

NSIS Working Group
Internet-Draft
Expires: December 23, 2002

X. Fu
Technical University Berlin
C. Kappler
H. Tschofenig
Siemens AG
June 24, 2002

Analysis on RSVP Regarding Multicast
draft-fu-rsvp-multicast-analysis-00.txt

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on December 23, 2002.

Copyright Notice

Copyright (C) The Internet Society (2002). All Rights Reserved.

Abstract

RSVP version 1 has been designed for optimally support multicast. However, in reality multicast is being used much less frequently than anticipated. Still, even for unicast (one sender, one receiver) communication, full-fledged multicast-enabled RSVP signaling must be used. As pointed out in the NSIS requirement draft, multicast would not be necessarily required for an NSIS signaling protocol. This draft analyses ingredients of RSVP Version 1 which are affected by multicast, and derives how these ingredients may look like if multicast is not supported. We find removing multicast capability

from RSVP will make it and its extensions considerably more lightweight.

Table of Contents

1.	Introduction	3
2.	Multicast-Influenced Ingredients in RSVP	4
2.1	Reservation Styles	4
2.2	Receiver-Initiated Reservation	4
2.3	Path/Resv Two-Pass Signaling Message Exchange	4
2.4	State Management in Routers	5
2.5	Error Handling	5
2.6	Non-RSVP Region Handling	5
3.	Removing Multicast in RSVP	6
3.1	No Reservation Styles	6
3.2	Sender-Initiated Reservation	6
3.3	Path One-Way Signaling Message for Reservation	6
3.4	Simpler State Management in Routers	7
3.5	Simplified Error Handling	7
3.6	Impact on Non-RSVP Region Problem	8
4.	Removing Multicast in Some RSVP Extensions	9
4.1	Proxy	9
4.2	Aggregation	9
5.	Benefits of Removing Multicast in RSVP	11
6.	Security Aspect	12
7.	Concluding Remarks	13
8.	Acknowledgement	14
	References	15
	Authors' Addresses	16
	Full Copyright Statement	17

1. Introduction

As a signaling protocol designed specifically for the Internet, RSVP Version 1 (RSVPv1) [8] distinguishes itself by a number of fundamental ways, particularly, soft state management, two-pass signaling message exchanges, receiver-based resource reservation and separation of QoS signaling from routing (in the sense that RSVP messages follow normal IP routing). However, RSVP end-to-end signaled QoS for the Internet has not become a reality. One reason for this is regarded the complexity brought by multicast [2][3]. In fact, vast majority of applications do not use multicast in reality. Some multicast protocols (e.g., PIM-SM [9]) even consider multicast as a function built on top of unicast routing rather than as an integral part of it. We notice that multicast is not listed as a (mandatory) requirement in the NSIS requirements draft [1].

This draft analyses ingredients of RSVP Version 1 which are needed to support multicast. Compared with standard RSVP, the following ingredients could be simplified or would even not be needed if multicast is removed from RSVP:

- o Reservation styles are not needed;
- o Sender-initiated rather than receiver-initiated;
- o Path one-way signaling instead of Path/Resv two-pass signaling message exchange (although ACK/NACK is still needed);
- o Only Path state is needed in routers, and NHOP/PHOP/LIH are not necessary in a Path state;
- o Error handling is simpler;
- o Non-RSVP region problems become easier.

This list might not be comprehensive, but we believe it covers main features. We find that removing multicast capability from RSVP will make it and its extensions on aggregation and proxy support considerably more light-weight.

[2. Multicast-Influenced Ingredients in RSVP](#)

This section analysis the ingredients in RSVPv1 [\[7\]](#)[\[8\]](#) influenced by multicast, they are: (1) reservation styles, (2) receiver-initiated reservation, (3) Path/Resv two-pass signaling message exchanges, (4) state management in routers, (5) error handling, and (6) non-RSVP region techniques. In this document terms "reservation" and "(QoS) signaling" are used interchangeably.

[2.1 Reservation Styles](#)

To accommodate the multipoint-to-multipoint multicast applications, RSVP was designed to support a vector of reservation attributes called the "style". There are two attributes in RSVPv1:

Sharing attribute, with value "Shared" (all senders share a single reservation) or "Distinct" (every sender has a seperate reservation);

Sender Selection attribute, with value "Explicit" (select explicitly a specific sender), "Wildcard" (applies to all senders).

[2.2 Receiver-Initiated Reservation](#)

Receiver-initiated is critical for RSVP to setup multicast sessions with a large number of receivers. RSVP is optimized for a large

number of receivers with heterogeneous request, and therefore deploys the receiver-initiated approach: a receiver initiates a reservation request at a leaf of the multicast distribution tree, travelling towards the sender. Whenever a reservation is found to already exist in a node in the distribution tree, the new request will be merged with the existing reservation. This could result in fewer signalling operations for the RSVP nodes close to the sender, and new receivers are handled completely by the receivers and the network without bothering the sender. In comparison, for sender-initiated reservation, when the reservation request travels up the multicast tree towards the intended heterogeneous receivers, it is necessary to distribute its reservation request information to all hops on the multicast tree between source and receivers. This implies gathering receivers' QoS information from receivers beforehand and results in more signaling overhead.

[2.3](#) Path/Resv Two-Pass Signaling Message Exchange

Since reservation requests are initiated by each receiver, to make sure the reservation request messages from a receiver follow exactly

the reverse routes of the data traffic from the sender(s), RSVP must establish a sink tree from each receiver to all senders to forward reservation messages. This is achieved by two-pass signaling message (Path and Resv) exchange process. Besides, to effectively signal QoS to the network, Path messages carry traffic characteristic information (Sender_TSpec) as well as network capability information gathered (ADSpec), while a Resv message carries QoS information (FlowSpec, which includes TSpec and RSpec) requested by the receiver. FlowSpecs should be merged when a Resv message is received by an RSVP router where multicast delivery replicates data packets.

[2.4](#) State Management in Routers

Each RSVP router uses a Path state to maintain a table of Logical Interface Handle (LIH, the outgoing interface of the previous RSVP hop), previous RSVP hop address (PHOP, used to route the Resv messages hop-by-hop in the reverse direction) and TSpec information for each multicast session. An RSVP router also uses a Resv state to maintain next RSVP hop address (NHOP, used to distinguish its route from other multicast session) and FlowSpec. These states are "soft" which will be time out, unless they are refreshed by the modified or

the same Path/Resv message as the first ones.

[2.5](#) Error Handling

Path errors are simple, because whenever a PathErr message is created, it will be just sent back to the sender.

For reservation errors (e.g., admission control fails), a ResvErr message should be reported to all of the responsible receivers. Furthermore, it may result in "killer problems", i.e., if the path towards the source has sufficient resources for the smaller of the reservations but not for the merged request for the multicast, then the effect of merging can be to deny reservations to both. Therefore, the processing of ResvErr messages in RSVPv1 is rather complex.

[2.6](#) Non-RSVP Region Handling

A subsequent problem due to RSVP two-pass signaling is illustrated in [7], where asymmetric routing for Path and Resv messages in a non-RSVP region can result in a Resv message arriving at a wrong interface of the previous RSVP hop (in the Path direction). To solve this, a LIH is stored in the Path state at next RSVP hop and directs a subsequent Resv message to be forwarded to the previous RSVP hop, then an appropriate reservation can be made to the right interface.

[3.](#) Removing Multicast in RSVP

This section analysis how RSVP would look like if removing its multicast capabilities and supporting only (1:1) unicast. For the convenience of the following discussion, the abbreviation "RSVPw/oMC" is used to stand for the possible feature set by removing multicast in RSVP(v1).

Like in RSVPv1, the soft-state mechanism would still be interesting to be used in RSVPw/oMC; QoS signaling in the RSVPw/oMC is also separated from routing in routers; the data flow definition can be the same (i.e., consists of a session = <dstIP, protoId, dstPort> and a filter spec = <srcIP, srcPort>).

However, RSVPw/oMC will bring a number of simplifications to RSVPv1. In the following paragraphs, these features are briefly described. Descriptions on other aspect of RSVPw/oMC can be found in subsequent sections of this document.

3.1 No Reservation Styles

Styles in RSVPv1 are used for specifying the sender set for a reservation request and whether these senders share a single reservation. In RSVPw/oMC a receiver has only one sender, therefore styles are not needed any more. Therefore all related burden like merging are released from RSVPw/oMC.

3.2 Sender-Initiated Reservation

Unlike in RSVPv1, it is not necessary for RSVPw/oMC to be receiver-initiated, because there is only one receiver, and no merging is necessary from the receiver to the sender. In fact, for unicast sessions, a sender typically does know its QoS requirement (maybe via higher layer) and data traffic characteristics, and since there is a need for QoS signaling in a quick way, it would be desirable to be sender-initiated.

3.3 Path One-Way Signaling Message for Reservation

In RSVPw/oMC, Path and Resv two-pass message exchange would be unnecessary, since no sink-tree is needed for a unicast session. Instead, one-way (Path message) would suffice for setting-up the reservation for a sender's session. On the other hand, to give the sender a chance to see whether and how his reservation request has been fulfilled, an acknowledge message (Ack/NAck) still is beneficial. Therefore, the needed message types in RSVPw/oMC would be:

Path: for one-way reservation and state refreshment. This message

in fact takes both responsibilities of Path and Resv messages in RSVPv1; the actual reverse directional Resv message in RSVPv1 is no longer necessary;

PathErr: for reporting errors in processing Path messages.

PathTear: for tear down of Path state. PathTear message is sent

by the sender towards the receiver or the IP address where the Path reservation request was rejected to explicitly delete or adapt reservation states;

(N)Ack: indicating whether a reservation request is accepted (Ack message) or rejected (NAck). The NAck message shares the same type as the Ack message, but the IP address where the Path reservation is rejected can be included. This provides the sender / higher layer a flexibility in deciding whether to still use the reserved segment for its session, or modify / release the reserved resources.

It is further possible to simplify and optimize traffic description and QoS specification by using a "QoS profile (with acceptable and desired QoS level)". To support different QoS provisioning techniques and optimize for the one-way signaling, the FlowSpec in Resv messages and the Sender_TSpec/ADSpec in Path messages of RSVPv1 could be unified to a "QoS profile", e.g., as defined in [4]. In the QoS profile of a Path message, the sender can state its desired QoS and (minimally) acceptable QoS, as well as its traffic characteristics (this allows the routers a flexibility to dynamically adapt its reservation in dynamic environments). A QoS profile with actually reserved QoS information can be attached in the Ack message.

[3.4](#) Simpler State Management in Routers

In RSVPw/oMC, there is no need to keep two states in routers: Path state and Resv state. Instead, just one Path state created by the Path message would be sufficient for RSVPw/oMC QoS signaling. Furthermore, no NHOP/PHOP is needed in the Path state, since the QoS signaling message (Path) is sent one-way from the sender towards the unicast receiver. Finally, hop-by-hop refreshes are simpler since Path/Resv refreshment messages neither need to be distributed towards the multiple receivers/senders nor need to be merged.

[3.5](#) Simplified Error Handling

Although there still are a few possibilities of error sources in RSVPw/oMC, most of the issues like killer problems become rather easier to handle since: (1) there is no need to merge states for multicast sessions in the RSVP routers, so the number of possible

error source decreases; (2) the error reports can be simply returned back to the unicast sender.

[3.6](#) Impact on Non-RSVP Region Problem

Since the Path message is used for one-way QoS signaling, while in the reverse direction a QoS signaling function is not needed, the problem described in [Section 2.6](#) would disappear in RSVPw/oMC. As a result the LIH information is not needed in the Path states.

[4.](#) Removing Multicast in Some RSVP Extensions

RSVP has been extended in various ways to provide better scalability and usefulness in certain scenarios. In this section we analysis how removing multicast impacts two important RSVP extensions: proxy and aggregation.

[4.1](#) Proxy

In cases where RSVP cannot be transported end-to-end, an RSVP proxy [\[6\]](#) provides a means to originate or receive RSVP messages on behalf of the end node(s), so that applications may still benefit from reservations that are not truly end-to-end. Removing multicast in principle would also apply to the RSVP proxy scheme.

An RSVPw/oMC proxy typically can be placed in the edge of an access network. However, the decision where to place the proxy functionality may be made considering other factors, e.g., as defined in [\[6\]](#).

In RSVPw/oMC, the procedure for incorporating a sender proxy would follow [\[6\]](#).

However, the RSVPv1 receiver proxy [\[6\]](#) should be modified to handle the following RSVPw/oMC messages:

Path message. RSVPw/oMC receiver proxy should originate an Ack/Nack message in response to an incoming Path message, on behalf of the receiver identified by the Path message.

PathTear and PathErr messages are treated as in normal RSVPw/oMC.

[4.2](#) Aggregation

RSVPw/oMC can make use of aggregation in a simpler way compared to RSVPv1. Possible changes to current RSVP aggregation techniques are described below.

The aggregator needs to change the protocol identifier of (e2e) RSVP messages into RSVP_e2e_ignore and decide which aggregate flow should the microflow be mapped into (e2ePath-->Path_Aggr per the processing for Resv message in [\[11\]](#)), and takes the (RSVPv1 deaggregator's)

responsibility of ensuring that there are sufficient resources reserved within the aggregation region to support the new e2e session. If there are not sufficient resources, a new/existing aggregated reservation should be created/readjusted by a Path_Aggr message from the aggregator towards deaggregator, and returned with

an Ack/NAck from the deaggregator or an interior RSVP router. Upon receipt of a Path_Aggr message, the interior routers decides whether to forward it on (when the reservation request is fulfilled) or return a NAck message towards the aggregator. If up to the deaggregator the aggregated reservation request is fulfilled, it acknowledges the aggregator with an Ack message and change the RSVP_e2e_ignore back to e2e Path. Upon receipt of an e2e Ack/NAck, a deaggregator now only needs to map the aggregated reservation to micro-reservations according to its policy. The Path states in interior routers of the aggregate region are maintained in either on-demand or in a more static, statistical way.

By RSVPw/OMC, when a Path message travels across an aggregation region, aggregation is much easier since for each aggregator, there is only one deaggregators. Hence, the challenges of complicated multicast processing described in [\[11\]](#) do not exist any longer.

[5.](#) Benefits of Removing Multicast in RSVP

According to the discussion above, RSVPw/oMC potentially has a number of advantages over RSVPv1:

Reduction in reservation set-up time. Because RSVPw/oMC does not need a reverse e2e Resv message after an e2e Path message, the reservation set-up time will be one-pass less than in RSVPv1.

Reduction of state in routers. There is only one Path state for a session; the PHOP/NHOP/LIH information is not needed.

Reduction in processing overhead due to (1) unified QoS profile, (2) no need to merge for multicast session and (3) simpler handling of error and non-RSVP regions.

6. Security Aspect

By removing multicast support from RSVP, message processing is simplified as explained throughout this document. However, there is little room for optimization in security aspect. Integrity and replay protection of RSVP signaling messages as offered by [RFC2747](#) [10] is still required to provide security on a hop-by-hop basis. Additionally, the integrity handshake mechanism used for recovering from host or router crash to offer sequence number resynchronization is required. In order to support policy based admission control and authentication of the user via Kerberos, digital signature or plain-text credentials the objects described in [12] have to be present. Multicast processing of the policy locator inside the POLICY_DATA object can be simplified in such a way that the policy locator is copied from the received to the new message. The same procedure has to be applied for the AUTH_DATA object which is also left unchanged and forwarded (i.e. copied to the new RSVP message).

Section 7 of [10] states that in the multicast case all receivers must share a single key with the Kerberos Authentication Server (i.e. a single Kerberos principal is used). Removing multicast allows avoiding the case where all receivers must share a single key.

Whether it is possible to simplify processing of POLICY_DATA objects altogether needs further investigation.

[7.](#) Concluding Remarks

This draft analyses ingredients of RSVP Version 1 which are influenced by multicast. We find the following ingredients may not be needed or can be simplified if multicast is not supported: two-pass (Path and Resv), receiver-initiated reservation; complex presentation and operation for signaled information and state management; reservation styles. Also, complexity in dealing with hop-by-hop refreshes, non-RSVP regions and errors would be largely reduced by removing multicast from RSVP. We find that removing multicast capability from RSVP will make it and its extensions on proxy aggregation considerably more light-weight.

[8.](#) Acknowledgement

The authors would like to thank Mehmet Ersue and Holger Karl for their comments to this draft.

- [1] Brunner, M., "Requirements for QoS Signaling Protocols", [draft-ietf-nsis-req-02](#) (work in progress), May 2002.
- [2] Hancock, R. and et al, "Towards a Framework for QoS Signaling in the Internet", [draft-hancock-nsis-framework-00](#) (work in progress), Feb 2002.
- [3] De Meer, H. and et al, "Analysis of Existing QoS Solutions", [draft-demeer-nsis-analysis-01](#) (work in progress), May 2002.
- [4] Westphal, C. and et al, "Context Relocation of QoS Parameters in IP Networks", [draft-westphal-seamoby-qos-relocate-00](#) (work in progress), Jul 2001.
- [5] Braden, B. and B. Lindell, "A Two-Level Architecture for Internet Signaling", [draft-braden-2level-signal-arch-00](#) (work in progress), Nov 2001.
- [6] Bernet, Y., Elfassy, N., Gai, S. and D. Dutt, "RSVP Proxy", [draft-ietf-rsvp-proxy-03](#) (work in progress), Mar 2002.
- [7] Braden, R., Estrin, D., Berson, S., Herzog, S. and D. Zappala, "The Design of the RSVP Protocol", ISI Final Technical Report, available at <http://www.isi.edu/div7/rsvp/pub.html>, Jul 1996.
- [8] Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), Sep 1997.
- [9] Estrin, D., Farinacci, D., Helmy, A., Thaler, D., Deering, S., Handley, M. and V. Jacobson, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification", [RFC 2362](#), Jun 1998.
- [10] Baker, F., Lindell, B. and M. Talwar, "RSVP Cryptographic Authentication", [RFC 2747](#), Jan 2000.
- [11] Baker, F., Iturralde, C., Faucheur, F. and B. Davie, "Aggregation of RSVP for IPv4 and IPv6 Reservations", [RFC 3175](#), Sep 2001.
- [12] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S. and R. Hess, "Identity Representation for RSVP", [RFC 3182](#), Oct 2001.

Authors' Addresses

Xiaoming Fu
Technical University Berlin
Skr. FT 5-2, Einsteinufer 25
Berlin 10587
Germany

Phone: +49-30-314-23827
EMail: fu@ee.tu-berlin.de

Cornelia Kappler
Siemens AG
Siemensdamm 62
Berlin 13623
Germany

Phone: +49-30-386-32894
EMail: Cornelia.Kappler@icn.siemens.de

Hannes Tschofenig
Siemens AG
Otto-Hahn-Ring 6
Munich 81739
Germany

Phone: +49-89-636-40390
EMail: Hannes.Tschofenig@mchp.siemens.de

Internet-Draft

RSVP Multicast Analysis

June 2002

Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS 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.

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.

