

TEAS WG  
Internet-Draft  
Intended status: Standards Track  
Expires: December 30, 2018

A. Deshmukh  
K. Kompella  
Juniper Networks, Inc.  
June 28, 2018

RSVP Extensions for RMR  
draft-ietf-teas-rsvp-rmr-extension-01

Abstract

Ring topology is commonly found in access and aggregation networks. However, the use of MPLS as the transport protocol for rings is very limited today. draft-ietf-mpls-rmr-02 describes a mechanism to handle rings efficiently using MPLS. This document describes the extensions to the RSVP-TE protocol for signaling MPLS label switched paths in rings.

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 December 30, 2018.

Copyright Notice

Copyright (c) 2018 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 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
1.1. Requirements Language . . . . .	3
2. Terminology . . . . .	3
3. RSVP Extensions . . . . .	4
3.1. Session Object . . . . .	4
3.2. SENDER_TEMPLATE, FILTER_SPEC Objects . . . . .	5
4. Ring Signaling Procedures . . . . .	5
4.1. Differences from regular RSVP-TE LSPs . . . . .	5
4.2. LSP signaling . . . . .	6
4.2.1. Path Propagation for RMR . . . . .	8
4.2.2. Resv Processing for RMR . . . . .	8
4.3. Protection . . . . .	9
4.4. Ring changes . . . . .	10
4.5. Express Links . . . . .	11
4.6. Bandwidth management . . . . .	11
5. Security Considerations . . . . .	13
6. Contributors . . . . .	13
7. IANA Considerations . . . . .	14
8. References . . . . .	14
8.1. Normative References . . . . .	14
8.2. Informative References . . . . .	14
Authors' Addresses . . . . .	15

## 1. Introduction

This document extends RSVP-TE [RFC3209] to establish label-switched path (LSP) tunnels in the ring topology. Rings are auto-discovered using the mechanisms mentioned in the [draft-ietf-mpls-rmr-02]. Either IS-IS [RFC5305] or OSPF[RFC3630] can be used as the IGP for auto-discovering the rings.

After the rings are auto-discovered, each node in the ring knows its clockwise(CW) and anti-clockwise (AC) ring neighbors and its ring links. All of the express links in the ring also get identified as part of the auto-discovery process. At this point, every node in the ring informs the RSVP protocol to begin the signaling of the ring LSPs.

Section 2 covers the terminology used in this document. Section 3 presents the RSVP protocol extensions needed to support MPLS rings. Section 4 describes the procedures of RSVP LSP signaling in detail.

### 1.1. Requirements Language

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. Terminology

Assuming there are  $n$  nodes in the network, a ring gets formed by a subset of those  $n$  nodes  $\{R_i, R_{i+1}, R_{i+2}, \dots, R_n\}$ . We define the direction from node  $R_i$  to  $R_{i+1}$  as "clockwise" (CW) and the reverse direction as "anti-clockwise" (AC). As there might be several rings in a graph, each ring is identified by its own distinct ring ID - RID.

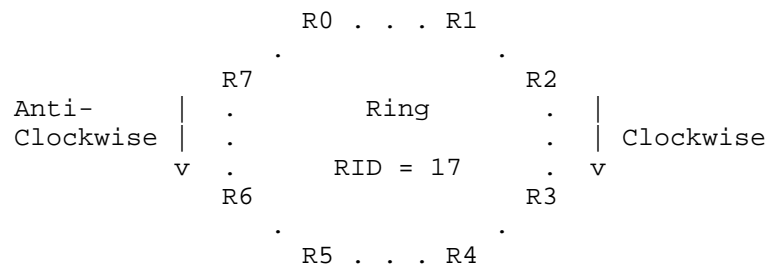


Figure 1: Ring with 8 nodes

The following terminology is used for ring LSPs:

**Ring ID (RID):** A non-zero number that identifies a ring; this is unique in a Service Provider's network. A node may belong to multiple rings.

**Ring node:** A member of a ring. Note that a device may belong to several rings.

**Node index:** A logical numbering of nodes in a ring, from zero up to one less than the ring size. Used purely for exposition in this document.

**Ring neighbors:** Nodes whose indices differ by one (modulo ring size).

**Ring links:** Links that connect ring neighbors.

Express links: Links that connect non-neighboring ring nodes.

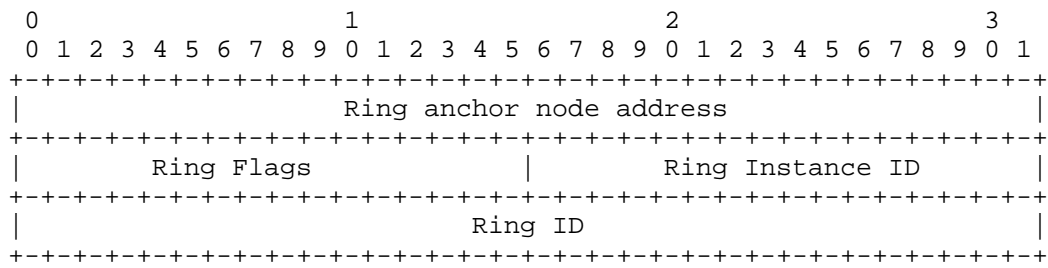
MP2P LSP: Each LSP in the ring is a multipoint to point LSP such that LSP can have multiple ingress nodes and one egress node.

### 3. RSVP Extensions

Due to the new ring LSP semantics, the signaling-message identification of ring LSPs will be different than the regular RSVP LSPs. So, a new C-Type is defined here for the SESSION object. This new C-Type will help to clearly differentiate ring LSPs from regular LSPs. In addition, new flags are introduced in the SESSION object to represent the ring direction of the corresponding Path message.

#### 3.1. Session Object

Class = SESSION, LSP\_TUNNEL\_IPv4 C-Type = TBD



SESSION Object

Ring anchor node address: IPv4 address of the anchor node. Each anchor node creates a LSP addressed to itself.

Ring Instance ID: A 16-bit identifier used in the SESSION. This Ring Instance ID is useful for graceful ring changes. If a new node is being added to the ring(resulting in signaling of a larger ring) or some existing node goes down(resulting in signaling of a smaller ring), in those cases, anchor node creates a new tunnel with a different Ring Instance ID.

Ring ID: A 32-bit number that identifies a ring; this is unique in some scope of a Service Provider's network. This number remains constant throughout the existence of ring.

Ring Flags: For each ring, the anchor node starts signaling of a ring LSP. Ring LSP named RLi, anchored on node Ri, consists of two counter-rotating unicast LSPs that start and end at Ri. One LSP will be in the clockwise direction and other LSP will be in

the anti-clockwise direction. A ring LSP is "multipoint": any node along the ring can use LSP RLi to send traffic to Ri; this can be in either the CW or AC directions, or both (i.e., load balanced). Two new flags are defined in the SESSION object which define the ring direction of the corresponding Path message.

ClockWise(CW) Direction 0x01: This flag indicates that the corresponding Path message is traveling in the ClockWise(CW) direction along the ring.

Anti-ClockWise(AC) Direction 0x02: This flag indicates that the corresponding Path message is traveling in the Anti-ClockWise(AC) direction along the ring.

### 3.2. SENDER\_TEMPLATE, FILTER\_SPEC Objects

There will be no changes to the SENDER\_TEMPLATE and FILTER\_SPEC objects. The format of the above 2 objects will be similar to the definitions in RFC 3209. [RFC3209] Only the semantics of these objects will slightly change. This will be explained in section Section 4.6 below.

## 4. Ring Signaling Procedures

A ring node indicates in its IGP updates the ring LSP signaling protocols that it supports. This can be LDP and/or RSVP-TE. Ideally, each node should support both. If the ring is configured with RSVP as the signaling protocol, then once a ring node R\_i knows the RID, its ring links and directions, it kicks off ring RSVP LSP signaling automatically.

### 4.1. Differences from regular RSVP-TE LSPs

Ring LSPs differ from regular RSVP-TE LSPs in several ways:

1. Ring LSPs (by construction) form a loop.
2. Ring LSPs are multipoint-to-point. Any ring node can inject traffic into a ring LSP.
3. The bandwidth of a ring LSP can change hop-by-hop.
4. Ring LSPs are protected without the use of bypass or detour LSPs. Protection is handled by the ring LSP traversing in the opposite direction.

#### 4.2. LSP signaling

After the ring auto-discovery process, each anchor node creates a LSP addressed to itself. This ring LSP contains of a pair of counter-rotating unicast LSPs. So, for a ring containing N nodes, there will be 2N total LSPs signaled.

There is no need for ERO object in the Path message. The Path message for ring LSPs has the following format:

```

<Path Message> ::= <Common Header> [ <INTEGRITY> ]
                        <SESSION> <RSVP_HOP>
                        <TIME_VALUES>
                        <LABEL_REQUEST>
                        [ <SESSION_ATTRIBUTE> ]
                        <sender descriptor list>

<sender descriptor list> ::= <sender descriptor>|
                                <sender descriptor list> <sender descri
ptor>

<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>

```

The anchor node creates 2 Path messages traveling in opposite directions. The SESSION format MUST be as per the description in Section 3.1. The anchor node which creates the LSP will insert it's own address in the "Ring anchor node address" field of the SESSION object. So effectively, the Path messages are addressed to the originating node itself.

The SESSION flags of these 2 Path messages are different. The Path message sent to the CW neighbor MUST have the CW flag set in the SESSION object to signal the LSP going in the clockwise direction. The Path message sent to the AC neighbor MUST have the AC flag set to signal the LSP in the anti-clockwise direction.

When an incoming Path message is received at the ring node Ri, it consults the results of auto-discovery to find the appropriate ring neighbor. If the incoming Path message has CW direction flag set, then Ri includes its own SENDER\_DESCRIPTOR in the path message and forwards the Path message to its CW ring neighbor(Ri+1). Similarly if the incoming Path message has AC direction flag set, then Ri includes its own SENDER\_TEMPLATE and forwards that Path message to it's AC ring neighbor(Ri-1). Thus, there is no need of ERO in the Path message. The Path message is routed locally at each ring based on the ring auto-discovery calculations.

The RESV message for ring LSPs also uses the new RING\_IPv4 SESSION object. When the Path message originated from the anchor node  $R_i$  reaches back to  $R_i$ ,  $R_i$  generates a Resv message. Note that this means that anchor node is both Ingress and Egress for the Path message. The Resv message copies the same ring flags as received in the corresponding Path message. So, a Resv message for a CW LSP goes in the AC direction (unlike the Path message, which goes CW). This is done to correctly match Path and corresponding Resv messages at transit ring nodes. Upon receiving Resv message with CW flag set, the ring node will forward the Resv message to its AC neighbor.

Each ring node  $R_i$  allocates CW and AC labels for each ring LSP  $RL_x(x$  between  $i..n$ ). As the signaling propagates around the ring, CW and AC labels are exchanged. When  $R_i$  receives CW and AC labels for LSP  $RL_x$  from its ring neighbors, primary and fast reroute (FRR) paths for  $RL_x$  are installed at  $R_i$ .

Consider the following three nodes of the ring, and their signaling interactions for LSP  $RL_5$  originating from anchor node  $R_5$ :

```

                P5_CW ->      P5_CW ->
                Q5_CW <-      Q5_CW <-
... ----- R7 ----- R8 ----- R9 ----- ...
                P5_AC <-      P5_AC <-
                Q5_AC ->      Q5_AC ->

```

P corresponds to the Path message and Q corresponds to the Resv message.

As explained above, an RMR LSP consists of two counter-rotating ring LSPs that start and end at the same node, say  $R_1$ . As such, this appears to cause a loop, something that is normally avoided by RSVP-TE. There are some benefits to this:

Having a ring LSP form a loop allows the anchor node  $R_1$  to ping itself and thus verify the end-to-end operation of the LSP. This, in conjunction with link-level OAM, offers a good indication of the operational state of the LSP. Also, having  $R_1$  to be the ingress means that  $R_1$  can initiate the Path messages for the two ring LSPs. This avoids  $R_1$  having to coordinate with its neighbors to signal the LSPs, and simplifies the case where a ring update changes  $R_1$ 's ring neighbors. The cost of this is a little more signaling and a couple more label entries in the LFIB. However, we will let experiences from implementation guide us when we evaluate this approach.

#### 4.2.1. Path Propagation for RMR

Ring LSPs are MP2P in nature. It means that every non-egress node is also an ingress and a merge-point for the LSP. Focussing on ring-LSP-0 (i.e ring-LSPs starting at R0):

```
R0---->R1---->R2---->R3---->R4---->R5---->R6--->R7--->R0(CW LSP)
R0---->R7---->R6---->R5---->R4---->R3---->R2--->R1--->R0(ACW LSP)
```

Each ring node inserts a new SENDER\_TEMPLATE object into an incoming Path message. The procedure for that is as follows:

When a ring node R3 receives a Path message initiated by anchor node R0(for anchor lsp "lsp0"), R3 SHOULD make a copy of the received Path message for "lsp0". R3 then inserts a new sender-template object into the Path message for "lsp0". In the sender-template object, R3 uses the sender address as the loopback address of node R3 and lsp-id = X. R3 then forwards this modified Path message to it's ring neighbor.

So at this point, when Path messages heads out at R3, there will be 4 different SENDER\_TEMPLATE objects in the outgoing Path message for lsp0:

```
-----
|SENDER_TEMPLATE_0 : SENDER_ADDRESS = R0, LSP_ID = 1 |
-----
|SENDER_TEMPLATE_1 : SENDER_ADDRESS = R1, LSP_ID = 1 |
-----
|SENDER_TEMPLATE_2 : SENDER_ADDRESS = R2, LSP_ID = 1 |
-----
|SENDER_TEMPLATE_3 : SENDER_ADDRESS = R3, LSP_ID = 1 |
-----
```

#### 4.2.2. Resv Processing for RMR

When Egress node R0 receives the modified Path message, it replies with the a Resv message containing multiple FLOW\_DESCRIPTOR objects. There should be 1 FLOW\_DESCRIPTOR object corresponding to each of the SENDER\_TEMPLATE object in the incoming Path message. The SESSION object of the Resv message will exactly match with the received Path message.

[RFC 3209] already supports receiving a Resv message with multiple flow-descriptors in it, as described in section 3.2 in that document. In each flow-descriptor there is a separate:



- a. FLOW\_SPEC object corresponding to the SENDER\_TSPEC that was sent in the Path message which could be admitted after admission-control downstream, and
- b. FILTER\_SPEC object corresponding to SENDER\_TEMPLATE that was sent in the Path message that could be admitted after admission-control downstream.

Each transit node removes the FLOW-DESCRIPTOR corresponding to itself from the Resv message before sending the Resv message upstream.

#### 4.3. Protection

In the rings, there are no protection LSPs -- no node or link bypass LSPs, no standby LSPs and no detours. Protection is via the "other" direction around the ring, which is why ring LSPs are in counter-rotating pairs. Protection works in the same way for link, node and ring LSP failures.

Since each ring LSP is a MP2P LSP, any ring node can inject traffic onto a LSP whose anchor might be a different ring node. To achieve the above, an ingress route will be installed as follows at every ring node J, for a given ring-LSP with anchor Rk (say 1.2.3.4).

```
1.2.3.4  -> (Push CL_J+1,K, NH: R_J+1)      # CW
          -> (Push AL_J-1,K, NH: R_J-1)      # AC

CL = Clockwise label
AL = Anti-Clockwise label
```

Traffic will either be load balanced in the CW and AC directions or the traffic will be sent on just CW or AC lsp based on parameters such as hop-count, policy etc.

Also, 2 transit routes will be installed for the anchor LSP transiting from node Rj as follows:

```

CL_J,K -> SWAP(CL_J+1,K, NH: R_J+1)      #CW
        -> SWAP(AL_J-1,K , NH: R_J-1)      #AC

CL = Clockwise label
AL = Anti-Clockwise label
CW NH has weight 1, AC NH has higher-weight.

AL_J,K -> SWAP(AL_J-1,K , NH: R_J-1)      #AC
        -> SWAP(CL_J+1,K, NH: R_J+1)      #CW

CL = Clockwise label
AL = Anti-Clockwise label
AC NH has weight 1, CW NH has higher weight.

```

Suppose a packet headed in anti-clockwise direction towards R5 and it arrives at node R7. Lets say that now R7 learns there is a link failure in the AC direction. R7 reroutes this packet back onto the clockwise direction. This reroute action is pre-programmed in the LFIB, to minimize the time between detection of a fault and the corresponding recovery action.

At this time, R7 also sends a notification to R0 that the AC direction is not working. R0 modifies it's ingress route(for R5 LSP) by removing the AC direction LSP's route. Thus, R0 switches traffic to the CW direction.

These notification propagate CW until each traffic source on the ring CW of the failure uses the CW direction. For RSVP-TE, this notification is sent in the form of PathErr message.

To provide this notification, the ring node detecting failure SHOULD send a Path Error message with error code of "Notify" and an error value field of ("Tunnel locally repaired"). This Path Error code and value is same as defined in RFC 4090[RFC4090] for the notification of local repair.

Note that the failure of a node or a link will not necessarily affect all ring LSPs. Thus, it is important to identify the affected LSPs and only switch the affected LSPs.

#### 4.4. Ring changes

A ring node can go down resulting in a smaller ring or a new node can be added to the ring which will increase the ring size. In both of

the above cases, the ring auto-discovery process SHOULD kick in and it SHOULD calculate a new ring with the changed ring nodes.

When the ring auto-discovery process is complete, IGP will signal RSVP to begin the MBB process for the existing ring LSPs. For this MBB process, the anchor node will create a new Path message with a different Ring Instance ID in the SESSION object. All other fields in the SESSION Object will remain same as the existing Path message(before the ring change).

This new Path message will then propagate along the ring neighbors in the same way as the original Path message. Each ring neighbor SHOULD forward the Path message to it's appropriate neighbor based on the new auto-discovery calculations.

For the ring links which are common between the old and new LSPs, the LSPs will share resources(SE style reservation) on those ring links. Note that here we are using Ring Instance ID in the SESSION object to share resources instead of the LSP\_ID in the SENDER\_TEMPLATE Object(which is used in RSVP-TE for sharing resources as described in RFC 3209 [RFC4090]). The LSP\_ID use is reserved for a different functionality as described in section Section 4.6.

#### 4.5. Express Links

The details for signaling over express links will be given in a future version.

#### 4.6. Bandwidth management

For RSVP-TE LSPs, bandwidths may be signaled in both directions. However, these are not provisioned either; rather, one does "reverse call admission control". When a service needs to use an LSP, the ring node where the traffic enters the ring attempts to increase the bandwidth on the LSP to the egress. If successful, the service is admitted to the ring.

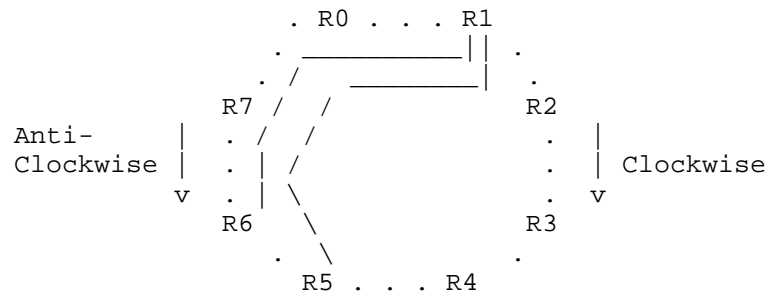


Figure 2: BW Management in Ring with 8 nodes

Let's say that Ring node R5 wants to increase the BW for the LSP whose egress is at node R1. To achieve this BW increase, Ring node R5 has to increase BW along the LSP anchored at node R1 (say lsp1).

R5 makes a copy of the existing ring Path message for lsp1. R5 then modifies the sender-template object from the copied Path message for "lsp1". In the sender-template object, R5 uses the sender address as the loopback address of node R5 and lsp-id = X+1. R5 also modifies the TSPEC object which represents the BW increase/decrease in this new Path message. R5 then forwards this new Path message to its ring neighbor. The original anchor Path message has sender address as loopback address of R1.

Now, let's say, node 5 wants to increase BW again for lsp1, then R5 adds a new SENDER\_TEMPLATE object in the existing Path message for "lsp1" with sender address as loopback of node 5 and lsp-id = X+2. So at this point, there will be 2 different SENDER\_TEMPLATE objects corresponding to node 5 in the outgoing path message.

```

-----
|SENDER_TEMPLATE_0 : SENDER_ADDRESS = R0, LSP_ID = 1 |
-----
|SENDER_TEMPLATE_1 : SENDER_ADDRESS = R1, LSP_ID = 1 |
-----
| ..... |
-----
|SENDER_TEMPLATE_5 : SENDER_ADDRESS = R5, LSP_ID = 1 |
-----
|SENDER_TEMPLATE_5 : SENDER_ADDRESS = R5, LSP_ID = 2 |
-----

```

Similarly, if node R6 wants to increase the BW for "lsp1", it SHOULD create a new Path message containing SENDER\_TEMPLATE object with sender address = loopback of node 6 and lsp-id = Y+1. Thus, it

should be noted that each ring-node independently tracks its own lsp-ID that is currently in-use on a given RMR sub-LSP. This lsp-ID value will (could) be different for each ring-node for a given ring sub-LSP.

If sufficient BW is available all the way towards ring node R1, then this new Path message reaches node R1. R1 generates a Resv message with the correct FILTER\_SPEC object corresponding to the received SENDER\_TEMPLATE object. This Resv message will also have the correct FLOWSPEC object as per the requested bandwidth.

If sufficient BW is not available at some downstream (say node R9), then ring node R9 SHOULD generate a PathErr message with the corresponding Sender Template Object. When node R5 receives this PathErr message, R5 understands that the BW increase was not successful. Note that the existing established bandwidths for lsp1 are not affected by this new PathErr message.

When ring node R5 no longer needs the BW reservation, then ring node R5 SHOULD originate a new Path message with the appropriate Sender Template Object containing 0 BW as described above. Every downstream node SHOULD then remove bandwidth allocated on the corresponding link on receipt of this Path message.

Also, note that as part of this BW increase or decrease process, any ring node does not actually change any label associated with the LSP. So, the label remains same as it was signaled initially when the anchor LSP came up.

## 5. Security Considerations

It is not anticipated that either the notion of MPLS rings or the extensions to various protocols to support them will cause new security loopholes. As this document is updated, this section will also be updated.

## 6. Contributors

Ravi Singh  
Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089  
USA

Email: ravis@juniper.net

Santosh Esale  
Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089  
USA

Email: [sesale@juniper.net](mailto:sesale@juniper.net)

Raveendra Torvi  
Juniper Networks, Inc.  
10 Technology Park Dr  
Westford, MA 01886  
USA

Email: [rtorvi@juniper.net](mailto:rtorvi@juniper.net)

## 7. IANA Considerations

Requests to IANA will be made in a future version of this document.

## 8. References

### 8.1. Normative References

- [I-D.ietf-mpls-rmr]  
Kompella, K. and L. Contreras, "Resilient MPLS Rings",  
draft-ietf-mpls-rmr-07 (work in progress), March 2018.
- [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>>.
- [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>>.

### 8.2. Informative References

- [I-D.dai-mpls-rsvp-te-mbb-label-reuse]  
Dai, M. and M. Chaudhry, "MPLS RSVP-TE MBB Label Reuse",  
draft-dai-mpls-rsvp-te-mbb-label-reuse-01 (work in  
progress), September 2015.
- [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>>.

- [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>>.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.
- [RFC4090] Pan, P., Ed., Swallow, G., Ed., and A. Atlas, Ed., "Fast Reroute Extensions to RSVP-TE for LSP Tunnels", RFC 4090, DOI 10.17487/RFC4090, May 2005, <<https://www.rfc-editor.org/info/rfc4090>>.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", RFC 5305, DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.

## Authors' Addresses

Abhishek Deshmukh  
Juniper Networks, Inc.  
10 Technology Park Dr  
Westford, MA 01886  
USA

Email: [adeshmukh@juniper.net](mailto:adeshmukh@juniper.net)

Kireeti Kompella  
Juniper Networks, Inc.  
1133 Innovation Way  
Sunnyvale, CA 94089  
USA

Email: [kireeti@juniper.net](mailto:kireeti@juniper.net)