# RSVP

# From Wikipedia, the free encyclopedia

The **Resource Reservation Protocol** (**RSVP**) is a [transport layer](https://en.wikipedia.org/wiki/Transport_layer)[[1]](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol#cite_note-1) [protocol](https://en.wikipedia.org/wiki/Communications_protocol) designed to reserve resources across a [network](https://en.wikipedia.org/wiki/Computer_networking) for an [integrated services](https://en.wikipedia.org/wiki/Integrated_services) [Internet](https://en.wikipedia.org/wiki/Internet). RSVP operates over an [IPv4](https://en.wikipedia.org/wiki/IPv4) or [IPv6](https://en.wikipedia.org/wiki/IPv6) [Internet Layer](https://en.wikipedia.org/wiki/Internet_Layer) and provides receiver-initiated setup of resource reservations for [multicast](https://en.wikipedia.org/wiki/IP_Multicast) or [unicast](https://en.wikipedia.org/wiki/Unicast" \o "Unicast) data flows with scaling and robustness. It does not transport application data but is similar to a control protocol, like [Internet Control Message Protocol](https://en.wikipedia.org/wiki/Internet_Control_Message_Protocol) (ICMP) or [Internet Group Management Protocol](https://en.wikipedia.org/wiki/Internet_Group_Management_Protocol) (IGMP). RSVP is described in [RFC 2205](https://tools.ietf.org/html/rfc2205).

RSVP can be used by either [hosts](https://en.wikipedia.org/wiki/Node_(networking)) or [routers](https://en.wikipedia.org/wiki/Router_(computing)) to request or deliver specific levels of quality of service ([QoS](https://en.wikipedia.org/wiki/Quality_of_service" \o "Quality of service)) for application data streams or flows. RSVP defines how applications place reservations and how they can relinquish the reserved resources once the need for them has ended. RSVP operation will generally result in resources being reserved in each node along a path.

RSVP is not a [routing protocol](https://en.wikipedia.org/wiki/Routing_protocol) and was designed to interoperate with current and future routing protocols.

RSVP by itself is rarely deployed in telecommunications networks today[*[citation needed](https://en.wikipedia.org/wiki/Wikipedia:Citation_needed" \o "Wikipedia:Citation needed)*] but the [traffic engineering](https://en.wikipedia.org/wiki/Teletraffic_engineering) extension of RSVP, or [RSVP-TE](https://en.wikipedia.org/wiki/RSVP-TE), is becoming more widely accepted nowadays in many QoS-oriented networks. [Next Steps in Signaling](https://en.wikipedia.org/wiki/Next_Steps_in_Signaling) (NSIS) is a replacement for RSVP.

## Main attributes

1. RSVP requests resources for [simplex](https://en.wikipedia.org/wiki/Simplex_communication) flows: a traffic stream in only one direction from sender to one or more receivers.
2. RSVP is not a routing protocol but works with current and future routing protocols.
3. RSVP is receiver oriented: in that the receiver of a data flow initiates and maintains the resource reservation for that flow.
4. RSVP maintains "[soft state](https://en.wikipedia.org/wiki/Soft_state_(computer_science))" (the reservation at each node needs a periodic refresh) of the host and routers' resource reservations, hence supporting dynamic automatic adaptation to network changes.
5. RSVP provides several reservation styles (a set of reservation options) and allows for future styles to be added to protocol revisions to fit varied applications.
6. RSVP transports and maintains traffic and policy control parameters that are opaque to RSVP.

## History and related standards

The basic concepts of RSVP were originally proposed in [RSVP93] (Zhang, L., Deering, S., Estrin, D., Shenker, S., and D. Zappala, "RSVP: A New Resource ReSerVation Protocol", IEEE Network, September 1993).

RSVP is described in a series of RFC documents from the IETF:

* [RFC 2205](https://tools.ietf.org/html/rfc2205): The version 1 functional specification was described in [RFC 2205](https://tools.ietf.org/html/rfc2205) (Sept. 1997) by [IETF](https://en.wikipedia.org/wiki/Internet_Engineering_Task_Force). Version 1 describes the interface to admission (traffic) control that is based "only" on resource availability. Later [RFC2750](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol#RFC2750) extended the admission control support.
* [RFC 2210](https://tools.ietf.org/html/rfc2210) defines the use of RSVP with controlled-load [RFC 2211](https://tools.ietf.org/html/rfc2211) and guaranteed [RFC 2212](https://tools.ietf.org/html/rfc2212) QoS control services. More details in [Integrated Services](https://en.wikipedia.org/wiki/Integrated_Services). Also defines the usage and data format of the data objects (that carry resource reservation information) defined by RSVP in [RFC 2205](https://tools.ietf.org/html/rfc2205).
* [RFC 2211](https://tools.ietf.org/html/rfc2211) specifies the network element behavior required to deliver Controlled-Load services.
* [RFC 2212](https://tools.ietf.org/html/rfc2212) specifies the network element behavior required to deliver guaranteed QoS services.
* [RFC 2750](https://tools.ietf.org/html/rfc2750) describes a proposed extension for supporting generic [policy based](https://en.wikipedia.org/wiki/Policy) admission control in RSVP. The extension included a specification of policy objects and a description on handling policy events. (January 2000).
* [RFC 3209](https://tools.ietf.org/html/rfc3209), "RSVP-TE: Extensions to RSVP for LSP Tunnels" (December 2001).
* [RFC 3473](https://tools.ietf.org/html/rfc3473), "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions" (January 2003).
* [RFC 3936](https://tools.ietf.org/html/rfc3936), "Procedures for Modifying the **R**esource re**S**er**V**ation **P**rotocol (RSVP)" (October 2004), describes current best practices and specifies procedures for modifying RSVP.
* [RFC 4495](https://tools.ietf.org/html/rfc4495), "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow" (May 2006), extends RSVP to enable the bandwidth of an existing reservation to be reduced instead of tearing down the reservation.
* [RFC 4558](https://tools.ietf.org/html/rfc4558), "Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement" (June 2006).

## Key concepts

The two key concepts of RSVP reservation model are flowspec and filterspec:

### Flowspec

RSVP reserves resources for a flow. A flow is identified by the destination address, the protocol identifier, and, optionally, the destination port. In [multiprotocol label switching](https://en.wikipedia.org/wiki/Multiprotocol_label_switching) (MPLS) a flow is defined as a [label switched path](https://en.wikipedia.org/wiki/Label_switched_path) (LSP). For each flow RSVP also identifies the particular quality of service required by the flow although it does not understand the specific information of the flow QoS. This QoS specific information is called a *flowspec* and RSVP passes the *flowspec* from the application to the hosts and routers along the path. Those systems then analyse the *flowspec* to accept and reserve the resources. A *flowspec* consists of:

1. Service class
2. Reservation spec - defines the QoS
3. Traffic spec - describes the data flow

### Filterspec

The *filterspec* defines the set of packets that shall be affected by a *flowspec* (i.e. the data packets to receive the QoS defined by the flowspec). A *filterspec* typically selects a subset of all the packets processed by a node. The selection can depend on any attribute of a packet (e.g. the sender IP address and port).

The currently defined RSVP reservation styles are:

1. Fixed filter - reserves resources for a specific flow.
2. Shared explicit - reserves resources for several flows and all share the resources
3. Wildcard filter - reserves resources for a general type of flow without specifying the flow; all flows share the resources

An RSVP reservation request consists of a *flowspec* and a *filterspec* and the pair is called a *flowdescriptor*. The effects at the node of each *spec* are that while the *flowspec* sets the parameters of the packet scheduler at a node, the *filterspec* sets the parameters at the packet classifier.

## Messages

There are two primary types of messages:

* Path messages (*path*)

The *path* message is sent from the sender host along the data path and stores the *path state* in each node along the path.

The *path state* includes the IP address of the previous node, and some data objects:

1. *sender template* to describe the format of the sender data in the form of a Filterspec [[2]](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol#cite_note-2)
2. *sender tspec* to describe the traffic characteristics of the data flow
3. *adspec* that carries advertising data (see [RFC 2210](https://tools.ietf.org/html/rfc2210) for more details).

* Reservation messages (*resv*)

The *resv* message is sent from the receiver to the sender host along the reverse data path. At each node the IP destination address of the *resv* message will change to the address of the next node on the reverse path and the IP source address to the address of the previous node address on the reverse path.

The *resv* message includes the *flowspec* data object that identifies the resources that the flow needs.

The data objects on RSVP messages can be transmitted in any order. For the complete list of RSVP messages and date objects see [RFC 2205](https://tools.ietf.org/html/rfc2205).

## Operation

An RSVP host that needs to send a data flow with specific QoS will transmit an RSVP *path* message every 30 seconds that will travel along the unicast or multicast routes pre-established by the working routing protocol. If the *path* message arrives at a router that does not understand RSVP, that router forwards the message without interpreting the contents of the message and will not reserve resources for the flow.

Those who want to listen to them send a corresponding *resv* (short for "Reserve") message which then traces the path backwards to the sender. The *resv* message contains the flow specs. When a router receives the RSVP *resv* message it will:

1. Make a reservation based on the request parameters. For this the [admission control](https://en.wikipedia.org/wiki/Call_Admission_Control) and [policy control](https://en.wikipedia.org/w/index.php?title=Policy_control&action=edit&redlink=1) process the request parameters and can either instruct the packet classifier to correctly handle the selected subset of data packets or negotiate with the upper layer how the packet handling should be performed. If they cannot support the reservation being requested, they send a reject message to let the listener know about it.
2. Forward the request upstream (in the direction of the sender). At each node the *resv* message *flowspec* can be modified by a forwarding node (e.g. in the case of a multicast flow reservation the reservations requests can be merged).
3. The routers then store the nature of the flow, and also police it. This is all done in soft state, so if nothing is heard for a certain length of time, then the reader will time out and the reservation will be cancelled. This solves the problem if either the sender or the receiver crash or are shut down incorrectly without first cancelling the reservation. The individual routers may, at their option, police the traffic to check that it conforms to the flow specs.

The *resv* message also has FilterSpec object; it defines the packets that will receive the requested QoS defined in the flowspec. A simple filter spec could be just the sender’s IP address and optionally its UDP or TCP port.

## Other features

* Integrity - RSVP messages are appended with a message digest created by combining the message contents and a shared key using a message digest algorithm (commonly [MD5](https://en.wikipedia.org/wiki/MD5)). The key can be distributed and confirmed using 2 message types: *integrity challenge request* and *integrity challenge response*.
* Error reporting - when a node detects an error, an error message is generated with an error code and is propagated upstream on the reverse path to the sender.
* Information on RSVP flow - two types of diagnostic messages allow a network operator to request the RSVP state information on a specific flow.
* Diagnostic facility - An extension to the standard which allows a user to collect information about the RSVP state along a path. [RFC2745 - RSVP Diagnostic Messages](http://ietfreport.isoc.org/rfc/rfc2745.txt)

## References

* 1. [**Jump up^**](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol#cite_ref-1) Garrett, Aviva; Drenan, Gary; Morris, Cris (2002).[*Juniper Networks Field Guide and Reference*](https://books.google.com/books?id=pIIu7IbUwIcC&pg=PA583). p. 583.
  2. [**Jump up^**](https://en.wikipedia.org/wiki/Resource_Reservation_Protocol#cite_ref-2) <https://tools.ietf.org/html/rfc2205#section-2>
* John Evans; Clarence Filsfils (2007).Deploying IP and MPLS QoS for Multiservice Networks: Theory and Practice. Morgan Kaufmann.[*ISBN*](https://en.wikipedia.org/wiki/International_Standard_Book_Number) [*0-12-370549-5*](https://en.wikipedia.org/wiki/Special:BookSources/0-12-370549-5).

## External links

* [*"Resource Reservation Protocol"*](http://docwiki.cisco.com/wiki/Resource_Reservation_Protocol). Cisco*. Retrieved 2011-02-16*.
* Naveen Joy (2002-06-17),[*RSVP provides quality of service*](http://www.networkworld.com/news/tech/2002/0617tech.html), Network World*, retrieved 2012-02-14*
* [*"RSVP Project"*](http://www.isi.edu/div7/rsvp/rsvp.html). USC Information Science Institute*. Retrieved 2011-02-16*.

### RFCs

* [RFC 2205](https://tools.ietf.org/html/rfc2205)
* [RFC 2210](https://tools.ietf.org/html/rfc2210)
* [RFC 2211](https://tools.ietf.org/html/rfc2211)
* [RFC 2212](https://tools.ietf.org/html/rfc2212)

## RSVP Configuration

Below is sample of RSVP configuration.

Here goes RSVP config

Here goes RSVP config

Here goes RSVP config

Configuration RSVP Configuration