Internet Draft
File: draft-nitsan-cops-rsvp-proxy-00.txt
Expiration Date: April 2000

Nitsan Elfassy
Dinesh Dutt
Cisco Systems

October 1999

COPS Extensions for RSVP Receiver Proxy Support

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.

Distribution of this memo is unlimited.

Copyright Notice

Abstract

This document proposes an extension to [COPS-RSVP] and [COPS] documents needed to support RSVP Receiver Proxy [RSVP-PROXY] and the Null Service Type [NULL-SERV].

Table of contents

Terminology

- RSVP: Resource ReSerVation Protocol.

- COPS: Common Open Policy Service.

- DSCP: DiffServ Code Point.

- Metering: the process of measuring the temporal properties (e.g., rate) of a traffic stream selected by a classifier. The instantaneous state of this process may be used to affect the operation of a marker, shaper, or dropper, and/or may be used for accounting and measurement purposes.

- Policing: the process of discarding packets (by a dropper) within a traffic stream in accordance with the state of a corresponding meter enforcing a traffic profile.

- Traffic conditioning: control functions performed to enforce rules specified in a TCA, including metering, marking, shaping, and policing.

- Microflow: A single instance of an application-to-application flow of packets which is identified by source address, source port, destination address, destination port and protocol id.

1. Introduction

RSVP Receiver Proxy [RSVP-PROXY] defines an extension to the RSVP message processing mainly designed to operate in conjunction with the Null Service Type [NULL-SERV]. Null Service type is a new service type proposed for use with RSVP to support applications which cannot quantify their resource requirements. The determination of resource requirements for these applications is left to the discretion of the network administrator.

The extension proposes that an intermediate router/switch receiving an RSVP Path message terminate the Path message instead of forwarding it all the way to the end destination. This router generates a proxy Resv message and sends it upstream. This originated Resv follows the same rules as any Resv message.

Existing COPS support for RSVP does not contain mechanisms to support this new functionality proposed by RSVP Receiver Proxy. This document proposes extensions to enable the use of COPS with RSVP Receiver Proxy.

2. Functionality Required to support RSVP Receiver Proxy

This section describes the nature of the additional information that needs to be exchanged between the PDP and the PEP to support RSVP Receiver Proxy and the Null Service Type.

2.1. Device capabilities

RSVP requires that network nodes be capable of reserving resources to support bandwidth allocation. These devices must also be capable of enforcing the traffic to the specified bandwidth - specified via TSPEC - such that they do not use more than their share of resources. Traffic exceeding the specified TSPEC is dropped. The bandwidth allocation and enforcement needs to be supported per each outgoing interface. For example, a multicast flow going out two separate interfaces, could have different resource requirements.

There are other capabilities such as marking that a device may or may not support. In order for the PDP to inform a PEP to enforce a decision, it would be useful for the PDP to know the capabilities of the PEP.

The device capabilities of interest follow.

2.1.1. Support for RSVP Receiver Proxy

Current IntServ capable nodes do not support the additional functionality specified by RSVP Receiver Proxy. Before the PDP can send a decision which uses this functionality, it is necessary for the PDP to know if the device supports it.

2.1.2. Support for Marking

This capability defines whether a node can mark packets and also the manner in which it can mark, using DSCP or only IP Precedence.

2.1.3. Support for Resource Reservation and Enforcement

This defines the ability of a node to reserve resources and enforce it. It also specifies whether the node can provide this functionality per each outgoing interface or only per input interface. The enforcement is accomplished using a meter and a policer.

2.2. Role Combinations

With the Null service type, the QoS assigned to a flow is upto the discretion of the network administrator. The network administrator may decide to use DiffServ to assign a QoS to the flow. The drafts related to provisioning of QoS policy in a DiffServ environment ([COPS-PR], [PIB]) specify that each interface has a set of roles associated with it. A role is simply a string that is associated with an interface and is used to group together interfaces that need to share a QoS policy. Each interface can have many roles. A "role combination" is an unordered set of roles.

Specifying the role combination associated with the ingress and egress interface associated with the Path message provides for consistency and compatbility with DiffServ policy.

2.3. Additional Decision Information

According to the new RSVP Receiver Proxy behavior, the RSVP Path message is not forwarded further. The node terminating the Path will instead originate the corresponding Resv message. This decision needs to be communicated to the PEP for a Path message.

It is the PDP that decides what policy objects need to be in the Resv message. The PDP needs to communicate these objects to the PEP.

3. COPS Objects Used To Communicate The Additional Information

The proposed extension defines new objects that are contained in the existing COPS objects. The objects used are:

o Stateless Decision object
o Client SI Named object
o Policy Data object [POL-EXT]
o DCLASS object [DCLASS]

Further explanation is provided in the following sections.

4. Definitions of the New Objects

4.1. PEP Capabilities

This section defines the objects used to communicate the RSVP-related device capabilities.

The container object used to communicate the Client capabilities is a Policy Data Object. The capability information is implemented as policy elements [POL-EXT].

The definitions of the new policy elements follow.

4.1.1 RSVP_PROXY_SUPPORT policy element

This policy element indicates if the PEP supports RSVP Receiver Proxy. This policy element MAY be sent in the Client Open message (in a POLICY DATA object that itself is encapsulated in COPS ClientSI Named object).

If the Client does not add the RSVP_PROXY_SUPPORT in the Client Open message, the PDP assumes that the PEP does not support RSVP Receiver Proxy.

```
    +-------------+-------------+-------------+-------------------+
    |        Length  = 8        | P-Type = RSVP_PROXY_SUPPORT     |
    +------+------+-------------+-------------+-------------------+
    |        Flags              |    /// Reserved ///             |
    +------+------+-------------+-------------+-------------------+
```

Length: 16 bits

The overall length of the policy element, in octets. Equals 8.

P-Type: 16 bits

RSVP_PROXY_SUPPORT policy element, as registered with IANA.

flags: 16 bits

The currently supported flags are:

0x00 - RSVP Receiver Proxy not supported
0x01 - RSVP Receiver Proxy supported

4.1.2. POLICING_SUPPORT policy element definition

This policy element indicates if the device supports metering and policing.

This policy element MAY be sent in REQ message or in the Client Open message. In case of the REQ message, the object is carried in a Named ClientSI Object following the Signaled ClientSI object that carries the RSVP message objects.

If the Client Open or REQ message does not contain the POLICING_SUPPORT policy element, the PDP assumes the PEP supports both input and output policing (the PEP could be running older code which does not define this object).

```
+-------------+-------------+-------------+-------------------+
|        Length  = 8        | P-Type = POLICING_SUPPORT       |
+------+------+-------------+-------------+-------------------+
|        Flags              |         /// reserved ////       |
+------+------+-------------+-------------+-------------------+
```

Length: 16 bits

The overall length of the policy element, in octets. Equals 8.

P-Type: 16 bits

POLICING_SUPPORT policy element, as registered with IANA.

flags: 16 bits

The currently supported flags are:

0x0 - No policing supported
0x1 - Only input-based policing
0x2 - Only output-based policing
0x3 - Both input and output-based policing

4.1.3. MARKING_SUPPORT policy element

This policy element indicates the marking capabilities of the PEP. Marking is defined as setting the ToS byte of a packet based on some defined rules.

This policy element MAY be sent in COPS REQ message or in the Client Open message. When the Client-Open or REQ message does not contain this element the PDP assumes the PEP has no marking capabilities.

```
    +-------------+-------------+-------------+-------------------+
    |        Length  = 8        | P-Type = MARKING_SUPPORT        |
    +------+------+-------------+-------------+-------------------+
    |        Flags              |         /// reserved ////       |
    +------+------+-------------+-------------+-------------------+
```

```
    Length: 16 bits

      The overall length of the policy element, in octets. Equals 8.

    P-Type: 16 bits

      MARKING_SUPPORT policy element, as registered with IANA.

    flags: 16 bits

      The currently supported flags are:
                0x0 - No Marking supported
                0x1 - Only IP Precedence Marking
                0x2 - DSCP based Marking
```

4.2. Role-Combination

As specified in section 2.2, it may also be useful add the role-combinations assigned to the ingress and egress interfaces as part of the information communicated to the PDP. Two new objects have been defined to carry this information.

The role-combination objects MAY be present in the REQ Message. The Named Client Specific Information Object (ClientSI Named) which carries the POLICY-DATA object also carries the role combination objects.

There are two role-combination objects defined, IN_ROLE_COMB and OUT_ROLE_COMB.

4.2.1. In Interface Role-Combination policy element

The format of In Interface Role Combination policy element is as follows:

```
    +-------------+-------------+-------------+-------------+
    | Length (variable)         | P-Type = IN_ROLE_COMB     |
    +------+------+-------------+-------------+-------------+
    |                 IN Role Combination                   |
    +------+------+-------------+-------------+-------------+
    |      .........                                        |
    +------+------+-------------+-------------+-------------+
```

Length: 16 bits

This is the overall length of the policy element, in octets. If the length in octets does not fall on a 32-bit word boundary, padding must be added to the end of the object so that it is aligned to the next 32-bit boundary.

P-Type: 16 bits

IN_ROLE_COMB policy element, as registered with IANA.

IN Role Combination: Role Combination string.

Role Combination is a display string as defined in [PIB].

IN_ROLE_COMB policy element MAY appear only once in the Policy Data object. If this element is absent in the REQ message, the PDP can assume a default IN Role-Combination. It is up to the PDP to figure out that default.

4.2.2. Out Interface Role Combination

The format of Out Interface Role Combination policy element is as follows:

```
+-------------+-------------+-------------+-------------+
| Length (variable)         | P-Type = OUT_ROLE_COMB    |
+------+------+-------------+-------------+-------------+
|                 OUT Role Combination                  |
+------+------+-------------+-------------+-------------+
|      .......                                          |
+------+------+-------------+-------------+-------------+
```

Length: 16 bits

This is the overall length of the policy element, in octets. If the length in octets does not fall on a 32-bit word boundary, padding must be added to the end of the object so that it is aligned to the next 32-bit boundary.

P-Type: 16 bits

OUT_ROLE_COMB policy element, as registered with IANA.

OUT Role Combination: Role Combination string.

OUT_ROLE_COMB policy element MAY appear only once in the Policy Data object. In the absence of this element in the REQ message, the PDP may assume a default OUT Role-Combination, which makes it a policy decision.

5. Communicating Additional Decisions In DEC Message

The current COPS for RSVP draft [COPS-RSVP] allows for the possibility of multiple context groups (section 3.6). We extend the use of multiple context groups to include the decision to originate a proxy Resv message.

When the PDP gets a Path IN context REQ message, it returns back a DEC message with a context group for the Path IN context, as specified in [COPS-RSVP]. In order to instruct the PEP to originate Resv, the PDP will add another context group for Resv OUT context.

Appearance of Resv OUT Decision context group in a DEC message sent for Path IN context, MUST be interpreted by the PEP as an instruction to install Resv state and originate a Resv upstream back to the previous hop defined in the Path message.

When Path IN context is "bundled" in the same REQ message with other contexts, the following rule applies:
The DEC message sent for this REQ MAY include a single Resv OUT Context Group and the PEP MUST take it as an extension to the Path IN Context Group.

5.1. Policy Information to be included in the returned Resv

The DEC message described in the previous section will include all the information to be sent back to the Sender inside the Resv. The container object for this information is the Replacement Decision object under the Resv OUT context group added to the DEC message. Among the objects that may populate the Replacement Decision object are Policy Data Object(s), DCLASS object and TSPEC object.

6. Illustrative Example

(Modified example from "COPS usage for RSVP" IETF draft). This section illustrates the steps in using COPS for controlling a unicast RSVP Receiver Proxy flow.

```
                    h1 ----> R1
                             |
                             |
                    h1 <-----+
```

Figure 1: Single PEP View

Assume that the PEP, R1 has two interfaces (if1, if2). Sender h1 sends to some receiver r1. R1 is a PEP along the path which supports RSVP Receiver Proxy. Let if1 be the interface on which h1 is connected to R1 and if2 be the outgoing interface associated with the receiver r1.

A. A Path message arrives from h1:

```
  PEP --> PDP  REQ := <Handle A>
               <Context: in & out, Path>
               <In-Interface if2> <Out-Interface if1>
               <ClientSI: all objects in Path message>
               <ClientSI: RSVP Receiver Proxy POLICY DATA -
                          IN & OUT-Role Comb)>

  PDP --> PEP  DEC := <Handle A>
               <Context: in , Path>
               <Decision: Command, Install>
               <Decision: Stateless, policy to the PEP itself>
               <Context: out, Resv>
               <Decision: Command, Install>
               <Decision: Replacement, policy objects for the Resv>
               <Context: out, Path>
               <Decision: Command, reject >
```

The decision message instructs the PEP to accept the Path message in, originate a Resv and to not forward the Path further.

7. Compatibility With Existing RSVP COPS Implementations

In order to inter-operate with existing RSVP COPS clients, the PDP must treat a Client-Open received with no capability objects specified as a device which does not support RSVP Receiver Proxy and send decisions which match the existing standard [COPS-RSVP]. The assumption made here is that clients which support the functionality detailed in this draft will also support the RSVP Receiver Proxy functionality.

If a PEP supporting RSVP Receiver Proxy talks to an older PDP, the PDP will ignore the capability objects sent. It will therefore treat all incoming messages as quantitative service type objects.

8. Security Considerations

This Section is TBD

9. References

[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

[RSVP-PROXY] Gai S., Dutt D., Elfassy N., Bernet Y., RSVP Receiver Proxy, <draft-sgai-rsvp-proxy-00.txt>, October 1999.

[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-06.txt>, February 1999.

[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

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

[COPS-PR] Reichmeyer F., Kwok Ho Chan, Durham D., Yavatkar R., Gai S., McCloghrie K., Herzog S., Smith A. "COPS Usage for Policy Provisioning", draft-sgai-cops-provisioning-00.txt, February 1999.

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

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

[PIB] M. Fine, K. McCloghrie et. al, "An Initial Policy Information Base For COPS-PR Clients and Servers", February 1999.

10. 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.

11. Author Information

Nitsan Elfassy
Cisco Systems, Inc.
4 Maskit St, P.O.Box 12497
Herzelia Pituach 46766,
Israel
Phone: +972 9 970 0066
email: nitsan@cisco.com

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

12. Full Copyright Statement