Network Working Group
Internet Draft
draft-sgai-rsvp-proxy-00.txt
Expiration Date: April 2000

Silvano Gai
Dinesh Dutt
Nitsan Elfassy
Cisco Systems
Yoram Bernet
Microsoft

October 1999

RSVP Receiver Proxy

Status of this Memo

Copyright Notice

Abstract

RSVP has been extended in several directions [Policy], [Identity], [DCLASS], [AggrRSVP], [DiffModel],[COPS-RSVP], These extensions have broadened the applicability of RSVP characterizing it as a signaling protocol usable outside the IntServ model.

With the addition of the "Null Service Type" [NullServ], RSVP is being adopted also by mission critical applications that require some form of prioritized service, but cannot readily specify their resource requirements. These applications do not need to set-up a reservation end-to-end, but only to signal to the network their policy information [Policy], [Identity] and obtain in response an applicable DSCP [DCLASS].

RSVP Receiver Proxy is an extension to the RSVP message processing (not to the protocol itself), mainly designed to operate in conjunction with the Null Service Type and with an extension of the COPS for RSVP protocol [COPS-RSVP-EXT].

Table of contents

1. Introduction

The IETF has come up with two architectures to support QoS in IP networks. IntServ (Integrated Services [RFC1633], [RFC2210]) is an architecture that provides the ability for applications to choose among multiple, controlled levels of delivery service for their data packets. It relies upon explicit signaling by applications to the network for the desired QoS. These applications typically know their traffic characteristics and have possibly strict latency requirements. Such applications require so called "tight QoS" or "quantitative QoS". RSVP is the protocol which can be used by applications to signal their QoS requirements to the network. Applications have to be modified to take advantage of the Integrated Services. The receivers control the QoS given to the data stream.

DiffServ (Differentiated Services, [RFC2474], [RFC2475]) is another IETF architecture for implementing scalable service differentiation in the Internet. There is no explicit signaling protocol used in DiffServ. The network is logically divided into edge devices and core devices. The edge devices attempt to recognize data flows and assign QoS based on this. They also assign a DSCP (DiffServ Code Point) in the DS byte of the packets (the byte that used to be called the TOS byte). Core devices use the DSCP to assign a QoS to the microflows. Applications typically do not have to be modified to take advantage of Differentiated Services. Receivers do not control the QoS given to the data stream.

The recognition of data flows and the assignment of an appropriate DSCP is a tricky task and often requires stateful inspection of flows and symmetrical routing paths. Moreover, application recognition is limited to the information present in the packet traversing the network and in most current network devices is further limited to what is in the IP/TCP/UDP headers. Application vendors desire to be able to assign QoS to their packets based on both information that may not be carried in the packet and information other than the

IP/TCP/UDP header fields. For example, a SAP print transaction may require a different treatment than a SAP database update. Similarly, if the user of the application is the CTO of the company, the priority assigned to such packets maybe different from that assigned to packets of the application being used by some other person in the company.

For this reason RSVP has been proposed also for mission critical applications (e.g. ERP) that require some form of prioritized service, but cannot readily specify their resource requirements. The ISSLL WG is discussing the specification of the Null Service Type as a way to use RSVP with a broader range of applications [NullServ].

Some of these applications have the requirement for the end-to-end message processing of RSVP. Others simply need to signal to the network their identity [Identity] and some additional policy information [Policy] related to the flows and obtaining from the network some decisions, e.g. the DSCP to be used [DCLASS].

RSVP Receiver Proxy is a proposal that mainly addresses this second type of applications, i.e., applications that simply want to use RSVP as a signaling protocol toward the network. For them, the end-to-end nature of RSVP is not interesting and often is perceived as a disadvantage, since it is characterized by a higher latency.

The RSVP Receiver Proxy:

- is an alternate way to process RSVP messages and policy information in the switch/routers;

- it does not require any change to the RSVP protocol;

- it does require an extension to the COPS for RSVP protocol [COPS-RSVP-EXT].

In general, "RSVP Proxy" should be symmetric, i.e., it may be useful to have RSVP Sender Proxy as well as RSVP Receiver Proxy. This document does not define RSVP Sender Proxy at this stage. If the document is accepted by the IETF community, the RSVP Sender Proxy can be added in the next version.

This document defines RSVP Receiver Proxy in association with the Null Service Type, but nothing prevents using this feature also in association with other service types, e.g. the Controlled Load service.

The following section uses an example in which the Receiver Proxy functionality is placed in the first hop switch/router. This is a possibility, but it is not a requirement. While designing a network the following trade-off should be considered:

- Proxying closer to the server reduces turn around time.

- Proxying further from the server enables additional downstream network elements to benefit from the information carried in the signaling messages, and to participate in the response.

- Proxying anywhere in the network enables the deployment of such applications in which only the server is required to signal, but

the client may remain unchanged.

The COPS-RSVP Extension [COPS-RSVP-EXT] should enable the network administrator to decide how to make the tradeoffs described above.

2. An overview of RSVP Receiver Proxy

With RSVP Receiver Proxy a switch/router acts as a proxy for the receiver, e.g. when it receives an RSVP Path message, it generates an RSVP Resv message on behalf of the receiver.

The generation of the Resv message is done under policy control, the switch/router may be programmed either to classify the packets marking them with an appropriate DSCP or to use the DCLASS object [DCLASS] to communicate the classification decision to the host.

The adoption of RSVP Receiver Proxy do not change the basic model of RSVP, i.e.:

- the handling of data flows is unidirectional. If the application data is strictly unidirectional it is sufficient to use RSVP only in one direction. In the case of bidirectional data, running RSVP only in one direction provides a certain performance benefit, but to get the maximum performance benefit it is necessary to use RSVP in both directions.

- The application on the host assumes the host model of RSVP, including the extensions proposed in [DiffModel], [Policy], [Identity], [NullServ].

- The message format and the message types are the same of RSVP, including the DCLASS object previously proposed in [DCLASS] and the Null Service Type [NullServ].

- The switch/router acts as a COPS client [COPS] in communicating with the policy server, i.e. it uses RSVP client for COPS [COPS-RSVP]. Certain extensions to COPS for RSVP are needed [COPS-RSVP-EXT], see Section 4.

- The classification of traffic cannot be more granular than microflow (the so called five-tuple) or in the case of IPSEC the four-tuple that includes the Parameter Index, or SPI, in place of the UDP/TCP-like ports [RFC2207].

- There is no special support for subflows (a set of packets inside a microflow). Of course, an application may send different Path messages for the same flow at different times, thus providing a support for subflows not overlapping in time.

3. Detailed description of the message processing

This sections details some of the message processing of a switch/router acting as RSVP Receiver Proxy. The description is mainly focused on the two fundamental messages in RSVP, i.e. the Path Message and the Resv message. Other messages are discussed in Section 4.4.

Figure 1 depicts a simple network topology (two hosts H1 & H2 and intermediate routers, R1-R5) that will be used in the explanation.

```
                    Path Message ----->
```

```
                           <----- Resv Message

                     +-----+                  +-----+
                     | PS1 |                  | PS1 |
                     +-----+                  +-----+
                    /   |    \                /   |
                   /    |     \              /    |
                  /     |      \            /     |
   +----+   +----+   +----+   +----+   +----+   +----+   +----+
   | H1 |---| R1 |---| R2 |---| R3 |---| R4 |---| R5 |---| H2 |
   +----+   +----+   +----+   +----+   +----+   +----+   +----+


     H1 ----> R1 ----> R2 ----> R3 ----> R4 ----> R5 ----> H2  (Regular
                                                            |   RSVP
                                                            v   case)
     H1 <---- R1 <---- R2 <---- R3 <---- R4 <---- R5 ----> H2


     H1 ----> R1
               |                       (RSVP Receiver Proxy)
               v
     H1 <---- R1


   Hx: Host x
   Ry: Router y
   PSz: Policy Server z
```

Figure 1: Possible Message Forwarding Behaviors in RSVP

Immediately below the network, the normal RSVP message processing is reported. The Path message goes hop-by-hop from H1 to H2. The Resv message uses the reverse path of the Path message and goes from H2 to H1. The interaction between the network devices and the policy servers is the one specified by COPS for RSVP ([COPS], [COPS-RSVP]).

With RSVP Receiver Proxy the propagation of the RSVP Path message is terminated in the router acting as a proxy. Any router in the network may act as RSVP Receiver Proxy, but it is a good design guideline to place the proxy functionality as close as possible to the sender. In our case R1 acts as a proxy for H2 under the control of a policy server.

For example, an application on H1 uses RSVP to signal parameters upon which to base the decision to assign the QoS for a microflow. The example assumes that the information needs to be used only by the edge network device and it is not required to propagate this further down the network

A possible sequence of steps consists of:

- The application on H1 indicates to the RSVP subsystem that it is a sender and specifies its traffic characteristics. It may specify additional parameters.

- This causes the RSVP subsystem on H1 to start transmitting RSVP Path messages in accordance with normal RSVP/SBM rules.

- The first hop switch/router (R1) receives this message and it communicates with the policy server for a decision on how to treat the Path message. It copies all the relevant information contained in the Path message to the policy server.

- The policy server communicates a decision to R1 to not forward the Path message, but instead to originate and send a Resv message to H1. H1 data traffic gets assigned the right DSCP by the switch/router as per the policy communicated by the policy server. The Resv message may also specify to the host the DSCP and shaping information to be associated with the microflow using the DCLASS object [DCLASS].

- On receiving the Resv message, H1 may start marking correctly the data traffic accordingly to the DSCP received in the Resv message.

4. The role of the policy server

To implement both RSVP and RSVP Receiver Proxy the policy server needs to specify a set of decisions [COPS-RSVP-EXT] which is extended compared to COPS-RSVP [COPS-RSVP]. If the decision is to accept the Path message, the decision message must specify how the network device behaves with respect to each of the following:

- Forwarding of the Path message;

- Originating a RSVP Resv message;

- Processing and possibly Forwarding a RSVP Resv message.

The decision may also possibly include the QoS specification to be associated with the flow identified in the Path message. This specification consists of a DSCP and possibly a TSPEC (as specified by RSVP [RFC2210]) for policing the traffic.

4.1 Generation of the Resv message by the Receiver Proxy

It maybe required that the network device originate a Resv message. This is a proxy Resv message in the sense that it is being generated by the network device and not by the actual receiver(s) identified in the RSVP Path message. The format of a Resv message is as follows (see [RFC2205] for details):

```
<Resv Message> ::=     <Common Header> [ <INTEGRITY> ]
   <SESSION> <RSVP HOP> <TIME_VALUES><DCLASS>
   [ <RESV_CONFIRM> ] [ <SCOPE> ] [ <POLICY_DATA>... ]
   <STYLE> <flow descriptor list>
```

- The network device puts its IP address and L2 address in the source IP and source mac-address fields. Since Resv messages follow Path messages, this would constitute a valid Resv message.

- The SESSION object can be copied from the Path message.

- The RSVP HOP object can be filled in with the IP address of the switch/router generating this Resv message.

- The TIME_VALUES object contains the refresh period. See below.

- The STYLE object is set to Wildcard Filter (WF) style indicating that the reservation is to be shared and that the sender is wildcarded. Associated with a WF style is a FLOWSPEC object which is encoded as specified in [RFC2210] or [NullServ].

- The SCOPE and RESV_CONFIRM objects need not be included in the Resv message.

- The POLICY_DATA objects will be as returned by the policy server.

- The Resv message may also contain the new DCLASS object is contained in the COPS decision message. The DCLASS object specifies the DSCP to be associated with the microflow for which the Path message was received.

- The Resv messages need to be originated and sent for each of the periodically-received Path messages.

4.2 Communication With the Policy Server

When a network device establishes the connection with the policy server, it sends a COPS Client-Open message for the RSVP client. It should indicate in this message whether the network device is capable of supporting only the base RSVP message processing or also the Receiver Proxy message processing. It can do this with in a capability list (that can accommodate also future extensions). To deal with existing clients, if the policy server does not receive a capability list, it should assume that it is communicating with a legacy RSVP client. The capability list can be included as part of the ClientSI object passed in the Client-Open message [COPS-RSVP-EXT].

On receiving a RSVP Path message, the network device sends a COPS REQ message to the policy server. This message will be the standard REQ message sent on receiving a RSVP Path message.

The DEC message returned by the policy server for this REQ message must contain the information needed to take the decisions listed in Section 4.

The DEC message SHOULD also contain a list of DSCP [DCLASS].

The DEC message may also contain bandwidth information to be associated with the microflow: communicating Shaping/limiting parameters to the network is a powerful Policy Management tool for the PDP/LPDP both for Qualitative and Quantitative services. This topic needs further study.

The network device must also be able to determine if a Path message is a refresh or a new one. It must communicate with the policy server only for new Path messages or for updated ones.

In the absence of a policy server or if the connection to the policy server is not up, the operation of RSVP Receiver Proxy depends on policy configuration local to the network device. For example, the network device may have a local configuration that specifies:

- o do not accept new flows;
- o honor existing flows until they time-out.

4.3 Enhancements To Existing Infrastructure

- o COPS for RSVP will have to be enhanced to support the new format for RSVP REQ and DEC message as stated in [COPS-RSVP-EXT].
- o When SBM is in use, it is possible that a device which does not support RSVP Receiver Proxy becomes the DSBM on the first-hop segment. This can be prevented by the network administrator by configuring the appropriate priority on the device with RSVP Receiver Proxy support.

4.4 Processing of other RSVP messages

This section details the processing of the protocol messages in RSVP other than Path and Resv. Only the differences in the processing from classical RSVP is specified.

- o PathTear message is honored and is forwarded or not similar to a Path message. The policy server is not contacted on receiving a PathTear message. This is consistent with the existing behavior of COPS for RSVP[RSVP-COPS].

- o PathErr messages are treated as in normal RSVP.

5. RSVP With Null Service Type

RSVP protocol can be represented as consisting of two parts: a message processing part and a resource allocation & resource enforcement part. The following are the minimal requirements for a network device to support RSVP Null Service Type:

- o The network device MUST implement the message processing part of the RSVP protocol. This includes the ability to receive and interpret a raw IP packet or UDP-based RSVP packet.

- o If the network device is a L2 device, it SHOULD implement SBM.

- o The network device SHOULD know how to talk to a policy server using COPS. Specifically, the network device SHOULD be able to talk to COPS as a RSVP client using the extensions defined in [COPS-RSVP-EXT].

- o The node SHOULD keep the RSVP state so that the following Path refresh won't cause a repetitive Path handling.

- o The network device SHOULD be able to generate a Resv message periodically in a coherent way with the RSVP soft state maintenance.

- o In the absence of a connection to the policy server, this network device depends on policy configuration local to the network device (see Section 4.2).

6. Security Considerations

RSVP messages contain an INTEGRITY object which authenticates the originating node and is also used to verify the contents of the message. Moreover the RSVP message SHOULD contain an IDENTITY object that SHOULD be authenticated. If the policy server does not implement any security mechanisms, it SHOULD use a clear text version of the user identity.

7. Intellectual Property Considerations

The IETF is being notified of intellectual property rights claimed in regard to some or all of the specification contained in this document. For more information consult the online list of claimed rights.

8. References

[COPS] Boyle, J., Cohen, R., Durham, D., Herzog, S., Raja, R., Sastry, A., "The COPS (Common Open Policy Service) Protocol", IETF <draft-ietf-rap-cops-07.txt>, August 1999.

[RFC1633] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," June 1994.

[RFC2205] Braden, R., Zhang, L., Berson, S., Herzog, S., and Jamin, S., "Resource Reservation Protocol (RSVP) Version 1 Functional Specification", IETF RFC 2205, Proposed Standard, September 1997.

[RFC2210] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," September 1997.

[RFC2474] K. Nichols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," December 1998.

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

[COPS-RSVP] Jim Boyle, Ron Cohen, David Durham, Shai Herzog, Raju Rajan, Arun Sastry, "COPS usage for RSVP," <draft-ietf-rap-cops-rsvp-05.txt>, June 14, 1999

[COPS-RSVP-EXT]   Nitsan Elfassy, Dinesh Dutt, "COPS Extensions for RSVP Receiver Proxy Support"," <draft-nitsan-cops-rsvp-proxy-00.txt>, October 1999.

[Policy]   Shai Herzog, "RSVP Extensions for Policy Control," Internet Draft., < draft-ietf-rap-rsvp-ext-06.txt>, April 1999.

[DiffModel] Y. Bernet, A. Smith, S. Blake, "A Conceptual Model for Diffserv Routers," Internet Draft, <draft-ietf-diffserv-model-00.txt>, June 1999.

[Identity]   Satyendra Yadav, Raj Yavatkar, Ramesh Pabbati, Peter Ford, Tim Moore, Shai Herzog, "Identity Representation for RSVP," Internet-Draft <draft-ietf-rap-rsvp-identity-05.txt>, September 1999.

[AggrRSVP]   Fred Baker, Carol Iturralde, Francois Le Faucheur, Bruce Davie, "Aggregation of RSVP for IP4 and IP6 Reservations," <draft-ietf-issll-rsvp-aggr-00.txt>, September 1999

[DCLASS]   Bernet, Y., "Usage and Format of the DCLASS Object With RSVP Signaling," <draft-ietf-issll-dclass-00.txt >, August 1999.

[NullServ]   Yoram Bernet, Andrew Smith, B. Davie, "Specification of the Null Service Type," <draft-ietf-issll-nullservice-00.txt>, September 1999

[RSVPDIFF]   Bernet, R. Yavatkar, P. Ford, F. Baker, L. Zhang, M. Speer, B. Braden, B. Davie, J. Wroclawski, E. Felstaine, "Integrated Services Operation Over Diffserv Networks," <draft-ietf-issll-diffserv-rsvp-03.txt>, September 1999

9. Author Information

Silvano Gai
Cisco Systems, Inc.
170 Tasman Dr.
San Jose, CA 95134-1706
Phone: (408) 527-2690
email: sgai@cisco.com

Dinesh Dutt
Cisco Systems, Inc.
170 Tasman Dr.
San Jose, CA 95134-1706
Phone: (408) 527-0955
email: ddutt@cisco.com

Nitsan Elfassy

Cisco Systems, Inc.
Cisco Systems, Inc.
170 Tasman Dr.
San Jose, CA 95134-1706
Phone: +972 9 970 0066
email: nitsan@cisco.com

Bernet, Yoram
Microsoft
One Microsoft Way,
Redmond, WA 98052
Phone: (425) 936-9568
Email: yoramb@microsoft.com

10. Full Copyright Statement