

A Resource Reservation Protocol (RSVP) Extension for the
Reduction of Bandwidth of a Reservation Flow

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document proposes an extension to the Resource Reservation Protocol (RSVPv1) to reduce the guaranteed bandwidth allocated to an existing reservation. This mechanism can be used to affect individual reservations, aggregate reservations, or other forms of RSVP tunnels. This specification is an extension of [RFC 2205](#).

Table of Contents

1. Introduction	2
1.1. Conventions Used in This Document	4
2. Individual Reservation Reduction Scenario	4
3. RSVP Aggregation Overview	6
3.1. RSVP Aggregation Reduction Scenario	8
4. Requirements for Reservation Reduction	9
5. RSVP Bandwidth Reduction Solution	10
5.1. Partial Preemption Error Code	11
5.2. Error Flow Descriptor	11
5.3. Individual Reservation Flow Reduction	11
5.4. Aggregation Reduction of Individual Flows	12
5.5. RSVP Flow Reduction Involving IPsec Tunnels	12
5.6. Reduction of Multiple Flows at Once	13
6. Backwards Compatibility	13
7. Security Considerations	14
8. IANA Considerations	15
9. Acknowledgements	15
10. References	15
10.1. Normative References	15
10.2. Informative References	16
Appendix A. Walking through the Solution	17

1. Introduction

This document proposes an extension to the Resource Reservation Protocol (RSVP) [1] to allow an existing reservation to be reduced in allocated bandwidth in lieu of tearing that reservation down when some of that reservation's bandwidth is needed for other purposes. Several examples exist in which this mechanism may be utilized.

The bandwidth allotted to an individual reservation may be reduced due to a variety of reasons such as preemption, etc. In such cases, when the entire bandwidth allocated to a reservation is not required, the reservation need not be torn down. The solution described in this document allows endpoints to negotiate a new (lower) bandwidth that falls at or below the specified new bandwidth maximum allocated by the network. Using a voice session as an example, this indication in RSVP could lead endpoints, using another protocol such as Session Initiation Protocol (SIP) [9], to signal for a lower-bandwidth codec and retain the reservation.

With RSVP aggregation [2], two aggregate flows with differing priority levels may traverse the same router interface. If that router interface reaches bandwidth capacity and is then asked to establish a new reservation or increase an existing reservation, the

router has to make a choice: deny the new request (because all resources have been utilized) or preempt an existing lower-priority reservation to make room for the new or expanded reservation.

If the flow being preempted is an aggregate of many individual flows, this has greater consequences. While [2] clearly does not terminate all the individual flows if an aggregate is torn down, this event will cause packets to be discarded during aggregate reservation reestablishment. This document describes a method where only the minimum required bandwidth is taken away from the lower-priority aggregated reservation and the entire reservation is not preempted. This has the advantage that only some of the microflows making up the aggregate are affected. Without this extension, all individual flows are affected and the deaggregator will have to attempt the reservation request with a reduced bandwidth.

RSVP tunnels utilizing IPsec [8] also require an indication that the reservation must be reduced to a certain amount (or less). RSVP aggregation with IPsec tunnels is being defined in [11], which should be able to take advantage of the mechanism created here in this specification.

Note that when this document refers to a router interface being "full" or "at capacity", this does not imply that all of the bandwidth has been used, but rather that all of the bandwidth available for reservation(s) via RSVP under the applicable policy has been used. Policies for real-time traffic routinely reserve capacity for routing and inelastic applications, and may distinguish between voice, video, and other real-time applications.

Explicit Congestion Notification (ECN) [10] is an indication that the transmitting endpoint must reduce its transmission. It does not provide sufficient indication to tell the endpoint by how much the reduction should be. Hence the application may have to attempt multiple times before it is able to drop its bandwidth utilization below the available limit. Therefore, while we consider ECN to be very useful for elastic applications, it is not sufficient for the purpose of inelastic application where an indication of bandwidth availability is useful for codec selection.

Section 2 discusses the individual reservation flow problem, while Section 3 discusses the aggregate reservation flow problem space. Section 4 lists the requirements for this extension. Section 5 details the protocol changes necessary in RSVP to create a reservation reduction indication. And finally, the appendix provides a walk-through example of how this extension modifies RSVP functionality in an aggregate scenario.

This document updates [RFC 2205](#) [1], as this mechanism affects the behaviors of the ResvErr and ResvTear indications defined in that document.

1.1. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [4].

2. Individual Reservation Reduction Scenario

Figure 1 is a network topology that is used to describe the benefit of bandwidth reduction in an individual reservation.

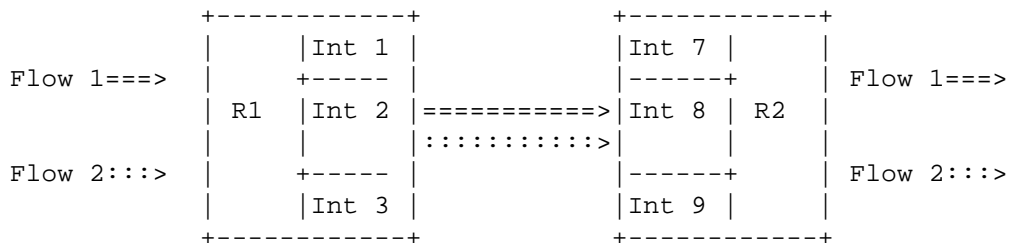


Figure 1. Simple Reservation Flows

Legend/Rules:

- Flow 1 priority = 300
- Flow 2 priority = 100
- Both flows are shown in the same direction (left to right). Corresponding flows in the reverse direction are not shown for diagram simplicity

RSVP is a reservation establishment protocol in one direction only. This split-path philosophy is because the routed path from one device to the other in one direction might not be the routed path for communicating between the same two endpoints in the reverse direction. End-systems must request 2 one-way reservations if that is what is needed for a particular application (like voice calls). Please refer to [1] for the details on how this functions. This example only describes the reservation scenario in one direction for simplicity's sake.

Figure 1 depicts 2 routers (R1 and R2) initially with only one flow (Flow 1). The flows are forwarded from R1 to R2 via Int 2. For this example, let us say that Flow 1 and Flow 2 each require 80 units of bandwidth (such as for the codec G.711 with no silence suppression).

Let us also say that the RSVP bandwidth limit for Int 2 of R1 is 100 units.

As described in [3], a priority indication is established for each flow. In fact, there are two priority indications:

- 1) one to establish the reservation, and
- 2) one to defend the reservation.

In this example, Flow 1 and Flow 2 have an 'establishing' and a 'defending' priority of 300 and 100, respectively. Flow 2 will have a higher establishing priority than Flow 1 has for its defending priority. This means that when Flow 2 is signaled, and if no bandwidth is available at the interface, Flow 1 will have to relinquish bandwidth in favor of the higher-priority request of Flow 2. The priorities assigned to a reservation are always end-to-end, and not altered by any routers in transit.

Without the benefit of this specification, Flow 1 will be preempted. This specification makes it possible for the ResvErr message to indicate that 20 units are still available for a reservation to remain up (the interface's 100 units maximum minus Flow 2's 80 units). The reservation initiating node (router or end-system) for Flow 1 has the opportunity to renegotiate (via call signaling) for acceptable parameters within the existing and available bandwidth for the flow (for example, it may decide to change to using a codec such as G.729)

The problems avoided with the partial failure of the flow are:

- Reduced packet loss, which results as Flow 1 attempts to reestablish the reservation for a lower bandwidth.
- Inefficiency caused by multiple attempts until Flow 1 is able to request bandwidth equal to or lower than what is available. If Flow 1 is established with much less than what is available then it leads to inefficient use of available bandwidth.

3. RSVP Aggregation Overview

The following network overview is to help visualize the concerns that this specification addresses in RSVP aggregates. Figure 2 consists of 10 routers (the boxes) and 11 flows (1, 2, 3, 4, 5, 9, A, B, C, D, and E). Initially, there will be 5 flows per aggregate (Flow 9 will be introduced to cause the problem we are addressing in this document), with 2 aggregates (X and Y); Flows 1 through 5 in aggregate X and Flows A through E in aggregate Y. These 2 aggregates will cross one router interface utilizing all available capacity (in this example).

RSVP aggregation (per [2]) is no different from an individual reservation with respect to being unidirectional.

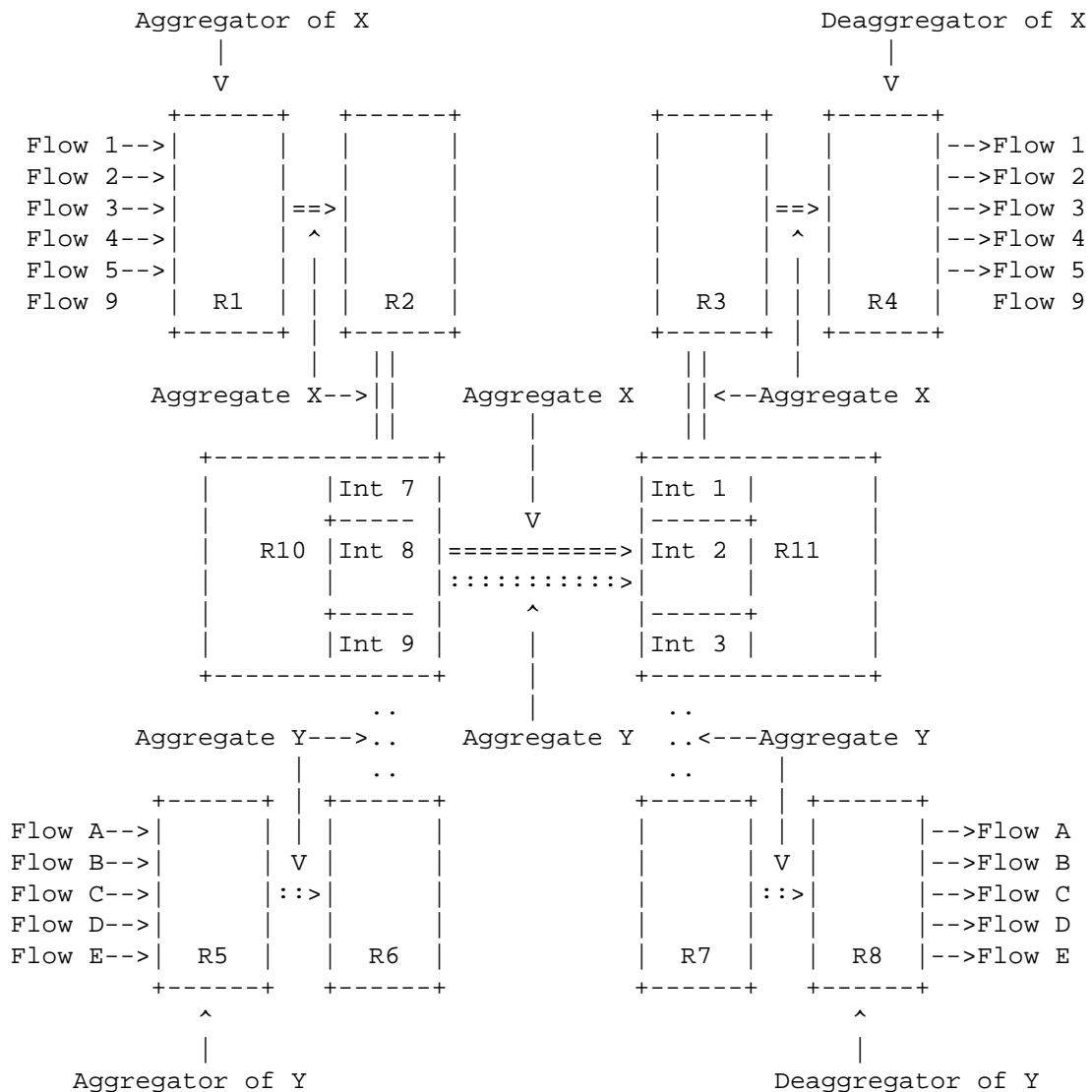


Figure 2. Generic RSVP Aggregate Topology

Legend/Rules:

- Aggregate X priority = 100
- Aggregate Y priority = 200
- All boxes are routers
- Both aggregates are shown in the same direction (left to right). Corresponding aggregates in the reverse direction are not shown for diagram simplicity.

The path for aggregate X is:

R1 => R2 => R10 => R11 => R3 => R4

where aggregate X starts in R1, and deaggregates in R4.

Flows 1, 2, 3, 4, 5, and 9 communicate through aggregate A.

The path for aggregate Y is:

R5 ::> R6 ::> R10 ::> R11 ::> R7 ::> R8

where aggregate Y starts in R5, and deaggregates in R8.

Flows A, B, C, D, and E communicate through aggregate B.

Both aggregates share one leg or physical link: between R10 and R11, thus they share one outbound interface: Int 8 of R10, where contention of resources may exist. That link has an RSVP capacity of 800 kbps. RSVP signaling (messages) is outside the 800 kbps in this example, as is any session signaling protocol like SIP.

3.1. RSVP Aggregation Reduction Scenario

Figure 2 shows an established aggregated reservation (aggregate X) between the routers R1 and R4. This aggregated reservation consists of 5 microflows (Flows 1, 2, 3, 4, and 5). For the sake of this discussion, let us assume that each flow represents a voice call and requires 80 kb (such as for the codec G.711 with no silence suppression). Aggregate X request is for 400 kbps (80 kbps * 5 flows). The priority of the aggregate is derived from the individual microflows that it is made up of. In the simple case, all flows of a single priority are bundled as a single aggregate (another priority level would be in another aggregate, even if traversing the same path through the network). There may be other ways in which the priority of the aggregate is derived, but for this discussion it is sufficient to note that each aggregate contains a priority (both hold and defending priority). The means of deriving the priority is out of scope for this discussion.

Aggregate Y, in Figure 2, consists of Flows A, B, C, D, and E and requires 400 kbps (80 kbps * 5 flows), and starts at R5 and ends R8. This means there are two aggregates occupying all 800 kbps of the RSVP capacity.

When Flow 9 is added into aggregate X, this will occupy 80 kbps more than Int 8 on R10 has available (880k offered load vs. 800k capacity) [1] and [2] create a behavior in RSVP to deny the entire aggregate Y

and all its individual flows because aggregate X has a higher priority. This situation is where this document focuses its requirements and calls for a solution. There should be some means to signal to all affected routers of aggregate Y that only 80 kbps is needed to accommodate another (higher priority) aggregate. A solution that accomplishes this reduction instead of a failure could:

- reduce significant packet loss of all flows within aggregate Y

During the re-reservation request period of time no packets will traverse the aggregate until it is reestablished.

- reduces the chances that the reestablishment of the aggregate will reserve an inefficient amount of bandwidth, causing the likely preemption of more individual flows at the aggregator than would be necessary had the aggregator had more information (that RSVP does not provide at this time)

During reestablishment of the aggregation in Figure 2 (without any modification to RSVP), R8 would guess at how much bandwidth to ask for in the new RESV message. It could request too much bandwidth, and have to wait for the error that not that much bandwidth was available; it could request too little bandwidth and have that aggregation accepted, but this would mean that more individual flows would need to be preempted outside the aggregate than were necessary, leading to inefficiencies in the opposite direction.

4. Requirements for Reservation Reduction

The following are the requirements to reduce the bandwidth of a reservation. This applies to both individual and aggregate reservations:

- Req#1 - MUST have the ability to differentiate one reservation from another. In the case of aggregates, it MUST distinguish one aggregate from other flows.
- Req#2 - MUST have the ability to indicate within an RSVP error message (generated at the router with the congested interface) that a specific reservation (individual or aggregate) is to be reduced in bandwidth.
- Req#3 - MUST have the ability to indicate within the same error message the new maximum amount of bandwidth that is available to be utilized within the existing reservation, but no more.

Req#4 - MUST NOT produce a case in which retransmitted reduction indications further reduce the bandwidth of a reservation. Any additional reduction in bandwidth for a specified reservation MUST be signaled in a new message.

RSVP messages are unreliable and can get lost. This specification should not compound any error in the network. If a reduction message were lost, another one needs to be sent. If the receiver ends up receiving two copies to reduce the bandwidth of a reservation by some amount, it is likely the router will reduce the bandwidth by twice the amount that was actually called for. This will be in error.

5. RSVP Bandwidth Reduction Solution

When a reservation is partially failed, a ResvErr (Reservation Error) message is generated just as it is done currently with preemptions. The ERROR_SPEC object and the PREEMPTION_PRI object are included as well. Very few additions/changes are needed to the ResvErr message to support partial preemptions. A new error subcode is required and is defined in [Section 5.1](#). The ERROR_SPEC object contained in the ResvErr message indicates the flowspec that is reserved. The bandwidth indication in this flowspec SHOULD be less than the original reservation request. This is defined in [Section 5.2](#).

A comment about RESV messages that do not use reliable transport: This document RECOMMENDS that ResvErr messages be made reliable by implementing mechanisms in [\[6\]](#).

The current behavior in RSVP requires a ResvTear message to be transmitted upstream when the ResvErr message is transmitted downstream (per [\[1\]](#)). This ResvTear message terminates the reservation in all routers upstream of the router where the failure occurred. This document requires that the ResvTear is only generated when the reservation is to be completely removed. In cases where the reservation is only to be reduced, routers compliant with this specification require that the ResvTear message MUST NOT be sent.

The appendix has been written to walk through the overall solution to the problems presented in [Sections 2 and 3](#). There is mention of this ResvTear transmission behavior in the appendix.

5.1. Partial Preemption Error Code

The ResvErr message generated due to preemption includes the ERROR_SPEC object as well as the PREEMPTION_PRI object. The format of ERROR_SPEC objects is defined in [1]. The error code listed in the ERROR_SPEC object for preemption [5] currently is as follows:

```
Errcode = 2 (Policy Control Failure) and  
ErrSubCode = 5 (ERR_PREEMPT)
```

The following error code is suggested in the ERROR_SPEC object for partial preemption:

```
Errcode = 2 (Policy Control Failure) and  
ErrSubCode = 102 (ERR_PARTIAL_PREEMPT)
```

There is also an error code in the PREEMPTION-PRI object. This error code takes a value of 1 to indicate that the admitted flow was preempted [3]. The same error value of 1 may be used for the partial preemption case as well.

5.2. Error Flow Descriptor

The error flow descriptor is defined in [1] and [7]. In the case of partial failure, the flowspec contained in the error flow descriptor indicates the highest average and peak rates that the preempting system can accept in the next RESV message. The deaggregator must reduce its reservation to a number less than or equal to that, whether by changing codecs, dropping reservations, or some other mechanism.

5.3. Individual Reservation Flow Reduction

When a router requires part of the bandwidth that has been allocated to a reservation be used for another flow, the router engages in the partial reduction of bandwidth as described in this document. The router sends a ResvErr downstream to indicate the partial error with the error code and subcode as described in [section 5.1](#). The flowspec contained in the ResvErr message will be used to indicate the bandwidth that is currently allocated.

The requesting endpoint that receives the ResvErr can then negotiate with the transmitting endpoint to lower the bandwidth requirement (by selecting another lower bandwidth codec, for example). After the negotiations, both endpoints will issue the RSVP PATH and RESV message with the new, lowered bandwidth.

5.4. Aggregation Reduction of Individual Flows

When a partial failure occurs in an aggregation scenario, the deaggregator receives the ResvErr message with the reduction indication from a router in the path of the aggregate. It then decides whether one or more individual flows from the aggregate are to be affected by this ResvErr message. The following choices are possible:

- o If that (deaggregator) router determines that one or more individual flow(s) are to partially failed, then it sends a ResvErr message with a reduced bandwidth indication to those individual flow(s). This is as per the descriptions in the previous section (5.3).
- o If that (deaggregator) router determines that one individual flow is to be preempted to satisfy the aggregate ResvErr, it determines which flow is affected. That router transmits a new ResvErr message downstream per [3]. That same router transmits a ResvTear message upstream. This ResvTear message of an individual flow does not tear down the aggregate. Only the individual flow is affected.
- o If that (deaggregator) router determines that multiple individual flows are to be preempted to satisfy the aggregate ResvErr, it chooses which flows are affected. That router transmits a new ResvErr message downstream as per [3] to each individual flow. The router also transmits ResvTear messages upstream for the same individual flows. These ResvTear messages of an individual flow do not tear down the aggregate. Only the individual flows are affected.

In all cases, the deaggregator lowers the bandwidth requested in the Aggregate Resv message to reflect the change.

Which particular flow or series of flows within an aggregate are picked by the deaggregator for bandwidth reduction or preemption is outside the scope of this document.

5.5. RSVP Flow Reduction Involving IPsec Tunnels

[RFC 2207](#) (per [8]) specifies how RSVP reservations function in IPsec data flows. The nodes initiating the IPsec flow can be an end-system like a computer, or it can router between two end-systems, or it can be an in-line bulk encryption device immediately adjacent to a router interface; [11] directly addresses this later scenario.

The methods of identification of an IPsec with reservation flow are different from non-encrypted flows, but how the reduction mechanism specified within this document functions is not.

An IPsec with reservation flow is, for all intents and purposes, considered an individual flow with regard to how to reduce the bandwidth of the flow. Obviously, an IPsec with reservation flow can be a series of individual flows or disjointed best-effort packets between two systems. But to this specification, this tunnel is an individual RSVP reservation.

Anywhere within this specification that mentions an individual reservation flow, the same rules of bandwidth reduction and preemption MUST apply.

5.6. Reduction of Multiple Flows at Once

As a cautionary note, bandwidth SHOULD NOT be reduced across multiple reservations at the same time, in reaction to the same reduction event. A router not knowing the impact of reservation bandwidth reduction on more than one flow may cause more widespread ill effects than is necessary.

This says nothing to a policy where preemption should or should not occur across multiple flows.

6. Backwards Compatibility

Backwards compatibility with this extension will result in RSVP operating as it does without this extension, and no worse. The two routers involved in this extension are the router that had the congested interface and the furthest downstream router that determines what to do with the reduction indication.

In the case of the router that experiences congestion or otherwise needs to reduce the bandwidth of an existing reservation:

- If that router supports this extension:
 - #1 - it generates the ResvErr message with the error code indicating the reduction in bandwidth.
 - #2 - it does not generate the ResvTear message.
- If that router does not support this extension, it generates both ResvErr and ResvTear messages according to [1].

In the case of the router at the extreme downstream of a reservation that receives the ResvErr message with the reduction indication:

- If that router does support this extension:

- #1 - it processes this error message and applies whatever local policy it is configured to do to determine how to reduce the bandwidth of this designated flow.

- If the router does not support this extension:

- #1 - it processes the ResvErr message according to [1] and all extensions it is able to understand, but not this extension from this document.

Thus, this extension does not cause ill effects within RSVP if one or more routers support this extension, and one or more routers do not support this extension.

7. Security Considerations

This document does not lessen the overall security of RSVP or of reservation flows through an aggregate.

If this specification is implemented poorly - which is never intended, but is a consideration - the following issues may arise:

- 1) If the ResvTear messages are transmitted initially (at the same time as the ResvErr messages indicating a reduction in bandwidth is necessary), all upstream routers will tear down the entire reservation. This will free up the total amount of bandwidth of this reservation inadvertently. This may cause the re-establishment of an otherwise good reservation to fail. This has the most severe affects on an aggregate that has many individual flows that would have remained operational.
- 2) Just as RSVP has the vulnerability of premature termination of valid reservations by rogue flows without authentication [12, 13], this mechanism will have the same vulnerability. Usage of RSVP authentication mechanisms is encouraged.

8. IANA Considerations

The IANA has assigned the following from [RFC 4495](#) (i.e., this document):

The following error code has been defined in the ERROR_SPEC object for partial reservation failure under "Errcode = 2 (Policy Control Failure)":

```
ErrSubCode = 102 (ERR_PARTIAL_PREEMPT)
```

The behavior of this ErrSubCode is defined in this document.

9. Acknowledgements

The authors would like to thank Fred Baker for contributing text and guidance in this effort and to Roger Levesque and Francois Le Faucheur for helpful comments.

10. References

10.1. Normative References

- [1] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), September 1997.
- [2] Baker, F., Iturralde, C., Le Faucheur, F., and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", [RFC 3175](#), September 2001.
- [3] Herzog, S., "Signaled Preemption Priority Policy Element", [RFC 3181](#), October 2001.
- [4] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [5] Herzog, S., "RSVP Extensions for Policy Control", [RFC 2750](#), January 2000.
- [6] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", [RFC 2961](#), April 2001.
- [7] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", [RFC 2210](#), September 1997.

- [8] Berger, L. and T. O'Malley, "RSVP Extensions for IPSEC Data Flows", [RFC 2207](#), September 1997.

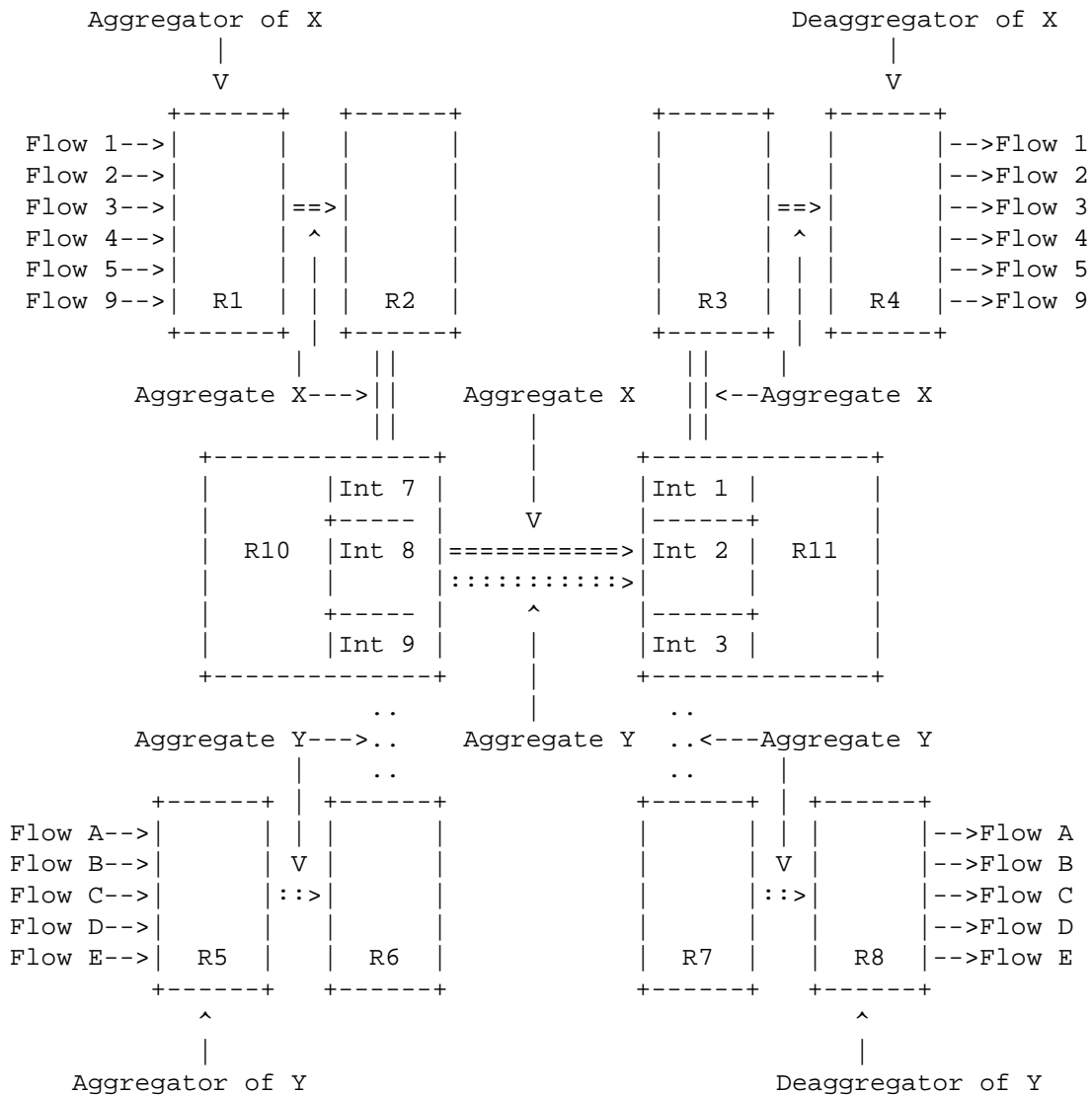
10.2. Informative References

- [9] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", [RFC 3261](#), June 2002.
- [10] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", [RFC 3168](#), September 2001.
- [11] Le Faucheur, F., Davie, B., Bose, P., Christou, C., and M. Davenport, "Generic Aggregate RSVP Reservations", Work in Progress, October 2005.
- [12] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", [RFC 2747](#), January 2000.
- [13] Braden, R. and L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value", [RFC 3097](#), April 2001.

Appendix A. Walking through the Solution

Here is a concise explanation of roughly how RSVP behaves with the solution to the problems presented in Sections 2 and 3 of this document. There is no normative text in this appendix.

Here is a duplicate of Figure 2 from [section 3](#) of the document body (to bring it closer to the detailed description of the solution).



Duplicate of Figure 2. Generic RSVP Aggregate Topology

Looking at Figure 2, aggregate X (with five 80 kbps flows) traverses:

R1 ==> R2 ==> R10 ==> R11 ==> R3 ==> R4

And aggregate Y (with five 80 kbps flows) traverses:

R5 ::> R6 ::> R10 ::> R11 ::> R7 ::> R8

Both aggregates are 400 kbps. This totals 800 kbps at Int 7 in R10, which is the maximum bandwidth that RSVP has access to at this interface. Signaling messages still traverse the interface without problem. Aggregate X is at a higher relative priority than aggregate Y. Local policy in this example is for higher relative priority flows to preempt lower-priority flows during times of congestion. The following points describe the flow when aggregate A is increased to include Flow 9.

- o When Flow 9 (at 80 kbps) is added to aggregate X, R1 will initiate the PATH message towards the destination endpoint of the flow. This hop-by-hop message will take it through R2, R10, R11, R3, and R4, which is the aggregate X path (that was built per [2] from the aggregate's initial setup) to the endpoint node.
- o In response, R4 will generate the RESV (reservation) message (defined behavior per [1]). This RESV from the deaggregator indicates an increase bandwidth sufficient to accommodate the existing 5 flows (1, 2, 3, 4, and 5) and the new flow (9), as stated in [2].
- o As mentioned before, in this example, Int 8 in R10 can only accommodate 800 kbps, and aggregates X and Y have each already established 400 kbps flows comprised of five 80 kbps individual flows. Therefore, R10 (the interface that detects a congestion event in this example) must make a decision about this new congestion generating condition in regard to the RESV message received at Int 8.
- o Local policy in this scenario is to preempt lower-priority reservations to place higher-priority reservations. This would normally cause all of aggregate Y to be preempted just to accommodate aggregate X's request for an additional 80 kbps.
- o This document defines how aggregate Y is not completely preempted, but reduced in bandwidth by 80 kbps. This is contained in the ResvErr message that R10 generates (downstream) towards R11, R7, and R8. See [section 5](#) for the details of the error message.

- o Normal operation of RSVP is to have the router that generates a ResvErr message downstream to also generate a ResvTear message upstream (in the opposite direction, i.e., towards R5). The ResvTear message terminates an individual flow or aggregate flow. This document calls for that message not to be sent on any partial failure of reservation.
- o R8 is the deaggregator of aggregate Y. The deaggregator controls all the parameters of an aggregate reservation. This will be the node that reduces the necessary bandwidth of the aggregate as a response to the reception of an ResvErr message (from R10) indicating such an action is called for. In this example, bandwidth reduction is accomplished by preempting an individual flow within the aggregate (perhaps picking on Flow D for individual preemption by generating a ResvErr downstream on that individual flow).
- o At the same time, a ResvTear message is transmitted upstream on that individual flow (Flow D) by R8. This will not affect the aggregate directly, but is an indication to the routers (and the source end-system) which individual flow is to be preempted.
- o Once R8 preempts whichever individual flow (or 'bandwidth' at the aggregate ingress), it transmits a new RESV message for that aggregate (Y), not for a new aggregate. This RESV from the deaggregator indicates a decrease in bandwidth sufficient to accommodate the remaining 4 flows (A, B, C, and E), which is now 320 kbps (in this example).
- o This RESV message travels the entire path of the reservation, resetting all routers to this new aggregate bandwidth value. This should be what is necessary to prevent a ResvTear message from being generated by R10 towards R6 and R5.

R5 will not know through this RESV message which individual flow was preempted. If in this example, R8 was given more bandwidth to keep, it might have transmitted a bandwidth reduction ResvErr indication towards the end-system of Flow D. In that case, a voice signaling protocol (such as SIP) could have attempted a renegotiation of that individual flow to a reduced bandwidth (say, but changing the voice codec from G.711 to G. 729). This could have saved Flow D from preemption.

Authors' Addresses

James M. Polk
Cisco Systems
2200 East President George Bush Turnpike
Richardson, Texas 75082 USA

EMail: jmpolk@cisco.com

Subha Dhesikan
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134 USA

EMail: sdhesika@cisco.com

Full Copyright Statement

Copyright (C) The Internet Society (2006).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.

Acknowledgement

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).