MPLS Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 02, 2014

R. Li
Q. Zhao
Huawei Technologies
C. Jacquenet
France Telecom Orange
E. Metz
KPN
B. Zhang
Telus Communications

July 01, 2013

Receiver-Driven Multicast Traffic-Engineered Label-Switched Paths
draft-lzj-mpls-receiver-driven-multicast-rsvp-te-03.txt

Abstract

This document describes extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for the setup of Receiver-Driven Traffic-Engineered point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP)Label Switched Paths (LSPs) in Multi-Protocol Label Switching (MPLS) and Generalized MPLS (GMPLS)networks.

Table of Contents

1. Introduction

Multiparty multimedia applications are getting great attentions in the telecom and datacom world. Such applications are QoS-demanding and can therefore benefit from the MPLS traffic engineering capabilities based on dynamic computation and establishment of MPLS LSPs to meet with application-specific QoS requirements. P2MP-TE [RFC4875] defines a procedure to set up point-to-multipoint LSPs from sender to receivers. This procedure works very well if the senders have a priori knowledge of all its receivers. Sometimes multicast data streams are required to get transported over both IP networks and MPLS networks, but MPLS networks have no priori knowledge about senders and receivers. In the IP networks, the receivers can join/leave a multicast distribution tree by PIM Join/Prune messages, and thus the multicast distribution tree is essentially receiver-driven. When such PIM Join/Prune messages arrive at an MPLS network border, we need a procedure to initiate and set up the multicast distribution tree in MPLS. This document extends RSVP-TE for initiation and setup of P2MP and MP2MP LSPs driven by receivers.

1.1. Motivation

IP multicast distribution trees are initiated by receivers and dynamic by nature. IP multicast applications are also sensative to bandwidth, especially in the area of residential IPTV services, where the delivery of multicast contents to several hundreds of thousands of IPTV receivers assumes the appropriate level of quality.

Current source-driven P2MP LSP establishment, as defined as in [RFC4875], assumes a priori knowledge of receiver locations, and the LSP signalling is initiated and driven by the data sender (headend). The priori knowledge of receiver locations is obtained either through static configuration or by using another protocol to discover such receivers. On the other hand, [RFC4875] does not address the MP2MP LSPs. Actually, there is no straightfoward way to support MP2MP applications by using P2MP LSP unless full-meshed P2MP LSPs are set up independently and separately.

The receiver-driven extension to RSVP-TE described in this document will support both P2MP LSPs and MP2MP LSPs. Moreover, it does not require the sender to know all the receivers' locations a priori. The protocols for discovery of receivers are not needed. It provides a natural mechanism to interwork with PIM dynamically.

1.2. Terminology

The following terms are used in this document:

- Sender: Sender refers to the Originator (and hence the Sender) of the content/payload, as defined in [RFC2205].

- Receiver: Receiver refers to the Receiver of the content/payload, as defined in [RFC2205].

- Upstream: The direction of flow from content Receiver toward content Sender, as defined in [RFC2205].

- Downstream: The direction of flow from content Sender toward content Receiver, as defined in [RFC2205].

- Path-Sender: The sender of RSVP PATH messages, with no correlation to the direction of content/payload flows. Its flow direction is irrelevant to that of Sender defined above. All other control messages discussed in this document will use this as the reference.

- Path-Receiver: The receiver of RSVP PATH messages, with no correlation to the direction of content/payload flows.

- Path-Initiator: The Path-Sender that originated a RSVP PATH message. This is different from Path-Sender in that an intermediate node can be a Path-Sender, but such an intermediate node cannot create and initiate the RSVP PATH message. A Path-Initator is a Path-Sender, but a Path-Sender doesn't have to be a Path-Initiator.

- o Path-Terminator: The Path-Receiver that does NOT propagate the Path message any further. This is different from Path-Receiver in that an intermediate node can be a Path-Receiver, but such an intermediate node will propagate the Path message to the next hop.

- o Root: A router where a multcast LSP tree is rooted at. Data enters the root and then is distributed to leaves along the P2MP/ MP2MP LSP.

1.3. Overview

Although the receiver-driven extensions to RSVP-TE as defined in this document use the existing sender-driven syntax, there are important semantic differences that need to be defined for correct interpretation and interoperability. In the receiver-driven context, we inverted the semantics of RSVP-TE messages, while keeping the syntax unchanged as much as possible. We will use mRSVP-TE to represent the RSVP-TE with receiver-driven extensions described in this document.

The following are some key differences that are specific to the receiver-driven paradigm:

- o The leaf router: the router that receives data/content/payload. In this document, the leaf router will initiate PATH messages. In some sense, the leaf router and the receiver mean the same thing. The term "receiver-driven" also means "leaf-driven".

- o L2S Destinations: routers where user data payload traffic enters the LSP. L2S means Leaf-to-Source. The source is the sender or root of a multicast stream.

- o RSVP P2MP PATH messages traverse from receivers to the root.

- o RSVP P2MP RESV messages traverse from the root to the leaf routers of the P2MP tree strcuture.

- o For P2MP LSP, a RSVP RESV message received by a router is interpreted as a successful resource reservation made by the upstream node.

- o For MP2MP LSP, a RSVP RESV message received by a router is interpreted as successful resource reservation made by the downstream node.

- o After a PATH message is received on an interface for P2MP LSP, label allocation on that interfaces is done prior to sending the corresponding RSVP PATH message upstream.

- After a PATH message is received on an interface for MP2MP LSP, label allocation on that interfaces is done prior to sending the corresponding RSVP RESV messages downstream.

- For P2MP LSP tree structures, a node receiving a RSVP PATH message first decides if this RSVP PATH message will make the said node a branch LSR or not. If it is not a branch LSR, it is a transit LSR. In the case that it will become a transit LSR because of this PATH message, it will, before sending the RSVP PATH message upstream, allocate required bandwidth on the interface on which the RSVP PATH message is received. The upstream node can send traffic soon after successfully reserving resources on the downstream link, on which the RSVP PATH message SHOULD be received. In the case that the node is already a branch or a transit node before it receives the PATH message, then it will allocate required bandwidth on the interface on which the RSVP PATH message is received, and send the RESV message to the node which sends the PATH message without propagating the PATH message further to the upstream node. For P2MP LSPs, a label is carried by the PATH message and should be used by the upstream node when distributing the data from upstream to downstream.

- For MP2MP LSP tree structures, a node will allocate required bandwidth on the interface through which the RSVP PATH message is sent before sending the RSVP PATH message upstream. A node receiving a RSVP PATH message MUST first decide if this RSVP PATH message will make the said node a branch LSR or not. In the case it will become a transit LSR because of this PATH message, then it will allocate required bandwidth on the interface on which the RSVP PATH message is received and will allocate required bandwidth on the interface through which the RSVP PATH message is sent, before sending the RSVP PATH message upstream. The downstream node can send traffic soon after successfully reserving bandwidth on the upstream link through which the RSVP PATH message SHOULD be sent. The upstream node can send traffic soon after successfully reserving bandwidth on the downstream link on which the RSVP PATH message SHOULD be received. In the case that the node is already a branch or a transit node before it receives the PATH message, then it will allocate required resources on the interface on which the RSVP PATH message is received, and send the RESV message to the node which sends the PATH message without propagating the PATH message further to the upstream node. The label carried by the PATH message should be used by the Path-Receiver node to forward data from the Path-Receiver node to the Path-Sender node, and the label carried by RESV messages should be used by its corresponding Path-Sender node to send data from the Path-Sender node to the Path-Receiver node.

- For the sake of readability, from now on all mRSVP-TE LSPs will be used to represent all P2MP and/or MP2MP LSPs in receiver-driven (RD) multicast P2MP/MP2MP MPLS environments. We will sometimes use RD P2MP TE LSP or RD MP2MP TE LSP to represent such receiver-driven multicast LSPs.

2. Receiver-Driven mRSVP-TE LSP Examples

In what follows we describe two examples to show how P2MP and MP2MP are set up, respectively. In both of such examples, Path messages are initiated by data receivers.

For the P2MP example, a Path message carries a label for the use of sending data downstream. And for the MP2MP example, both Path message and Resv message carries a label for sending data downstream and upstream.

2.1. P2MP Example

```
                       Sender/Source/Path Terminator/Ingress Router
                    +---------+
                    |    R1   |
                    +-----+---+
                             _
                       \  \ /\
                        \  \  \ Path Message w/ Label OBJECT
                 Resv    \  \  \   (msg2)
                 Message  \  \  \
                  (msg3)   \  \  \
                           _\/ \  \
                        +----------------+ Path Remerge
                        |        R3      | Creates Branch Point
                        +----------------+
                              _          _
                        /  /  /\   \  \ /\
                       /  /  /      \  \  \ Path Message (msg1)
        Resv Message  /  /  /   msg4 \  \  \ w/ Label OBJECT
             (msg6)  /  /  /          \  \  \
                    /  /  /Path Msg    \ \  \
                   /  /  / (msg5)       \  \  \
                 \/_ /  / w/Label OBJ   _\/ \  \
               +----------+            +---+-----+
               |    R4    |            |   R5    |
               +----------+            +---------+
               Path Initiator          Path Initiator
               Originator ID = R4      Originator ID = R5
```

```
               L2S Destination = R1    L2S Destination = R1
               Session = S             Session = S
```

Figure 1: P2MP Example

In Figure 1, when R5 is added as the first leaf of a mulitcast distribution tree (multicast LSP), the message flow goes as follows: R5->msg1->R3->msg2->R1->msg3->R3->msg4->R5. When the leaf R4 is added, the message flow goes from R4->msg5->R3->msg6->R4. In this case, when R3 receives msg5, R3 finds out that a multicast LSP has already been set up for the same session and the same source. Therefore, R3 finds itself a branch node for leaf R4 and R5, so it will terminate the PATH message and build the corresponding RESV message and send it back to R4. The association of the LSP initiated by R4 to the existing multicast LSP is determined based on the processing of the SESSION object and L2S_SUB_LSP object from the mRSVP-TE message. The SESSION object and the L2S_SUB_LSP objects are documented later in this draft.

2.2. MP2MP Example

```
                    Root/Path Terminator/Ingress Router
                    +---------+
                    |    R1   |
                    +-----+---+
                             _
                       \  \ /\
                        \  \  \ Path-mp2mp Message w/ Label OBJECT
                 Resv    \  \  \   (msg2)
                 Message  \  \  \
                  (msg3)   \  \  \
           w/ Label OBJECT _\/ \  \
                        +----------------+ Path-mp2mp
                        |        R3      | (Branch Point)
                        +----------------+
                              _          _
                        /  /  /\   \  \ /\
                       /  /  /      \  \  \ Path-mp2mp Message (msg1)
        Resv Message  /  /  /   msg4 \  \  \  (msg1)
             (msg6)  /  /  /          \  \  \   w/ Label OBJECT
     w/ Label OBJECT/  /  /Path-mp2mp  \  \  \
                   /  /  / Message      \  \  \
                  /  /  / (msg5)         \  \  \
                \/_ /  /  w/ Label OBJ   _\/ \  \
               +----------+           +---+-----+
```

```
               |    R4    |            |   R5    |
               +----------+            +---------+
               Path-mp2mp Initiator    Path-mp2mp Initiator
               Originator ID = R4      Originator ID = R5
               L2S Destination = R1    L2S Destination = R1
               Session = S             Session = S
```

Figure 2: MP2MP Example

For MP2MP, the root address should be specified. It is something similar to RP in PIM, but it doesn't need the Register message. In one-to-many applications, the root should be the same as the Sender, while in many-to-many applications, the root could be any router, but should be selected in the same way as RP is selected in PIM. In Figure 2, R1 is specified as the root. When R5 is added as the first leaf (as both a sender and a receiver) of an MP2MP multicast LSP, the message flow goes from R5->msg1->R3->msg2->R1->msg3->R3->msg4->R5. When the leaf R4 (as both a sender and a receiver)is added, the message flow goes from R4->msg5->R3->msg6->R4. In this case, when R3 receives msg5, R3 finds out that an MP2MP mulitcast LSP has already been set up for the same session and the same root and R3 will become the branch LSR for the leaf R4 and R5, so it will terminate the PATH message, build a RESV message and send the RESV message back to R4. The association of the LSP initiated by R4 to the existing MP2MP LSP is determined based on the processing of the SESSION object and the S2L_SUB_LSP from the mRSVP-TE message. The SESSION objects and the L2S_SUB_LSP objects are further documented later in this draft.

3. Signaling Protocol Extensions

The RSVP-TE with receiver-driven extensions (mRSVP-TE) is similar to the RSVP-TE protocol as specified in [RFC4875], [RFC3473] and [RFC3209], but differs in that the data receivers of an LSP tunnel initiate the Path messages toward the data sender (or the root of a mulitcast LSP). Compared with [RFC4875], mRSVP-TE can also be used to set up MP2MP LSPs.

In the context of the receiver-driven RSVP-TE, the Receiver is the Path-Originator. The Path messages go from the Receivers towards the Sender. The Resv messages flow in the opposite direction as compared to the Path messages, i.e. Resv messages are generated by the Sender or a branch LSR. Path messages flow in opposite directions as cmpared with those of the multicast stream distributions, while Resv messages flow in the same directions as the multicast streams.

In the context of the receiver-driven RSVP-TE, a Path message will be terminated at the "root" of the multicast distribution tree

(multicast LSP) or at an intermediate node if the intermediate node has received another Path message from another receiver for the same multicast distribution tree. When an intermediate node receives two or more Path messages for the same multicast distribution tree, the intermediate node will merge them together. Whether two Path messages should be merged depends on the information encoded in the SESSION and L2S-SUB-LSP objects. The SESSION object encodes multicast group information and the L2S-SUB-LSP (leaf-to-source sub-lsp) object encodes the multicast source or multicast root information.

The following sections describe the receiver-driven extensions to the RSVP-TE protocol. When there is no difference in the protocol, the usage of [RFC4875] is assumed.

3.1. Mechanisms

3.1.1. Sessions

As specified in [RFC2205], a session is a data flow with a particular destination and transport-layer protocol. In the context of multicast, the data flow is essentially a multicast distribution tree rooted at the P2MP source or MP2MP root.

For the sake of reliability, two or more sources/roots may be deployed to distribute the same multicast streams. A mulitcast stream is often represented by a mulitcast group address. In this document, we will encode the mulitcast group address in the SESSION object and the mulitcast source/root address in the leaf-to-source sub-LSP object. Note that the same session can have different sources/roots, and the same sources/roots can have different sessions.

In the context of the receiver-driven mRSVP-TE, the processing of SESSION objects is different from that of SESSION objects in sender-driven RSVP-TE [RFC4875]. In order to distinguish them, we will employ different C-Types of SESSIONs. In this document we will document SESSION objects for native IPv4/IPv6 multicast applications. For new and more applications, new types of SESSION objects will be added.

Following the method used by RSVP-TE and P2MP RSVP-TE, this draft documents the use of some new SESSION C-Type as follows:

```
  Class Name = SESSION
   C-Type
     XX+0   mRSVP_TE_P2MP_LSP_TUNNEL_IPv4 C-Type
```

```
XX+1   mRSVP_TE_P2MP_LSP_TUNNEL_IPv6 C-Type
XX+2   mRSVP_TE_MP2MP_LSP_TUNNEL_IPv4 C-Type
XX+3   mRSVP_TE_MP2MP_LSP_TUNNEL_IPv6 C-Type
```

Where XX is a number to be allocated by IANA.

Figure 3: New C-Types of SESSIONs

The new SESSION C-Type MUST be used in all receiver-driven P2MP RSVP-TE messages.

3.1.2. L2S Sub-LSPs

A multicast LSP is composed of one or more leaf-to-source sub-LSPs, which are merged together at the branch nodes. There are two ways to identify each such sub-LSP:

- From the Sender's perspective, each sub-LSP is identified by the SESSION object, the SENDER_TEMPLATE object and S2L_SUB_LSP object, as specified in [RFC 4875]. The SESSION object encodes P2MP ID, Tunnel ID, and Extended Tunnel ID. The P2MP ID is unique within the scope of the sender (ingress LSR) and remains constant throughout the lifetime of the P2MP tree structure. The Extended Tunnel ID, which remains constant throughout the lifetime of the P2MP tree structure, and which should contain the sender's address to make sure the identifier is globally unique. Finally, the Tunnel ID, also remains constant throughout the lifetime of the P2MP tree structure. The SENDER_TEMPLATE object contains the ingress LSR source address. The S2L_SUB_LSP contains the destination address of the sub-LSP.

- From the Receiver's perspective, each sub-LSP is identified by a new SESSION object, a new SENDER_TEMPLATE object and a new L2S_SUB_LSP object. The SESSION object, different from the one used in typical sender-driven environments, contains information to be used as the key to associate different PATH messages originated from different leaves. The SENDER_TEMPLATE object contains the Path-Originator's address, which is actually the Data Receiver. For P2MP LSP, the L2S_SUB_LSP contains the source address of the sub-LSP, i.e. the data Sender's address. For MP2MP LSP, the L2S_SUB_LSP contains the root address of the sub-LSP. The root address could be any router. The SESSION, SENDER_TEMPLATE and L2S_SUB_LSP all together will identify the multicast stream, the multicast stream's source, and a mulitcast stream's receiver

This document takes the approach from the Receiver's perspective. The approach from the Sender's perspective is documented in [RFC 4875].

Once an LSR receives a receiver-driven Path message with the SESSION object and L2S_SUB_LSP object, the LSR should be able to use the SESSION object and L2S_SUB_LSP object to determine whether the sub-LSP signaled by this Path message should be merged with existing multicast LSPs.

3.1.3. Path Originator and Data Receiver

In the context of the receiver-driven RSVP-TE, a Path Originator is also a Data Receiver. This document will document a new type of SENDER_TEMPLATE object, which contains the Path-Originator's IP address and describes the identity of the Path Originator.

In [RFC 2205] and [RFC 4875], the "sender" is both a path originator and a data sender. In the receiver-driven context, path originators and data senders may be different. For P2MP, path originators are actually the data receivers. For MP2MP, path originators are also both the data senders and data receivers.

In this document, we will use the same Object Class SENDER_TEMPLATE with a different C-Type to represent and identify Path Originator. In the case of P2MP LSP, the SENDER_TEMPLATE describes the identify of a data receiver. In the case of MP2MP, the SENDER_TEMPLATE describes the identify of an LSR which work as both a data sender and a data receiver.

All of the SESSION object, L2S_SUB_LSP object and SENDER_TEMPLATE object together contained in a Path message will uniquely identify a leaf-to-source sub-LSP.

3.1.4. Explicit Routing

An EXPLICIT_ROUTE Object (ERO) is used to optionally specify the explicit route of an L2S sub-LSP. Each signaled ERO corresponds to a particular L2S_SUB_LSP object. Details of explicit route encoding are specified in section 4.5 of [RFC4875], but they are encoded in a reverse order in the receiver-driven context.

When a Path message signals a L2S sub-LSP, the EXPLICIT_ROUTE object encodes the path from the leaf to the root LSR. The Path message also includes the L2S_SUB_LSP object for the L2S sub-LSP being signaled. The < [<EXPLICIT_ROUTE>], <L2S_SUB_LSP>> tuple represents the L2S sub-LSP and is referred to as the sub-LSP descriptor.

The absence of the ERO should be interpreted as requiring hop-by-hop reverse-forwarding for the sub-LSP based on the root address field of the L2S_SUB_LSP object.

3.2. Path Messages

The mechanism specified in this document allows a multicast P2MP/MP2MP LSP to be signaled using one or more Path messages. Each Path message may signal one L2S sub-LSPs.

A receiver-driven P2MP MPLS-TE LSP uses the Path message to carry the LABEL object upstream from the Receiver towards the Sender. With a receiver-driven usage of the RSVP PATH messages, the LABEL_REQUEST object carried by the PATH message is no longer mandatory, it becomes optional for receiver-driven PATH messages, as specified in Figure 4:

```
       <Path Message> ::=     <Common Header> [ <INTEGRITY> ]
                            [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
                            [ <MESSAGE_ID> ]
                            <SESSION> <RSVP_HOP>
                            <TIME_VALUES>
                            [ <EXPLICIT_ROUTE> ]
                            [ <LABEL_REQUEST> ]
                            [ <PROTECTION> ]
                            [ <LABEL_SET> ... ]
                            [ <SESSION_ATTRIBUTE> ]
                            [ <NOTIFY_REQUEST> ]
                            [ <ADMIN_STATUS> ]
                            [ <POLICY_DATA> ... ]
                            <sender descriptor>
                            [<L2S_SUB_LSP>]
```

Figure 4: Path Message Extensions

The SESSION object encodes information about the being-signalled multicast stream. The SESSION object together with L2S_SUB_LSP will be used as the key to associate different sub-LSPs to the same multicast LSP.

Using [RFC4875] as the base specification, the LABEL object is added to the <sender descriptor> as specified in Figure 5:

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

```
                                     [ <ADSPEC> ]
                                     [ <RECORD_ROUTE> ]
                                     [ <SUGGESTED_LABEL> ]
                                     [ <RECOVERY_LABEL> ]
                                     <LABEL>
```

Figure 5: Sender Descriptor

The LABEL object is defined in section 4.1 of [RFC3209]

Note that the receiver-driven Path messages convey the LABEL_REQUEST as an optional object. If the Path message signals a P2MP LSP, the LABEL_REQUEST in the Path message is not used. If the Path message signals an MP2MP, the LABEL_REQUEST is needed to ask for labels from its upstream LSR.

3.3. Resv Messages

Receiver-driven P2MP RSVP-TE does not need any change to the basic RESV messages specified in section 6.1 of [RFC4875], as long as the receiver-driven SESSION objects of the new C-Types are used.

For receiver-driven P2MP LSPs, the Path message carries the LABEL object, and thus the Resv message doesn't have to carry the LABEL object anymore. But for MP2MP LSPs, both Path and Resv messages will carry LABEL objects for sending and receiving purposes, respectively. Within the context of MP2MP LSPs, one of the directions is established as per [RFC3209]. Thus, this document is changing the use of the LABEL object in the FF Flow Descriptor and SE Filter Spec from mandatory to optional, as specified in Figure 6:

```
   <FF flow descriptor> ::= [ <FLOWSPEC> ] <FILTER_SPEC> [ <LABEL> ]
                            [ <RECORD_ROUTE> ]
                            [ <L2S_SUB_LSP> ]

   <SE filter spec> ::=     <FILTER_SPEC> [ <LABEL> ] [ <RECORD_ROUTE> ]
                            [ <L2S_SUB_LSP> ]
```

Figure 6: Resv Message Extensions

3.4. PathErr Messages

The receiver-driven PathErr messages have the same syntax and utilization as the PathErr message described in [RFC4875], with the difference in the <sender descriptor> carried by the PathErr message. The receiver-driven PathErr message will use the <sender descriptor> defined in this document, the same as that carried by the Path messages which the PathErr messages correspond to.

3.5. ResvErr Message

The receiver-driven ResvErr messages have the same syntax and utilization as the ResvErr message described in [RFC4875]. But the ResvErr messages will be processed as per this document, given that the <FF flow descriptor> and the <SE filter spec> can optionally contain the LABEL object instead of mandating the use of the LABEL object. The optional use of the LABEL object is conditioned by the nature of the multicast LSP, either uni-directional (P2MP) or bi-directional (MP2MP).

3.6. PathTear Messages

The receiver-driven PathTear messages have the same syntax and utilization as the PathTear messages described in [RFC4875] except for the <sender descriptor> carried by the PathTear messages. The receiver-driven PathTear messages will use <sender descriptor> defined in this document, the same as that carried by the Path messages which the PathTear messages correspond to.

4. New and Updated Objects

4.1. SESSION Objects

An mRSVP-TE LSP SESSION object is used to represent a multicast stream whose traffic will be carried by the multicast LSP being set up by the mRSVP-TE. The object still uses the existing SESSION C-Num assigned for RSVP-TE, but new C-Types are defined for the new purposes. Different from the values in the existing point-to-point or point-to-multipoint RSVP-TE SESSION object, the new objects defined by the new C-Types will encode "multicasting" information. The new SESSION object will have enough information so that the Path-Receivers can use the SESSION objects together with L2S_SUB_LSP to determine whether or not to associate different Path messages from different leaves to the same P2MP/MP2MP LSP. The combination of the SESSION object, the SENDER_TEMPLATE object and the L2S_SUB_LSP object will uniquely identify a single L2S sub-LSP.

For native IPv4/IPv6 multicast, IPv4/IPv6 (S, G) or (*, G, RP) will be encoded in the SESSION object for P2MP or MP2MP LSPs. In what follows we specify such session objects for IPv4/IPv6 P2MP and MP2MP

applications in the context of receiver-driven RSVP-TE. Other SESSION objects in the receiver-driven context are defined in other documents.

4.1.1. P2MP LSP for IPv4 SESSION Objects

Class = SESSION, mRSVP_TE_P2MP_LSP_TUNNEL_IPv4 C-Type = TBD.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Multicast Group Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7: P2MP LSP for IPv4 SESSION Objects

4.1.2. MP2MP LSP for IPv4 SESSION Objects

Class = SESSION, mRSVP_TE_MP2MP_LSP_TUNNEL_IPv4 C-Type = TBD.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Multicast Group Address                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 8: MP2MP LSP for IPv4 SESSION Objects

The MP2MP LSP for IPv4 SESSION objects are of the same format as P2MP LSP for IPv4 SESSION objects, but their C-Types are different.

4.1.3. P2MP LSP for IPv6 SESSION Objects

This is the same as the P2MP LSP for IPv4 SESSION object with the difference that the IPv6 multicast group addresses are 16-byte long.

Class = SESSION, mRSVP_TE_P2MP_LSP_TUNNEL_IPv6 C-Type = TBD.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Multicast Group Address (16 bytes)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 9: P2MP LSP for IPv6 SESSION Objects

4.1.4. MP2MP LSP for IPv6 SESSION Objects

Class = SESSION, mRSVP_TE_MP2MP_LSP_TUNNEL_IPv6 C-Type = TBD.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|            Multicast Group Address (16 bytes)                 |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 10: MP2MP LSP for IPv6 SESSION Objects

4.2. SENDER_TEMPLATE Objects

The SENDER_TEMPLATE object contains the Path-Initiator LSR address. In this document, the Path-Initiator is the same as the Leaf Router or Data Receiver. The LSP ID can be changed to allow a sender to do a certain level of resource sharing. Thus, multiple instances of the same mutlicast LSP can be created, each with a different LSP ID. The instances can share resources with each other. The L2S sub-LSPs corresponding to a particular instance use the same LSP ID.

4.2.1. Multicast LSP IPv4 SENDER_TEMPLATE Objects

Class = SENDER_TEMPLATE, mRSVP_TE_LSP_TUNNEL_IPv4 C-Type = TBD.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   IPv4 Leaf Router Address                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|       Reserved                |            LSP ID             |
```

```
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 11: mRSVP-TE Multicast LSP SENDER_TEMPLATE Objects

IPv4 Leaf Router Address: The IPv4 address of the Data Receiver.

LSP ID: A 2-byte identifier that can be changed to allow it to share resources with itself. Its usage is the same as that described in [RFC3209].

4.2.2. Multicast LSP IPv6 SENDER_TEMPLATE Objects

Class = SENDER_TEMPLATE, mRSVP-TE_LSP_TUNNEL_IPv6 C-Type = TBD.

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                                                               |
      +                                                               +
      |                   IPv6 Leaf Router address                    |
      +                                                               +
      |                            (16 bytes)                         |
      +                                                               +
      |                                                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |       Reserved                |             LSP ID            |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 12: mRSVP-TE LSP IPv6 SENDER_TEMPLATE Objects

IPv6 Leaf Router Address: The IPv6 address of the Data Receiver.

LSP ID: A 2-byte identifier that can be changed to allow it to share resources with itself. Its usage is the same as that described in [RFC3209].

4.3. L2S_SUB_LSP Objects

An L2S_SUB_LSP object identifies a particular L2S sub-LSP belonging to a multicast LSP, as explained earlier in this document.

4.3.1. L2S_SUB_LSP IPv4 Objects

L2S_SUB_LSP Class = TBD, L2S_SUB_LSP_IPv4 C-Type = TBD.

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                   IPv4 L2S Sub-LSP Root Address               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 13: L2S_SUB_LSP IPv4 Objects

IPv4 L2S Sub-LSP Root Address: IPv4 address of the L2S sub-LSP sender.

4.3.2. L2S_SUB_LSP IPv6 Objects

L2S_SUB_LSP Class = TBD, L2S_SUB_LSP_IPv6 C-Type = TBD

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |        IPv6 L2S Sub-LSP Root Address (16 bytes)               |
    |                                                               |
    |                                                               |
    |                                                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 14: L2S_SUB_LSP IPv6 Object

4.4. FILTER_SPEC Objects

The FILTER_SPEC object is canonical to the SENDER_TEMPLATE object.

4.4.1. mRSVP-TE LSP_IPv4 FILTER_SPEC Objects

Class = FILTER_SPEC, P2MP LSP_IPv4 C-Type = TBD.

The format of the mRSVP-TE LSP_IPv4 FILTER_SPEC object is identical to the mRSVP_TE_LSP_TUNNEL_IPv4 SENDER_TEMPLATE object.

4.4.2. mRSVP-TE LSP_IPv6 FILTER_SPEC Objects

The format of the mRSVP-TE LSP_IPv6 FILTER_SPEC object is identical to the mRSVP_TE_LSP_TUNNEL_IPv6 SENDER_TEMPLATE object.

5. Applications

There are two basic applications for receiver-driven RSVP-TE: inter-work with PIM and Multicast VPN.

5.1. Interwork with PIM

Some multicast applications may involve several domains, some of which are operated with PIM while others are enabled with RSVP-TE. This requires the multicast distribution trees to be computed and set up across different domains with PIM and MPLS configured in different domains. When a PIM Join message is received at the border of the MPLS domain, information encoded from the PIM Join message can be encoded as a receiver-driven RSVP-TE Path message which will set up a multicast distribution LSP across the MPLS domain. The root of such a multicast LSP can encode a PIM Join message by using the information encoded in the RSVP-TE Path message. The result of doing so will enable to build a mulitcast distribution tree across both IP and MPLS domains. The multicast tree will consist of a set of IP multicast sub-trees built by PIM and a set of MPLS multicast LSPs built by the receiver-driven RSVP-TE.

5.2. Multicast VPN

An L3VPN service that supports multicast is known as a Multicast VPN, or MVPN for short. An MVPN needs to connect multiple customer sites where some hosts may be senders, may be receivers and may be both senders and receivers. [RFC 6513] specifies protocols and procedures for Multicast in BGP/MPLS IP VPN, and [RFC 6514] describes the BGP encodings and procedures for exchanging the information elements required by Multicast in MPLS/BGP IP VPNs as specified in RFC 6513.

Consider an MVPN with two or more senders. If P2MP RSVP-TE is used to build the multicast distribution tree for multicast in MPLS/BGP IP VPNs, we will need two or more P2MP LSPs, each such P2MP LSP for each sender, which will increase the forwarding states in core routers. The more senders, the more P2MP LSPs, and the more forwarding states. Instead, we can use the extension and the procedure described in this document to set up a single MP2MP LSP no matter how many senders there are. The use of MP2MP will greatly reduce the number of P2MP LSPs and the forwarding states for multicast in BGP/MPLS IP VPNs.

6. Fast Re-Route Considerations

The Fast Re-Route mechanisms and procedures specified in [RFC 4090] will not be applicable to the receiver-driven extension to RSVP-TE described in this document, since their Path/Resv messages are sent in different directions.

Extensions to mRSVP-TE to support Fast Re-Route are described in the document [I-D.zlj-mpls-mpls-mrsvp-te-frr].

7. Backward Compatibility

A receiver-driven P2MP LSP mechanism uses different C-Types than those in the sender-driven P2MP RSVP-TE. If LSRs do not recognize the receiver-driven C-Types, they will not support the receiver-driven extensions described in this document. LSRs that do not support receiver-driven P2MP-TE LSP, send Path Error [TBD] back to the Path Originator.

The complete discussion on the backward compatibility will be provided in the Next version of the document.

8. Acknowledgements

We would like to thank Lin Han, Katherine Zhao, Robert Tao, Lou Berger and Eric Osborne for their comments, questions, and suggestions on our earlier drafts and presentations in IETF meetings.

9. IANA Considerations

This section is TBD.

10. Security Considerations

How a receiver is authenticated is outside the scope of this document. But we will briefly summarize the requirements which are detailed in the requirements draft.

It is a requirement that any mRSVP-TE solution developed to meet some or all of the requirements expressed in this document MUST include mechanisms to enable the secure establishment and management of mRSVP-TE MPLS-TE LSPs. This includes, but is not limited to:

- o A receiver MUST be authenticated before it is allowed to establish mRSVP-TE LSP with its source, in addition to hop-by-hop security issues identified by in RFC 3209 and RFC 4206.

- o mechanisms to ensure that the ingress LSR of a P2MP LSP is identified;

- o mechanisms to ensure that communicating signaling entities can verify each other's identities;

- o mechanisms to ensure that control plane messages are protected against spoofing and tampering;

o mechanisms to ensure that unauthorized leaves or branches are not added to the mRSVP-TE LSP; and

o mechanisms to protect signaling messages from snooping.

o Note that mRSVP-TE signaling mechanisms built on P2P RSVP-TE signaling are likely to inherit all the security techniques and problems associated with RSVP-TE. These problems may be exacerbated in mRSVP-TE situations where security relationships may need to maintained between an ingress LSR and multiple egress LSRs. Such issues are similar to security issues for IP multicast.

o It is a requirement that documents offering solutions for P2MP LSPs MUST have detailed security sections.

11. References

11.1. Normative References

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

[RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", RFC 2702, September 1999.

[RFC3031] Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol Label Switching Architecture", RFC 3031, January 2001.

[RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC4461] Yasukawa, S., "Signaling Requirements for Point-to-Multipoint Traffic-Engineered MPLS Label Switched Paths (LSPs)", RFC 4461, April 2006.

[RFC4875] Aggarwal, R., Papadimitriou, D., and S. Yasukawa, "Extensions to Resource Reservation Protocol - Traffic Engineering (RSVP-TE) for Point-to-Multipoint TE Label Switched Paths (LSPs)", RFC 4875, May 2007.

[RFC4420] Farrel, A., Papadimitriou, D., Vasseur, J., and A. Ayyangar, "Encoding of Attributes for Multiprotocol Label Switching (MPLS) Label Switched Path (LSP) Establishment Using Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4420, February 2006.

[RFC4206] Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP) Hierarchy with Generalized Multi-Protocol Label Switching (GMPLS) Traffic Engineering (TE)", RFC 4206, October 2005.

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

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

[RFC3471] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Functional Description", RFC 3471, January 2003.

11.2. Informative References

[I-D.zlj-mpls-mrsvp-te-frr] Zhao, K., Li, R., and C. Jacquenet, "Fast Reroute Extensions to Receiver-Driven RSVP-TE for Multicast Tunnels", draft-zlj-mpls-mrsvp-te-frr-00 (work in progress), July 2012.

[RFC3468] Andersson, L. and G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group decision on MPLS signaling protocols", RFC 3468, February 2003.

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3564] Le Faucheur, F. and W. Lai, "Requirements for Support of Differentiated Services-aware MPLS Traffic Engineering", RFC 3564, July 2003.

[RFC5467] Berger, L., Takacs, A., Caviglia, D., Fedyk, D., and J. Meuric, "GMPLS Asymmetric Bandwidth Bidirectional Label Switched Paths (LSPs)", RFC 5467, March 2009.

[RFC6513] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", RFC 6513, February 2012.

[RFC6514] Aggarwal, R., Rosen, E., Morin, T., and Y. Rekhter, "BGP Encodings and Procedures for Multicast in MPLS/BGP IP VPNs", RFC 6514, February 2012.

Authors' Addresses

Renwei Li
Huawei Technologies
2330 Central Expressway
Santa Clara, CA 95050
USA

Email: renwei.li@huawei.com

Quintin Zhao
Huawei Technologies
Boston, MA
USA

Email: quintin.zhao@huawei.com

Christian Jacquenet
France Telecom Orange
4 rue du Clos Courtel
35512 Cesson Sevigne,
France

Email: christian.jacquenet@orange-ftgroup.com

Eduard Metz
KPN
The Netherlands

Email: eduard.metz@kpn.com

Boris Zhang
Telus Communications
200 Consilium PL Floor 15
Toronto, ON M1H 3J3
Canada

Email: Boris.Zhang@telus.com