

Internet Draft  
Expiration: October 1999  
File: [draft-yuhara-rsvp-refresh-00.txt](#)

M. Yuhara  
M. Tomikawa

Fujitsu Laboratories Ltd.

April 1999

## **RSVP Extensions for ID-based Refreshes**

### 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/1id-abstracts.txt>

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

### Abstract

The purpose of this document is to provide a discussion on the refresh overhead reduction of RSVP. Specifically, this document proposes some extensions to deal with route change while maintaining the advantage of ID-based refreshes. For environments where PATH refreshes must be used to detect route change, ID-only refreshes are used to decrease the message size. For environments where route change is informed to RSVP process by some other means, a new message is introduced to invalidate ID.

### **1. Introduction**

The combination of RSVP [RFC2205, [RFC2209](#)] and intserv [RFC2210, [RFC2215](#)] was proposed to provide better quality of service than traditional best effort service. When deploying this combination over the internet, however, there are several concerns [[RFC 2208](#)], and the scalability problem is regarded as the biggest problem. The

differentiated services (diffserv) [[RFC2475](#)] has challenged this problem. Diffserv is expected to replace the role of intserv in the internet backbones. However, to provide the end-to-end QoS, we still need an end-to-end signaling protocol, and the only usable signaling protocol now is RSVP. In fact, [[Baker](#)] and other drafts propose to use RSVP as a signaling protocol for diffserv.

If we assume the aggregation region of [[Baker](#)] is a DS domain, RSVP will be used (a) between a DS boundary node of a DS domain and an external node which may be in a different DS domain or in an intserv domain, (b) between DS boundary nodes of the same DS domain (for individual reservation, bypassing interior nodes), (c) between two RSVP neighbor nodes within the same DS domain (for behavior aggregates).

Because the number of diffserv codepoints and the number of boundary nodes are limited, (c) should not cause a scalability problem.

Scalability problems in the case of (a) and (b), however, still exist. The refresh reduction draft [[Berger](#)] alleviates some of the problems using several techniques. The techniques include refresh suppression and packaging multiple RSVP messages into one message (called aggregate messages; should not be confused with the reservation aggregation in [[Baker](#)]).

The purpose of this document is to provide some discussion on that draft and show some enhancements. Although multicast case will be mentioned, it is not discussed extensively.

The extensions introduced in this draft surely complicate the programming of RSVP process, but alleviate the RSVP processing cost and the bandwidth required for refresh messages in the environments where route change must be detected via PATH messages. Whether such environments will exist in the core of the future internet, remains to be seen.

## **2. Definition of route change**

Let us defined what "route change at a node" means in the following discussion. Route is said to have changed at an RSVP node when:

- after sending the first PATH message for a session from this node,
- the stream of packets for which this PATH message is intended still goes through this node and
- the next RSVP-capable node (or set of nodes of a multicast group) for this stream has changed, or the outgoing interface (or interfaces) for this stream has changed.

Note that this definition only concerns outgoing route from this



node. When the packets under consideration are no longer received by this node due to a route change, the route is considered to have changed at some upstream node, not at this node.

There are three cases for route change:

- (case 1) There is no route change for this outgoing interface. An example is an outgoing interface of a diffserv boundary node that connects to another diffserv boundary. Since this kind of link tends to be stable for a long period, we can consider that there is no route change at this node. Maintenance shutdown may be required, but that can be considered as a rare exception and may be manually handled (for example, shutting down rsvp processes on both sides).
- (case 2) The route change may happen, but the necessary route change information is provided to the rsvp process as soon as possible. The information may be provided by means of routing protocols or management protocols, or by measurements. In this case, if the no-refresh option (Last\_Refresh) [[Berger](#)] is enabled and the (old) next hop has already registered an ID-based state for PATH of this stream, that state must be explicitly revoked. Otherwise, the state would remain there forever. We will discuss this case in [section 4](#).
- (case 3) The route change may happen, and the necessary route change information is NOT provided to the rsvp process. In this case, we have to rely on the PATH refreshes. However, by exchanging the ID between RSVP neighbor nodes, we can drastically reduce the amount of PATH message size. We will discuss this case first.

### **[3. ID-only PATH refreshes](#)**

In (case 3), route change is detected using PATH refreshes, thus we cannot eliminate PATH refreshes. In [[Baker](#)], detection of route change within a domain depends on the refreshes of individual PATH messages. An example of route change is shown below. Individual PATH messages are not interpreted by interior nodes (Rx) but by boundary nodes (ingress, egress1, egress2).



Before route change

```
[ingress] -----> [R1] -----> [R2] -----> [R3] -----> [egress1]
```

After route change

```
[ingress] -----> [R1] -----> [R2] ---+
                                   |
                                   +--> [R4] -----> [egress2]
```

In general, the interior routing protocol does not inform the ingress node of the route change at R2 (although some protocols could do so, such protocols cannot be assumed in general).

Since the number of reservations on a DS boundary node may get large, we still want the processing overhead and bandwidth overhead be small to alleviate the scalability problem.

The ID-based refresh in [[Berger](#)] helps to quickly find a corresponding state in the receiving node and to instantly check the equality of the previous message and the incoming message (Last\_Refresh flag off).

In this draft, we further reduce the overhead by removing unnecessary information from the PATH refresh. A PATH message with an ID consists of the following elements (size is shown by the number of 4-byte words).

	Words	
-----		
Common Header	2	
INTEGRITY	0-	(9 for HMAC-MD5)
MESSAGE_ID	2	[ <a href="#">Berger</a> ]
SESSION	3	(IPv4)
RSVP_HOP	3	(IPv4)
TIME_VALUES	2	
POLICY_DATA	0-	
SENDER_TEMPLATE	3	(IPv4)
SENDER_TSPEC	9	(INTSERV)
ADSPEC	about 21	(21 for General Param, Guaranteed Service, Controlled-Load Service, no override)

Some objects are optional and vary in size. In particular, some people has been worried that a POLICY\_DATA object gets large (possibly gets larger than MTU). So a PATH message could be very large.



Now, we propose ID-only PATH refreshes. ID-only PATH refresh reduces the PATH message size, once the PATH is confirmed. The ACK mechanism in [Berger] is used for confirmation. ID-only PATH message looks like this:

```
<Path Refresh Message> ::= <Common Header> [ <INTEGRITY> ]
                                <MESSAGE_REFRESH>
                                <SESSION> <RSVP_HOP>
```

MESSAGE\_ID ACK object is defined in [Berger]. The MESSAGE\_REFRESH object is similar to the MESSAGE\_ID object in [Berger] but has a different class number:

#### MESSAGE\_REFRESH class

Class-Num is of 0bbbbbbb form (must report error if this object is unknown) and should be assigned by IANA.

#### REFRESH\_ID object

Class = MESSAGE\_REFRESH class, C-Type = 1

```
+-----+-----+-----+-----+
|  Reserved   |                Message ID                |
+-----+-----+-----+-----+
```

Reserved: reserved and must be zero.

Message ID: Latest ID for this PATH state. The ID must have been acknowledged by the next hop(s).

The IP header is the same as normal PATH messages. The IP destination address is the session address. Unfortunately, since PATH REFRESH messages must detect route change, PATH REFRESH messages cannot be aggregated in a single message [Berger], whose destination address is the address of the common next hop.

The recipient hop of this PATH REFRESH message must behave as if the full PATH message previously received with this Message ID was received, if such message was received from the same RSVP\_HOP and this node has sent its acknowledgement. However, this node does not have to send another acknowledgment even if the original message requested it.

In this way, the total message size is reduced significantly. If we assume the numbers above, the message is reduced from 45 words to 10 words. If the POLICY\_DATA object is very large, the reduction is more





effective. The smaller gets the message size, the smaller gets the message handling cost. In addition, the bandwidth required for PATH refreshes is reduced.

The rest of this section describes how route change is handled when ID-only PATH refresh is used.

If a PATH receiving node has never heard of the Message ID from the RSVP\_HOP, or if this node has not send an acknowledgment for the Message ID, then this node considers it detects a route change at RSVP\_HOP, and explicitly requests the RSVP\_HOP node to send a full PATH message. The request message is sent to the sending node (RSVP\_HOP) as follows:

```
<Refresh Request Message> ::= <Common Header> [ <INTEGRITY> ]
                                [ <MESSAGE_ID ACK> ... ]
                                <MESSAGE_REQUEST>
                                <SESSION> <RSVP_HOP>
```

#### REFRESH\_REQUEST object

Class = MESSAGE\_REFRESH class, C-Type = 2

```
+-----+-----+-----+-----+
|  Reserved   |           Message ID           |
+-----+-----+-----+-----+
```

Reserved: reserved and must be zero.

Message ID: The received Message ID that should be sent again in full format.

If a PATH sending node receives Refresh Request message with an ID registered in this node, the node must send a full PATH message with a newly allocated ID.

The PATH state in the receiving node will timeout if PATH or PATH Refresh messages are not received for a long time as defined in [\[RFC2205\]](#). Or, it may be explicitly revoked by Invalidate ID Messages described in the next section.

An RSVP node that does not support this extension will report a Path Error when it receives a Path Refresh message, since it does not know the REFRESH\_ID object and its class is requesting an error report if the object is unknown to the node. The SESSION and RSVP\_HOP objects are included in the Path Refresh Message so that such node can



construct a Path Error message. The PATH sending host which receives a Path Error must first check if the previous PATH message was an ID-only PATH, and if so, it should not forward the Path Error message upstream, but should reset the ID-based state and send a full PATH message. If the error reporting node only supports conventional RSVP, it will not send back an acknowledgment, so the sending node continues to send full PATH messages.

We could use Path Error in place of Refresh Request. However, we provide a separate message type for easier handling.

The above route change handling for ID-only refreshes can also be used for multicast sessions when ID-only refreshes are used. If a new node joins a multicast group down-stream of a PATH sending node, and the rsvp process of the PATH sending node is not informed of the change, then the new node will request a full PATH message or will report an error, since it receives unknown ID-only PATH refreshes. The PATH sending node can try to re-establish ID-based relations with all neighbor members.

#### **4. Route change handling when refresh is suppressed**

This section discusses the extension to [Berger] for previously described (case 2). In this case, The route change may happen, but the information on route change at this node is provided to the rsvp process. With this information, the node can re-establish the PATH state with a new next hop.

However, there is no means to revoke the old state in the old next hop. We cannot use a Path Tear message for this purpose, because Path Tear has the ultimate destination as its IP address and it would now be routed to the new next hop. We need a message that can directly reach the old next hop. To this end, we define INVALIDATE\_ID RSVP messages.

```
<Invalidate ID Message> ::= <Common Header> [ <INTEGRITY> ]  
  
[ <MESSAGE_ID ACK> ... ]  
  
<MESSAGE_ID>  
  
<INVALIDATE_ID>
```

INVALIDATE\_ID object

Class = MESSAGE\_REFRESH class, C-Type = 3



```

+-----+-----+-----+-----+
|  Reserved   |           Message ID           |
+-----+-----+-----+-----+

```

Reserved: reserved and must be zero.

Message ID: The Message ID that should be invalidated.

The host must retransmit the Invalidate ID Message until it is acknowledged, or a maximum retry limit is reached (or Hello Message has timed-out and all relating states are cleared). Since this message must be acknowledged, the message ID for this message is included in the message.

If a host receives an Invalidate ID Message that has a matching state, the host must invalidate the state and send an acknowledgment. The host must be prepared to repeatedly receive the same INVALIDATE\_ID messages, since the acknowledgment may be lost in the network.

## 6. Possible modification to the RSVP aggregate reservation

The RSVP aggregate reservation proposed in [Baker] installs states in DS-ingress and DS-egress boundary nodes to aggregate reservations. Since this state handling is very similar to the one for ID-based refreshes, reservation aggregation mechanism can relay on the ID-based states.

In [Baker], Path Error messages with error code of NEW-AGGREGATE-NEEDED are used for two purposes. One is for the ingress node to identify an egress node for the session. This Path Error can be replaced by an acknowledgment for the PATH message. The other is for an egress node to request re-establishment of the states when it receives an unknown PATH (due to route change). This Path Error can be replaced by a Refresh Request message. By combining [Baker] and [Berger] in this way, we can eliminate the duplicate state management.

## Security Considerations

No new security issues are raised in this document. See [RFC2205] for a general discussion on RSVP security issues.

## References

[Baker] Baker, F., Iturralde, C., Le Faucheur, F., Davie, B., "Aggregation of RSVP for IP4 and IP6 Reservations", Work In Progress, [draft-baker-rsvp-aggregation-00.txt](#), February 1999.



[Berger] Berger, L., Gan, D., Swallow, G., "RSVP Refresh Reduction Extensions", [draft-berger-rsvp-refresh-reduct-00.txt](#), Work In Progress, March 1999.

[Bernet] Bernet, Y., Yavatkar, R., Ford, P., Baker, F., Zhang, L., Nichols, K., Speer, M., Braden, R., "Interoperation of RSVP/Int-Serv and Diff-Serv Networks", Work In Progress, [draft-ietf-diffserv-rsvp-02.txt](#), February 1999.

[Pan] Pan, P., Schulzrinne, H., Guerin, R., "Staged Refresh Timers for RSVP", [draft-pan-rsvp-timer-00.txt](#), Work In Progress (already expired), November 1997.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., Jamin, S., "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", [RFC 2205](#), September 1997.

[RFC2209] Braden, R., Zhang, L., "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules", [RFC 2209](#), September 1997.

[RFC2210] Wroclawski, J., "The Use of RSVP with IETF Integrated Services", [RFC 2210](#), September 1997.

[RFC2208] Baker, F., Braden, B., Bradner, S., O'Dell, M., Romanow, A., Weinrib, A., Zhang, L., "Resource ReSerVation Protocol (RSVP) Version 1 Applicability Statement Some Guidelines on Deployment", [RFC 2208](#), September 1997.

[RFC2215] Shenker, S., Wroclawski, J., "General Characterization Parameters for Integrated Service Network Elements", [RFC 2215](#), September 1997.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., Weiss, W., "An Architecture for Differentiated Services", [RFC 2475](#), December 1998.

#### Author's Address

Masanobu Yuhara  
Fujitsu Laboratories Ltd.  
4-1-1 Kamikodanaka, Nakahara-ku  
Kawasaki, 211-8588, Japan  
Phone: +81-44-777-1111  
Email: yuhara@flab.fujitsu.co.jp

Mayumi Tomikawa  
Fujitsu Laboratories Ltd.





4-1-1 Kamikodanaka, Nakahara-ku  
Kawasaki, 211-8588, Japan  
Phone: +81-44-777-1111  
Email: ohya@flab.fujitsu.co.jp