

TEAS WG
Internet-Draft
Updates: [4875](#) (if approved)
Intended status: Standards Track
Expires: August 2, 2019

Z. Zhang
A. Deshmukh
R. Singh
Juniper Networks
January 29, 2019

RSVP-TE P2MP Tunnels on RMR
draft-zzhang-teas-rmr-rsvp-p2mp-02

Abstract

This document specifies the optimization in RSVP-TE P2MP tunnel signaling over Resilient MPLS Rings (RMR).

Requirements Language

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 [RFC2119](#).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

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."

This Internet-Draft will expire on August 2, 2019.

Copyright Notice

Copyright (c) 2019 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect

Internet-Draft

rmr-rsvp-p2mp

January 2019

to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
2.	Specification	4
2.1.	RMR Object	4
2.2.	Procedures	4
2.2.1.	PATH Message/State	4
2.2.2.	RESV Message/State	5
3.	Security Considerations	5
4.	Acknowledgements	6
5.	Normative References	6
	Authors' Addresses	6

[1.](#) Introduction

Traditional RSVP-TE P2MP tunnel signaling could be quite involving. With RMR, this could be significantly simplified:

There is no need for ERO/RRO/SERO/SRRO or hop by hop routing. The tunnel ingress simply sends PATH messages in one or both directions of the ring, depending on how leaves are best reached. The <S2L Sub-LSP Descriptor List> only needs to list the tunnel leaves, and a transit router does not need to "branch" a PATH message into multiple ones. Therefore, unless there are many tunnel leaves on a huge ring, a single PATH message is enough. In the rare situation of a large tunnel with many leaves to list, a small number of PATH messages should suffice. Additionally, there is no need to signal and maintain individual sub-LSPs (one for each leaf) any more. As a result, corresponding PATH/RESV state is also reduced. Each node only needs to maintain a single PATH state and a single RESV state for each P2MP tunnel, and the RESV state does not need to track individual leaves - it just need to track if a RESV is received from downstream and/or if this node itself is a leaf.

A RESV message is triggered to the PHOP when the RESV state is first created (either because the node is a leaf or because a RESV message is received from downstream) and it is refreshed periodically. A RESV Tear is sent when the RESV state is deleted (when the node is no

longer a Leaf and the RESV from downstream has timed out or a RESV Tear is received).

Optionally, the tunnel ingress may not need to list any/all leaves. It could simply send the PATH message around the ring, with the <S2L

Sub-LSP Descriptor List> listing the root itself. Through methods outside the scope of this document, a node determines if it is a leaf of the tunnel, and if yes, it will send back a RESV message. With this, a single PATH message is surely enough.

In this document, leaves in <S2L Sub-LSP Descriptor List> are referred to as explicit leaves, and leaves not listed there but self-determined by ring nodes are referred to as implicit leaves. There could be both explicit and implicit leaves for a tunnel. The ingress allows implicit leaves by including itself as the last one in the <S2L Sub-LSP Descriptor List>.

Optionally, the RESV message could also include a <S2L Sub-LSP Descriptor List> to list all the leaves on the established tunnel so that the each node knows its downstream leaves. In that case, when the set of downstream leaves changes, a RESV message with the new <S2L Sub-LSP Descriptor List> is triggered.

Adding/removing explicit leaves is straightforward. The ingress simply sends a triggered PATH message with new <S2L Sub-LSP Descriptor List>. As it passes around the ring, each node determines if it is an explicit leaf and updates its state accordingly. The triggered PATH message does not have to go all the way to the last leaf - if on a node the <S2L Sub-LSP Descriptor List> in the would-be-sent PATH message is the same as what was sent before, the triggered PATH message will not be sent further.

To indicate that the tunnel signaling is with above mentioned RMR optimizations, a new object is included in the PATH message to specify the Ring ID and direction.

Link/Node protection is achieved by tunneling packets to the next node using the Ring LSP to that node in the other direction. This does not need any additional signaling but is based on a reasonable premise that unicast Ring LSPs are always in place. Once the ingress learns the failure (through IGP discovery or through other error

detection/notification mechanisms), global repair kicks in to reach some leaves via PATH message sent in the other direction. Before global repair is finished, traffic continues to flow in the original path except that at the failure point it is tunneled to the next node.

If an RMR is just part of a general RSVP network the optimization can also be applied on the ring nodes. If the tunnel ingress knows the leaves that are on the ring, it could put all those leaves in the single PATH message and construct the ERO/SERO only towards the entry point on the ring. The entry point then includes the RMR object in the PATH messages that it sends. For leaves beyond the ring, the

ingress may include the exit points on the ring as loose hops in the ERO/SERO, and when a ring node needs to send the PATH message off the ring, it removes the RMR object. Details will be provided in future revisions of this document.

[2. Specification](#)

[2.1. RMR Object](#)

The RMR object is a new object of the following:

- o Class Name: RMR
- o Class-Num: TBA1 (to be assigned by IANA)
- o C-Type: TBA2 (to be assigned by IANA)

The format of the object content following the common object header is the following:

```

      0               1               2               3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Ring ID (4 octets)                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|D| Flags          |          Reserved          |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

Following the 4-octect Ring ID, there is an 8-bit Flags field. The

first bit of the Flags field indicates the direction. If it is set, it is clockwise direction. Otherwise, it is anti-clockwise.

[2.2.](#) Procedures

This section describes the differences in the procedures for ring nodes to set up RSVP-TE P2MP tunnels across the ring, compared to the conventional non-RMR-aware case. For now it is assumed that all nodes (ingress, transit, and leaves) on the tunnel are on the ring.

More details will be provided in future revisions.

[2.2.1.](#) PATH Message/State

The tunnel ingress includes the RMR object with the Ring ID and the direction flag bit set accordingly. The explicit tunnel leaves are encoded in the <S2L Sub-LSP Descriptor List>, and no ERO/SERO is included. If the tunnel allows implicit leaves, the descriptor list encodes the ingress itself as the last element. The message is sent

to the next node on the ring in the direction specified in the RMR object, w/o using ERO/SERO or hop-by-hop routing.

When a node receives a PATH message with the RMR object, it checks if itself is listed in the <S2L Sub-LSP Descriptor List>, or if the <S2L Sub-LSP Descriptor List> encodes the tunnel ingress as the last element and this node itself is an implicit leaf. If yes, it creates corresponding RESV state and sends a RESV message to the PHOP.

The receiving node removes itself from the <S2L Sub-LSP Descriptor List> in the PATH message, and saves the list locally. The PATH message is sent to the next node on the ring in the specified direction with the saved <S2L Sub-LSP Descriptor List>, if one of the following conditions is met:

- o The <S2L Sub-LSP Descriptor List> encodes the tunnel ingress itself as the last element.
- o The <S2L Sub-LSP Descriptor List> is not empty and either the PATH state is newly created or the <S2L Sub-LSP Descriptor List> is different from the previously saved one.

If <S2L Sub-LSP Descriptor List> is empty and different from the previously saved one, a PATH Teardown is sent instead.

[2.2.2.](#) RESV Message/State

A ring node may know that it is a leaf when the PATH message is first processed as described in the previous section. In case of implicit leaves, it may become a leaf after the PATH messages has been processed. A non-leaf node may also receive a RESV message from its NHOP. In all these cases, the node creates RESV state and sends a RESV message to the PHOP, w/o encoding RR0/SRR0.

If a ring node was a leaf but stops being a leaf, either because it is no longer listed in the <S2L Sub-LSP Descriptor List> or it is no longer an implicit leaf, it removes/updates corresponding local state. A RESV Teardown is sent to the PHOP if there is no RESV received from its downstream either.

[3.](#) Security Considerations

This document does not introduce new security risks.

[4.](#) Acknowledgements

[5.](#) Normative References

[I-D.ietf-mpls-rmr]

Kompella, K. and L. Contreras, "Resilient MPLS Rings",
[draft-ietf-mpls-rmr-09](#) (work in progress), January 2019.

[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>>.

[RFC4875] Aggarwal, R., Ed., Papadimitriou, D., Ed., and S. Yasukawa, Ed., "Extensions to Resource Reservation

Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", [RFC 4875](#), DOI 10.17487/RFC4875, May 2007, <<https://www.rfc-editor.org/info/rfc4875>>.

Authors' Addresses

Zhaohui Zhang
Juniper Networks

EMail: zzhang@juniper.net

Abhishek Deshmukh
Juniper Networks

EMail: adeshmukh@juniper.net

Ravi Singh
Juniper Networks

EMail: ravis@juniper.net