

Network Working Group
Internet-Draft
Expires: April 16, 2003

X. Fu
University of Goettingen
C. Kappler
H. Tschofenig
Siemens AG
October 16, 2002

Analysis on RSVP Regarding Multicast
draft-fu-rsvp-multicast-analysis-01.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 April 16, 2003.

Copyright Notice

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

Abstract

RSVP version 1 has been designed for optimum support multicast. However, in reality multicast is being used much less frequently than anticipated. Still, even for unicast (one sender, one receiver) 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 in the generic RSVP signaling protocol and

Internet-Draft

RSVP Multicast Analysis

October 2002

adapt related functionalities accordingly - we call the resulting feature set "RSVP Lite", a potentially more light-weight version of RSVP.

Table of Contents

1.	Introduction	3
2.	Multicast-Influenced Ingredients in RSVP	4
2.1	Reservation Styles and Scope Object	4
2.2	Limitation on Receiver-Initiated Reservation	4
2.3	Coupled QoS Specification with the Generic Signaling Protocol	4
2.4	Complex State Merging Operation in the Routers for Signaling	5
2.5	Killer Problems and Error/Failure Handling	5
3.	Towards a Light-weight RSVP: RSVP Lite	6
3.1	No Reservation Styles and Scope Object	6
3.2	Decoupling Signaled Data from Signaling	6
3.3	No Restriction on Sender- or Receiver-Initiation	6
3.4	Simpler State Management in Routers	7
3.5	Simplified Error Handling	7
3.6	Message Types	7
3.7	Mobility Consideration	8
3.8	Multicast Consideration	8
3.9	Potential Advantages of RSVP Lite	8
4.	Security Considerations	10
5.	Acknowledgement	11
	References	12
	Authors' Addresses	13
	Full Copyright Statement	14

1. Introduction

As a signaling protocol designed specifically for the Internet, RSVP Version 1 (RSVPv1) [\[1\]](#) [\[2\]](#) [\[3\]](#) 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 [\[8\]](#). In fact, vast majority of applications do not use multicast in reality. Some multicast protocols (e.g., PIM-SM [\[4\]](#)) 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 [\[7\]](#).

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

- o Unnecessary reservation styles and scope object in the signaling protocol;
- o Limitation on receiver-initiation;
- o Coupled QoS specification and signaling support;
- o Complex state structure and merging operation in the routers for the signaling;
- o Killer problems and error/failure handling.

This list might not be comprehensive, but we believe it covers main features. We find that removing multicast capability from RSVP

generic signaling (and adpting related functionalities accordingly) will make it considerably more light-weight.

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

This section analysis the ingredients in RSVPv1 influenced by multicast.

[2.1 Reservation Styles and Scope Object](#)

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 Limitation on 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](#) Coupled QoS Specification with the Generic Signaling Protocol

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. These specs are particularly designed for multicast QoS resource reservation in the Integrated Service model,

where FlowSpecs should be merged when a Resv message is received by an RSVP router where multicast delivery replicates data packets.

[2.4](#) Complex State Merging Operation in the Routers for Signaling

Each RSVP router uses a Path state to maintain a table of TSpec information for each multicast session, in addition to outgoing interfaces of the previous RSVP hop and previous RSVP hop addresses. An RSVP router also uses a Resv state to maintain next RSVP hop address (used to distinguish its route from other multicast session) and FlowSpec. When a router finds multicast delivery replicates data packets, those specs related to the downstream reservations should be merged, and appropriate reservation parameters should be passed upstream.

[2.5](#) Killer Problems and Error/Failure Handling

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 error processing of in RSVPv1 is rather complex. Furthermore, a blockade state is introduced to solve the one of the killer problems (KR-II), which is denial of (reservation) service by a large and failing reservation. After a reservation's failure is recorded in the blockade state created by ResvError messages, merging will take this into account.

[3.](#) Towards a Light-weight RSVP: RSVP Lite

Based on the analysis above, this section describes how RSVP would look like if removing its multicast capabilities and adapting other related functionalities of RSVP. For the convenience of the following discussion, the abbreviation "RSVP Lite" is used to stand for the possible feature set that follows.

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

However, RSVP Lite will bring a number of simplifications to RSVPv1. In the following paragraphs, these features are briefly described.

[3.1](#) No Reservation Styles and Scope Object

Styles in RSVPv1 are used for specifying the sender set for a reservation request and whether these senders share a single reservation. Beside, a scope object is used in Resv messages to carry an explicit list of sender hosts towards which the information in the message is to be forwarded. In RSVP Lite a receiver has only one sender, therefore styles are not needed any more. Then all related burden like merging are released from RSVP Lite.

[3.2](#) Decoupling Signaled Data from Signaling

Path/Resv messages will not be responsible for creation and merging of multicast reservation sink trees, they are only responsible for creation and maintaining the appropriate states. The signaled data are handled by upper level protocol(s) being carried by RSVP Lite (e.g., ALSP described in [\[10\]](#)). Typically, one-way service signaling (Path message) would suffice for setting-up the states from any node towards another node in the Internet. However, to give the signaling source a chance to see whether and how its service signaling (Path) has been succeed, a confirm message (Resv) still is necessary for RSVP Lite, with an optional object to record the last failed node (when failed).

[3.3](#) No Restriction on Sender- or Receiver-Initiation

Unlike in RSVPv1, it is not necessary for RSVP Lite to be always receiver-initiated, because there is only one receiver, and no merging is necessary from the receiver to the sender. It is up to the upper level of signaling protocols to decide which way to initiate. In addition, RSVP Lite does not rely on the multicast

membership, therefore can initiate and respond the signaling at any node in the Internet, i.e., the placement of its signaling source and destination is flexible. Thus it is able to support other signaling models like host-to-network and edge-to-edge in a simple way in addition to end-to-end signaling.

[3.4](#) Simpler State Management in Routers

In RSVP Lite, there is no need to keep blockade states in routers, only Path and Resv states would suffice. Hop-by-hop refreshes are also 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 RSVP Lite, 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](#) Message Types

The possible types for messages involved in the signaling process of RSVP Lite are listed as follows.

Path (or Request): for signaling request and state refreshment. This message in fact takes only part of responsibilities of Path messages in RSVP, and the real semantic of the signaled data is defined by separate protocols. We call these separate protocols ``client protocols'' and an example is the QoS client protocol. These client protocol elements are encoded into the Path (Request) messages - and Resv (Response) message when necessary - which are similar to the concept of ALSP [[10](#)].

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 (Request) was rejected to explicitly delete or adapt associated states;

Resv (or Response): indicating whether a signaling request is accepted and whether to adapt in the reverse direction (``Commit'') or rejected (``Reject'). The actual reverse directional functionalities (mainly reserve for resources) of

Resv message in RSVP is no longer necessary. However, in RSVP Lite, two-pass message exchange would be still necessary for setting up proper states in the nodes along the data path from the signaling requester to the signaling receiver. A Resv (Response) message with ``reject'' flag set shares the same type as the ``Commit'' one, but the IP address where the Path reservation is rejected can be included. This provides the signaling sender a flexibility in deciding whether to still use the signaled segment for its session, or to clean up the established states along the path.

For QoS signaling purposes, it is further possible to simplify and optimize traffic description and QoS specification in the QoS client protocol by introducing a ``QoS profile (with acceptable and desired QoS level)'' into the QoS-client protocol for RSVP Lite. To support different QoS provisioning techniques and optimize the signaling procedure, the FlowSpec in Resv messages and the SenderTSpec/ADSpec in Path messages of RSVP could be unified to a ``QoS profile'', e.g., as defined in [14]. In the QoS profile of a Path (Request) 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 Resv (Response) message.

[3.7](#) Mobility Consideration

TBD - Ideas of existing RSVP mobility schemes e.g., "Localized RSVP" [12], "Mobile Extension for RSVP" [13] can be ported. It is also possible for RSVP Lite to release the unused states and establish new states without much pain following the ideas of mobility/route change handling in CASP design [11]; special attention should be paid to session identifiers.

[3.8](#) Multicast Consideration

TBD - It would be useful for RSVPm/oMC to provide limited support for some of multicast models. One possibility is to provide a special interface to access the multicast routing tables. A limited support for SSM multicast [9] is provided in CASP [11] and its idea may also applies for RSVP Lite.

[3.9](#) Potential Advantages of RSVP Lite

According to the discussion above, RSVP Lite potentially has a number of advantages over RSVPv1:

Reduction in signaling set-up time. Because RSVP Lite can be

Internet-Draft

RSVP Multicast Analysis

October 2002

sender- and receiver-initiated without additional message exchange, the signaling set-up time will be one-pass less than in RSVPv1 in a sender-initiated signaling scenario.

Different modularity and more flexibility. RSVP Lite does not intend to optimize for multicast, instead keeping the RSVP Lite signaling integrants as minimal as possible, while freely putting initiator and responder. Decoupling signaled data and signaling message makes it more flexible and applicable to many other signaling purposes besides QoS signaling.

Reduction in processing overhead due to (1) no need to carry unnecessary component and interpret signaled data in the basic signaling protocol (RSVP Lite), (2) no need to merge for multicast sessions and (3) simpler handling of errors and failures.

[4](#). Security Considerations

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](#) [5] 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 [6] 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 [5] 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.

[5.](#) Acknowledgement

Mehmet Ersue and Robert Hancock provided useful comments.

References

- [1] Braden, R., Estrin, D., Berson, S., Herzog, S. and D. Zappala, "The Design of the RSVP Protocol", ISI Final Technical Report , Jul 1996.
- [2] Zhang, L., Deering, S., Estrin, D. and D. Zappala, "RSVP: A New Resource Reservation Protocol", IEEE Network, Volume 7, Pages 8-18, Sep 1993.
- [3] Braden, R., Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), Sep 1997.
- [4] 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.
- [5] Baker, F., Lindell, B. and M. Talwar, "RSVP Cryptographic Authentication", [RFC 2747](#), Jan 2000.

- [6] Yadav, S., Yavatkar, R., Pabbati, R., Ford, P., Moore, T., Herzog, S. and R. Hess, "Identity Representation for RSVP", [RFC 3182](#), Oct 2001.
- [7] Brunner, M., "Requirements for Signaling Protocols", [draft-ietf-nsis-req-04](#) (work in progress), Aug 2002.
- [8] Freytsis, I. and et al, "Next Steps in Signaling: Framework", [draft-ietf-nsis-fw-00](#) (work in progress), Oct 2002.
- [9] Bhattacharyya, S. and et al, "An Overview of Source-Specific Multicast(SSM) Deployment", [draft-ietf-ssm-overview-03](#) (work in progress), Mar 2002.
- [10] Braden, B. and B. Lindell, "A Two-Level Architecture for Internet Signaling", [draft-braden-2level-signal-arch-00](#) (work in progress), Nov 2001.
- [11] Schulzrinne, H. and et al, "CASP - Cross-application Signaling Protocol", [draft-schulzrinne-nsis-casp-00](#) (work in progress), Sep 2002.
- [12] Manner, J. and et al, "Localized RSVP", [draft-manner-lrsvp-00](#) (work in progress), May 2002.
- [13] Shen, C. and et al, "Mobility Extensions to RSVP in an RSVP-

Fu, et al.

Expires April 16, 2003

[Page 12]

Internet-Draft

RSVP Multicast Analysis

October 2002

Mobile IPv6 Framework", [draft-shen-nsis-rsvp-mobileipv6-00](#) (work in progress), Jul 2002.

- [14] Westphal, C. and et al, "Context Relocation of QoS Parameters in IP Networks", [draft-westphal-seamoby-qos-relocate-00](#) (work in progress), Jul 2001.

Authors' Addresses

Xiaoming Fu
 University of Goettingen
 Telematics Group
 Lotzestr. 16-18
 Goettingen 37083

Germany

E-Mail: fu@cs.uni-goettingen.de

Cornelia Kappler
Siemens AG
Siemensdamm 62
Berlin 13623
Germany

E-Mail: Cornelia.Kappler@icn.siemens.de

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

E-Mail: Hannes.Tschofenig@mchp.siemens.de

Fu, et al.

Expires April 16, 2003

[Page 13]

Internet-Draft

RSVP Multicast Analysis

October 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.