Internet Engineering Task Force (IETF)
Request for Comments: 8370
Category: Standards Track
ISSN: 2070-1721

V. Beeram, Ed.
Juniper Networks
I. Minei
R. Shakir
Google, Inc
D. Pacella
Verizon
T. Saad
Cisco Systems

May 2018

Techniques to Improve the Scalability of RSVP-TE Deployments

Abstract

Networks that utilize RSVP-TE LSPs are encountering implementations that have a limited ability to support the growth in the number of LSPs deployed.

This document defines two techniques, Refresh-Interval Independent RSVP (RI-RSVP) and Per-Peer Flow Control, that reduce the number of processing cycles required to maintain RSVP-TE LSP state in Label Switching Routers (LSRs) and hence allow implementations to support larger scale deployments.

Status of This Memo

This is an Internet Standards Track document.

This document is a product of the Internet Engineering Task Force (IETF). It represents the consensus of the IETF community. It has received public review and has been approved for publication by the Internet Engineering Steering Group (IESG). Further information on Internet Standards is available in Section 2 of RFC 7841.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at https://www.rfc-editor.org/info/rfc8370.

Table of Contents

1. Introduction

Networks that utilize RSVP-TE [RFC3209] LSPs are encountering implementations that have a limited ability to support the growth in the number of LSPs deployed.

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 beyond a certain scale may still cause significant disruption to a Label Switching Router (LSR).

This document builds on existing scaling work and analysis and defines protocol extensions to help RSVP-TE deployments push the envelope further on scaling by increasing the threshold above which an LSR struggles to achieve sufficient processing to maintain LSP state.

This document defines two techniques, Refresh-Interval Independent RSVP (RI-RSVP) and Per-Peer Flow Control, that cut down the number 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). This document defines a unique RSVP Capability [RFC5063] for each technique (support for the CAPABILITY object is a prerequisite for implementing these techniques). Note that the Per-Peer Flow-Control technique requires the RI-RSVP technique as a prerequisite. In order to reap maximum scaling benefits, it is strongly recommended that implementations support both techniques and have them enabled by default. Both techniques are fully backward compatible and can be deployed incrementally.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

2. Required Support for RFC 2961

The techniques defined in Sections 3 and 4 are based on proposals made in [RFC2961]. Implementations of these techniques need to support the RSVP messages and procedures defined in [RFC2961] with some minor modifications and alterations to recommended time intervals and iteration counts (see Appendix A for the set of recommended defaults).

2.1. Required Functionality from RFC 2961

An implementation that supports the techniques discussed in Sections 3 and 4 must support the functionality described in [RFC2961] as follows:

- It MUST indicate support for RSVP Refresh Overhead Reduction extensions (as specified in Section 2 of [RFC2961]).
- It MUST support receipt of any RSVP Refresh Overhead Reduction message as defined in [RFC2961].
- It MUST initiate all RSVP Refresh Overhead Reduction mechanisms as defined in [RFC2961] (including the SRefresh message) with the default behavior being to initiate the mechanisms; however, a configuration override should be offered.
- It MUST support reliable delivery of Path/Resv and the corresponding Tear/Err messages (as specified in Section 4 of [RFC2961]).
- It MUST support retransmission of all unacknowledged RSVP-TE messages using exponential backoff (as specified in Section 6 of [RFC2961]).

2.2. Making Acknowledgements Mandatory

The reliable message delivery mechanism specified in [RFC2961] states that "Nodes receiving a non-out of order [sic] 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 the techniques discussed in Sections 3 and 4, 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 MESSAGE_ID_ACK object can be packed with other MESSAGE_ID_ACK or MESSAGE_ID_NACK objects and sent in an Ack message (or piggybacked in any other RSVP message).

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.

3. Refresh-Interval Independent RSVP (RI-RSVP)

The RSVP protocol relies on periodic refreshes for state synchronization between RSVP neighbors and recovery from lost RSVP messages. It relies on a 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] talk about increasing the value of the refresh interval to provide linear improvement of transmission overhead, but they also point out the degree of functionality that is lost by doing so. This section revisits this notion, but also sets out additional requirements 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 of the requirements specified in Section 2.
- MUST make the default value of the configurable refresh interval (R) be a large value (tens of minutes). A default value of 20 minutes is RECOMMENDED by this document.
- MUST use a separate shorter refresh interval for refreshing state associated with unacknowledged Path/Resv (uR) messages. A default value of 30 seconds 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 all the Path and Resv states learned via the failed signaling adjacency have timed out.
- MUST make use of the Hello session based on the Node-ID ([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 default value of 5 milliseconds proposed in Section 5.3 of [RFC3209]).
- MUST indicate support for RI-RSVP via the CAPABILITY object [RFC5063] in Hello messages.

3.1. Capability Advertisement

An implementation supporting the RI-RSVP technique MUST set a new flag, RI-RSVP Capable, in the CAPABILITY object signaled in Hello messages. The following bit indicates that the sender supports RI-RSVP:

Bit Number 28 (0x0008) - RI-RSVP Capable (I-bit)

Any node that sets the new I-bit in its CAPABILITY object MUST also set the Refresh-Reduction-Capable bit [RFC2961] in the common header of all RSVP-TE messages. If a peer sets the I-bit in the CAPABILITY object but does not set the Refresh-Reduction-Capable bit, then the RI-RSVP functionality MUST NOT be activated for that peer.

3.2. Compatibility

The RI-RSVP functionality MUST NOT be activated with a peer that does not indicate support for this functionality. Inactivation of the RI-RSVP functionality MUST result in the use of the traditional smaller refresh interval [RFC2205].

4. Per-Peer Flow Control

The functionality discussed in this section provides an RSVP speaker with the ability to apply back pressure to its peer(s) to reduce/eliminate a significant portion of the RSVP-TE control message load.

An implementation that supports Per-Peer Flow Control:

- MUST support all of the requirements specified in Section 2.
- MUST support RI-RSVP (Section 3).
- MUST treat lack of ACKs from a peer as an indication of a 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. 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 Hello messages and messages carrying Acknowledgements over other RSVP messages.

- 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 this technique via the CAPABILITY object [RFC5063] in Hello messages.

4.1. Capability Advertisement

An implementation supporting the Per-Peer Flow-Control technique MUST set a new flag, Per-Peer Flow-Control Capable, in the CAPABILITY object signaled in Hello messages. The following bit indicates that the sender supports Per-Peer Flow Control:

Bit Number 27 (0x0010) - Per-Peer Flow-Control Capable (F-bit)

Any node that sets the new F-bit in its CAPABILITY object MUST also set the Refresh-Reduction-Capable bit in the common header of all RSVP-TE messages. If a peer sets the F-bit in the CAPABILITY object but does not set the Refresh-Reduction-Capable bit, then the Per-Peer Flow-Control functionality MUST NOT be activated for that peer.

4.2. Compatibility

The Per-Peer Flow-Control functionality MUST NOT be activated with a peer that does not indicate support for this functionality. If a peer hasn't indicated that it is capable of participating in Per-Peer Flow Control, then it SHOULD NOT be assumed that the peer would always acknowledge a non-out-of-order message containing a MESSAGE_ID object with the ACK_Desired flag set.

5. IANA Considerations

5.1. Capability Object Values

IANA maintains the "Capability Object values" subregistry [RFC5063] within the "Resource Reservation Protocol (RSVP) Parameters" registry <http://www.iana.org/assignments/rsvp-parameters>. IANA has assigned two new Capability Object Value bit flags as follows:

Bit Number	Hex Value	Name	Reference
28	0x0008	RI-RSVP Capable (I)	Section 3
27	0x0010	Per-Peer Flow-Control Capable (F)	Section 4

6. 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.

7. References

7.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC2205] Braden, R., Ed., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, DOI 10.17487/RFC2205, September 1997, <https://www.rfc-editor.org/info/rfc2205>.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F., and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, DOI 10.17487/RFC2961, April 2001, <https://www.rfc-editor.org/info/rfc2961>.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, DOI 10.17487/RFC3209, December 2001, <https://www.rfc-editor.org/info/rfc3209>.

[RFC4558] Ali, Z., Rahman, R., Prairie, D., and D. Papadimitriou, "Node-ID Based Resource Reservation Protocol (RSVP) Hello: A Clarification Statement", RFC 4558, DOI 10.17487/RFC4558, June 2006, <https://www.rfc-editor.org/info/rfc4558>.

[RFC5063] Satyanarayana, A., Ed. and R. Rahman, Ed., "Extensions to GMPLS Resource Reservation Protocol (RSVP) Graceful Restart", RFC 5063, DOI 10.17487/RFC5063, October 2007, <https://www.rfc-editor.org/info/rfc5063>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

7.2. Informative References

[RFC5439] Yasukawa, S., Farrel, A., and O. Komolafe, "An Analysis of Scaling Issues in MPLS-TE Core Networks", RFC 5439, DOI 10.17487/RFC5439, February 2009, <https://www.rfc-editor.org/info/rfc5439>.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, DOI 10.17487/RFC5920, July 2010, <https://www.rfc-editor.org/info/rfc5920>.

Appendix A. Recommended Defaults

a. Refresh Interval (R) - 20 minutes (Section 3):
Given that an implementation supporting RI-RSVP doesn’t rely on refreshes for state sync between peers, the function of the RSVP refresh interval is analogous to that of IGP refresh interval (the default of which is typically in the order of tens 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 3):

[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 in the vicinity of the IGP hello timeout value.

c. Retry-Limit (Rl) - 7 (Section 4):
Choosing 7 as the retry-limit results in an overall rapid retransmit phase of 31.5 seconds. This matches up with the hello timeout of 31.5 seconds.

d. Refresh Interval for refreshing state associated with unacknowledged Path/Resv messages (uR) - 30 seconds (Section 3):
The recommended refresh interval (R) value of 20 minutes (for an implementation supporting RI-RSVP) cannot be used for refreshing state associated with unacknowledged Path/Resv messages. This document recommends the use of the traditional default refresh interval value of 30 seconds for uR.

Acknowledgements

The authors would like to thank Yakov Rekhter for initiating this work and providing valuable input. They would like to thank Raveendra Torvi and Chandra Ramachandran for participating in the many discussions that led to the techniques discussed in this document. They would also like to thank Adrian Farrel, Lou Berger, and Elwyn Davies for providing detailed review comments and text suggestions.

Contributors

Markus Jork
Juniper Networks
Email: mjork@juniper.net

Ebben Aries
Juniper Networks
Email: exa@juniper.net

Authors' Addresses

Vishnu Pavan Beeram (editor)
Juniper Networks

Email: vbeeram@juniper.net

Ina Minei
Google, Inc

Email: inaminei@google.com

Rob Shakir
Google, Inc

Email: rjs@rob.sh

Dante Pacella
Verizon

Email: dante.j.pacella@verizon.com

Tarek Saad
Cisco Systems

Email: tsaad@cisco.com