TEAS Working Group
Internet Draft
Intended status: Proposed Standard

Vishnu Pavan Beeram
Juniper Networks
Ina Minei
Google, Inc
Rob Shakir
Jive Communications
Dante Pacella
Verizon
Tarek Saad
Cisco Systems

Expires: September 21, 2016

March 21, 2016

Implementation Recommendations to Improve the Scalability of RSVP-TE Deployments

draft-ietf-teas-rsvp-te-scaling-rec-01

Status of this Memo

Copyright Notice

Abstract

The scale at which RSVP-TE Label Switched Paths (LSPs) get deployed is growing continually and the onus is on RSVP-TE implementations across the board to keep up with this increasing demand.

This document makes a set of implementation recommendations to help RSVP-TE deployments push the envelope on scaling and advocates the use of a couple of techniques - "Refresh Interval Independent RSVP (RI-RSVP)" and "Per-Peer flow-control" - for improving scaling.

Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [RFC2119].

Table of Contents

1. Introduction

The scale at which RSVP-TE [RFC3209] Label Switched Paths (LSPs) get deployed is growing continually and there is considerable onus on RSVP-TE implementations across the board to keep up with this increasing demand in scale.

The set of RSVP Refresh Overhead Reduction procedures [RFC2961] serves as a powerful toolkit for RSVP-TE implementations to help cover a majority of the concerns about soft-state scaling. However, even with these tools in the toolkit, analysis of existing implementations [RFC5439] indicates that the processing required under certain scale may still cause significant disruption to an LSR.

This document builds on the scaling work and analysis that has been done so far and makes a set of concrete implementation recommendations to help RSVP-TE deployments push the envelope further on scaling - push higher the threshold above which an LSR struggles to achieve sufficient processing to maintain LSP state.

This document advocates the use of a couple of techniques - "Refresh Interval Independent RSVP (RI-RSVP)" and "Per-Peer flow-control" - for significantly cutting down the amount of processing cycles required to maintain LSP state. "RI-RSVP" helps completely eliminate RSVP's reliance on refreshes and refresh-timeouts while "Per-Peer Flow-Control" enables a busy RSVP speaker to apply back pressure to its peer(s). In order to reap maximum scaling benefits, it is strongly RECOMMENDED that implementations support both the techniques, but it is possible for an implementation to support just one but not the other.

2. Recommendations

2.1. "RFC2961 specific" Recommendations

The implementation recommendations discussed in this section are based on the proposals made in [RFC2961] and act as pre-requisites

for implementing either or both of the techniques discussed in Sections 2.2 and 2.3.

2.1.1. Basic Pre-Requisites

An implementation that supports either or both of the techniques discussed in Sections 2.2 and 2.3:

- SHOULD indicate support for RSVP Refresh Overhead Reduction extensions (as specified in Section 2 of [RFC2961]) by default, with the ability to override the default via configuration.

- MUST support reliable delivery of Path/Resv and the corresponding Tear/Err messages using the procedures specified in [RFC2961].

- MUST support retransmit of all RSVP-TE messages using exponential-backoff, as specified in Section 6 of [RFC2961].

2.1.2. Making Acknowledgements mandatory

The reliable message delivery mechanism specified in [RFC2961] states that "Nodes receiving a non-out of order message containing a MESSAGE_ID object with the ACK_Desired flag set, SHOULD respond with a MESSAGE_ID_ACK object."

In an implementation that supports either or both of the techniques discussed in Sections 2.2 and 2.3, nodes receiving a non-out of order message containing a MESSAGE ID object with the ACK-Desired flag set, MUST respond with a MESSAGE_ID_ACK object. This improvement to the predictability of the system in terms of reliable message delivery is key for being able to take any action based on a non-receipt of an ACK.

2.1.3. Clarifications on reaching Rapid Retry Limit (Rl)

According to section 6 of [RFC2961] "The staged retransmission will continue until either an appropriate MESSAGE_ID_ACK object is received, or the rapid retry limit, Rl, has been reached." The following clarifies what actions, if any, a router should take once Rl has been reached.

If it is the retransmission of Tear/Err messages and Rl has been reached, the router need not take any further actions. If it is the retransmission of Path/Resv messages and Rl has been reached, then the router starts periodic retransmission of these messages. The

retransmitted messages MUST carry MESSAGE_ID object with ACK_Desired flag set. This periodic retransmission SHOULD continue until an appropriate MESSAGE_ID ACK object is received indicating acknowledgement of the (retransmitted) Path/Resv message. The configurable periodic retransmission interval SHOULD be less than the regular refresh interval. A default periodic retransmission interval of 30 seconds is RECOMMENDED by this document.

2.2. Refresh Interval Independent RSVP

The RSVP protocol relies on periodic refreshes for state synchronization between RSVP neighbors and for recovery from lost RSVP messages. It relies on refresh timeout for stale state cleanup. The primary motivation behind introducing the notion of "Refresh Interval Independent RSVP" (RI-RSVP) is to completely eliminate RSVP's reliance on refreshes and refresh timeouts. This is done by simply increasing the refresh interval to a fairly large value. [RFC2961] and [RFC5439] do talk about increasing the value of the refresh-interval to provide linear improvement on transmission overhead, but also point out the degree of functionality that is lost by doing so. This section revisits this notion, but also proposes sufficient recommendations to make sure that there is no loss of functionality incurred by increasing the value of the refresh interval.

An implementation that supports RI-RSVP:

- MUST support all the recommendations made in Section 2.1

- MUST make the default value of the configurable refresh interval

be a large value (10s of minutes). A default value of 20 minutes is RECOMMENDED by this document.

- MUST implement coupling the state of individual LSPs with the state of the corresponding RSVP-TE signaling adjacency. When an RSVP-TE speaker detects RSVP-TE signaling adjacency failure, the speaker MUST act as if the all the Path and Resv state learnt via the failed signaling adjacency has timed out.

- MUST make use of Node-ID based Hello Session ([RFC3209], [RFC4558]) for detection of RSVP-TE signaling adjacency failures; A default value of 9 seconds is RECOMMENDED by this document for the configurable node hello interval (as opposed to the 5ms default value proposed in Section 5.3 of [RFC3209]).

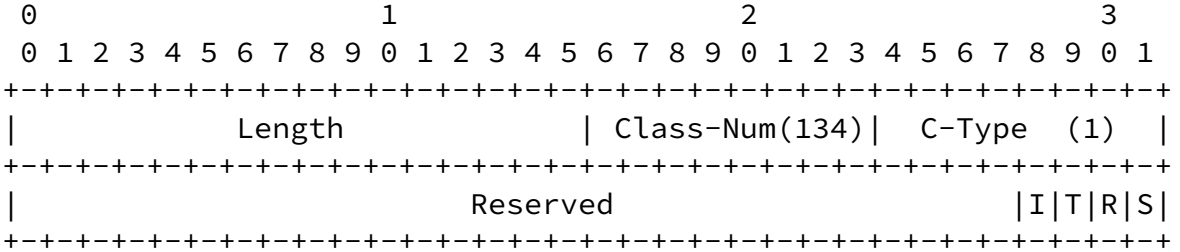
- (If Bypass FRR [RFC4090] is supported,) MUST implement procedures specified in [RI-RSVP-FRR] which describes methods to facilitate FRR that works independently of the refresh-interval.

- MUST indicate support for RI-RSVP via the CAPABILITY object in Hello messages.

2.2.1. Capability Advertisement

An implementation supporting the RI-RSVP recommendations MUST set a new flag "RI-RSVP Capable" in the CAPABILITY object signaled in Hello messages.

The new flag that will be introduced to CAPABILITY object is specified below.



I bit

Indicates that the sender supports RI-RSVP.

Any node that sets the new I-bit in its CAPABILITY object MUST also set Refresh-Reduction-Capable bit in common header of all RSVP-TE messages.

2.2.2. Compatibility

The RI-RSVP functionality MUST be activated only between peers that indicate their support for this functionality. The RI-RSVP specific Bypass FRR procedures discussed in [RI-RSVP-FRR] introduce a few new protocol extensions and those MUST get activated only if the participating nodes support RI-RSVP functionality.

2.3. Per-Peer RSVP Flow Control

The set of recommendations discussed in this section provide an RSVP speaker with the ability to apply back pressure to its peer(s) to reduce/eliminate RSVP-TE control plane congestion.

An implementation that supports "Per-Peer RSVP Flow Control":

- MUST support all the recommendations made in Section 2.1

- MUST use lack of ACKs from a peer as an indication of peer's RSVP-TE control plane congestion. If congestion is detected, the local system MUST throttle RSVP-TE messages to the affected peer. This MUST be done on a per-peer basis. (Per-peer throttling MAY be implemented by a traffic shaping mechanism that proportionally reduces the RSVP signaling packet rate as the number of outstanding Acks increases. And when the number of outstanding Acks decreases, the send rate would be adjusted up again.)

- SHOULD use a Retry Limit (Rl) value of 7 (Section 6.2 of [RFC2961], suggests using 3).

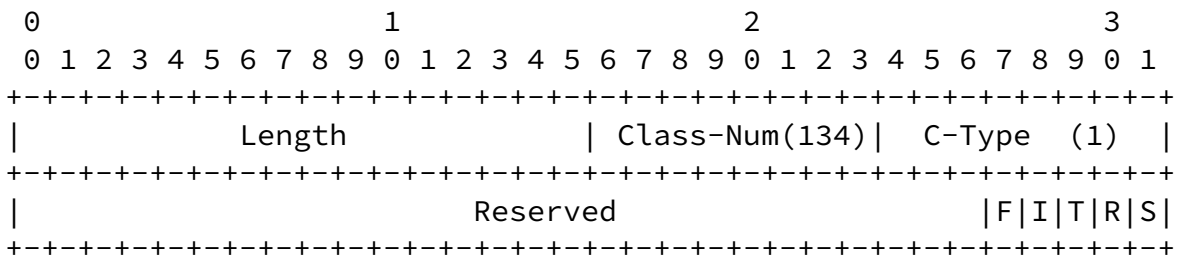
- SHOULD prioritize Tear/Error over trigger Path/Resv (messages that bring up new LSP state) sent to a peer when the local system detects RSVP-TE control plane congestion in the peer.

- MUST indicate support for all recommendations in this section via the CAPABILITY object in Hello messages.

2.3.1. Capability Advertisement

An implementation supporting the "Per-Peer Flow Control" recommendations MUST set a new flag "Per-Peer Flow Control Capable" in the CAPABILITY object signaled in Hello messages.

The new flag that will be introduced to CAPABILITY object is specified below.



F bit

Indicates that the sender supports Per-Peer RSVP Flow Control

Any node that sets the new I-bit in its CAPABILITY object MUST also set Refresh-Reduction-Capable bit in common header of all RSVP-TE messages.

2.3.2. Compatibility

The "Per-Peer Flow Control" functionality MUST be activated only if both peers support it. If a peer hasn't indicated that it is capable of participating in "Per-Peer Flow Control", then it is risky to assume that the peer would always acknowledge a non-out of order message containing a MESSAGE ID object with the ACK-Desired flag set.

2.4. Other Recommendations

The following scaling recommendations have no interdependency with any of the techniques/recommendations specified in Sections 2.2 and 2.3. These are stand-alone functionalities that help improve RSVP-TE scalability.

2.4.1. Summary FRR

If Bypass FRR [RFC4090] is supported by an implementation, it SHOULD support the procedures discussed in [SUMMARY-FRR]. These procedures reduce the amount of RSVP signaling required for Fast Reroute procedures and subsequently improve the scalability of RSVP-TE signaling when undergoing FRR convergence post a link or node failure.

3. Security Considerations

This document does not introduce new security issues. The security considerations pertaining to the original RSVP protocol [RFC2205] and RSVP-TE [RFC3209] and those that are described in [RFC5920] remain relevant.

4. IANA Considerations

4.1. Capability Object Values

IANA maintains all the registries associated with "Resource Reservation Protocol (RSVP) Paramaters" (see http://www.iana.org/assignments/rsvp-parameters/rsvp-parameters.xhtml). "Capability Object Values" Registry (introduced by [RFC5063]) is one of them.

IANA is requested to assign two new Capability Object Value bit flags as follows:

Bit	Hex	Name	Reference

Number	Value		
TBA	TBA	RI-RSVP Capable (I)	Section 2.2.1
TBA	TBA	Per-Peer Flow Control Capable (F)	Section 2.3.1

5. References

5.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, R., "Resource Reservation Protocol (RSVP)", RFC 2205, September 1997.

[RFC2961] Berger, L., "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[RFC3209] Awduche, D., "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC4090] Pan, P., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May 2005.

[RFC4558] Ali, Z., "Node-ID Based Resource Reservation (RSVP) Hello: A Clarification Statement", RFC 4558, June 2006.

[RFC5063] Satyanarayana, A., "Extensions to GMPLS Resource Reservation Protocol Graceful Restart", RFC5063, October 2007.

[RI-RSVP-FRR] Ramachandran, C., "Refresh Interval Independent FRR

Facility Protection", draft-chandra-mpls-ri-rsvp-frr, (work in progress)

[SUMMARY-FRR] Taillon, M., "RSVP-TE Summary Fast Reroute Extensions for LSP Tunnels", draft-mtaillon-mpls-summary-frr-rsvpte, (work in progress)

5.2. Informative References

[RFC5439] Yasukawa, S., "An Analysis of Scaling Issues in MPLS-TE

Core Networks", RFC 5439, February 2009.

[RFC5920] Fang, L., "Security Framework for MPLS and GMPLS Networks", RFC5920, July 2010.

6. Acknowledgments

The authors would like to thank Yakov Rekhter for initiating this work and providing valuable inputs. They would like to thank Raveendra Torvi and Chandra Ramachandran for participating in the many discussions that led to the recommendations made in this document. They would also like to thank Adrian Farrel for providing detailed review comments.

Appendix A. Recommended Defaults

(a) Refresh-Interval (R)- 20 minutes (Section 2.2)
Given that an implementation supporting RI-RSVP doesn't rely on refreshes for state sync between peers, the RSVP refresh interval is sort of analogous to IGP refresh interval, the default of which is typically in the order of 10s of minutes. Choosing a default of 20 minutes allows the refresh timer to be randomly set to a value in the range [10 minutes (0.5R), 30 minutes (1.5R)].

(b) Node Hello-Interval - 9 Seconds (Section 2.2)
[RFC3209] defines the hello timeout as 3.5 times the hello interval. Choosing 9 seconds for the node hello-interval gives a hello timeout of 3.5*9 = 31.5 seconds. This puts the hello timeout value to be in the same ballpark as the IGP hello timeout value.

(c) Retry-Limit (Rl) - 7 (Section 2.3)
Choosing 7 as the retry-limit results in an overall rapid retransmit phase of 31.5 seconds. This nicely matches up with the 31.5 seconds hello timeout.

(d) Periodic Retransmission Interval - 30 seconds (Section 2.1.3)
If the Retry-Limit (Rl) is 7, then it takes about 30 (31.5 to be precise) seconds for the 7 rapid retransmit steps to max out. (The last delay from message 6 to message 7 is 16 seconds). The 30 seconds interval also matches the traditional default refresh time.

Contributors

Markus Jork
Juniper Networks
Email: mjork@juniper.net

Ebben Aries
Juniper Networks
Email: exa@juniper.net

Authors' Addresses

Vishnu Pavan Beeram (Ed)
Juniper Networks
Email: vbeeram@juniper.net

Ina Minei
Google, Inc
Email: inaminei@google.com

Rob Shakir
Jive Communications, Inc
Email: rjs@rob.sh

Dante Pacella
Verizon
Email: dante.j.pacella@verizon.com

Tarek Saad
Cisco Systems
Email: tsaad@cisco.com