Chapter 40

Resource Reservation Protocol (RSVP)

Introduction

This chapter describes the *Resource Reservation Protocol* (RSVP), support for RSVP on the router, and how to configure and manage the router to provide RSVP services.

Traditional IP-based protocols typically provide *best-effort Quality of Service* (QoS) to Internet applications. The next-hop forwarding approach of IP inherently creates delays or bursts of traffic. Conventional Internet applications like Telnet, FTP, and email tolerate best-effort delivery and busty traffic. However, traffic must be transmitted continuously (*streamed)* for voice and video applications. This requires a guaranteed QoS in terms of minimum available bandwidth and/or maximum delay over the entire path of the traffic. RSVP is a signalling protocol that provides this QoS by enabling receivers of traffic flows to reserve resources for the flow along the flow's path. RSVP is not a traffic delivery protocol or a routing protocol. It does not deliver the application's traffic to its destination or manage the routing of the data packets; this is left to existing transport and routing protocols.

In addition to RSVP, the router also implements an RSVP proxy agent that enables hosts and applications that do not support RSVP to take advantage of RSVP services. The proxy agent listens for specific application flows and generates the necessary resource reservations on behalf of the application.

Resource Reservation Protocol (RSVP)

RSVP enables the receiver of a traffic flow to make the resource reservations necessary to ensure that the receiver obtains the desired QoS for the traffic flow.

An RSVP *session* is an application data stream defined by the destination IP address, transport protocol ID and optional transport-protocol-specific port number. Within a session there are one or more senders. Packets from a single sender to a single destination belong to a *data flow.* A data flow is identified by a *filter spec*, which consists of the sender's IP address and an optional transport-protocol-specific port number (Figure 40-1 on page 40-3).

For example, in a video conferencing application, the session is the flow of data to one video conference client from all other clients participating in the conference. The traffic from one client to another within the conference is the data flow.

Each sender in an RSVP session periodically transmits *Path* messages to the same (unicast or multicast) destinations of the data flow (the receivers). Path messages follow the same route through the network as the data flow but are quite separate from the data flow (Figure 40-2 on page 40-3). Path messages contain the previous hop address, a session identifier, a sender template (the sender's IP address and port number), and a sender *TSpec*. The session and sender template together identify a data flow. The TSpec specifies the upper bounds of the characteristics of the traffic the sender is transmitting (e.g. maximum data rate, burstiness, maximum delay, etc.).

Figure 40-1: RSVP sessions with multicast receiver addresses

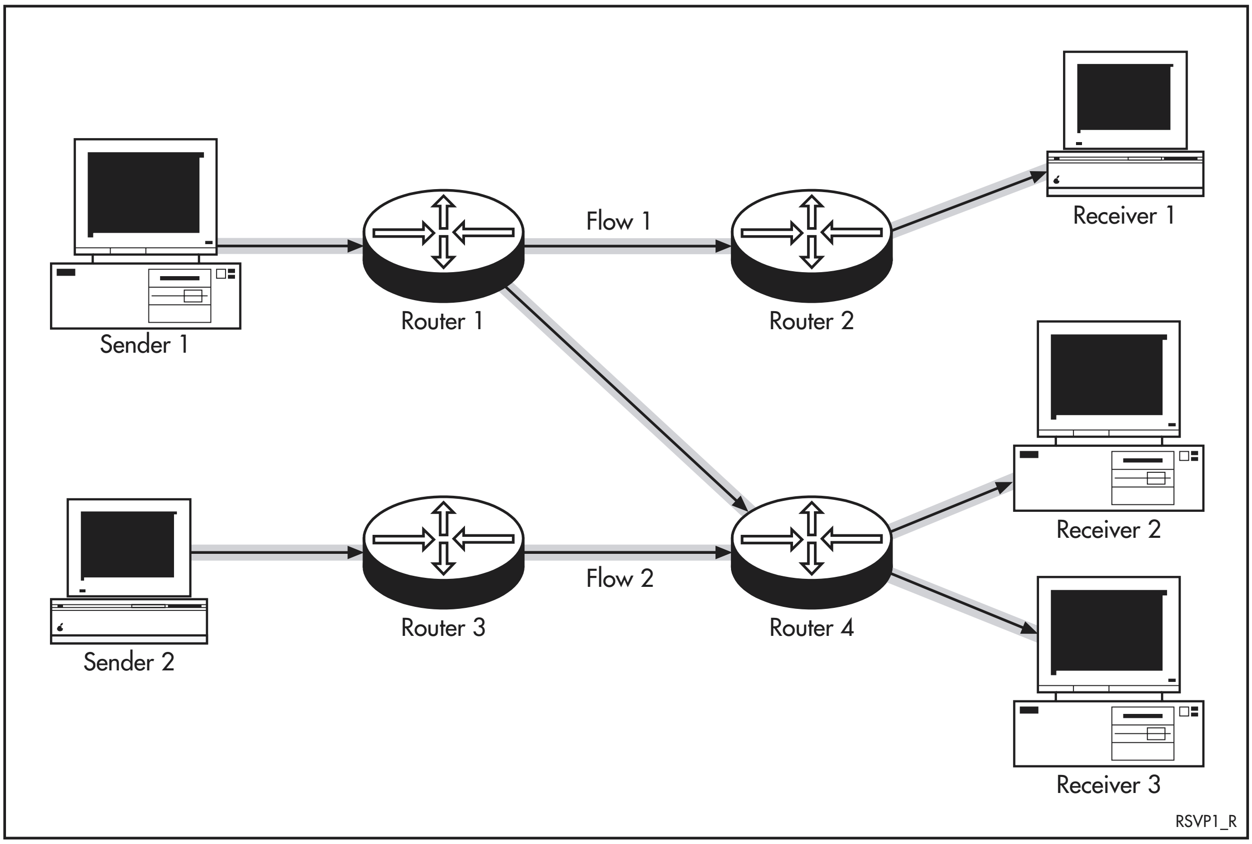
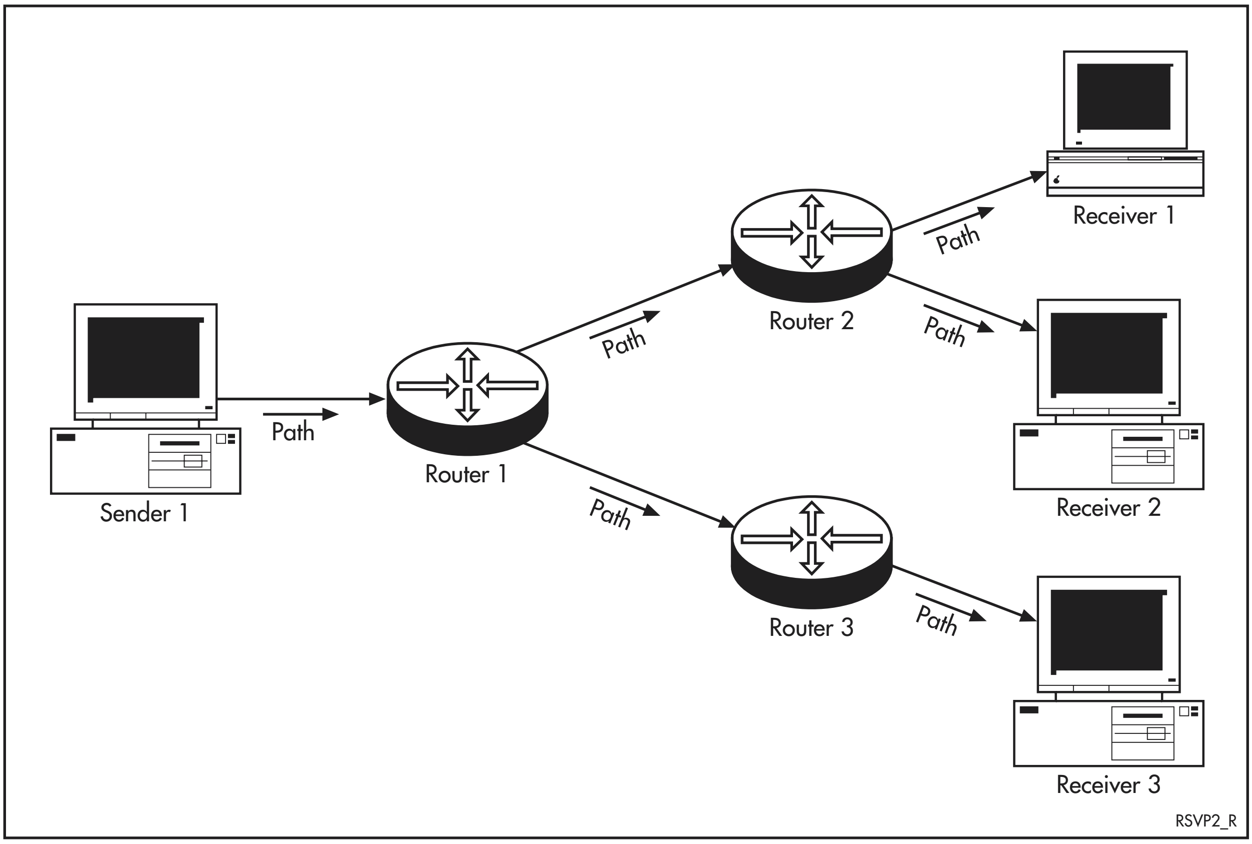


Figure 40-2: RSVP Path messages travel from the sender to the receiver



When an intermediate router on the data flow path receives a Path message it captures the data to create a path state. The router replaces the previous hop address in the Path message with its own IP address and forwards the Path message to the next hop on the route to the destination. By this mechanism the intermediate routers record the route back upstream from the receiver to the sender, and the information required to recognise the sender's data flow in order to provide it with the necessary resources.

When the receiver of the data flow receives a Path message it responds by transmitting a *Resv* message back to the previous hop address to actually request the resource reservation. The Resv message includes the *reservation style*, a *flowspec* and a *filter spec*.

The reservation style determines the flows in the session to which this reservation applies. Three styles are currently defined:

- The Fixed-Filter (FF) style reserves distinct resources for each sender.
- The Shared-Explicit (SE) style explicitly lists multiple senders who share the same resource reservation.
- The Wildcard-Filter (WF) style reserves resources that are shared by all flows in the session.

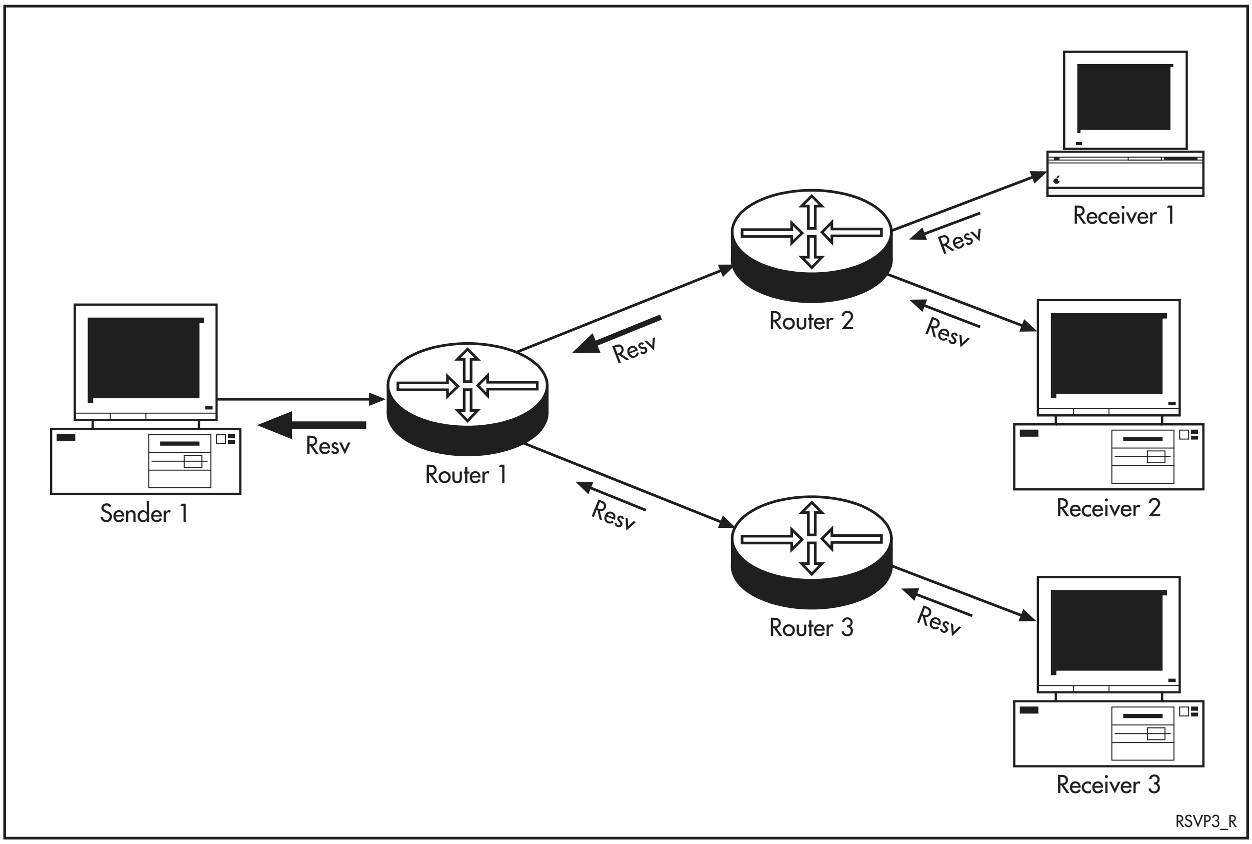
The flowspec defines the QoS that the receiver requires for the data flow, and the filter spec defines the packets in the data flow that are to receive the requested QoS. At each intermediate router, the filter spec is recorded and the necessary resources are allocated to provide the requested QoS. The Resv message is then forwarded to the previous hop address learned from the Path message. The filter spec and session (i.e. source and destination addresses) are used to identify packets belonging to the flow and therefore the resources to be allocated to those packets, and together with the QoS reservations made, define a reservation state.

Resv messages are transmitted upstream against the data flow, from the receiver to the sender (Figure 40-3 on page 40-5). In a multicast network, an intermediate router may receive Resv messages from one or more downstream routers or receivers. Multiple reservations of the same reservation style are merged to create a single reservation request that fulfils the QoS required by the merged requests. The merged reservation request is encapsulated in a single Resv message that is then forwarded by the router to the next upstream hop. In this manner, reservations from multiple multicast sender addresses are aggregated together as they traverse the network back to the sender.

RSVP uses a *soft state* approach to managing the state of reservations in routers and hosts. Path and Resv messages must be periodically re-transmitted to refresh the path and reservation states on each sender, intermediate router and receiver. If the state of a particular path or reservation on a host or router expires before refresh messages arrive, the path or reservation state is deleted and the associated resources are relinquished. Subsequent Path and Resv refresh messages generated by the host or router reflect the changed resource requirements, and the new resource requirements are propagated along the data flow path as Path and Resv messages traverse the network.

A host or router can remove a path or reservation state immediately, rather than waiting for the state to timeout, by sending a teardown message. A *PathTear* message travels downstream towards all receivers, deleting path state and all dependant reservation state along the way. A *ResvTear* message travels upstream towards all senders, deleting reservation state along the way.

Figure 40-3: RSVP Resv messages travel from the receiver to the sender



RSVP uses two error messages to respond to Path and Resv messages. *PathErr* messages are sent upstream to the sender that created the error and do not change path state in the routers they pass. Since a reservation error may be the result of merging a number of reservation requests, *ResvErr* messages are sent downstream to all responsible senders.

RSVP on the Router

The router’s implementation of RSVP conforms to the requirements for routers defined in RFC 2205. The router implements multicasting for routers as defined in RFCs 1112 and 1812, and supports the Controlled Load Service (RFC 2211) as the mechanism for delivering QoS.

RSVP is disabled by default and all traffic flows are given the standard best-effort QoS. RSVP is enabled or disabled by using the commands:

```
enable rsvp
disable rsvp
```

When RSVP is enabled, RSVP messages are processed and reservations are made. Disabling RSVP prevents RSVP messages being processed, and therefore prevents reservations from being made. The current state of RSVP can be displayed by using the command:

```
show rsvp
```

Each router interface that is to receive and process RSVP messages and accept reservation requests must be enabled. Interfaces are enabled or disabled for RSVP by using the commands:

```
enable rsvp interface={interface|dynamic}
disable rsvp interface={interface|dynamic}
```

where *interface* is an existing static IP interface and **dynamic** specifies any dynamic interfaces that may be created in future.

The RSVP attributes for an interface can be set by using the command:

```
set rsvp interface={interface|dynamic} [maxbandwidth=0..100]
    [encapsulation={udp|raw}]
```

The configuration and current state of interfaces used by RSVP can be displayed by using the command:

```
show rsvp interface
```

Debugging can be enabled or disabled on a per-interface basis by using the commands:

```
enable rsvp debug={all|event|packet|resv|state}
    interface={interface|dynamic}
disable rsvp debug={all|event|packet|resv|state}
    interface={interface|dynamic}
```

where EVENT displays information about RSVP state machine events, PKT displays information about RSVP messages sent and received, RESV displays information about reservations created, modified or deleted, and STATE displays information about RSVP control state blocks.

Information about Path messages sent and received, and current path states, can be displayed by using the command:

```
show rsvp path
```

Information about Resv messages sent and received, and current reservation states, can be displayed by using the command:

```
show rsvp resv
```

The following command displays counters for the RSVP process:

```
show rsvp counter
```

RSVP Proxy Agent

The router’s RSVP implementation includes an RSVP proxy agent that enables hosts and senders that do not directly support RSVP to take advantage of the QoS support provided by RSVP.

Two proxies are provided—a sender proxy for the sender of a traffic flow that is to have reservations made against it, and a receiver proxy for the receivers of the traffic.

The RSVP sender proxy listens for traffic with a destination address, protocol and port number matching a configured session profile. When such traffic is detected, the RSVP sender proxy creates an RSVP session and starts generating RSVP Path messages to the destination. Router along the data flow path process these Path messages as normal.

The RSVP receiver proxy listens for Path messages with a destination matching a configured session profile. When such Path messages are detected, the proxy agent starts generating the required Resv messages back towards the sender. Intermediate routers process these Resv messages as normal. Resv messages received from downstream routers and receivers are merged by RSVP with Resv messages generated by the local proxy agent and forwarded upstream as normal.

When IP traffic (traffic that is not RSVP protocol messages) is received, the proxy agent checks whether a Path proxy session already exists for this flow. If a session exists, the packet is forwarded using the reserved QoS. Otherwise the packet is compared with each enabled Path proxy to determine whether it falls within the parameters of the proxy entry. The protocol must match, the destination address and port must fall within the entry's session range, and the source address and port must fall within the entry's path ID range. If a match is found a new Path proxy session is created and Path messages are sent to the destination address.

In all cases, the original packet or PATH message is also processed as normal by IP or RSVP. In particular, the IP traffic is forwarded as normal, and the PATH message is passed on to its destination.

The router can act as a proxy agent for traffic that passes through the router. For example, the proxy agent does not listen promiscuously to LAN traffic and acts as a proxy agent for devices on the LAN that do not use the router to forward the traffic.

The RSVP proxy agent is enabled or disabled by using the commands:

```
enable rsvp proxy
disable rsvp proxy
```

The proxy agent is disabled by default.

Sender and receiver proxies are created by using the commands:

```
create rsvp proxy=name type=path path=ipadd[/port]
    [pmask=ipadd] tspec=tspec [timeout=10..1800]
    session=protocol,ipadd[/port[-port]] [smask=ipadd]

create rsvp proxy=name type=resv interface=interface
    style={wf|se|ff} flowspec=flowspec
    session=protocol,ipadd[/port[-port]] [smask=ipadd]
```

Existing sender and receiver proxies can be modified by using the commands:

```
set rsvp proxy=name [path=ipadd[/port]] [pmask=ipadd]
    [tspec=tspec] [timeout=10..1800] [session=protocol,ipadd[/
    port[-port]]] [smask=ipadd]
set rsvp proxy=name [flowspec=flowspec]
    [session=protocol,ipadd[/port[-port]]] [smask=ipadd]
```

A proxy is temporarily enabled or disabled by using the commands:

```
enable rsvp proxy[=name]
disable rsvp proxy[=name]
```

Disabling a proxy terminates activate sessions and stops the listening process from activating new sessions.

A proxy can be destroyed by using the command:

```
destroy rsvp proxy=name
```

A proxy can be reset by using the command:

```
reset rsvp proxy[=name]
```

This command terminates all activate sessions, and if the proxy is enabled, restarts the listening process. New sessions can still be initiated. Resetting a proxy that is currently enabled is equivalent to disabling and then enabling the proxy.

The current state and configuration of all sender and receiver proxies can be displayed by using the command:

```
show rsvp proxy[=name]
```

The following command displays activity counters for all configured sender and receiver proxies:

```
show rsvp proxy counter
```

Command Reference

This section describes the commands available on the router to enable, configure, control and monitor RSVP and the RSVP proxy agent.

RSVP requires IP routing and IP interfaces to be enabled and configured. See Chapter 14, Internet Protocol (IP) for the commands required to enable and configure IP routing and IP interfaces.

The shortest valid command is denoted by capital letters in the Syntax section. See “Conventions” on page xcv of Preface in the front of this manual for details of the conventions used to describe command syntax. See Appendix A, Messages for a complete list of error messages and their meanings.

create rsvp proxy

Syntax

```
CREate RSVP PROXy=name TYpe=PATH PATH=ipadd[/port]
    [PMask=ipadd] TSPEC=tspec [TIMeout=10..1800]
    SEssion=protocol,ipadd[/port[-port]] [SMask=ipadd]

CREate RSVP PROXy=name TYpe=RESV INTerface=interface
    STYLe={WF|SE|FF} FLOWspec=flowspec
    SEssion=protocol,ipadd[/port[-port]] [SMask=ipadd]
```

where:

- *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen ("-") and the underscore ("_").
- *ipadd* is an IP address in dotted decimal notation.
- *port* is an IP port number from 0 to 65535.
- *tspec* is an RSVP TSpec expressed in the format [t,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>]. All values are decimal numbers.

- *protocol* is an IP protocol name or number.
- *interface* is a valid interface name formed by concatenating a layer 2 interface type, an interface instance, and optionally a hyphen followed by a logical interface number from 0 to 3. If a logical interface is not specified, 0 is assumed.
- *flowspec* is an RSVP FlowSpec expressed in the format [cl,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>]. All values are decimal numbers.

Description This command adds an RSVP proxy entry for a sender of traffic or a receiver, determined by the value of the TYPE parameter. The valid parameters differ depending on the type of proxy entry. For PATH proxies, valid parameters are PATH, PMASK, TSPEC, TIMEOUT and SESSION. For RESV proxies, valid parameters are INTERFACE, STYLE, FLOWSPEC and SESSION.

For a proxy entry of type PATH, the router starts monitoring all IP traffic passing through the router for traffic that matches the PATH, PMASK, SESSION, and SMASK parameters. That is, traffic from the specified source IP address to the specified destination IP address, with the specified protocol and matching the range of port numbers specified. When such a traffic flow is detected, the router adds an entry for the exact traffic flow to its internal table and starts sending PATH messages containing the specified TSPEC to the specified destination.

The router normally receives and processes PATH messages passing through it. For a proxy entry of type RESV, an internal entry is created against which PATH messages are checked. When PATH messages whose destination characteristics match this entry are detected, an actual RSVP session is created that starts sending RESV messages to the source of the PATH messages.

The filter specification in proxy generated RESV messages is determined by the actual source address of the PATH messages. The flow specification in RESV messages is determined by both the information found in the PATH message's TSpec and the flow specification information in the proxy entry. The smaller of each of the parameters in the flow specification is used.

Note that a number of actual traffic flows may have the router acting as a proxy depending on the patterns given in the PATH and SESSION parameters.

The PROXY parameter specifies a name by which this proxy entry is known. The name must be unique among all RSVP proxy entries on this router, including those of type PATH and those of type RESV.

The TYPE parameter specifies the type of proxy to be added. A PATH proxy entry is a proxy entry for the sender of traffic. The router sends PATH messages as a result of finding a match to this proxy entry. A RESV proxy entry is a proxy entry for the receiver. The router sends RESV messages as a result of finding a match to this proxy entry. When a proxy entry has been created, the type cannot be changed.

The PATH parameter specifies the source of the traffic for which the router acts as a proxy in terms of an IP address and optionally a port number. If a port is not specified, all sender port numbers match.

The PMASK parameter specifies an IP address mask for use with the IP address specified by the PATH parameter. To determine whether a particular IP source address matches the path address, the source IP address is first ANDed with the address mask and then compared with the path address.

The SESSION parameter specifies the destination of the traffic flow for this proxy entry in terms of the IP protocol and destination IP address. A single TCP or UDP port number, or a range of port numbers may also be specified. If the protocol is not TCP or UDP, port numbers may not be specified. If the protocol is TCP or UDP and the port number is not specified, the default is the range 0 to 65535.

The SMASK parameter specifies an IP address mask for use with the IP address specified by the SESSION parameter. To determine whether a particular IP destination address matches the session address, the destination IP address is first ANDed with the address mask and then compared with the session address.

The TIMEOUT parameter specifies the timeout period, in seconds, that the router uses to stop performing proxy services for sessions based on this proxy entry. As long as traffic flows for the session, the router continues to send PATH messages. When traffic stops flowing and the timeout expires, the router ends the session and stops sending PATH messages. This initiates the end of the reservation. The default is 120.

The TSPEC parameter specifies the TSpec to place in PATH messages initiated from this proxy entry. The value is expressed in the format [t,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] where <*r*> is the token bucket rate in bytes of IP datagrams per second, <*d*> is the token bucket depth in bytes, <*p*> is the peak rate in bytes of IP datagrams per second, <*m*> is the minimum policed unit in bytes, and <*M*> is the maximum policed unit in bytes.

The INTERFACE parameter specifies the interface to which an RSVP proxy of type RESV applies. To act as a proxy, the router must be sending PATH messages over the interface, and the RESV messages created by the action of the proxy entry are processed in the router as having been received via this interface. Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

The interface must already exist. To see a list of all currently available interfaces, use the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The STYLE parameter specifies the style of reservation requests. If WF is specified, Wildcard-Filter style reservation requests are made that create a single reservation shared by flows from all upstream senders. If FF is specified, Fixed-Filter style reservation requests are made that create distinct reservations for data packets from a particular sender. If SE is specified, Shared-Explicit style reservation requests are made that create a single reservation shared by flows from explicitly selected upstream senders.

The FLOWSPEC parameter specifies a flow specification for this reservation style. The value is expressed in the format [cl,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] where <*r*> is the token bucket rate in bytes of IP datagrams per second, <*d*> is the token bucket depth in bytes, <*p*> is the peak rate in bytes of IP datagrams per second, <*m*> is the minimum policed unit in bytes, and <*M*> is the maximum policed unit in bytes.

Examples To add a proxy entry for a sender of traffic, use the command:

```
cre rsvp prox=video_23 ty=path path=192.168.1.1/1657
```

```
        se=udp,224.0.0.123/435 tspec=[t,4000,1000,4000,150,1600]
```

To add a proxy entry for a receiver of traffic, use the command:

```
    cre rsvp prox=dataflow3 ty=resv ses=udp,192.18.4.5/324
        int=eth0 styl=wf flow=[cl,2000,2000,4000,100,2000]
```

Related Commands

destroy rsvp proxy
disable rsvp proxy
enable rsvp proxy
reset rsvp proxy
set rsvp proxy
show rsvp proxy

destroy rsvp proxy

Syntax

`DESTroy RSVP PROXy=`*`name`*

where *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), hyphen ("-"), and underscore ("_").

Description

This command destroys an RSVP proxy entry. The entry must already exist on the router.

The PROXY parameter specifies the name of the proxy entry to be destroyed. The proxy entry must exist on the router and may be of type RESV or PATH.

Any proxy sessions that have been created as a result of a match to the proxy entry being destroyed are left running. To shut down the proxy sessions as well, use the **disable rsvp proxy** command on page 40-13 first.

Examples

To destroy a proxy entry named "video_23", use the command:

```
    dest rsvp prox=video_23
```

Related Commands

create rsvp proxy
disable rsvp proxy
enable rsvp proxy
show rsvp proxy

disable rsvp

Syntax

```
DISable RSVP
```

Description

This command disables RSVP on the router. RSVP messages are no longer processed, and reservations are not made. RSVP is disabled by default.

Examples

To disable RSVP, use the command:

```
    dis rsvp
```

Related Commands **enable rsvp**
show rsvp

disable rsvp debug

Syntax

```
DISable RSVP DEBug[={ALL|EVEnt|PKT|RESv|STAte}]
    [INTerface={interface|DYNAMIC}]
```

where *interface* is a valid interface name

Description This command disables the display of RSVP debugging messages.

The DEBUG parameter specifies the type of debug messages to be disabled. If ALL is specified, all debugging messages are disabled. If EVENT is specified, debug messages for RSVP state machine events are disabled. If PKT is specified, packet trace information for all RSVP messages sent and received on the interface is disabled. If RESV is specified, debug messages about any reservations made, changed or removed are disabled. If STATE is specified, debug messages about changes in RSVP control state blocks are disabled.

The INTERFACE parameter specifies an interface for which RSVP debugging is to be disabled. The specified interface must exist. If an interface is not specified then debugging is disabled on all interfaces. If DYNAMIC is specified then debugging is disabled on any dynamic interfaces created in the future. Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

To see a list of current valid interfaces, use the **show rsvp interface** command on page 40-23 or the **show interface** command on page 7-66 of Chapter 7, Interfaces.

Examples To disable RSVP state debugging on all interfaces, use the command:

```
dis rsvp deb=sta
```

Related Commands **enable rsvp debug**
show rsvp

disable rsvp interface

Syntax

```
DISable RSVP INTerface={interface|DYNAMIC}
```

where *interface* is a valid interface name

Description This command disables RSVP over the specified interface. An interface that is disabled does not accept reservation requests but still processes RSVP packets.

The INTERFACE parameter specifies the interface on which RSVP is to be disabled. The specified interface must exist. If DYNAMIC is specified, then RSVP is disabled on dynamic interfaces created in the future. Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

To see a list of current valid interfaces, use the **show rsvp interface** command on page 40-23 or the **show interface** command on page 7-66 of Chapter 7, Interfaces.

Examples To disable RSVP on interface PPP0, use the command:

```
dis rsvp int=ppp0
```

Related Commands

enable rsvp interface
show rsvp
show rsvp interface

disable rsvp proxy

Syntax

```
DISable RSVP PROXy[=name]
```

where *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), hyphen ("-") and underscore ("_").

Description This command disables the specified RSVP proxy entry, or the entire RSVP proxy agent in the router. If a proxy entry is specified it must already exist.

The PROXY parameter specifies the name of the proxy entry to be disabled. The proxy entry can be a PATH proxy entry or a RESV proxy entry. If a value is not specified, the entire RSVP proxy agent is disabled.

The effect of disabling one proxy entry is to terminate any existing proxy sessions that resulted from a match with the proxy entry, and to stop matches with the proxy entry from initiating new proxy sessions.

The effect of disabling the entire RSVP proxy agent is the same as individually disabling all currently enabled RSVP proxy entries: all proxy sessions are terminated and no new proxy sessions are initiated.

Examples To disable a proxy entry named "video_23", use the command:

```
dis rsvp prox=video_23
```

Related Commands

enable rsvp proxy
reset rsvp proxy
show rsvp proxy

enable rsvp

Syntax

```
ENAble RSVP
```

Description This command enables RSVP on the router. RSVP messages are processed and reservations are made. RSVP is disabled by default.

Interfaces must be enabled separately for reservations to be made on those interfaces.

Examples To enable RSVP, use the command:

```
ena rsvp
```

Related Commands

disable rsvp
enable rsvp interface
show rsvp

enable rsvp debug

Syntax

```
ENAble RSVP DEBug[={ALL|EVEnt|PKT|RESv|STAte}]
    [INTerface={interface|DYNAMIC}]
```

where *interface* is a valid interface name

Description This command enables the display of RSVP debugging messages for RSVP activity on the specified interface. Debugging information is sent to the asynchronous terminal port or Telnet session from which the command was executed.

The DEBUG parameter specifies the type of debug messages to be enabled. If ALL is specified, all debugging messages are displayed. If EVENT is specified, RSVP state machine events are displayed. If PKT is specified, packet trace information for all RSVP messages sent and received on the interface is displayed. If RESV is specified, any reservations made, changed or removed are displayed. If STATE is specified, debug messages about changes in RSVP control state blocks are displayed.

The INTERFACE parameter specifies the interface on which RSVP debugging is to be enabled. The specified interface must exist. If DYNAMIC is specified then debugging is enabled on any dynamic interfaces created in the future (until debugging is disabled). Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

To see a list of current valid interfaces, use the **show rsvp interface** command on page 40-23 or the **show interface** command on page 7-66 of Chapter 7, Interfaces.

Examples To enable RSVP packet debugging on interface PPP0, use the command:

```
ena rsvp deb=pkt int=ppp0
```

Related Commands **disable rsvp debug**
show rsvp

enable rsvp interface

Syntax

```
ENAble RSVP INTerface={interface|DYNAMIC}
```

where *interface* is a valid interface name

Description This command enables RSVP over the specified interface. An interface that is enabled accepts reservation requests. Interfaces that are disabled do not accept reservation requests but still process RSVP packets.

The INTERFACE parameter specifies the interface on which RSVP is to be enabled. The specified interface must exist. If DYNAMIC is specified, then RSVP is enabled on dynamic interfaces created in the future. Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

To see a list of current valid interfaces, use the **show rsvp interface** command on page 40-23 or the **show interface** command on page 7-66 of Chapter 7, Interfaces.

Examples To enable RSVP on interface PPP0, use the command:

```
ena rsvp int=ppp0
```

Related Commands **disable rsvp interface**
show rsvp
show rsvp interface

enable rsvp proxy

Syntax

```
ENAble RSVP PROXy[=name]
```

where *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), hyphen ("-") and underscore ("_").

Description This command enables the specified RSVP proxy entry, or the entire RSVP proxy agent. If a proxy entry is specified it must already exist.

The PROXY parameter specifies the name of the proxy entry to be enabled. The proxy entry can be a PATH proxy entry or a RESV proxy entry. If a value is not specified, the entire RSVP proxy agent is enabled.

The effect of enabling one proxy entry is to allow the router to initiate proxy sessions as a result of a match with the proxy entry. The RSVP proxy agent must also be enabled for this to occur.

The effect of enabling the entire RSVP proxy agent is to allow the router to initiate proxy sessions as a result of a match with any enabled proxy entry.

The RSVP proxy agent is disabled by default. RSVP proxy entries are enabled by default when they are created.

Examples To enable a proxy entry named "video_23", use the command:

```
ena rsvp prox=video_23
```

Related Commands

disable rsvp proxy
reset rsvp proxy
show rsvp proxy

reset rsvp proxy

Syntax `RESET RSVP PROXy[=name]`

where *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), hyphen ("-") and underscore ("_").

Description This command resets the specified RSVP proxy entry, or the entire RSVP proxy agent. If a proxy entry is specified it must already exist.

The PROXY parameter specifies the name of the proxy entry to be reset. The proxy entry can be a PATH proxy entry or a RESV proxy entry. If a value is not specified, the entire RSVP proxy agent is reset.

The effect of resetting one proxy entry is to terminate any existing proxy sessions that resulted from a match with the proxy entry. New proxy sessions can be initiated if the proxy entry is enabled.

The effect of resetting the entire RSVP proxy agent is the same as individually resetting all RSVP proxy entries: all proxy sessions are terminated. New proxy sessions are initiated for enabled proxy entries.

Examples To reset a proxy entry named "video_23", use the command:

```
reset rsvp prox=video_23
```

Related Commands

disable rsvp proxy
enable rsvp proxy
show rsvp proxy

set rsvp interface

Syntax

```
SET RSVP INTerface={interface|DYNAMIC}
   [MAXbandwidth=0..100] [ENCapsulation={UDP|RAW}]
```

where *interface* is a valid interface name

Description

This command sets RSVP attributes for the specified interface.

The INTERFACE parameter specifies the interface for which RSVP attributes are to be set. The specified interface must exist. If DYNAMIC is specified the attributes of any dynamic interfaces created in the future are set. Existing dynamic interfaces are not affected. Valid interfaces are:

- eth (e.g. eth0)
- PPP (e.g. ppp0)
- VLAN (e.g. vlan1)

To see a list of current valid interfaces, use the **show rsvp interface** command on page 40-23 or the **show interface** command on page 7-66 of Chapter 7, Interfaces.

The MAXBANDWIDTH parameter specifies the maximum amount of bandwidth, as a percentage of interface bandwidth, that may be allocated on the interface. The default is 75%.

The ENCAPSULATION parameter specifies the encapsulation type to use when transmitting RSVP messages over this interface. If RAW is specified, the raw IP encapsulation is used. If UDP is specified, the router transmits messages in the UDP encapsulation, as well as the raw IP encapsulation. The default is RAW.

Examples

To use UDP encapsulation for RSVP on interface PPP0, use the command:

```
set rsvp int=ppp0 enc=udp
```

Related Commands

disable rsvp interface
enable rsvp interface
show rsvp
show rsvp interface

set rsvp proxy

Syntax

```
SET RSVP PROXy=name [PATH=ipadd[/port]] [PMask=ipadd]
   [TSPEC=tspec] [TIMeout=10..1800]
   [SEssion=protocol,ipadd[/port[-port]]] [SMask=ipadd]

SET RSVP PROXy=name [FLOWspec=flowspec]
   [SEssion=protocol,ipadd[/port[-port]]] [SMask=ipadd]
```

where:

- *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen ("-") and the underscore ("_").
- *ipadd* is an IP address in dotted decimal notation.
- *port* is an IP port number from 0 to 65535.
- *tspec* is an RSVP TSpec expressed in the format [t,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>]. All values are decimal numbers.
- *protocol* is an IP protocol name or number.
- *flowspec* is an RSVP FlowSpec expressed in the format [cl,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>]. All values are decimal numbers.

Description This command change the parameters for the specified RSVP proxy entry. The proxy entry must already exist.

The PROXY parameter specifies the name of the proxy entry. The proxy entry must exist in the router and may be of type RESV or PATH.

The PATH parameter specifies the source of the traffic for which the router acts as a proxy in terms of an IP address and optionally a port number. If a port is not specified, all sender port numbers match.

The PMASK parameter specifies an IP address mask for use with the IP address specified by the PATH parameter. To determine whether a particular IP source address matches the path address, the source IP address is first ANDed with the address mask and then compared with the path address.

The SESSION parameter specifies the destination of the traffic flow for this proxy entry in terms of the IP protocol and destination IP address. A single TCP or UDP port number, or a range of port numbers may also be specified. If the protocol is not TCP or UDP, port numbers may not be specified. If the protocol is TCP or UDP and the port number is not specified, the default is the range 0 to 65535.

The SMASK parameter specifies an IP address mask for use with the IP address specified by the SESSION parameter. To determine whether a particular IP destination address matches the session address, the destination IP address is first ANDed with the address mask and then compared with the session address.

The TIMEOUT parameter specifies the timeout period in seconds that the router uses to stop performing proxy services for sessions based on this proxy entry. As long as traffic flows for the session, the router continues to send PATH messages. When traffic stops flowing and the timeout expires, the router ends the session and stops sending PATH messages. This initiates the end of the reservation. The default is 120.

The TSPEC parameter specifies the TSpec to place in PATH messages initiated from this proxy entry. The value is expressed in the format [t,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] where <*r*> is the token bucket rate in bytes of IP datagrams per second, <*d*> is the token bucket depth in bytes, <*p*> is the peak rate in bytes of IP datagrams per second, <*m*> is the minimum policed unit in bytes, and <*M*> is the maximum policed unit in bytes.

The FLOWSPEC parameter specifies a flow specification for this reservation style. The value is expressed in the format [cl,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] where <*r*> is the token bucket rate in bytes of IP datagrams per second, <*d*> is the token bucket depth in bytes, <*p*> is the peak rate in bytes of IP datagrams per second, <*m*> is the minimum policed unit in bytes, and <*M*> is the maximum policed unit in bytes.

Any parameter of an RSVP proxy entry may be changed, except the proxy type (PATH or RESV) and the reservation style for RESV type proxy entries. Only options that relate to the type of proxy can be specified. Existing RSVP proxy sessions initiated as a result of the proxy entry remain unchanged. New sessions created as a result of the proxy entry take on the new parameters.

Examples To change the timeout of a PATH proxy, use the command:

```
set rsvp prox=video_23 tim=200
```

Related Commands

create rsvp proxy
destroy rsvp proxy
disable rsvp proxy
enable rsvp proxy
reset rsvp proxy
show rsvp proxy

show rsvp

Syntax `SHow RSVP`

Description This command displays general RSVP configuration and status information (Figure 40-4 on page 40-20, Table 40-1 on page 40-20).

Figure 40-4: Example output from the **show rsvp** command

```
RSVP
----------------------------------------
Enabled ...................... No
Debug ........................ None
Debug Port ................... None
Number of Reservations ....... 0

Interfaces:
Number ....................... 1
Number Enabled ............... 0

State:
Number of Sessions ........... 0
Number of Paths .............. 0
Number of Reservations ....... 0
----------------------------------------
```

Table 40-1: Parameters in the output of the **show rsvp** command

Parameter	Meaning
Enabled	Whether RSVP is enabled.
Debug	List of the debug modes currently enabled; one or more of "None", "Resv", "Packet", "Event" and "State".
Debug Port	Port where debug messages are being sent or "None".
Number of reservations	Number of resource reservations currently active on the router.
Interfaces	Summary information about RSVP interfaces.
Number	Number of IP interfaces that RSVP knows about.
Number Enabled	Number of IP interfaces enabled for use by RSVP.
State	Summary RSVP state information.
Number of Sessions	Number of RSVP sessions currently active.
Number of Paths	Number of RSVP PATH states currently active.
Number of Reservations	Number of RSVP RESV states currently active.

Examples To display the current status of RSVP, use the command:

```
sh rsvp
```

Related Commands

disable rsvp
disable rsvp debug
disable rsvp interface
enable rsvp
enable rsvp debug
enable rsvp interface
show rsvp counter
show rsvp interface
show rsvp path
show rsvp resv

show rsvp counter

Syntax `SHow RSVP COUnter`

Description This command displays counters for RSVP (Figure 40-5 on page 40-21, Table 40-2 on page 40-21).

Figure 40-5: Example output from the **show rsvp counter** command

```
RSVP Counters
--------------------------------------------------------------------------------
Packets:
  inPath ....................... 2063       outPath ...................... 2113
  inResv ....................... 2087       outResv ...................... 2105
  inPathErr ....................... 0       outPathErr ...................... 0
  inResvErr ....................... 0       outResvErr ...................... 0
  inPathTear ...................... 0       outPathTear ..................... 0
  inResvTear ...................... 0       outResvTear ..................... 0
  inResvConf ...................... 0       outResvConf ..................... 0
  fwdResv ......................... 0       fwdPathErr ...................... 0
  fwdResvErr ...................... 0       fwdResvTear ..................... 0

Errors:
  errVersion ...................... 0       errChecksum ..................... 0
  errLength ....................... 0       errBadObject .................... 0
  errOrder ........................ 0       errUnknownObject ................ 0
  errNoSession .................... 0       errNoStyle ...................... 0
  errNoFlowDesc ................... 0       errBadStyle ..................... 0
  errDestPortConflict ............. 0       errSrcPortConflict .............. 0
  errNoSessionState ............... 0       errStyleConflict ................ 0
  errNoPathState .................. 0       errTTLZero ...................... 0
  errNoRoute ...................... 0
--------------------------------------------------------------------------------
```

Table 40-2: Parameters in the output of the **show rsvp counter** command

Parameter	Meaning
inPath	The number of PATH messages received by RSVP.
inResv	The number of RESV messages received by RSVP.
inPathErr	The number of PATH error messages received by RSVP.
inResvErr	The number of RESV error messages received by RSVP.
inPathTear	The number of PATH teardown messages received by RSVP.
inResvTear	The number of RESV teardown messages received by RSVP.
inResvConf	The number of RESV confirmation messages received by RSVP.
fwdResv	The number of RESV messages forwarded by RSVP.
fwdResvErr	The number of RESV error messages forwarded by RSVP.
outPath	The number of PATH messages sent by RSVP.
outResv	The number of RESV messages sent by RSVP.
outPathErr	The number of PATH error messages sent by RSVP.
outResvErr	The number of RESV error messages sent by RSVP.
outPathTear	The number of PATH teardown messages sent by RSVP.

Table 40-2: Parameters in the output of the **show rsvp counter** command (continued)

Parameter	Meaning
outResvTear	The number of RESV teardown messages sent by RSVP.
outResvConf	The number of RESV confirmation messages sent by RSVP.
fwdPathErr	The number of PATH error messages forwarded by RSVP.
fwdResvTear	The number of RESV teardown messages forwarded by RSVP.
errVersion	The number of RSVP messages received with an invalid version number.
errLength	The number of RSVP messages received with inconsistent object lengths.
errOrder	The number of RSVP messages received with objects in an incorrect order.
errNoSession	The number of RSVP messages received that do not contain a SESSION object.
errNoFlowDesc	The number of RSVP messages received that should contain at least one FLOWDESC object but do not.
errDestPortConflict	The number of RSVP messages received that have inconsistency between different representations of the session port.
errNoSessionState	The number of RSVP messages received for which a session control block cannot be found.
errNoPathState	The number of RSVP messages received for which a path control block cannot be found.
errNoRoute	The number of RSVP messages that could not be sent from the router due to a route to the destination not being found.
errChecksum	The number of RSVP messages received with an invalid checksum.
errBadObject	The number of RSVP messages received with invalid or unrecognised objects in them.
errUnknownObject	The number of RSVP messages received with unknown objects of a class that indicates that the router must reject the packet.
errNoStyle	The number of RSVP messages received that should contain a STYLE object but don't.
errBadStyle	The number of RSVP messages received with a STYLE object of an unrecognised type.
errSrcPortConflict	The number of RSVP messages received with an error or inconsistency in the source port.
errStyleConflict	The number of RSVP messages received with a conflict between the style in the message and the existing reservation state.
errTTLZero	The number of RSVP PATH TEAR messages received with a TTL of 0.

Examples To display RSVP counters, use the command:

```
sh rsvp cou
```

Related Commands
show rsvp
show rsvp interface
show rsvp path
show rsvp resv

show rsvp interface

Syntax `SHow RSVP INTerface`

Description This command displays the RSVP attributes and status of all IP interfaces on the router, as well as dynamic interfaces (Figure 40-6 on page 40-23, Table 40-3 on page 40-23).

Figure 40-6: Example output from the **show rsvp interface** command

```
RSVP Interfaces

                    Maximum       Reserved       No. Of
Interface  Enabled  Bandwidth(%)  Bandwidth(%)   Reservations  Debug     Encap
------------------------------------------------------------------------------
Dynamic      No        75             0              0         None      RAW
eth0         Yes       75             0              1         None      RAW
ppp0         Yes       75             0              0         None      RAW
------------------------------------------------------------------------------
```

Table 40-3: Parameters in the output of the **show rsvp interface** command

Parameter	Meaning
Interface	The name of the interface, or "Dynamic" for dynamically created interfaces.
Enabled	Whether the interface is enabled for RSVP.
Maximum Bandwidth	The maximum bandwidth of the interface available for RSVP reservations, expressed as a percentage of the interface bandwidth.
Reserved Bandwidth	The currently reserved bandwidth on the interface, expressed as a percentage of the interface bandwidth. "<1" is displayed for reservations that utilise between 0 and 1% of the interface bandwidth.
No. of Reservations	The current number of active reservations on the interface.
Debug	The debug status of the interface; one or more of "None", "Resv", "Packet" or "Pkt+Resv".
Encap	The RSVP encapsulation in use on the interface; one of "RAW" or "UDP".

Examples To display RSVP parameters for all IP interfaces, use the command:

```
sh rsvp int
```

Related Commands

disable rsvp interface
enable rsvp interface
set rsvp interface
show rsvp
show rsvp counter
show rsvp path
show rsvp resv

show rsvp path

Syntax `SHow RSVP PATH`

Description This command displays the RSVP sessions currently active on the router. These are derived from the reception of PATH messages from outside the router, or from local senders of PATH messages (Figure 40-7 on page 40-24, Table 40-4 on page 40-24).

Figure 40-7: Example output from the **show rsvp path** command

```
Path State

Session                    Senders              TSpec
------------------------------------------------------------------------------
192.168.3.1:20 TCP         192.168.1.1:20       [t 4000 1000 4000 100 2000]
224.0.0.5:1022 TCP         202.32.19.7          [t 2400 500 3000 345 1000]
                           202.32.19.8          [t 1000 500 1500 345 1000]
                           202.32.19.9          [t 2000 500 2400 345 1000]
------------------------------------------------------------------------------
```

Table 40-4: Parameters in the output of the **show rsvp path** command

Parameter	Meaning
Session	The session (destination of the traffic), expressed as the destination IP address in dotted decimal notation, the destination IP port and the IP protocol.
Senders	A list of the IP addresses of senders of traffic for this session.
TSpec	For each sender, the TSpec that appears in the PATH messages, expressed in the format [t *<r>* *<d>* *<p>* *<m>* *<M>*] where: *<r>* is the bucket rate, in bytes of IP datagrams per second *<d>* is the bucket depth, in bytes *<p>* is the peak rate, in bytes of IP datagrams per second *<m>* is the minimum policed unit, in bytes *<M>* is the maximum policed, units in bytes

Examples To display information about current RSVP sessions, use the command:

```
sh rsvp path
```

Related Commands show rsvp
show rsvp counter
show rsvp interface
show rsvp resv

show rsvp proxy

Syntax `SHow RSVP PROXy[=`*`name`*`]`

where *name* is a character string 1 to 15 characters long. Valid characters are uppercase and lowercase letters, digits (0–9), the hyphen ("-"), and the underscore ("_").

Description This command displays information about the specified or all RSVP proxy entries (Figure 40-8 on page 40-25, Table 40-5 on page 40-26).

The PROXY parameter specifies the name of a single proxy entry to display.

Figure 40-8: Example output from the **show rsvp proxy** command

```
RSVP proxy agent

Enabled ................ Yes

RSVP PATH proxy entries
--------------------------------------------------------------------------
video_23p
  Enabled ..... Yes
  Matches ..... 12
  Session ..... TCP,224.0.0.0/12345-54321
    Mask ...... 255.255.255.0
  Timeout ..... 50s
  PathID ...... 202.123.112.0/1128
    Mask ...... 255.255.255.255
  TSpec ....... [t,40000000000,1000000000,4000000000,150,2000]
--------------------------------------------------------------------------

RSVP RESV proxy entries
-----------------------------------------------------------------------------
video_1r
  Enabled ..... Yes
  Matches ..... 12
  Session ..... TCP,224.0.0.0/12345-54321
    Mask ...... 255.255.255.0
  Interface ... ppp0
  Style ....... WF
  Flow spec ... [cl,400000,400000,400000,150,1500]
-----------------------------------------------------------------------------
```

Table 40-5: Parameters in the output of the **show rsvp proxy** command

Parameter	**Meaning**
RSVP proxy agent	Information about the status of the RSVP proxy agent.
Enabled	Whether the RSVP proxy agent is enabled or disabled.
RSVP PATH proxy entries	Information about PATH proxy entries.
RSVP RESV proxy entries	Information about RESV proxy entries.
<*name*>	Name of the RSVP proxy entry.
Enabled	Whether the RSVP proxy entry is enabled or disabled.
Matches	Number of matches to this pattern and the number of proxy sessions initiated through this pattern.
Session	Session pattern for this proxy entry.
Session/Mask	Address mask for the session address.
Timeout	Timeout in seconds for this RSVP proxy entry. Valid for a PATH type proxy entry.
PathID	Path ID for this proxy entry. Valid for a PATH type proxy entry.
Path/Mask	Address mask for the path ID address. Valid for a PATH type proxy entry.
TSpec	TSpec for this proxy entry, expressed in the [t,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] format where: <*r*> is the token bucket rate in bytes of IP datagrams per second <*d*> is the token bucket depth in bytes <*p*> is the peak rate in bytes of IP datagrams per second <*m*> is the minimum policed unit in bytes <*M*> is the maximum policed unit in bytes. Valid for a PATH type proxy entry.
Interface	Interface from which RESV messages generated by this RESV proxy entry is seen to be arriving. In other words, the device for which this entry is acting as a proxy is reachable via this interface. Valid for a RESV type proxy entry.
Style	Reservation style for this proxy entry; one of `"WF"`, `"SE"` or `"FF"`. Valid for a RESV type proxy entry.
Flow spec	Flow specification for this proxy entry, expressed in the [cl,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] format where: <*r*> is the token bucket rate in bytes of IP datagrams per second <*d*> is the token bucket depth in bytes <*p*> is the peak rate in bytes of IP datagrams per second <*m*> is the minimum policed unit in bytes <*M*> is the maximum policed unit in bytes Valid for a RESV type proxy entry.

Examples To display all current proxy entries, use the command:

```
sh rsvp prox
```

Related Commands

- create rsvp proxy
- destroy rsvp proxy
- disable rsvp proxy
- enable rsvp proxy
- set rsvp proxy
- show rsvp proxy counter

show rsvp proxy counter

Syntax

```
SHow RSVP PROXy COUnter
```

Description

This command displays RSVP proxy counters (Figure 40-9 on page 40-27, Table 40-6 on page 40-27).

Figure 40-9: Example output from the **show rsvp proxy counter** command

```
RSVP proxy counters

Enabled PATH entries .... 1        Enabled RESV entries .... 3
Disabled PATH entries ... 0        Disabled RESV entries ... 0

Current PATH sessions ... 3        Current RESV sessions ... 2
Total PATH sessions ... 234        Total RESV sessions ... 342
```

Table 40-6: Parameters in the output of the **show rsvp proxy counter** command

Parameter	Meaning
Enabled PATH entries	Number of RSVP proxy entries of type PATH that are enabled.
Disabled PATH entries	Number of RSVP proxy entries of type PATH that are disabled.
Current PATH sessions	Number of current RSVP PATH proxy sessions active.
Total PATH sessions	Total number of RSVP PATH proxy sessions that have been initiated by the router.
Enabled RESV entries	Number of RSVP proxy entries of type RESV that are enabled.
Disabled RESV entries	Number of RSVP proxy entries of type RESV that are disabled.
Current RESV sessions	Number of current RSVP RESV proxy sessions active.
Total RESV sessions	Total number of RSVP RESV proxy sessions that have been initiated by the router.

Examples

To show the RSVP proxy counters, use the command:

```
sh rsvp prox cou
```

Related Commands

- create rsvp proxy
- destroy rsvp proxy
- disable rsvp proxy
- enable rsvp proxy
- set rsvp proxy
- show rsvp proxy

show rsvp resv

Syntax `SHow RSVP RESV`

Description This command displays information about current RSVP reservations (Figure 40-10 on page 40-28, Table 40-7 on page 40-28).

Figure 40-10: Example output from the **show rsvp resv** command

```
Reservation State

Session                    Interface  Style  Flow Spec
--------------------------------------------------------------------------------
192.168.3.1:20 TCP         ppp0        WF    [cl,4000,1000,Inf,100,1500]
--------------------------------------------------------------------------------
```

Table 40-7: Parameters in the output of the **show rsvp resv** command

Parameter	Meaning
Session	Session (destination of the traffic), expressed as the destination IP address in dotted decimal notation, the destination IP port and the IP protocol.
Interface	Interface to which this reservation applies.
Style	Style of reservation; one of "WF" (Wildcard Filter), "SE" (Shared Explicit) and "FF" (Fixed Filter).
Flow Spec	Flow specification for the reservation, expressed in the [cl,<*R*>,<*S*>,<*r*>,<*d*>,<*p*>,<*m*>,<*M*>] format where: <*R*> is the reservation rate, in bytes of IP datagrams per second (Guaranteed Service only) <*S*> is the slack term, in microseconds (Guaranteed Service only) <*r*> is the bucket rate, in bytes of IP datagrams per second <*d*> is the bucket depth, in bytes <*p*> is the peak rate, in bytes of IP datagrams per second <*m*> is the minimum policed unit, in bytes <*M*> is the maximum policed units, in bytes

Examples To display the current RSVP reservations, use the command:

```
sh rsvp resv
```

Related Commands show rsvp
show rsvp counter
show rsvp interface
show rsvp path