

When you configure RSVP graceful restart, Cisco IOS XR software sends and expects node-id address based Hello messages (that is, Hello Request and Hello Ack messages). The RSVP graceful restart Hello session is not established if the neighbor router does not respond with a node-id based Hello Ack message.

You can also configure graceful restart to respond (send Hello Ack messages) to interface-address based Hello messages sent from a neighbor router in order to establish a graceful restart Hello session on the neighbor router. If the neighbor router does not respond with node-id based Hello Ack message, however, the RSVP graceful restart Hello session is not established.

Cisco IOS XR software provides two commands to configure graceful restart:

• **signalling hello graceful-restart**

• **signalling hello graceful-restart interface-based**

Note   By default, graceful restart is disabled. To enable interface-based graceful restart, you must first enable standard graceful restart. You cannot enable interface-based graceful restart independently.
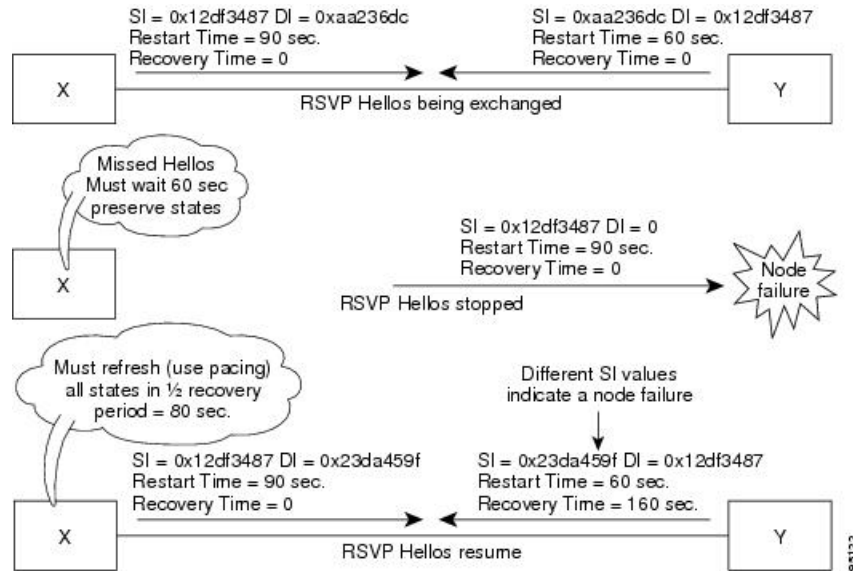
Related Topics

Graceful Restart: Figure

Figure 2: Node Failure with RSVP

This figure illustrates how RSVP graceful restart handles a node failure condition.

RSVP graceful restart requires the use of RSVP hello messages. Hello messages are used between RSVP neighbors. Each neighbor can autonomously issue a hello message containing a hello request object. A receiver that supports the hello extension replies with a hello message containing a hello acknowledgment (ACK) object. This means that a hello message contains either a hello Request or a hello ACK object. These two objects have the same format.

The restart cap object indicates a node's restart capabilities. It is carried in hello messages if the sending node supports state recovery. The restart cap object has the following two fields:

Restart Time

Time after a loss in Hello messages within which RSVP hello session can be reestablished. It is possible for a user to manually configure the Restart Time.

Recovery Time

Time that the sender waits for the recipient to re-synchronize states after the re-establishment of hello messages. This value is computed and advertised based on number of states that existed before the fault occurred.

For graceful restart, the hello messages are sent with an IP Time to Live (TTL) of 64. This is because the destination of the hello messages can be multiple hops away. If graceful restart is enabled, hello messages (containing the restart cap object) are send to an RSVP neighbor when RSVP states are shared with that neighbor.

Restart cap objects are sent to an RSVP neighbor when RSVP states are shared with that neighbor. If the neighbor replies with hello messages containing the restart cap object, the neighbor is considered to be graceful restart capable. If the neighbor does not reply with hello messages or replies with hello messages that do not contain the restart cap object, RSVP backs off sending hellos to that neighbor. If graceful restart is disabled, no hello messages (Requests or ACKs) are sent. If a hello Request message is received from an unknown neighbor, no hello ACK is sent back.

ACL-based Prefix Filtering

RSVP provides for the configuration of extended access lists (ACLs) to forward, drop, or perform normal processing on RSVP router-alert (RA) packets. Prefix filtering is designed for use at core access routers in order that RA packets (identified by a source/destination address) can be seamlessly forwarded across the core from one access point to another (or, conversely to be dropped at this node). RSVP applies prefix filtering rules only to RA packets because RA packets contain source and destination addresses of the RSVP flow.

Note   RA packets forwarded due to prefix filtering must not be sent as RSVP bundle messages, because bundle messages are hop-by-hop and do not contain RA. Forwarding a Bundle message does not work, because the node receiving the messages is expected to apply prefix filtering rules only to RA packets.

For each incoming RSVP RA packet, RSVP inspects the IP header and attempts to match the source/destination IP addresses with a prefix configured in an extended ACL. The results are as follows:

- If an ACL does not exist, the packet is processed like a normal RSVP packet.

- If the ACL match yields an explicit permit (and if the packet is not locally destined), the packet is forwarded. The IP TTL is decremented on all forwarded packets.

- If the ACL match yields an explicit deny, the packet is dropped.

If there is no explicit permit or explicit deny, the ACL infrastructure returns an implicit (default) deny. RSVP can be configured to drop the packet. By default, RSVP processes the packet if the ACL match yields an implicit (default) deny.

Related Topics

RSVP MIB

RFC 2206, RSVP Management Information Base Using SMIv2 defines all the SNMP MIB objects that are relevant to RSVP. By implementing the RSVP MIB, you can perform these functions:

- Specifies two traps (NetFlow and LostFlow) which are triggered when a new flow is created or deleted.

- Lets you use SNMP to access objects belonging to RSVP.

Related Topics

Information About Implementing RSVP Authentication

Before implementing RSVP authentication, you must configure a keychain first. The name of the keychain must be the same as the one used in the keychain configuration. For more information about configuring keychains, see *System Security Configuration Guide for Cisco CRS Routers*.

Note RSVP authentication supports only keyed-hash message authentication code (HMAC) type algorithms.

To implement RSVP authentication on Cisco IOS XR software, you must understand the following concepts:

RSVP Authentication Functions

You can carry out these tasks with RSVP authentication:

- Set up a secure relationship with a neighbor by using secret keys that are known only to you and the neighbor.

- Configure RSVP authentication in global, interface, or neighbor configuration modes.

- Authenticate incoming messages by checking if there is a valid security relationship that is associated based on key identifier, incoming interface, sender address, and destination address.

- Add an integrity object with message digest to the outgoing message.

- Use sequence numbers in an integrity object to detect replay attacks.

RSVP Authentication Design

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

The RSVP 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.

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 to validate the digital signature in the received RSVP message.

Network administrators manually configure a common key for each RSVP neighbor on the shared network.

The following reasons explain how to choose between global, interface, or neighbor configuration modes:

- Global configuration mode is optimal when a router belongs to a single security domain (for example, part of a set of provider core routers). A single common key set is expected to be used to authenticate all RSVP messages.

- Interface, or neighbor configuration mode, is optimal when a router belongs to more than one security domain. For example, a provider router is adjacent to the provider edge (PE), or a PE is adjacent to an edge device. Different keys can be used but not shared.

Global configuration mode configures the defaults for interface and neighbor interface modes. These modes, unless explicitly configured, inherit the parameters from global configuration mode, as follows:

- Window-size is set to 1.

- Lifetime is set to 1800.

- **key-source key-chain** command is set to none or disabled.

Related Topics

Global, Interface, and Neighbor Authentication Modes

You can configure global defaults for all authentication parameters including key, window size, and lifetime. These defaults are inherited when you configure authentication for each neighbor or interface. However, you can also configure these parameters individually on a neighbor or interface basis, in which case the global values (configured or default) are no longer inherited.

Note RSVP uses the following rules when choosing which authentication parameter to use when that parameter is configured at multiple levels (interface, neighbor, or global). RSVP goes from the most specific to least specific; that is, neighbor, interface, and global.

Global keys simplify the configuration and eliminate the chances of a key mismatch when receiving messages from multiple neighbors and multiple interfaces. However, global keys do not provide the best security.

Interface keys are used to secure specific interfaces between two RSVP neighbors. Because many of the RSVP messages are IP routed, there are many scenarios in which using interface keys are not recommended. If all keys on the interfaces are not the same, there is a risk of a key mismatch for the following reasons:

- When the RSVP graceful restart is enabled, RSVP hello messages are sent with a source IP address of the local router ID and a destination IP address of the neighbor router ID. Because multiple routes can exist between the two neighbors, the RSVP hello message can traverse to different interfaces.

- When the RSVP fast reroute (FRR) is active, the RSVP Path and Resv messages can traverse multiple interfaces.

- When Generalized Multiprotocol Label Switching (GMPLS) optical tunnels are configured, RSVP messages are exchanged with router IDs as the source and destination IP addresses. Since multiple control channels can exist between the two neighbors, the RSVP messages can traverse different interfaces.

Neighbor-based keys are particularly useful in a network in which some neighbors support RSVP authentication procedures and others do not. When the neighbor-based keys are configured for a particular neighbor, you are advised to configure all the neighbor's addresses and router IDs for RSVP authentication.

Related Topics

Security Association

A security association (SA) is defined as a collection of information that is required to maintain secure communications with a peer to counter replay attacks, spoofing, and packet corruption.

This table lists the main parameters that define a security association.

Table 1: Security Association Main Parameters

Parameter	Description
src	IP address of the sender.
dst	IP address of the final destination.
interface	Interface of the SA.
direction	Send or receive type of the SA.
Lifetime	Expiration timer value that is used to collect unused security association data.
Sequence Number	Last sequence number that was either sent or accepted (dependent of the direction type).
key-source	Source of keys for the configurable parameter.
keyID	Key number (returned form the key-source) that was last used.
digest	Algorithm last used (returned from the key-source).
Window Size	Specifies the tolerance for the configurable parameter. The parameter is applicable when the direction parameter is the receive type.
Window	Specifies the last *window size* value sequence number that is received or accepted. The parameter is applicable when the direction parameter is the receive type.

An SA is created dynamically when sending and receiving messages that require authentication. The neighbor, source, and destination addresses are obtained either from the IP header or from an RSVP object, such as a HOP object, and whether the message is incoming or outgoing.

When the SA is created, an expiration timer is created. When the SA authenticates a message, it is marked as recently used. The lifetime timer periodically checks if the SA is being used. If so, the flag is cleared and is cleaned up for the next period unless it is marked again.

This table shows how to locate the source and destination address keys for an SA that is based on the message type.

Table 2: Source and Destination Address Locations for Different Message Types

Message Type	Source Address Location	Destination Address Location
Path	HOP object	SESSION object
PathTear	HOP object	SESSION object

Message Type	Source Address Location	Destination Address Location
PathError	HOP object	IP header
Resv	HOP object	IP header
ResvTear	HOP object	IP header
ResvError	HOP object	IP header
ResvConfirm	IP header	CONFIRM object
Ack	IP header	IP header
Srefresh	IP header	IP header
Hello	IP header	IP header
Bundle	—	—

Related Topics

Key-source Key-chain

The key-source key-chain is used to specify which keys to use.

You configure a list of keys with specific IDs and have different lifetimes so that keys are changed at predetermined intervals automatically, without any disruption of service. Rollover enhances network security by minimizing the problems that could result if an untrusted source obtained, deduced, or guessed the current key.

RSVP handles rollover by using the following key ID types:

• On TX, use the youngest eligible key ID.

• On RX, use the key ID that is received in an integrity object.

For more information about implementing keychain management, see *System Security Configuration Guide for Cisco CRS Routers*.

Related Topics

Guidelines for Window-Size and Out-of-Sequence Messages

These guidelines are required for window-size and out-of-sequence messages:

- Default window-size is set to 1. If a single message is received out-of-sequence, RSVP rejects it and displays a message.

- When RSVP messages are sent in burst mode (for example, tunnel optimization), some messages can become out-of-sequence for a short amount of time.

- Window size can be increased by using the **window-size** command. When the window size is increased, replay attacks can be detected with duplicate sequence numbers.

Related Topics

Caveats for Out-of-Sequence

These caveats are listed for out-of-sequence:

- When RSVP messages traverse multiple interface types with different maximum transmission unit (MTU) values, some messages can become out-of-sequence if they are fragmented.

- Packets with some IP options may be reordered.

- Change in QoS configurations may lead to a transient reorder of packets.

- QoS policies can cause a reorder of packets in a steady state.

Because all out-of-sequence messages are dropped, the sender must retransmit them. Because RSVP state timeouts are generally long, out-of-sequence messages during a transient state do not lead to a state timeout.

How to Implement RSVP

RSVP requires coordination among several routers, establishing exchange of RSVP messages to set up LSPs. Depending on the client application, RSVP requires some basic configuration, as described in these topics:

Configuring Traffic Engineering Tunnel Bandwidth

To configure traffic engineering tunnel bandwidth, you must first set up TE tunnels and configure the reserved bandwidth per interface (there is no need to configure bandwidth for the data channel or the control channel).

Cisco IOS XR software supports two MPLS DS-TE modes: Prestandard and IETF.

✎

Note	For prestandard DS-TE you do not need to configure bandwidth for the data channel or the control channel. There is no other specific RSVP configuration required for this application. When no RSVP bandwidth is specified for a particular interface, you can specify zero bandwidth in the LSP setup if it is configured under RSVP interface configuration mode or MPLS-TE configuration mode.

Related Topics

Configuring a Prestandard DS-TE Tunnel

Configuring an IETF DS-TE Tunnel Using RDM

Configuring an IETF DS-TE Tunnel Using MAM

Confirming DiffServ-TE Bandwidth

Perform this task to confirm DiffServ-TE bandwidth.

In RSVP global and subpools, reservable bandwidths are configured per interface to accommodate TE tunnels on the node. Available bandwidth from all configured bandwidth pools is advertised using IGP. RSVP signals the TE tunnel with appropriate bandwidth pool requirements.

SUMMARY STEPS

1. **configure**
2. **rsvp**
3. **interface** *type interface-path-id*
4. **bandwidth** *total-bandwidth max-flow* **sub-pool** *sub-pool-bw*
5. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	
Step 2	**rsvp** **Example:** RP/0/RP0/CPU0:router(config)# **rsvp**	Enters RSVP configuration mode.
Step 3	**interface** *type interface-path-id* **Example:** RP/0/RP0/CPU0:router(config-rsvp)# **interface pos 0/2/0/0**	Enters interface configuration mode for the RSVP protocol.
Step 4	**bandwidth** *total-bandwidth max-flow* **sub-pool** *sub-pool-bw* **Example:**	Sets the reservable bandwidth, the maximum RSVP bandwidth available for a flow and the sub-pool bandwidth on this interface.

	Command or Action	Purpose
	RP/0/RP0/CPU0:router(config-rsvp-if)# **bandwidth 1000 100 sub-pool 150**	
Step 5	**commit**	

Related Topics

Differentiated Services Traffic Engineering

Bandwidth Configuration (MAM): Example, on page 29

Bandwidth Configuration (RDM): Example, on page 29

Configuring MPLS O-UNI Bandwidth

For this application you do not need to configure bandwidth for the data channel or the control channel. There is no other specific RSVP configuration needed for this application.

Enabling Graceful Restart

Perform this task to enable graceful restart for implementations using both node-id and interface-based hellos.

RSVP graceful restart provides a control plane mechanism to ensure high availability, which allows detection and recovery from failure conditions while preserving nonstop forwarding services.

SUMMARY STEPS

1. **configure**
2. **rsvp**
3. **signalling graceful-restart**
4. **signalling graceful-restart interface-based**
5. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**rsvp** **Example:** RP/0/RP0/CPU0:router(config)# **rsvp**	Enters the RSVP configuration mode.
Step 3	**signalling graceful-restart** **Example:** RP/0/RP0/CPU0:router(config-rsvp)# **signalling graceful-restart**	Enables the graceful restart process on the node.

	Command or Action	Purpose
Step 4	**signalling graceful-restart interface-based** **Example:** RP/0/RP0/CPU0:router(config-rsvp)# **signalling graceful-restart interface-based**	Enables interface-based graceful restart process on the node.
Step 5	**commit**	

Related Topics

Configuring ACL-based Prefix Filtering

Two procedures are provided to show how RSVP Prefix Filtering is associated:

Configuring ACLs for Prefix Filtering

Perform this task to configure an extended access list ACL that identifies the source and destination prefixes used for packet filtering.

Note The extended ACL needs to be configured separately using extended ACL configuration commands.

SUMMARY STEPS

1. **configure**
2. **rsvp**
3. **signalling prefix-filtering access-list**
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	
Step 2	**rsvp** **Example:** RP/0/RP0/CPU0:router(config)# **rsvp**	Enters the RSVP configuration mode.

	Command or Action	Purpose
Step 3	**signalling prefix-filtering access-list** **Example:** `RP/0/RP0/CPU0:router(config-rsvp)# signalling prefix-filtering access-list banks`	Enter an extended access list name as a string.
Step 4	**commit**	

Related Topics

Configuring RSVP Packet Dropping

Perform this task to configure RSVP to drop RA packets when the ACL match returns an implicit (default) deny.

The default behavior performs normal RSVP processing on RA packets when the ACL match returns an implicit (default) deny.

SUMMARY STEPS

1. **configure**
2. **rsvp**
3. **signalling prefix-filtering default-deny-action**
4. **commit**

DETAILED STEPS

|	Command or Action	Purpose
Step 1	**configure**	
Step 2	**rsvp** **Example:** `RP/0/RP0/CPU0:router(config)# rsvp`	Enters the RSVP configuration mode.
Step 3	**signalling prefix-filtering default-deny-action** **Example:** `RP/0/RP0/CPU0:router(config-rsvp)# signalling prefix-filtering default-deny-action`	Drops RA messages.
Step 4	**commit**	

Related Topics

Verifying RSVP Configuration

Figure 3: Sample Topology

This figure illustrates the topology.
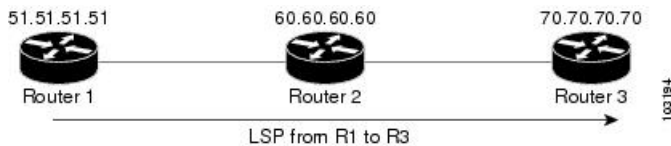
Perform the following steps to verify RSVP configuration.

SUMMARY STEPS

1. **show rsvp session**
2. **show rsvp counters messages summary**
3. **show rsvp counters events**
4. **show rsvp interface** *type interface-path-id* [**detail**]
5. **show rsvp graceful-restart**
6. **show rsvp graceful-restart** [**neighbors** *ip-address* | **detail**]
7. **show rsvp interface**
8. **show rsvp neighbor**

DETAILED STEPS

Step 1   **show rsvp session**

Verifies that all routers on the path of the LSP are configured with at least one Path State Block (PSB) and one Reservation State Block (RSB) per session.

Example:

```
RP/0/RP0/CPU0:router# show rsvp session

Type Destination Add DPort Proto/ExtTunID PSBs RSBs Reqs
 ---- -------------- ----- -------------- ----- ----- ----- LSP4
 172.16.70.70 6 10.51.51.51 1 1 0
```

In the example , the output represents an LSP from ingress (head) router 10.51.51.51 to egress (tail) router 172.16.70.70. The tunnel ID (also called the *destination port*) is 6.

Example:

```
If no states can be found for a session that should be up, verify the
     application (for example, MPLS-TE and O-UNI) to see if
     everything is in order. If a session has one PSB but no RSB, this indicates
     that either the Path message is not making it to the egress (tail) router or
```

> the reservation message is not making it back to the router R1 in question.

Go to the downstream router R2 and display the session information:

Example:

```
If R2 has no PSB, either the path message is not making it to the
    router or the path message is being rejected (for example, due to lack of
    resources). If R2 has a PSB but no RSB, go to the next downstream router R3
    to investigate. If R2 has a PSB and an RSB, this means the reservation is
    not making it from R2 to R1 or is being rejected.
```

Step 2 **show rsvp counters messages summary**

Verifies whether the RSVP message is being transmitted and received.

Example:

```
RP/0/RP0/CPU0:router# show rsvp counters messages summary

All RSVP Interfaces Recv Xmit Recv Xmit Path 0 25
    Resv 30 0 PathError 0 0 ResvError 0 1 PathTear 0 30 ResvTear 12 0
    ResvConfirm 0 0 Ack 24 37 Bundle 0 Hello 0 5099 SRefresh 8974 9012
    OutOfOrder 0 Retransmit 20 Rate Limited 0
```

Step 3 **show rsvp counters events**

Verifies how many RSVP states have expired. Because RSVP uses a soft-state mechanism, some failures will lead to RSVP states to expire due to lack of refresh from the neighbor.

Example:

```
RP/0/RP0/CPU0:router# show rsvp counters events

mgmtEthernet0/0/0/0 tunnel6 Expired Path states 0 Expired
    Path states 0 Expired Resv states 0 Expired Resv states 0 NACKs received 0
    NACKs received 0 POS0/3/0/0                        POS0/3/0/1 Expired
    Path states 0 Expired Path states 0 Expired Resv states 0 Expired Resv
    states 0 NACKs received 0 NACKs received 0 POS0/3/0/2
                            POS0/3/0/3 Expired Path states 0 Expired Path
    states 0 Expired Resv states 0 Expired Resv states 1 NACKs received 0 NACKs
    received 1
```

Step 4 **show rsvp interface** *type interface-path-id* [**detail**]

Verifies that refresh reduction is working on a particular interface.

Example:

```
RP/0/RP0/CPU0:router# show rsvp interface pos0/3/0/3 detail

INTERFACE: POS0/3/0/3 (ifh=0x4000D00). BW
    (bits/sec): Max=1000M. MaxFlow=1000M. Allocated=1K (0%). MaxSub=0.
    Signalling: No DSCP marking. No rate limiting. States in: 1. Max missed
    msgs: 4. Expiry timer: Running (every 30s). Refresh interval: 45s. Normal
    Refresh timer: Not running. Summary refresh timer: Running. Refresh
    reduction local: Enabled. Summary Refresh: Enabled (4096 bytes max).
    Reliable summary refresh: Disabled. Ack hold: 400 ms, Ack max size: 4096
    bytes. Retransmit: 900ms. Neighbor information: Neighbor-IP Nbor-MsgIds
```

```
States-out Refresh-Reduction Expiry(min::sec) -------------- --------------
---------- ------------------ ---------------- 64.64.64.65 1 1 Enabled
14::45
```

Step 5   **show rsvp graceful-restart**

Verifies that graceful restart is enabled locally.

Example:

```
RP/0/RP0/CPU0:router# show rsvp graceful-restart

Graceful restart: enabled Number of global
    neighbors: 1 Local MPLS router id: 10.51.51.51 Restart time: 60 seconds
    Recovery time: 0 seconds Recovery timer: Not running Hello interval: 5000
    milliseconds Maximum Hello miss-count: 3
```

Step 6   **show rsvp graceful-restart** [**neighbors** *ip-address* | **detail**]

Verifies that graceful restart is enabled on the neighbor(s). These examples show that neighbor 192.168.60.60 is not responding to hello messages.

Example:

```
RP/0/RP0/CPU0:router# show rsvp graceful-restart neighbors 192.168.60.60

Neighbor App State Recovery Reason
    Since LostCnt --------------- ----- ------ -------- ------------
    ------------------- -------- 192.168.60.60 MPLS INIT DONE N/A 12/06/2003
    19:01:49 0
RP/0/RP0/CPU0:router# show rsvp graceful-restart neighbors detail

Neighbor: 192.168.60.60 Source: 10.51.51.51
    (MPLS) Hello instance for application MPLS Hello State: INIT (for 3d23h)
    Number of times communications with neighbor lost: 0 Reason: N/A Recovery
    State: DONE Number of Interface neighbors: 1 address: 10.64.64.65 Restart
    time: 0 seconds Recovery time: 0 seconds Restart timer: Not running Recovery
    timer: Not running Hello interval: 5000 milliseconds Maximum allowed missed
    Hello messages: 3
```

Step 7   **show rsvp interface**

Verifies the available RSVP bandwidth.

Example:

```
RP/0/RP0/CPU0:router# show rsvp interface

Interface MaxBW MaxFlow Allocated MaxSub -----------
    -------- -------- --------------- -------- Et0/0/0/0 0 0 0 ( 0%) 0 PO0/3/0/0
    1000M 1000M 0 ( 0%) 0 PO0/3/0/1 1000M 1000M 0 ( 0%) 0 PO0/3/0/2 1000M 1000M
    0 ( 0%) 0 PO0/3/0/3 1000M 1000M 1K ( 0%) 0
```

Step 8   **show rsvp neighbor**

Verifies the RSVP neighbors.

Example:

```
RP/0/RP0/CPU0:router# show rsvp neighbor detail
Global Neighbor: 40.40.40.40 Interface Neighbor: 1.1.1.1
      Interface: POS0/0/0/0 Refresh Reduction: "Enabled" or "Disabled". Remote
      epoch: 0xXXXXXXXX Out of order messages: 0 Retransmitted messages: 0
      Interface Neighbor: 2.2.2.2 Interface: POS0/1/0/0 Refresh Reduction:
      "Enabled" or "Disabled". Remote epoch: 0xXXXXXXXX Out of order messages: 0
      Retransmitted messages: 0
```

Related Topics

Overview of RSVP for MPLS-TE and MPLS O-UNI , on page 2

Enabling RSVP Traps

With the exception of the RSVP MIB traps, no action is required to activate the MIBs. This MIB feature is automatically enabled when RSVP is turned on; however, RSVP traps must be enabled.

Perform this task to enable all RSVP MIB traps, NewFlow traps, and LostFlow traps.

SUMMARY STEPS

1. **configure**
2. **snmp-server traps rsvp lost-flow**
3. **snmp-server traps rsvp new-flow**
4. **snmp-server traps rsvp all**
5. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	
Step 2	**snmp-server traps rsvp lost-flow** **Example:** RP/0/RP0/CPU0:router(config)# **snmp-server traps rsvp lost-flow**	Sends RSVP notifications to enable RSVP LostFlow traps.
Step 3	**snmp-server traps rsvp new-flow** **Example:** RP/0/RP0/CPU0:router(config)# **snmp-server traps rsvp new-flow**	Sends RSVP notifications to enable RSVP NewFlow traps.
Step 4	**snmp-server traps rsvp all** **Example:** RP/0/RP0/CPU0:router(config)# **snmp-server traps**	Sends RSVP notifications to enable all RSVP MIB traps.

	Command or Action	Purpose
	`rsvp all`	
Step 5	commit	

Related Topics

How to Implement RSVP Authentication

There are three types of RSVP authentication modes—global, interface, and neighbor. These topics describe how to implement RSVP authentication for each mode:

Configuring Global Configuration Mode RSVP Authentication

These tasks describe how to configure RSVP authentication in global configuration mode:

Enabling RSVP Authentication Using the Keychain in Global Configuration Mode

Perform this task to enable RSVP authentication for cryptographic authentication by specifying the keychain in global configuration mode.

Note   You must configure a keychain before completing this task (see *System Security Configuration Guide for Cisco CRS Routers*).

SUMMARY STEPS

1. **configure**
2. **rsvp authentication**
3. **key-source key-chain** *key-chain-name*
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	configure	

Step 2	**rsvp authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# rsvp authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-auth)#`
		Enters RSVP authentication configuration mode.

	Command or Action	**Purpose**
Step 3	**key-source key-chain** *key-chain-name* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-auth)# key-source key-chain mpls-keys`	Specifies the source of the key information to authenticate RSVP signaling messages. *key-chain-name* Name of the keychain. The maximum number of characters is 32.
Step 4	**commit**	

Related Topics

Configuring a Lifetime for RSVP Authentication in Global Configuration Mode

Perform this task to configure a lifetime value for RSVP authentication in global configuration mode.

SUMMARY STEPS

1. **configure**
2. **rsvp authentication**
3. **life-time** *seconds*
4. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	

Step 2	**rsvp authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# rsvp authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-auth)#`
		Enters RSVP authentication configuration mode.

Step 3	**life-time** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-auth)# life-time 2000`	Controls how long RSVP maintains security associations with other trusted RSVP neighbors. *seconds* Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. Range is from 30 to 86400. The default value is 1800.
Step 4	**commit**	

Related Topics

Configuring the Window Size for RSVP Authentication in Global Configuration Mode

Perform this task to configure the window size for RSVP authentication in global configuration mode.

SUMMARY STEPS

1. **configure**
2. **rsvp authentication**
3. **window-size** N
4. **commit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**configure**	

Step 2	**rsvp authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# rsvp authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-auth)#`
		Enters RSVP authentication configuration mode.

Step 3	**window-size** N **Example:** `RP/0/RP0/CPU0:router(config-rsvp-auth)# window-size 33`	Specifies the maximum number of RSVP authenticated messages that can be received out-of-sequence. N Size of the window to restrict out-of-sequence messages. The range is from 1 to 64. The default value is 1, in which case all out-of-sequence messages are dropped.
Step 4	**commit**	

Related Topics

Configuring an Interface for RSVP Authentication

These tasks describe how to configure an interface for RSVP authentication:

Specifying the RSVP Authentication Keychain in Interface Mode

Perform this task to specify RSVP authentication keychain in interface mode.

You must configure a keychain first (see *System Security Configuration Guide for Cisco CRS Routers*).

SUMMARY STEPS

1. **configure**
2. **rsvp interface** *type interface-path-id*
3. **authentication**
4. **key-source key-chain** *key-chain-name*
5. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	

Step 2	**rsvp interface** *type interface-path-id*
	
	 Example:
	
	 `RP/0/RP0/CPU0:router(config)# rsvp interface POS 0/2/1/0`
	 `RP/0/RP0/CPU0:router(config-rsvp-if)#`
		Enters RSVP interface configuration mode.

Step 3	**authentication**
	
	 Example:
	
	 `RP/0/RP0/CPU0:router(config-rsvp-if)# authentication`
	 `RP/0/RP0/CPU0:router(config-rsvp-if-auth)#`
		Enters RSVP authentication configuration mode.

Step 4	**key-source key-chain** *key-chain-name* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-auth)# key-source key-chain mpls-keys`	Specifies the source of the key information to authenticate RSVP signaling messages. *key-chain-name* Name of the keychain. The maximum number of characters is 32.
Step 5	**commit**	

Related Topics

Configuring a Lifetime for an Interface for RSVP Authentication

Perform this task to configure a lifetime for the security association for an interface.

SUMMARY STEPS

1. **configure**
2. **rsvp interface** *type interface-path-id*
3. **authentication**

4. **life-time** *seconds*

5. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	
Step 2	**rsvp interface** *type interface-path-id* **Example:** `RP/0/RP0/CPU0:router(config)# `**`rsvp interface POS`** **`0/2/1/0`** `RP/0/RP0/CPU0:router(config-rsvp-if)#`	Enters RSVP interface configuration mode.

Step 3	**authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config-rsvp-if)#`
	`authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-if-auth)#`
		Enters RSVP authentication configuration mode.

Step 4	**life-time** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-auth)#` **`life-time 2000`**	Controls how long RSVP maintains security associations with other trusted RSVP neighbors. *seconds* Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. Range is from 30 to 86400. The default value is 1800.
Step 5	**commit**	

Related Topics

Configuring the Window Size for an Interface for RSVP Authentication

Perform this task to configure the window size for an interface for RSVP authentication to check the validity of the sequence number received.

SUMMARY STEPS

1. **configure**
2. **rsvp interface** *type interface-path-d*
3. **authentication**
4. **window-size** *N*
5. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	
Step 2	**rsvp interface** *type interface-path-d* **Example:** `RP/0/RP0/CPU0:router(config)# `**`rsvp interface POS 0/2/1/0`** `RP/0/RP0/CPU0:router(config-rsvp-if)#`	Enters RSVP interface configuration mode.
Step 3	**authentication** **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if)# `**`authentication`** `RP/0/RP0/CPU0:router(config-rsvp-if-auth)#`	Enters RSVP interface authentication configuration mode.
Step 4	**window-size** N **Example:** `RP/0/RP0/CPU0:router(config-rsvp-if-auth)# `**`window-size 33`**	Specifies the maximum number of RSVP authenticated messages that can be received out-of-sequence. N Size of the window to restrict out-of-sequence messages. The range is from 1 to 64. The default value is 1, in which case all out-of-sequence messages are dropped.
Step 5	**commit**	

Related Topics

Guidelines for Window-Size and Out-of-Sequence Messages, on page 11

RSVP Authentication by Using All the Modes: Example, on page 34

RSVP Authentication for an Interface: Example, on page 33

Configuring RSVP Neighbor Authentication

These tasks describe how to configure the RSVP neighbor authentication:

- Specifying the Keychain for RSVP Neighbor Authentication, on page 25

- Configuring a Lifetime for RSVP Neighbor Authentication, on page 26

- Configuring the Window Size for RSVP Neighbor Authentication, on page 27

Specifying the Keychain for RSVP Neighbor Authentication

Perform this task to specify the keychain RSVP neighbor authentication.

You must configure a keychain first (see *System Security Configuration Guide for Cisco CRS Routers*).

SUMMARY STEPS

1. **configure**
2. **rsvp neighbor** *IP-address* **authentication**
3. **key-source key-chain** *key-chain-name*
4. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	

Step 2	**rsvp neighbor** *IP-address* **authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# rsvp neighbor 1.1.1.1`
	`authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#`
		Enters neighbor authentication configuration mode. Use the **rsvp neighbor** command to activate RSVP cryptographic authentication for a neighbor.
		
		IP address
		
		IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces.
		
		authentication
		
		Configures the RSVP authentication parameters.

Step 3	**key-source key-chain** *key-chain-name*
	
	Example:
	
	`RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#`
	`key-source key-chain mpls-keys`
		Specifies the source of the key information to authenticate RSVP signaling messages.
		
		key-chain-name
		
		Name of the keychain. The maximum number of characters is 32.

Step 4	**commit**	

Related Topics

Configuring a Lifetime for RSVP Neighbor Authentication

Perform this task to configure a lifetime for security association for RSVP neighbor authentication mode.

SUMMARY STEPS

1. **configure**
2. **rsvp neighbor** *IP-address* **authentication**
3. **life-time** *seconds*
4. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	
Step 2	**rsvp neighbor** *IP-address* **authentication** **Example:** `RP/0/RP0/CPU0:router(config)# `**`rsvp neighbor 1.1.1.1`** **`authentication`** `RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#`	Enters RSVP neighbor authentication configuration mode. Use the **rsvp neighbor** command to specify a neighbor under RSVP. *IP address* IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces. **authentication** Configures the RSVP authentication parameters.
Step 3	**life-time** *seconds* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#` **`life-time 2000`**	Controls how long RSVP maintains security associations with other trusted RSVP neighbors. The argument specifies the *seconds* Length of time (in seconds) that RSVP maintains idle security associations with other trusted RSVP neighbors. Range is from 30 to 86400. The default value is 1800.
Step 4	**commit**	

Related Topics

Security Association, on page 9

RSVP Authentication Global Configuration Mode: Example, on page 33

Configuring the Window Size for RSVP Neighbor Authentication

Perform this task to configure the RSVP neighbor authentication window size to check the validity of the sequence number received.

SUMMARY STEPS

1. **configure**
2. **rsvp neighbor** *IP address* **authentication**
3. **window-size** *N*
4. **commit**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	**configure**	

	Command or Action	**Purpose**

Step 2	**rsvp neighbor** *IP address* **authentication**
	
	Example:
	
	`RP/0/RP0/CPU0:router(config)# rsvp neighbor 1.1.1.1 authentication`
	`RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)#`
		Enters RSVP neighbor authentication configuration mode. Use the **rsvp neighbor** command to specify a neighbor under RSVP.
		
		IP address
		
		   IP address of the neighbor. A single IP address for a specific neighbor; usually one of the neighbor's physical or logical (loopback) interfaces.
		
		authentication
		
		   Configures the RSVP authentication parameters.

Step 3	**window-size** *N* **Example:** `RP/0/RP0/CPU0:router(config-rsvp-nbor-auth)# window-size 33`	Specifies the maximum number of RSVP authenticated messages that is received out-of-sequence. *N* Size of the window to restrict out-of-sequence messages. The range is from 1 to 64. The default value is 1, in which case all out-of-sequence messages are dropped.
Step 4	**commit**	

Related Topics

Verifying the Details of the RSVP Authentication

To display the security associations that RSVP has established with other RSVP neighbors, use the **show rsvp authentication** command.

Eliminating Security Associations for RSVP Authentication

To eliminate RSVP authentication SA's, use the **clear rsvp authentication** command. To eliminate RSVP counters for each SA, use the **clear rsvp counters authentication** command.

Configuration Examples for RSVP

Sample RSVP configurations are provided for some of the supported RSVP features.

- #unique_180

- #unique_181

- #unique_182

Bandwidth Configuration (Prestandard): Example

The example shows the configuration of bandwidth on an interface using prestandard DS-TE mode. The example configures an interface for a reservable bandwidth of 7500, specifies the maximum bandwidth for one flow to be 1000 and adds a sub-pool bandwidth of 2000.

```
rsvp interface hundredGigE 0/0/0/0
 bandwidth 7500 1000 sub-pool 2000
```

Bandwidth Configuration (MAM): Example

The example shows the configuration of bandwidth on an interface using MAM. The example shows how to limit the total of all RSVP reservations on the hundredGigE 0/0/0/0 interface to 7500 kbps, and allow each single flow to reserve no more than 1000 kbps.

```
rsvp interface hundredGigE 0/0/0/0
 bandwidth mam 7500 1000
```

Related Topics

Confirming DiffServ-TE Bandwidth, on page 12

Differentiated Services Traffic Engineering

Bandwidth Configuration (RDM): Example

The example shows the configuration of bandwidth on an interface using RDM. The example shows how to limit the total of all RSVP reservations on the hundredGigE 0/0/0/0 interface to 7500 kbps, and allows each single flow to reserve no more than 1000 kbps.

```
rsvp interface hundredGigE 0/0/0/0
 bandwidth rdm 7500 1000
```

Related Topics

Confirming DiffServ-TE Bandwidth, on page 12

Differentiated Services Traffic Engineering

Refresh Reduction and Reliable Messaging Configuration: Examples

Refresh reduction feature as defined by RFC 2961 is supported and enabled by default. The examples illustrate the configuration for the refresh reduction feature. Refresh reduction is used with a neighbor only if the neighbor supports it also.

Refresh Interval and the Number of Refresh Messages Configuration: Example

The example shows how to configure the refresh interval to 30 seconds on POS 0/3/0/0 and how to change the number of refresh messages the node can miss before cleaning up the state from the default value of 4 to 6.

```
rsvp interface pos 0/3/0/0
 signalling refresh interval 30
 signalling refresh missed 6
```

Retransmit Time Used in Reliable Messaging Configuration: Example

The example shows how to set the retransmit timer to 2 seconds. To prevent unnecessary retransmits, the retransmit time value configured on the interface must be greater than the ACK hold time on its peer.

```
rsvp interface pos 0/4/0/1
 signalling refresh reduction reliable retransmit-time 2000
```

Acknowledgement Times Configuration: Example

The example shows how to change the acknowledge hold time from the default value of 400 ms, to delay or speed up sending of ACKs, and the maximum acknowledgment message size from default size of 4096 bytes. The example shows how to change the acknowledge hold time from the default value of 400 ms and how to delay or speed up sending of ACKs. The maximum acknowledgment message default size is from 4096 bytes.

```
rsvp interface pos 0/4/0/1
 signalling refresh reduction reliable ack-hold-time 1000
rsvp interface pos 0/4/0/1
 signalling refresh reduction reliable ack-max-size 1000
```

Note Ensure retransmit time on the peers' interface is at least twice the amount of the ACK hold time to prevent unnecessary retransmissions.

Summary Refresh Message Size Configuration: Example

The example shows how to set the summary refresh message maximum size to 1500 bytes.

```
rsvp interface pos 0/4/0/1
 signalling refresh reduction summary max-size 1500
```

Disable Refresh Reduction: Example

If the peer node does not support refresh reduction, or for any other reason you want to disable refresh reduction on an interface, the example shows how to disable refresh reduction on that interface.

```
rsvp interface pos 0/4/0/1
 signalling refresh reduction disable
```

Configure Graceful Restart: Examples

RSVP graceful restart is configured globally or per interface (as are refresh-related parameters). These examples show how to enable graceful restart, set the restart time, and change the hello message interval.

Enable Graceful Restart: Example

The example shows how to enable the RSVP graceful restart by default. If disabled, enable it with the following command.

```
rsvp signalling graceful-restart
```

Related Topics

Enable Interface-Based Graceful Restart: Example

The example shows how to enable the RSVP graceful restart feature on an interface.

```
RP/0/RP0/CPU0:router#configure
RP/0/RP0/CPU0:router(config-rsvp)#interface bundle-ether 17
RP/0/RP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart ?
  interface-based  Configure Interface-based Hello
RP/0/RP0/CPU0:router(config-rsvp-if)#signalling hello graceful-restart interface-based
RP/0/RP0/CPU0:router(config-rsvp-if)#
```

Related Topics

Change the Restart-Time: Example

The example shows how to change the restart time that is advertised in hello messages sent to neighbor nodes.

```
rsvp signalling graceful-restart restart-time 200
```

Change the Hello Interval: Example

The example shows how to change the interval at which RSVP graceful restart hello messages are sent per neighbor, and change the number of hellos missed before the neighbor is declared down.

```
  rsvp signalling hello graceful-restart refresh interval 4000
  rsvp signalling hello graceful-restart refresh misses 4
```

Configure ACL-based Prefix Filtering: Example

The example shows when RSVP receives a Router Alert (RA) packet from source address 1.1.1.1 and 1.1.1.1 is not a local address. The packet is forwarded with IP TTL decremented. Packets destined to 2.2.2.2 are dropped. All other RA packets are processed as normal RSVP packets.

```
show run ipv4 access-list
  ipv4 access-list rsvpacl
  10 permit ip host 1.1.1.1 any
  20 deny ip any host 2.2.2.2
  !
show run rsvp
  rsvp
  signalling prefix-filtering access-list rsvpacl
  !
```

Related Topics

Configuring ACLs for Prefix Filtering, on page 14

ACL-based Prefix Filtering, on page 6

Set DSCP for RSVP Packets: Example

The configuration example sets the Differentiated Services Code Point (DSCP) field in the IP header of RSVP packets.

```
rsvp interface pos0/2/0/1
 signalling dscp 20
```

Related Topics

Configuring RSVP Packet Dropping, on page 15

Overview of RSVP for MPLS-TE and MPLS O-UNI , on page 2

Enable RSVP Traps: Example

The example enables the router to send all RSVP traps:

```
configure
 snmp-server traps rsvp all
```

The example enables the router to send RSVP LostFlow traps:

```
configure
 snmp-server traps rsvp lost-flow
```

The example enables the router to send RSVP RSVP NewFlow traps:

```
configure
 snmp-server traps rsvp new-flow
```

Related Topics

Enabling RSVP Traps, on page 19

RSVP MIB, on page 6

Configuration Examples for RSVP Authentication

These configuration examples are used for RSVP authentication:

RSVP Authentication Global Configuration Mode: Example

The configuration example enables authentication of all RSVP messages and increases the default lifetime of the SAs.

```
rsvp
 authentication
  key-source key-chain default_keys
  life-time 3600
 !
!
```

Note   The specified keychain (default_keys) must exist and contain valid keys, or signaling will fail.

Related Topics

RSVP Authentication for an Interface: Example

The configuration example enables authentication of all RSVP messages that are being sent or received on one interface only, and sets the window-size of the SAs.

```
rsvp
 interface GigabitEthernet0/6/0/0
  authentication
   window-size 64
  !
 !
```

Note Because the key-source keychain configuration is not specified, the global authentication mode keychain is used and inherited. The global keychain must exist and contain valid keys or signaling fails.

Related Topics

RSVP Neighbor Authentication: Example

The configuration example enables authentication of all RSVP messages that are being sent to and received from only a particular IP address.

```
rsvp
 neighbor 10.0.0.1
  authentication
   key-source key-chain nbr_keys
  !
 !
!
```

Related Topics

RSVP Authentication by Using All the Modes: Example

The configuration example shows how to perform the following functions:

- Authenticates all RSVP messages.

- Authenticates the RSVP messages to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to nbr_keys, SA lifetime is set to 3600, and the default window-size is set to 1.

- Authenticates the RSVP messages not to or from 10.0.0.1 by setting the keychain for the **key-source key-chain** command to default_keys, SA lifetime is set to 3600, and the window-size is set 64 when using GigabitEthernet0/6/0/0; otherwise, the default value of 1 is used.

```
rsvp
 interface GigabitEthernet0/6/0/0
  authentication
   window-size 64
  !
 !
 neighbor 10.0.0.1
  authentication
   key-source key-chain nbr_keys
  !
```

```
 !
 authentication
  key-source key-chain default_keys
  life-time 3600
 !
 !
```

Note   If a keychain does not exist or contain valid keys, this is considered a configuration error because signaling fails. However, this can be intended to prevent signaling. For example, when using the above configuration, if the nbr_keys does not contain valid keys, all signaling with 10.0.0.1 fails.

Related Topics

Configuring the Window Size for RSVP Authentication in Global Configuration Mode, on page 22

Configuring the Window Size for an Interface for RSVP Authentication, on page 24

Configuring the Window Size for RSVP Neighbor Authentication, on page 27

Guidelines for Window-Size and Out-of-Sequence Messages, on page 11

Specifying the RSVP Authentication Keychain in Interface Mode, on page 22

Global, Interface, and Neighbor Authentication Modes, on page 8

Configuring a Lifetime for an Interface for RSVP Authentication, on page 23

RSVP Authentication Design, on page 7

Additional References

For additional information related to implementing GMPLS UNI, refer to the following references:

Related Documents

Related Topic	Document Title
GMPLS UNI commands	*GMPLS UNI Commands* module in *MPLS Command Reference for the Cisco CRS Router*
MPLS Traffic Engineering commands	*MPLS Traffic Engineering commands* module in *MPLS Command Reference for the Cisco CRS Router*
RSVP commands	*RSVP commands* module in *MPLS Command Reference for the Cisco CRS Router*
Getting started material	*Cisco IOS XR Getting Started Guide for the Cisco CRS Router*
Information about user groups and task IDs	*Configuring AAA Services* module in *System Security Configuration Guide for Cisco CRS Routers*

Standards

Standard	Title
OIF UNI 1.0	*User Network Interface (UNI) 1.0 Signaling Specification*

MIBs

MIBs	MIBs Link
—	To locate and download MIBs using Cisco IOS XR software, use the Cisco MIB Locator found at the following URL and choose a platform under the Cisco Access Products menu: http://cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml

RFCs

RFCs	Title
RFC 3471	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description*
RFC 3473	*Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions*
RFC 4208	*Generalized Multiprotocol Label Switching (GMPLS) User-Network Interface (UNI): Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Support for the Overlay Model*
RFC 4872	*RSVP-TE Extensions in Support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS) Recovery*
RFC 4874	*Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)*
RFC 6205	*Generalized Labels for Lambda-Switch-Capable (LSC) Label Switching Routers*

Technical Assistance

Description	Link
The Cisco Technical Support website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access even more content.	http://www.cisco.com/techsupport