RSVP-TE: Extensions to RSVP for Multicast LSP Tunnels

draft-cheng-mpls-rsvp-multicast-er-00.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."

To view the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in an Internet-Drafts Shadow Directory, see http://www.ietf.org/shadow.html.

Abstract

Currently the RSVP-TE ([7]) defines explicit route object and related procedure for traffic engineering based MPLS tunnels but for unicast situations only. This document extends the RSVP-TE's explicit route capability to multicast applications. The mechanism proposed in this document is also intended for multicast applications in GMPLS networks.

Table of Contents

1.0 Introduction .............................................. 3
2.0 Overview .............................................. 3
3. Tree Explicit Route Object (TERO) .......................... 4
3.1 Applicability ............................................ 5
3.2 Semantics of the Tree Explicit Route Object .............. 5
3.3 Tree Explicit Route Subobjects ........................... 5
3.3.1 Subobjects with Numbered Link ......................... 6
3.3.1.1 Subobject: IPv4 Prefix ............................ 6
3.3.1.2 Subobject: IPv6 Prefix ............................ 7
3.3.1.3 Subobject: Autonomous System Number ............... 7
3.3.2 Subobject with Unnumbered Link ....................... 7
3.4 Processing of the Tree Explicit Route Object ............ 8
4. Tree Record Route Object (TRRO) ........................... 8
4.1 Applicability ........................................... 9
4.2 Semantics of the Tree Record Route Object ............... 9
4.3 Tree Record Route Subobjects ........................... 9
4.3.1 Subobjects with Numbered Link ............................ 9
4.3.1.1 Subobject: IPv4 address ................................ 10
4.3.1.2 Subobject: IPv6 address ................................ 10
4.3.1.3 Subobject with Label ................................... 10
4.3.2 Subobject with Unnumbered Link ........................... 11
4.4  Processing of the Tree Record Route Object ............... 11
5. Teardown ................................................................ 11
6. Security Considerations ............................................ 12
7. RSVP Message Formats ............................................. 12
8.0 References ........................................................... 13
9.0 Author’s Address .................................................... 14
1.0 Introduction

There are various IP multicast protocols that have been defined including DVMRP ([1]), PIM-SM ([2]), CBT ([3]), etc. However, these multicast routing protocols are currently used for forwarding IP datagrams in networks not with MPLS based, or at least without traffic engineering considerations. Moreover, sometimes it requires mechanisms other than multicast routing protocols to specify the routing path for multicast traffic.

The MPLS architecture ([4]) states clearly that the MPLS technology is applied to IP multicast networks but much work is required in the area including traffic engineering. The GMPLS architecture ([5]) lists the multicast in GMPLS networks is an open issue.

To support multicast in MPLS/GMPLS networks, some additions are required in the signaling protocols. The RSVP ([6]) is one of the key signaling protocols. The RSVP has actually already supported multicast applications but does not support explicit routing capability. The RSVP-TE ([7]) adds the explicit routing capability to be used in the MPLS/GMPLS networks but for unicast applications only.

This document proposes the explicit routing capability in addition to those already defined in the [7] to support multicast applications in MPLS/GMPLS networks.

To build a point-to-multipoint LSP, the information provided by IP routing protocols with traffic engineering extensions, i.e., OSPF-TE ([10, [11]) or IS-IS-TE ([12], [13]) is required similarly to the point-to-point LSP case, i.e., the path selection criteria is based on constraints on network topology, traffic engineering, policy, etc.

2. Overview

The RSVP-TE document ([7]) defines the EXPLICIT_ROUTE object that included in the Path message to specify the routing path for a MPLS TE tunnel across a network. It also defines RECORD_ROUTE object included in the Path and Resv messages to record the route for a given MPLS TE tunnel. This document defines the TREE_EXPLICIT_ROUTE object (TERO) that included in the Path message to specify a MPLS multicast TE tunnel across a network, and the TREE_RECORD_ROUTE object that included in the Path and Resv messages to record the route for a given MPLS multicast TE tunnel.

A MPLS multicast TE tunnel is realized by a point-to-multipoint LSP that can be represented by a multicast distribution tree with the ingress LER as the tree root, and all the egress LERs as the tree leafs. The distribution tree can be calculated from the network management station as a pinned tree, or by the ingress LER. A leaf node can at anytime join and leave a multicast distribution tree. The actual mechanism for the tree calculation and its
algorithm is outside the scope of this document.

A MPLS multicast TE tunnel can be initiated from the ingress LER, to build a so-called root-initiated distribution tree, or initiated from the egress LER, to build a so-called leaf-initiated distribution tree. This document only discusses root-initiated distribution tree using RSVP. Leaf-initiated distribution tree using RSVP is for further study.

The MPLS multicast TE tunnel as discussed in this document is intended for unidirectional traffic only.

A RSVP Path message containing TREE_EXPLICIT_ROUTE (TERO) object is originated from the ingress LER and forwarded to all the leaf nodes along the point-to-multipoint distribution path represented by the set of subobjects contained in the TERO object. A leaf node is a LSR in the network that claims the reachability to the IP multicast address that contained in the Session object. All TE links that on the tree satisfies with the traffic engineering parameters as contained in the Path message. Additional tree leaves and their associated branches, if any, may be added to the TERO when new leaf node(s) claims the reachability to the same IP multicast address in the subsequent refresh Path messages. Leaf nodes that no longer reachable towards the destination IP multicast address will be initially included in the TERO object with a flag indicating as a dropped party, and removed completely in subsequent Path messages.

The handing for the RSVP Resv message for multicast TE tunnel is the same as that as for the point-to-point TE tunnel except at a branch node, the Resv messages received from all the down stream LSRs are "merged" and there is only one Resv message sent to its own upstream LSR.

The TREE_RECORD_ROUTE object (TRRO) is also introduced and may be included in the Path and Resv messages with similar semantics as the RECORD_ROUTE object used in the point-to-point TE tunnel situation.

3. Tree Explicit Route Object (TERO)

Tree explicit route are specified via the TREE_EXPLICIT_ROUTE object (TERO). The TERO has the following format:

```
Class = TBA, C_Type = 1

0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                                                               |
|                                                               |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Subobjects
```

<table>
<thead>
<tr>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>9</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
| +++++++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| //                        (Subobjects) //
| +++++++-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+

Subobjects
The contents of a TERO are a series of variable-length data items called TERO subobject as defined in the Section 3.3 below.

3.1 Applicability

The TERO object is intended to be used only for multicast situations when the multicast distribution tree is specified by the root node. Also it is used when all LSRs that are included in the multicast tree support this object.

The TERO will be included in the RSVP Path message to specify a point-to-multipoint and unidirectional LSP.

3.2 Semantics of the Tree Explicit Route Object

A TERO contains a set of subobjects that represent a multicast distribution tree or tree branch. Each subobject represents a tree node that contains generally the same information as the Explicit Route Subobject as defined in the [7], plus others as detailed in the Section 3.3. The difference is the TERO contains a point-to-multipoint LSP path represented by a whole or partial distribution tree, not a hop-by-hop linear path.

```
  A
   |
  +-----------------------------+
  |   |   |   |   |   |
  B  G  H  K  M
   |   |   |   |   |
  +-----+-----+-----+
  |   |   |   |   |
  C  D  I  J  N  P  Q
   |   |   |   |
  +---+---+---+
  |   |   |   |
  E  F  L  O
```

The tree nodes appear in the TERO in depth-first order. E.g., if a distribution tree is as shown as in the above figure, the TREE_EXPLICIT_ROUTE object is encoded as follows:

```
{ A(3), B(2), C(0), D(1), E(0), F(0), G(0), H(1), I(0), J(0), K(0), M(2), N(1), L(0), O(0), P(0), Q(0) }
```

where, the number in the parenthesis after each node notation stands for the number of tree levels below that node.

3.3 Tree Explicit Route Subobjects

The contents of a TERO consist of a series of subobjects and
collectively, they specify a point-to-multipoint distribution tree or a sub-tree.

3.3.1 Subobjects with Numbered Link

The RSVP Explicit Route subobjects defined in the [7] can be extended to support the subobjects with numbered links with the details as follows:

The Explicit Route subobjects defined in the [7] have a form as:

```
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|  Type     |     Length    |    (Subobject contents)       |
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

where, the Type indicates the type of contents of the subobject with values currently only for unicast traffic as follows:

1 IPv4 prefix
2 IPv6 prefix
32 Autonomous system number

This document adds the following values for the Type:

17 IPv4 prefix for a hop on a multicast LSP
18 IPv4 prefix for a hop on a multicast LSP
64 Autonomous system number for a hop on a multicast LSP

3.3.1.1 Subobject: IPv4 Prefix

```
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|  Type (17)  |     Length    |    IPv4 address (4 bytes)     |
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   IPv4 address (continued)    | Prefix Length |   Reserved    |
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|D|         Reserved            |       Sub-Tree Depth          |
+-----+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

This new subobject is extended from the ER subobject for numbered links using IPv4 prefix, with additional fields as follows:

D - an one-bit flag to indicate this hop is dropped from the tree.

Sub-Tree Depth - a 2-byte field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.
3.3.1.2 Subobject: IPv6 Prefix

This new subobject is extended from the ER subobject for numbered links using IPv6 prefix, with an additional 2-byte reserved field and the Sub-Tree Depth field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.

3.3.1.3 Subobject: Autonomous System Number

This new subobject is extended from the ER subobject for numbered links using Autonomous System, with an additional 2-byte reserved field and the Sub-Tree Depth field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.

3.3.2 Subobject with Unnumbered Link

The RSVP Explicit Route subobject defined in the [8] for unnumbered links can be extended to support the Tree Explicit Route Subobject.

The Type defined in the [8] is 4 for ERO on unnumbered links and this document adds a new subobject with Type as 20. There is also an additional 2-byte reserved field and the Sub-Tree Depth field that indicates the number of subobjects under this hop.
Refer to the [8] for meanings of the rest of the data fields.

3.4 Processing of the Tree Explicit Route Object

The processing rules for the TERO are similar to that for the EXPLICIT_ROUTE object as documented in the [7] with the following additions:

1) The node receiving a Path message containing an TERO must be able to locate the TERO subobject corresponding to itself. Since the order of the subobject is with depth-first, the associated subobject may or may not be the first one on the list.

2) Once the receiving node of a Path message locates itself on the list of the TERO, it needs to delete all sibling branches that do not belong to the subtree where the receiving node is the root, before other handlings including adding more subobjects on the tree, locate the next hops, etc.

3) When a receiving node needs to add subobjects to the TERO, it must take a position for itself as a root of a tree branch, by including all the required intermediate nodes to get to all the leaf nodes that it sees.

4) If one of the next subobject included in the TERO has the "D" bit set, the same handling as for the Path Tear message in the unicast LSP is taken in regard to that next hop.

5) If the subobject that corresponding to the receiving node of the Path message has the "D" bit set, it is equivalent to receiving a Path Tear message.

4. Tree Record Route Object (TRRO)

Multicast LSP can be recorded via the TREE_RECORD_ROUTE object (TRRO). Optionally, labels may also be recorded. The TRRO has the following format:

Class = TBA, C_Type = 1
Subobjects

The contents of a TRRO are a series of variable-length data items called TRRO subobjects as defined in the Section 4.3.

The TRRO can be present in both RSVP Path and Resv messages. If a Path or Resv message contains multiple RROs, only the first TRRO is meaningful. Subsequent TRROs need to be ignored and not be propagated.

4.1 Applicability

The applicability of the TRRO is similar to that of the RRO as described in the [7], i.e., the TRRO can be used for loop detection, multicast LSP trace and feedback as the TERO.

4.2 Semantics of the Tree Record Route Object

The TRRO consists of a series of Tree Record Route Subobjects. Each subobject represents a node on a multicast TE path, and collectively, a TRRO represents the actual path for a point-to-multipoint LSP throughout a MPLS/GMPLS network.

4.3 Tree Record Route Subobjects

The contents of a TRRO consist of a series of Tree Record Route Subobjects that collectively specifies a point-to-multipoint LSP.

4.3.1 Subobjects with Numbered Link

The RSVP Route Record subobjects defined in the [7] can be extended to support the TRRO subobjects with numbered link. The subobjects defined in [7] to support RRO for unicast tunnels include the following:

1 IPv4 prefix
2 IPv6 prefix
3 Label

This document adds the following values for the Type:

17 IPv4 prefix for a hop on a multicast LSP
18 IPv4 prefix for a hop on a multicast LSP
19 Label for a hop on a multicast LSP
### 4.3.1.1 Subobject: IPv4 address

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This new subobject is extended from the RRO subobject for numbered links using IPv4 prefix, with an additional field as follows:

Sub-Tree Depth - a 2-byte field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.

### 4.3.1.2 Subobject: IPv6 address

<table>
<thead>
<tr>
<th></th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

This new subobject is extended from the RRO subobject for numbered links using IPv6 prefix, with an additional field as follows:

Sub-Tree Depth - a 2-byte field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.

### 4.3.1.3 Subobject with Label
This new subobject is extended from the RRO subobject for labels with an additional field as follows:

Sub-Tree Depth - a 2-byte field that indicates the number of subobjects under this hop.

Refer to the [7] for meanings of the rest of the data fields.

4.3.2 Subobject with Unnumbered Links

The RSVP Record Route subobject defined in the [8] for unnumbered link can be extended to support the Tree Route Record subobject.

The Type defined in the [8] is 4 for RRO on unnumbered links and this document adds a new subobject with Type as 20. There is also an additional 2-byte reserved field and the Sub-Tree Depth field that indicates the number of subobjects under this hop.

Refer to the [8] for meanings of the rest of the data fields.

4.4 Processing of the Tree Record Route Object

The processing rules for the TRRO are similar to that for the RRO as described in the [7] except the path information that recorded at each node may be limited. At a node, the Path TRRO will have the route from the ingress LER to this node; the Resv TRRO will have the route from this node to all the leaf nodes belong to the sub-tree rooted at this node. Only the ingress LER has the complete multicast distribution tree from itself to all leaf nodes.

5. Teardown
A point-to-multipoint LSP setup using TERO may be torn down or partially torn down through three methods depending on the circumstances.

First, a Path Tear message can be sent by the ingress LER or any node that is on the multicast distribution tree. The message will traverse down the tree or a sub-tree, with replication at each branch node, resulting the Path State being removed immediately at each node.

Second, a Resv Tear message can be sent by any node on the multicast distribution tree and traverse upstream, but with possible merge at each branch node. The result is to prune the reservation state toward the ingress LER as far as possible as described in the [6]. In particular, when a single leaf node decides to leave a given multicast group, it can sends a Resv Tear message towards its upstream node.

Third, the ingress LER can flag the "D" bit in one or more subobjects of the TERO included in the Path message during the refresh procedure. The effect of the "D" bit on the associated node upon arriving is the same as if receiving a Path Tear message, and the upstream node also needs to prune this node from its subtree.

6. Security Considerations

Security issues are not discussed in this document.

7. RSVP Message Formats

This section presents the RSVP message related formats as modified by this document. Unmodified formats are not listed. Note due to the intention that the mechanism proposed in this document also used in GMPLS networks, the changes are made on the top of the RSVP messages as defined in the [9].

The format of a Path message is as follows:

```plaintext
<Path Message> ::= <Common Header> [ <INTEGRITY> ]
                    [ [MESSAGE_ID_ACK] | MESSAGE_ID_NACK ] ... ]
                    [ MESSAGE_ID ]
                    <SESSION> <RSVP_HOP>
                    <TIME_VALUES>
                    [ [EXPLICIT_ROUTE] | TREE_EXPLICIT_ROUTE ]
                    <LABEL_REQUEST>
                    [ PROTECTION ]
                    [ LABEL_SET ... ]
                    [ SESSION_ATTRIBUTE ]
                    [ NOTIFY_REQUEST ]
                    [ ADMIN_STATUS ]
                    [ POLICY_DATA ... ]
                    <sender descriptor>
```

[Page 12]
The format of the sender description for unidirectional LSPs is:

\[
\text{<sender descriptor>} ::= \text{<SENDER\_TEMPLATE>} \text{<SENDER\_TSPEC>}
\]
\[
\quad [ \text{<ADSPEC>} ]
\quad [ \text{<RECORD\_ROUTE>} | \text{<TREE\_RECORD\_ROUTE>} ]
\quad [ \text{<SUGGESTED\_LABEL>} ]
\quad [ \text{<RECOVERY\_LABEL>} ]
\]

The format of a Resv message is as follows:

\[
\text{<Resv Message>} ::= \text{<Common Header>} [ \text{<INTEGRITY>} ]
\]
\[
\quad [ [\text{<MESSAGE\_ID\_ACK>} | \text{<MESSAGE\_ID\_NACK}>] ... ]
\quad [ \text{<MESSAGE\_ID>} ]
\quad \text{<SESSION>} \text{<RSVP\_HOP>}
\quad \text{<TIME\_VALUES>}
\quad [ \text{<RESV\_CONFIRM>} ] [ \text{<SCOPE>} ]
\quad [ \text{<NOTIFY\_REQUEST>} ]
\quad [ \text{<ADMIN\_STATUS>} ]
\quad [ \text{<POLICY\_DATA>} ... ]
\quad \text{<STYLE>} \text{<flow descriptor list>}
\]

\[
\text{<flow descriptor list>} ::= \text{<FF flow descriptor list>}
\quad | \text{<SE flow descriptor>}
\]

\[
\text{<FF flow descriptor list>} ::= \text{<FLOWSPEC>} \text{<FILTER\_SPEC>}
\]
\[
\quad \text{<LABEL>}
\quad [ \text{<RECORD\_ROUTE>} | \text{<TREE\_RECORD\_ROUTE>} ]
\quad | \text{<FF flow descriptor list>}
\quad \text{<FF flow descriptor>}
\]

\[
\text{<FF flow descriptor>} ::= [ \text{<FLOWSPEC>} ] \text{<FILTER\_SPEC>} \text{<LABEL>}
\]
\[
\quad [ \text{<RECORD\_ROUTE>} | \text{<TREE\_RECORD\_ROUTE>} ]
\]

\[
\text{<SE flow descriptor>} ::= \text{<FLOWSPEC>} \text{<SE filter spec list>}
\]

\[
\text{<SE filter spec list>} ::= \text{<SE filter spec>}
\quad | \text{<SE filter spec list> <SE filter spec>}
\]

\[
\text{<SE filter spec>} ::= \text{<FILTER\_SPEC>} \text{<LABEL>}
\]
\[
\quad [ \text{<RECORD\_ROUTE>} | \text{<TREE\_RECORD\_ROUTE>} ]
\]

8.0 References


9.0 Authors’ Addresses

Dean Cheng
Polaris Networks Inc.
6810 Santa Teresa Blvd.
San Jose, CA 95119
Phone:  +1 408 284 8061
Email:  dcheng@polarisnetworks.com