Network Working Group
Internet Draft
Expiration Date: July 2003

P. Pan (Ciena)
H. Schulzrinne (Columbia U)

An Evaluation on RSVP Transport Mechanism

draft-pan-nsis-rsvp-transport-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/1id-abstracts.txt

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

Abstract

In this memo, we look into some of the transport-layer design issues in the original RSVP as defined in RFC2205 [RFC2205]. Based on the observation, we conclude that the current RSVP transport mechanism may not be adequate to support some applications. We recommend to use a different transport layer protocol such as TCP for next-generation signaling protocols.

1. Introduction

RSVP [RFC2205] was originally designed to support real-time applications over the Internet. Over the past several years, the demand for multicast-capable real-time teleconferencing, which many people had envisioned to be one of the key Internet applications that could benefit from network-wide deployment of RSVP, has never materialized. Instead, RSVP-TE [RFC3209], a RSVP extension for traffic engineering, has been widely deployed by a large number of network providers to support MPLS applications.

Currently, there are a number of applications, such as mobile networking and residential broadband network access, that could benefit from having a signaling protocol that can reserve network resources for user application sessions. However, before we jump into the conclusion of using or not using RSVP to support such applications, we need to evaluate some of the design choices made in RSVP. One important aspect is its transport mechanism.

In this memo, we look into some of the transport-layer design issues in the original RSVP. Based on these observation, we conclude that the current RSVP transport mechanism may not be adequate to support the new applications. We recommend to use a different transport layer protocol such as TCP for next-generation signaling protocols.

2. RSVP Transport Mechanism Issues

2.1. RSVP Messaging Reliability

RSVP messages are defined as a new IP protocol (that is, a new ptype in the IP header). RSVP Path messages must be delivered end-to-end. In order for the transit routers to intercept the Path messages, a new IP Router Alert option [RFC2113] was introduced. This design is simple to implement and efficient to run. As shown from the experiments in [IP-OPT], IP option processing introduces very little overhead on a FreeBSD box with minor kernel changes.

However, RSVP does not have a good message delivery mechanism. If a message is lost on the wire, the next re-transmit cycle by the network would be one soft-state refresh interval later. By default, a soft-state refresh interval is 30 seconds.

To overcome this problem, we have introduced a staged refresh timer mechanism [STAGED], which has been defined as a RSVP extension in [RFC2961]. The staged refresh timer mechanism retransmits RSVP

messages until the receiving node acknowledges. It can address the reliability problem in RSVP.

However, during its implementation, a lot of of effort had to be spent on per-session timer maintenance, message retransmission (e.g., avoid message bursts) and message sequencing. In addition, we have to make an effort to try to separate the transport functions from protocol processing. For example, if a protocol extension requires a natural RSVP session time-out (such as RSVP-TE one-to-one fast-reroute [FAST-REROUTE]), we have to turn off the staged refresh timers.

In summary, in trying to introduce reliability in RSVP, we are getting closer to reinvent TCP. Certainly, if TCP, SCTP or similar protocols is the transport protocol for RSVP, the message reliability would not have been an issue.

2.2. RSVP Message Packing

According to RSVP [RFC2205], each RSVP message can only contain information for one session. In a network that has a reasonably large number of RSVP sessions, this constraint imposes a heavy processing burden on the routers. Many router OS is based on UNIX. From [IP-OPT], we have noticed that the UNIX socket I/O processing is not very sensitive to packet size. In fact, processing small packets takes almost as much CPU overhead as processing large ones. However, processing too many individual messages can easily cause congestion at socket I/O interfaces.

To overcome this problem, RFC2961 introduced the message bundling mechanism. The bundling mechanism packs multiple RSVP messages between two adjacent nodes into a single packet. In one deployed router platform, the bundling mechanism has improved the number of RSVP sessions that a router can handle from 2,000 to over 7,000.

2.3. RSVP MTU Problem

RSVP does not support message fragmentation and reassembly at protocol level. If the size of a RSVP message is larger than the link MTU, the message will be fragmented. And the routers simply cannot detect and process RSVP message fragments.

There is no solution for the MTU problem. Fortunately, at places where RSVP-TE has been used, either the amount of per-session RSVP data is never too large, or the link MTU is adjustable - PPP and

Frame Relay can always increase or decrease the MTU sizes. For example, on some routers, a Frame Relay interface can support the link MTU size up to 9600 bytes. Currently, the RSVP MTU problem is not a realistic concern in MPLS networks.

However, when and if RSVP is used for end-user applications, where network security is an essential and critical concern, it is possible that the size of RSVP messages can be larger than the link MTU. It is important to notice that end-users are most likely to have to deal with a small 1500-byte Ethernet MTU.

Once again, if RSVP is operated on top of TCP or similar protocols, there would be no MTU issue here.

2.4. RSVP-TE vs. Signaling Protocol for Real-Time Applications

RSVP-TE works in an environment that is different from what the original RSVP has been designed for: in MPLS networks, the RSVP sessions that are used to support Label-Switched-Paths (LSP's) do not change frequently.

In fact, the network operators typically set up the MPLS LSP's in such a way that they cannot switch too quickly. For example, the operators often regulate the CSPF (Constraint-based Shortest Path First, a routing algorithm operates from the network edge to compute the "most" optimal routes for the LSP's) computation interval to prevent or delay large volume of user traffic to shift from one session to the other during LSP path optimization. As a result, RSVP-TE does not have to handle a large amount of "triggered" (new or modified) messages. Most of the messages are refresh messages, which can be handled by the mechanisms introduced in RFC2961. In particular, in the Summary Refresh extension [RFC2961], each RSVP refresh message can be represented as a 4-byte ID. The routers can simply exchange the ID's to refresh RSVP sessions. With the full implementation of RFC2961, MPLS routers do not have any RSVP scaling issue. On one deployed router platform, it can support over 50,000 RSVP sessions in a stable backbone network.

On the other hand, in many of the new applications where a signaling protocol is required, the user session duration can be relatively short. The dynamics of adding/dropping user sessions could introduce a large number of "triggered" messages in the network. This can clearly introduce a substantial amount of processing overhead to the routers. This is one area where a new signaling protocol may be needed to reduce the processing complexity in the resource reservation process.

3. Where Do We Go From Here

A good signaling protocol should be transparent or oblivious to the applications. On the other hand, the design of a signaling protocol must take the intended and potential applications into consideration.

With the addition of RFC2961, RSVP-TE is sufficient to support its intended application, MPLS, within the backbone. There is no significant transport-layer problem that needs to be solved.

In the last several years, a number of new applications has been developed and they require the use of IP signaling. One example is midcom, which has been designed for firewall control. There are also some far-out applications such as depositing active network code on network devices. It is likely that the next-generation signaling protocols will have to deal with the network security problems. The MTU problem prevents the re-use of the existing RSVP transport mechanism.

If a new transport protocol is needed, the protocol must be able to handle the following:

- reliable messaging (Section 2.1);

- message packing (Section 2.2);

- the MTU problem (Section 2.3);

- small triggered message volume (Section 2.4).

TCP satisfies all the criteria. TCP-based signaling/routing protocols have been deployed in the Internet for years. BGP [BGP] and LDP [RFC3036] establish peering relationship between network nodes over TCP sessions. Various control information, such as routes and MPLS labels, can be exchanged between the nodes. It is quite possible that any given node may have many peers over a large number of TCP sessions. Peering and session management thus become an important implementation issue. However, this can be handled with some proper software techniques.

3.1. What About RSVP

Many applications and features have been developed on top of RSVP. This is largely because RSVP is designed as an application-neutral protocol. A great deal has learned from RSVP design, development and deployment.

We should note that RSVP has already been defined to run over UDP (albeit apparently little used). Adding or swapping another transport protocol, such as TCP, below should be relatively painless.

Hence, one idea would be to run RSVP over TCP, and change RSVP protocol to support new applications. Another possibility is to define new signaling protocols but use some of the RSVP data elements (session description, flow spec, etc.).

4. Transport-layer Protocol Swapping

So far, we have explained some of the problems that would prevent RSVP as a generic signaling protocol in the Internet. However, we should also realize that not all applications are likely to have the MTU problem, and not all applications require the messaging reliability to be accomplished over IP.

For example, in Radio Area Networks (RAN's), all messages between base stations and clients can be exchanged reliably at MAC layer. Thus singling over TCP simply introduces unnecessary processing overhead and consumes additional bandwidth.

Since the signaling messages are transported hop-by-hop, one flexible solution is to swap transport-layer protocol at every hop along a signaling path. As illustrated in the figure below, when signaling between A-B-C-D, we can run signaling over a TCP session between B-C and C-D, while sending control messages directly over IP between A-B.

```
+-+         +-+         +-+         +-+
|A|--(RAN)--|B|--(WAN)--|C|--(LAN)--|D|
+-+         +-+         +-+         +-+
```

However, this would require all nodes along the signaling path to be aware of the type of transport protocol of their neighbors. This can be accomplished through either static configuration or dynamic capability negotiation. Either mechanism is straight-forward. Capability negotiation has been designed and implemented in BGP and RSVP-TE.

5. Acknowledgments

We thank Georgios Karagiannis for valuable input.

6. References

[RFC2205] R. Braden, Ed., et al, "Resource ReSerVation protocol (RSVP) -- version 1 functional specification," RFC2205.

[RFC3209] D. Awduche, et al, "RSVP-TE: Extensions to RSVP for LSP Tunnels".

[RFC2113] D. Katz, "IP Router Alert Option".

[IP-OPT] P. Pan, H. Schulzrinne, "PF_IPOPTION: A kernel extension for IP option packet processing", Technical Memorandum 10009669-02TM, Bell Labs, Lucent Technologies, Murray Hill, NJ, June 2000.

[STAGED] P. Pan, H. Schulzrinne, "Staged refresh timers for {RSVP}", Global Internet, Phoenix, Arizona, Nov. 1997.

[RFC2961] L. Berger, et al, "RSVP refresh overhead reduction extensions", RFC 2961.

[FAST-REROUTE] P. Pan, et al, "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", draft-ietf-mpls-rsvp-lsp-fastreroute-01.txt.

[RFC3036] L. Andersson, et al, "LDP Specification".

[BGP] Y. Rekhter, et al, "A Border Gateway Protocol 4 (BGP-4)", draft-ietf-idr-bgp4-18.txt, 2002.

7. Author Information

Ping Pan
CIENA Corp.
5965 **Silver Creek Valley Road**
San Jose, CA 95134
USA
Email: ppan@ciena.com

Henning Schulzrinne
Dept. of Computer Science
Columbia University
1214 **Amsterdam Avenue**
New York, NY 10027
USA
Email: hgs@cs.columbia.edu