Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 25, 2008

I. Minei (Editor)
K. Kompella
Juniper Networks
I. Wijnands (Editor)
B. Thomas
Cisco Systems, Inc.
February 22, 2008

Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths
draft-ietf-mpls-ldp-p2mp-04

Status of this Memo

Copyright Notice

Abstract

This document describes extensions to the Label Distribution Protocol (LDP) for the setup of point to multi-point (P2MP) and multipoint-to-multipoint (MP2MP) Label Switched Paths (LSPs) in Multi-Protocol Label Switching (MPLS) networks. The solution relies on LDP without requiring a multicast routing protocol in the network. Protocol elements and procedures for this solution are described for building such LSPs in a receiver-initiated manner. There can be various applications for P2MP/MP2MP LSPs, for example IP multicast or support for multicast in BGP/MPLS L3VPNs. Specification of how such applications can use a LDP signaled P2MP/MP2MP LSP is outside the scope of this document.

Table of Contents

1. Introduction

The LDP protocol is described in [1]. It defines mechanisms for setting up point-to-point (P2P) and multipoint-to-point (MP2P) LSPs in the network. This document describes extensions to LDP for setting up point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) LSPs. These are collectively referred to as multipoint LSPs (MP LSPs). A P2MP LSP allows traffic from a single root (or ingress) node to be delivered to a number of leaf (or egress) nodes. A MP2MP LSP allows traffic from multiple ingress nodes to be delivered to multiple egress nodes. Only a single copy of the packet will be sent on any link traversed by the MP LSP (see note at end of Section 2.4.1). This is accomplished without the use of a multicast protocol in the network. There can be several MP LSPs rooted at a given ingress node, each with its own identifier.

The solution assumes that the leaf nodes of the MP LSP know the root node and identifier of the MP LSP to which they belong. The mechanisms for the distribution of this information are outside the scope of this document. The specification of how an application can use a MP LSP signaled by LDP is also outside the scope of this document.

Interested readers may also wish to peruse the requirements draft [9] and other documents [8] and [10].

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [2].

1.2. Terminology

The following terminology is taken from [9].

P2P LSP: An LSP that has one Ingress LSR and one Egress LSR.

P2MP LSP: An LSP that has one Ingress LSR and one or more Egress LSRs.

MP2P LSP: An LSP that has one or more Ingress LSRs and one unique Egress LSR.

MP2MP LSP: An LSP that connects a set of leaf nodes, acting indifferently as ingress or egress.

MP LSP: A multipoint LSP, either a P2MP or an MP2MP LSP.

Ingress LSR: Source of the P2MP LSP, also referred to as root node.

Egress LSR: One of potentially many destinations of an LSP, also referred to as leaf node in the case of P2MP and MP2MP LSPs.

Transit LSR: An LSR that has one or more directly connected downstream LSRs.

Bud LSR: An LSR that is an egress but also has one or more directly connected downstream LSRs.

2. Setting up P2MP LSPs with LDP

A P2MP LSP consists of a single root node, zero or more transit nodes and one or more leaf nodes. Leaf nodes initiate P2MP LSP setup and tear-down. Leaf nodes also install forwarding state to deliver the traffic received on a P2MP LSP to wherever it needs to go; how this is done is outside the scope of this document. Transit nodes install MPLS forwarding state and propagate the P2MP LSP setup (and tear-down) toward the root. The root node installs forwarding state to map traffic into the P2MP LSP; how the root node determines which traffic should go over the P2MP LSP is outside the scope of this document.

2.1. Support for P2MP LSP setup with LDP

Support for the setup of P2MP LSPs is advertised using LDP capabilities as defined in [6]. An implementation supporting the P2MP procedures specified in this document MUST implement the procedures for Capability Parameters in Initialization Messages.

A new Capability Parameter TLV is defined, the P2MP Capability. Following is the format of the P2MP Capability Parameter.

```
        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
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |1|0| P2MP Capability (TBD IANA) |     Length (= 1)             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
       |1| Reserved    |
       +-+-+-+-+-+-+-+-+
```

The P2MP Capability TLV MUST be supported in the LDP Initialization Message. Advertisement of the P2MP Capability indicates support of the procedures for P2MP LSP setup detailed in this document. If the peer has not advertised the corresponding capability, then no label messages using the P2MP FEC Element should be sent to the peer.

2.2. The P2MP FEC Element

For the setup of a P2MP LSP with LDP, we define one new protocol entity, the P2MP FEC Element to be used as a FEC Element in the FEC TLV. Note that the P2MP FEC Element does not necessarily identify the traffic that must be mapped to the LSP, so from that point of view, the use of the term FEC is a misnomer. The description of the P2MP FEC Element follows.

The P2MP FEC Element consists of the address of the root of the P2MP LSP and an opaque value. The opaque value consists of one or more LDP MP Opaque Value Elements. The opaque value is unique within the context of the root node. The combination of (Root Node Address, Opaque Value) uniquely identifies a P2MP LSP within the MPLS network.

The P2MP FEC Element is encoded as follows:

```
       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
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |P2MP Type (TBD)|        Address Family         | Address Length|
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      ~                       Root Node Address                       ~
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |    Opaque Length              |    Opaque Value ...           |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
      ~                                                               ~
      |                                                               |
      |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
      |                               |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The type of the P2MP FEC Element is to be assigned by IANA.

Address Family: Two octet quantity containing a value from ADDRESS FAMILY NUMBERS in [3] that encodes the address family for the Root LSR Address.

Address Length: Length of the Root LSR Address in octets.

Root Node Address: A host address encoded according to the Address Family field.

Opaque Length: The length of the Opaque Value, in octets.

Opaque Value: One or more MP Opaque Value elements, uniquely identifying the P2MP LSP in the context of the Root Node. This is described in the next section.

If the Address Family is IPv4, the Address Length MUST be 4; if the Address Family is IPv6, the Address Length MUST be 16. No other Address Lengths are defined at present.

If the Address Length doesn't match the defined length for the Address Family, the receiver SHOULD abort processing the message containing the FEC Element, and send an "Unknown FEC" Notification message to its LDP peer signaling an error.

If a FEC TLV contains a P2MP FEC Element, the P2MP FEC Element MUST be the only FEC Element in the FEC TLV.

2.3. The LDP MP Opaque Value Element

The LDP MP Opaque Value Element is used in the P2MP and MP2MP FEC Elements defined in subsequent sections. It carries information that is meaningful to leaf (and bud) LSRs, but need not be interpreted by non-leaf LSRs.

The LDP MP Opaque Value Element is encoded as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Type(TBD)     | Length                        | Value ...     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
~                                                               ~
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |

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

Type: The type of the LDP MP Opaque Value Element is to be assigned by IANA.

Length: The length of the Value field, in octets.

Value: String of Length octets, to be interpreted as specified by the Type field.

2.3.1. The Generic LSP Identifier

The generic LSP identifier is a type of Opaque Value Element encoded as follows:

Type: 1 (to be assigned by IANA)

Length: 4

Value: A 32bit integer, unique in the context of the root, as identified by the root's address.

This type of Opaque Value Element is recommended when mapping of traffic to LSPs is non-algorithmic, and done by means outside LDP.

2.4. Using the P2MP FEC Element

This section defines the rules for the processing and propagation of the P2MP FEC Element. The following notation is used in the processing rules:

1. P2MP FEC Element <X, Y>: a FEC Element with Root Node Address X and Opaque Value Y.

2. P2MP Label Map <X, Y, L>: a Label Map message with a FEC TLV with a single P2MP FEC Element <X, Y> and Label TLV with label L.

3. P2MP Label Withdraw <X, Y, L>: a Label Withdraw message with a FEC TLV with a single P2MP FEC Element <X, Y> and Label TLV with label L.

4. P2MP LSP <X, Y> (or simply <X, Y>): a P2MP LSP with Root Node Address X and Opaque Value Y.

5. The notation L' -> {<I1, L1> <I2, L2> ..., <In, Ln>} on LSR X means that on receiving a packet with label L', X makes n copies of the packet. For copy i of the packet, X swaps L' with Li and sends it out over interface Ii.

The procedures below are organized by the role which the node plays in the P2MP LSP. Node Z knows that it is a leaf node by a discovery process which is outside the scope of this document. During the course of protocol operation, the root node recognizes its role because it owns the Root Node Address. A transit node is any node (other than the root node) that receives a P2MP Label Map message (i.e., one that has leaf nodes downstream of it).

Note that a transit node (and indeed the root node) may also be a leaf node.

2.4.1. Label Map

The following lists procedures for generating and processing P2MP Label Map messages for nodes that participate in a P2MP LSP. An LSR should apply those procedures that apply to it, based on its role in the P2MP LSP.

For the approach described here we use downstream assigned labels. On Ethernet networks this may be less optimal, see Section 6.

2.4.1.1. Determining one's 'upstream LSR'

A node Z that is part of P2MP LSP <X, Y> determines the LDP peer U which lies on the best path from Z to the root node X. If there are more than one such LDP peers, only one of them is picked. U is Z's "Upstream LSR" for <X, Y>.

When there are several candidate upstream LSRs, the LSR MAY select one upstream LSR using the following procedure:

1. The candidate upstream LSRs are numbered from lower to higher IP address

2. The following hash is performed: H = (Sum Opaque value) modulo N, where N is the number of candidate upstream LSRs

3. The selected upstream LSR U is the LSR that has the number H.

This allows for load balancing of a set of LSPs among a set of candidate upstream LSRs, while ensuring that on a LAN interface a single upstream LSR is selected.

2.4.1.2. Leaf Operation

A leaf node Z of P2MP LSP <X, Y> determines its upstream LSR U for <X, Y> as per Section 2.4.1.1, allocates a label L, and sends a P2MP Label Map <X, Y, L> to U.

2.4.1.3. Transit Node operation

Suppose a transit node Z receives a P2MP Label Map <X, Y, L> from LDP peer T. Z checks whether it already has state for <X, Y>. If not, Z allocates a label L', and installs state to swap L' with L over interface I associated with peer T. Z also determines its upstream LSR U for <X, Y> as per Section 2.4.1.1, and sends a P2MP Label Map <X, Y, L'> to U.

If Z already has state for <X, Y>, then Z does not send a Label Map message for P2MP LSP <X, Y>. All that Z needs to do in this case is update its forwarding state. Assuming its old forwarding state was L'-> {<I1, L1> <I2, L2> ..., <In, Ln>}, its new forwarding state becomes L'-> {<I1, L1> <I2, L2> ..., <In, Ln>, <I, L>}.

2.4.1.4. Root Node Operation

Suppose the root node Z receives a P2MP Label Map <X, Y, L> from peer T. Z checks whether it already has forwarding state for <X, Y>. If not, Z creates forwarding state to push label L onto the traffic that Z wants to forward over the P2MP LSP (how this traffic is determined is outside the scope of this document).

If Z already has forwarding state for <X, Y>, then Z adds "push label L, send over interface I" to the nexthop, where I is the interface associated with peer T.

2.4.2. Label Withdraw

The following lists procedures for generating and processing P2MP Label Withdraw messages for nodes that participate in a P2MP LSP. An LSR should apply those procedures that apply to it, based on its role in the P2MP LSP.

2.4.2.1. Leaf Operation

If a leaf node Z discovers (by means outside the scope of this document) that it is no longer a leaf of the P2MP LSP, it SHOULD send a Label Withdraw <X, Y, L> to its upstream LSR U for <X, Y>, where L is the label it had previously advertised to U for <X, Y>.

2.4.2.2. Transit Node Operation

If a transit node Z receives a Label Withdraw message <X, Y, L> from a node W, it deletes label L from its forwarding state, and sends a Label Release message with label L to W.

If deleting L from Z's forwarding state for P2MP LSP <X, Y> results in no state remaining for <X, Y>, then Z propagates the Label Withdraw for <X, Y>, to its upstream T, by sending a Label Withdraw <X, Y, L1> where L1 is the label Z had previously advertised to T for <X, Y>.

2.4.2.3. Root Node Operation

The procedure when the root node of a P2MP LSP receives a Label Withdraw message are the same as for transit nodes, except that it would not propagate the Label Withdraw upstream (as it has no upstream).

2.4.2.4. Upstream LSR change

If, for a given node Z participating in a P2MP LSP <X, Y>, the upstream LSR changes, say from U to U', then Z MUST update its forwarding state by deleting the state for label L, allocating a new label, L', for <X,Y>, and installing the forwarding state for L'. In addition Z MUST send a Label Map <X, Y, L'> to U' and send a Label Withdraw <X, Y, L> to U.

3. Shared Trees

The mechanism described above shows how to build a tree with a single root and multiple leaves, i.e., a P2MP LSP. One can use essentially the same mechanism to build Shared Trees with LDP. A Shared Tree can

be used by a group of routers that want to multicast traffic among themselves, i.e., each node is both a root node (when it sources traffic) and a leaf node (when any other member of the group sources traffic). A Shared Tree offers similar functionality to a MP2MP LSP, but the underlying multicasting mechanism uses a P2MP LSP. One example where a Shared Tree is useful is video-conferencing. Another is Virtual Private LAN Service (VPLS) [7], where for some types of traffic, each device participating in a VPLS must send packets to every other device in that VPLS.

One way to build a Shared Tree is to build an LDP P2MP LSP rooted at a common point, the Shared Root (SR), and whose leaves are all the members of the group. Each member of the Shared Tree unicasts traffic to the SR (using, for example, the MP2P LSP created by the unicast LDP FEC advertised by the SR); the SR then splices this traffic into the LDP P2MP LSP. The SR may be (but need not be) a member of the multicast group.

A major advantage of this approach is that no further protocol mechanisms beyond the one already described are needed to set up a Shared Tree. Furthermore, a Shared Tree is very efficient in terms of the multicast state in the network, and is reasonably efficient in terms of the bandwidth required to send traffic.

A property of this approach is that a sender will receive its own packets as part of the multicast; thus a sender must be prepared to recognize and discard packets that it itself has sent. For a number of applications (for example, VPLS), this requirement is easy to meet. Another consideration is the various techniques that can be used to splice unicast LDP MP2P LSPs to the LDP P2MP LSP; these will be described in a later revision.

4. Setting up MP2MP LSPs with LDP

An MP2MP LSP is much like a P2MP LSP in that it consists of a single root node, zero or more transit nodes and one or more leaf LSRs acting equally as Ingress or Egress LSR. A leaf node participates in the setup of an MP2MP LSP by establishing both a downstream LSP, which is much like a P2MP LSP from the root, and an upstream LSP which is used to send traffic toward the root and other leaf nodes. Transit nodes support the setup by propagating the upstream and downstream LSP setup toward the root and installing the necessary MPLS forwarding state. The transmission of packets from the root node of a MP2MP LSP to the receivers is identical to that for a P2MP LSP. Traffic from a leaf node follows the upstream LSP toward the root node and branches downward along the downstream LSP as required to reach other leaf nodes. Mapping traffic to the MP2MP LSP may

happen at any leaf node. How that mapping is established is outside the scope of this document.

Due to how a MP2MP LSP is built a leaf LSR that is sending packets on the MP2MP LSP does not receive its own packets. There is also no additional mechanism needed on the root or transit LSR to match upstream traffic to the downstream forwarding state. Packets that are forwarded over a MP2MP LSP will not traverse a link more than once, with the exception of LAN links which are discussed in Section 4.3.1

4.1. Support for MP2MP LSP setup with LDP

Support for the setup of MP2MP LSPs is advertised using LDP capabilities as defined in [6]. An implementation supporting the MP2MP procedures specified in this document MUST implement the procedures for Capability Parameters in Initialization Messages.

A new Capability Parameter TLV is defined, the MP2MP Capability. Following is the format of the MP2MP Capability Parameter.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0| MP2MP Capability (TBD IANA) |    Length (= 1)              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1| Reserved    |
+-+-+-+-+-+-+-+-+
```

The MP2MP Capability TLV MUST be supported in the LDP Initialization Message. Advertisement of the MP2MP Capability indicates support of the procedures for MP2MP LSP setup detailed in this document. If the peer has not advertised the corresponding capability, then no label messages using the MP2MP upstream and downstream FEC Elements should be sent to the peer.

4.2. The MP2MP downstream and upstream FEC Elements.

For the setup of a MP2MP LSP with LDP we define 2 new protocol entities, the MP2MP downstream FEC and upstream FEC Element. Both elements will be used as FEC Elements in the FEC TLV. Note that the MP2MP FEC Elements do not necessarily identify the traffic that must be mapped to the LSP, so from that point of view, the use of the term FEC is a misnomer. The description of the MP2MP FEC Elements follow.

The structure, encoding and error handling for the MP2MP downstream and upstream FEC Elements are the same as for the P2MP FEC Element described in Section 2.2. The difference is that two new FEC types

are used: MP2MP downstream type (TBD) and MP2MP upstream type (TBD).

If a FEC TLV contains an MP2MP FEC Element, the MP2MP FEC Element MUST be the only FEC Element in the FEC TLV.

4.3. Using the MP2MP FEC Elements

This section defines the rules for the processing and propagation of the MP2MP FEC Elements. The following notation is used in the processing rules:

1. MP2MP downstream LSP <X, Y> (or simply downstream <X, Y>): an MP2MP LSP downstream path with root node address X and opaque value Y.

2. MP2MP upstream LSP <X, Y, D> (or simply upstream <X, Y, D>): a MP2MP LSP upstream path for downstream node D with root node address X and opaque value Y.

3. MP2MP downstream FEC Element <X, Y>: a FEC Element with root node address X and opaque value Y used for a downstream MP2MP LSP.

4. MP2MP upstream FEC Element <X, Y>: a FEC Element with root node address X and opaque value Y used for an upstream MP2MP LSP.

5. MP2MP Label Map downstream <X, Y, L>: A Label Map message with a FEC TLV with a single MP2MP downstream FEC Element <X, Y> and label TLV with label L.

6. MP2MP Label Map upstream <X, Y, Lu>: A Label Map message with a FEC TLV with a single MP2MP upstream FEC Element <X, Y> and label TLV with label Lu.

7. MP2MP Label Withdraw downstream <X, Y, L>: a Label Withdraw message with a FEC TLV with a single MP2MP downstream FEC Element <X, Y> and label TLV with label L.

8. MP2MP Label Withdraw upstream <X, Y, Lu>: a Label Withdraw message with a FEC TLV with a single MP2MP upstream FEC Element <X, Y> and label TLV with label Lu.

The procedures below are organized by the role which the node plays in the MP2MP LSP. Node Z knows that it is a leaf node by a discovery process which is outside the scope of this document. During the course of the protocol operation, the root node recognizes its role because it owns the root node address. A transit node is any node (other then the root node) that receives a MP2MP Label Map message (i.e., one that has leaf nodes downstream of it).

Note that a transit node (and indeed the root node) may also be a leaf node and the root node does not have to be an ingress LSR or leaf of the MP2MP LSP.

4.3.1. MP2MP Label Map upstream and downstream

The following lists procedures for generating and processing MP2MP Label Map messages for nodes that participate in a MP2MP LSP. An LSR should apply those procedures that apply to it, based on its role in the MP2MP LSP.

For the approach described here if there are several receivers for a MP2MP LSP on a LAN, packets are replicated over the LAN. This may not be optimal; optimizing this case is for further study, see [4].

4.3.1.1. Determining one's upstream MP2MP LSR

Determining the upstream LDP peer U for a MP2MP LSP <X, Y> follows the procedure for a P2MP LSP described in Section 2.4.1.1.

4.3.1.2. Determining one's downstream MP2MP LSR

A LDP peer U which receives a MP2MP Label Map downstream from a LDP peer D will treat D as downstream MP2MP LSR.

4.3.1.3. MP2MP leaf node operation

A leaf node Z of a MP2MP LSP <X, Y> determines its upstream LSR U for <X, Y> as per Section 4.3.1.1, allocates a label L, and sends a MP2MP Label Map downstream <X, Y, L> to U.

Leaf node Z expects an MP2MP Label Map upstream <X, Y, Lu> from node U in response to the MP2MP Label Map downstream it sent to node U. Z checks whether it already has forwarding state for upstream <X, Y>. If not, Z creates forwarding state to push label Lu onto the traffic that Z wants to forward over the MP2MP LSP. How it determines what traffic to forward on this MP2MP LSP is outside the scope of this document.

4.3.1.4. MP2MP transit node operation

When node Z receives a MP2MP Label Map downstream <X, Y, L> from peer D associated with interface I, it checks whether it has forwarding state for downstream <X, Y>. If not, Z allocates a label L' and installs downstream forwarding state to swap label L' with label L over interface I. Z also determines its upstream LSR U for <X, Y> as per Section 4.3.1.1, and sends a MP2MP Label Map downstream <X, Y, L'> to U.

If Z already has forwarding state for downstream <X, Y>, all that Z needs to do is update its forwarding state. Assuming its old forwarding state was L'-> {<I1, L1> <I2, L2> ..., <In, Ln>}, its new forwarding state becomes L'-> {<I1, L1> <I2, L2> ..., <In, Ln>, <I, L>}.

Node Z checks whether it already has forwarding state upstream <X, Y, D>. If it does, then no further action needs to happen. If it does not, it allocates a label Lu and creates a new label swap for Lu from the label swap(s) from the forwarding state downstream <X, Y>, omitting the swap on interface I for node D. This allows upstream traffic to follow the MP2MP tree down to other node(s) except the node from which Z received the MP2MP Label Map downstream <X, Y, L>. Node Z determines the downstream MP2MP LSR as per Section 4.3.1.2, and sends a MP2MP Label Map upstream <X, Y, Lu> to node D.

Transit node Z will also receive a MP2MP Label Map upstream <X, Y, Lu> in response to the MP2MP Label Map downstream sent to node U associated with interface Iu. Node Z will add label swap Lu over interface Iu to the forwarding state upstream <X, Y, D>. This allows packets to go up the tree towards the root node.

4.3.1.5. MP2MP root node operation

4.3.1.5.1. Root node is also a leaf

Suppose root/leaf node Z receives a MP2MP Label Map downstream <X, Y, L> from node D associated with interface I. Z checks whether it already has forwarding state downstream <X, Y>. If not, Z creates forwarding state for downstream to push label L on traffic that Z wants to forward down the MP2MP LSP. How it determines what traffic to forward on this MP2MP LSP is outside the scope of this document. If Z already has forwarding state for downstream <X, Y>, then Z will add the label push for L over interface I to it.

Node Z checks if it has forwarding state for upstream <X, Y, D>. If not, Z allocates a label Lu and creates upstream forwarding state to push Lu with the label push(s) from the forwarding state downstream

<X, Y>, except the push on interface I for node D. This allows upstream traffic to go down the MP2MP to other node(s), except the node from which the traffic was received. Node Z determines the downstream MP2MP LSR as per section Section 4.3.1.2, and sends a MP2MP Label Map upstream <X, Y, Lu> to node D. Since Z is the root of the tree Z will not send a MP2MP downstream map and will not receive a MP2MP upstream map.

4.3.1.5.2. Root node is not a leaf

Suppose the root node Z receives a MP2MP Label Map downstream <X, Y, L> from node D associated with interface I. Z checks whether it already has forwarding state for downstream <X, Y>. If not, Z creates downstream forwarding state and installs a outgoing label L over interface I. If Z already has forwarding state for downstream <X, Y>, then Z will add label L over interface I to the existing state.

Node Z checks if it has forwarding state for upstream <X, Y, D>. If not, Z allocates a label Lu and creates forwarding state to swap Lu with the label swap(s) from the forwarding state downstream <X, Y>, except the swap for node D. This allows upstream traffic to go down the MP2MP to other node(s), except the node is was received from. Root node Z determines the downstream MP2MP LSR D as per Section 4.3.1.2, and sends a MP2MP Label Map upstream <X, Y, Lu> to it. Since Z is the root of the tree Z will not send a MP2MP downstream map and will not receive a MP2MP upstream map.

4.3.2. MP2MP Label Withdraw

The following lists procedures for generating and processing MP2MP Label Withdraw messages for nodes that participate in a MP2MP LSP. An LSR should apply those procedures that apply to it, based on its role in the MP2MP LSP.

4.3.2.1. MP2MP leaf operation

If a leaf node Z discovers (by means outside the scope of this document) that it is no longer a leaf of the MP2MP LSP, it SHOULD send a downstream Label Withdraw <X, Y, L> to its upstream LSR U for <X, Y>, where L is the label it had previously advertised to U for <X,Y>.

Leaf node Z expects the upstream router U to respond by sending a downstream label release for L and a upstream Label Withdraw for <X, Y, Lu> to remove Lu from the upstream state. Node Z will remove label Lu from its upstream state and send a label release message with label Lu to U.

4.3.2.2. MP2MP transit node operation

If a transit node Z receives a downstream label withdraw message <X, Y, L> from node D, it deletes label L from its forwarding state downstream <X, Y> and from all its upstream states for <X, Y>. Node Z sends a label release message with label L to D. Since node D is no longer part of the downstream forwarding state, Z cleans up the forwarding state upstream <X, Y, D> and sends a upstream Label Withdraw for <X, Y, Lu> to D.

If deleting L from Z's forwarding state for downstream <X, Y> results in no state remaining for <X, Y>, then Z propagates the Label Withdraw <X, Y, L> to its upstream node U for <X,Y>.

4.3.2.3. MP2MP root node operation

The procedure when the root node of a MP2MP LSP receives a label withdraw message is the same as for transit nodes, except that the root node would not propagate the Label Withdraw upstream (as it has no upstream).

4.3.2.4. MP2MP Upstream LSR change

The procedure for changing the upstream LSR is the same as documented in Section 2.4.2.4, except it is applied to MP2MP FECs, using the procedures described in Section 4.3.1 through Section 4.3.2.3.

5. The LDP MP Status TLV

An LDP MP capable router MAY use an LDP MP Status TLV to indicate additional status for a MP LSP to its remote peers. This includes signaling to peers that are either upstream or downstream of the LDP MP capable router. The value of the LDP MP status TLV will remain opaque to LDP and MAY encode one or more status elements.

The LDP MP Status TLV is encoded as follows:

```
     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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |1|0| LDP MP Status Type(TBD)   |            Length             |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                           Value                               |
    ~                                                               ~
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LDP MP Status Type: The LDP MP Status Type to be assigned by IANA.

Length: Length of the LDP MP Status Value in octets.

Value: One or more LDP MP Status Value elements.

5.1. The LDP MP Status Value Element

The LDP MP Status Value Element that is included in the LDP MP Status TLV Value has the following encoding.

```
    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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | Type(TBD)     | Length                        | Value ...     |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
    ~                                                               ~
    |                                                               |
    |                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |                               |

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

Type: The type of the LDP MP Status Value Element is to be assigned
   by IANA.

Length: The length of the Value field, in octets.

Value: String of Length octets, to be interpreted as specified by the Type field.

5.2. LDP Messages containing LDP MP Status messages

The LDP MP status message may appear either in a label mapping message or a LDP notification message.

5.2.1. LDP MP Status sent in LDP notification messages

An LDP MP status TLV sent in a notification message must be accompanied with a Status TLV. The general format of the Notification Message with an LDP MP status TLV is:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|   Notification (0x0001)     |      Message Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Message ID                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Status TLV                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   LDP MP Status TLV                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Optional LDP MP FEC TLV                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                 Optional Label TLV                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Status TLV status code is used to indicate that LDP MP status TLV and an additional information follows in the Notification message's "optional parameter" section. Depending on the actual contents of the LDP MP status TLV, an LDP P2MP or MP2MP FEC TLV and Label TLV may also be present to provide context to the LDP MP Status TLV. (NOTE: Status Code is pending IANA assignment).

Since the notification does not refer to any particular message, the Message Id and Message Type fields are set to 0.

5.2.2. LDP MP Status TLV in Label Mapping Message

An example of the Label Mapping Message defined in RFC3036 is shown below to illustrate the message with an Optional LDP MP Status TLV present.

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|0|   Label Mapping (0x0400)    |      Message Length           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Message ID                                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     FEC TLV                                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Label TLV                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Optional LDP MP Status TLV                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     Additional Optional Parameters            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

6. Upstream label allocation on a LAN

On a LAN the upstream LSR will send a copy of the packet to each receiver individually. If there is more then one receiver on the LAN we don't take full benefit of the multi-access capability of the network. We may optimize the bandwidth consumption on the LAN and replication overhead on the upstream LSR by using upstream label allocation [4]. Procedures on how to distribute upstream labels using LDP is documented in [5].

6.1. LDP Multipoint-to-Multipoint on a LAN

The procedure to allocate a context label on a LAN is defined in [4]. That procedure results in each LSR on a given LAN having a context label which, on that LAN, can be used to identify itself uniquely. Each LSR advertises its context label as an upstream-assigned label, following the procedures of [5]. Any LSR for which the LAN is a downstream link on some P2MP or MP2MP LSP will allocate an upstream-assigned label identifying that LSP. When the LSR forwards a packet downstream on one of those LSPs, the packet's top label must be the LSR's context label, and the packet's second label is the label identifying the LSP. We will call the top label the "upstream LSR label" and the second label the "LSP label".

6.1.1. MP2MP downstream forwarding

The downstream path of a MP2MP LSP is much like a normal P2MP LSP, so we will use the same procedures as defined in [5]. A label request for a LSP label is send to the upstream LSR. The label mapping that is received from the upstream LSR contains the LSP label for the

MP2MP FEC and the upstream LSR context label. The MP2MP downstream path (corresponding to the LSP label) will be installed the context specific forwarding table corresponding to the upstream LSR label. Packets sent by the upstream router can be forwarded downstream using this forwarding state based on a two label lookup.

6.1.2. MP2MP upstream forwarding

A MP2MP LSP also has an upstream forwarding path. Upstream packets need to be forwarded in the direction of the root and downstream on any node on the LAN that has a downstream interface for the LSP. For a given MP2MP LSP on a given LAN, exactly one LSR is considered to be the upstream LSR. If an LSR on the LAN receives a packet from one of its downstream interfaces for the LSP, and if it needs to forward the packet onto the LAN, it ensures that the packet's top label is the context label of the upstream LSR, and that its second label is the LSP label that was assigned by the upstream LSR.

Other LSRs receiving the packet will not be able to tell whether the packet really came from the upstream router, but that makes no difference in the processing of the packet. The upstream LSR will see its own upstream LSR in the label, and this will enable it to determine that the packet is traveling upstream.

7. Root node redundancy

The root node is a single point of failure for an MP LSP, whether this is P2MP or MP2MP. The problem is particularly severe for MP2MP LSPs. In the case of MP2MP LSPs, all leaf nodes must use the same root node to set up the MP2MP LSP, because otherwise the traffic sourced by some leafs is not received by others. Because the root node is the single point of failure for an MP LSP, we need a fast and efficient mechanism to recover from a root node failure.

An MP LSP is uniquely identified in the network by the opaque value and the root node address. It is likely that the root node for an MP LSP is defined statically. The root node address may be configured on each leaf statically or learned using a dynamic protocol. How leafs learn about the root node is out of the scope of this document.

Suppose that for the same opaque value we define two (or more) root node addresses and we build a tree to each root using the same opaque value. Effectively these will be treated as different MP LSPs in the network. Once the trees are built, the procedures differ for P2MP and MP2MP LSPs. The different procedures are explained in the sections below.

7.1. Root node redundancy - procedures for P2MP LSPs

Since all leafs have set up P2MP LSPs to all the roots, they are prepared to receive packets on either one of these LSPs. However, only one of the roots should be forwarding traffic at any given time, for the following reasons: 1) to achieve bandwidth savings in the network and 2) to ensure that the receiving leafs don't receive duplicate packets (since one cannot assume that the receiving leafs are able to discard duplicates). How the roots determine which one is the active sender is outside the scope of this document.

7.2. Root node redundancy - procedures for MP2MP LSPs

Since all leafs have set up an MP2MP LSP to each one of the root nodes for this opaque value, a sending leaf may pick either of the two (or more) MP2MP LSPs to forward a packet on. The leaf nodes receive the packet on one of the MP2MP LSPs. The client of the MP2MP LSP does not care on which MP2MP LSP the packet is received, as long as they are for the same opaque value. The sending leaf MUST only forward a packet on one MP2MP LSP at a given point in time. The receiving leafs are unable to discard duplicate packets because they accept on all LSPs. Using all the available MP2MP LSPs we can implement redundancy using the following procedures.

A sending leaf selects a single root node out of the available roots for a given opaque value. A good strategy MAY be to look at the unicast routing table and select a root that is closest in terms of the unicast metric. As soon as the root address of the active root disappears from the unicast routing table (or becomes less attractive) due to root node or link failure, the leaf can select a new best root address and start forwarding to it directly. If multiple root nodes have the same unicast metric, the highest root node addresses MAY be selected, or per session load balancing MAY be done over the root nodes.

All leafs participating in a MP2MP LSP MUST join to all the available root nodes for a given opaque value. Since the sending leaf may pick any MP2MP LSP, it must be prepared to receive on it.

The advantage of pre-building multiple MP2MP LSPs for a single opaque value is that convergence from a root node failure happens as fast as the unicast routing protocol is able to notify. There is no need for an additional protocol to advertise to the leaf nodes which root node is the active root. The root selection is a local leaf policy that does not need to be coordinated with other leafs. The disadvantage of pre-building multiple MP2MP LSPs is that more label resources are used, depending on how many root nodes are defined.

8. Make Before Break (MBB)

An LSR selects as its upstream LSR for a MP LSP the LSR that is its next hop to the root of the LSP. When the best path to reach the root changes the LSR must choose a new upstream LSR. Sections Section 2.4.2.4 and Section 4.3.2.4 describe these procedures.

When the best path to the root changes the LSP may be broken temporarily resulting in packet loss until the LSP "reconverges" to a new upstream LSR. The goal of MBB when this happens is to keep the duration of packet loss as short as possible. In addition, there are scenarios where the best path from the LSR to the root changes but the LSP continues to forward packets to the prevous next hop to the root. That may occur when a link comes up or routing metrics change. In such a case a new LSP should be established before the old LSP is removed to limit the duration of packet loss. The procedures described below deal with both scenarios in a way that an LSR does not need to know which of the events described above caused its upstream router for an MBB LSP to change.

This MBB procedures are an optional extension to the MP LSP building procedures described in this draft.

8.1. MBB overview

The MBB procedues use additional LDP signaling.

Suppose some event causes a downstream LSR-D to select a new upstream LSR-U for FEC-A. The new LSR-U may already be forwarding packets for FEC-A; that is, to downstream LSR's other than LSR-D. After LSR-U receives a label for FEC-A from LSR-D, it will notify LSR-D when it knows that the LSP for FEC-A has been established from the root to itself. When LSR-D receives this MBB notification it will change its next hop for the LSP root to LSR-U.

The assumption is that if LSR-U has received an MBB notification from its upstream router for the FEC-A LSP and has installed forwarding state the LSP it is capable of forwarding packets on the LSP. At that point LSR-U should signal LSR-D by means of an MBB notification that it has become part of the tree identified by FEC-A and that LSR-D should initiate its switchover to the LSP.

At LSR-U the LSP for FEC-A may be in 1 of 3 states.

1. There is no state for FEC-A.

2. State for FEC-A exists and LSR-U is waiting for MBB notification that the LSP from the root to it exists.

3. State for FEC-A exists and the MBB notification has been received.

After LSR-U receives LSR-D's Label Mapping message for FEC-A LSR-U MUST NOT reply with an MBB notification to LSR-D until its state for the LSP is state #3 above. If the state of the LSP at LSR-U is state #1 or #2, LSR-U should remember receipt of the Label Mapping message from LSR-D while waiting for an MBB notification from its upstream LSR for the LSP. When LSR-U receives the MBB notification from its upstream LSR it transitions to LSP state #3 and sends an MBB notification to LSR-D.

8.2. The MBB Status code

As noted in Section 8.1, the procedures to establish an MBB MP LSP are different from those to establish normal MP LSPs.

When a downstream LSR sends a Label Mapping message for MP LSP to its upstream LSR it MAY include an LDP MP Status TLV that carries a MBB Status Code to indicate MBB procedures apply to the LSP. This new MBB Status Code MAY also appear in an LDP Notification message used by an upstream LSR to signal LSP state #3 to the downstream LSR; that is, that the upstream LSR's state for the LSP exists and that it has received notification from its upstream LSR that the LSP is in state #3.

The MBB Status is a type of the LDP MP Status Value Element as described in Section 5.1. It is encoded as follows:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| MBB Type = 1  |      Length = 1               | Status code   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

MBB Type: Type 1 (to be assigned by IANA)

Length: 1

Status code: 1 = MBB request

2 = MBB ack

8.3. The MBB capability

An LSR MAY advertise that it is capable of handling MBB LSPs using the capability advertisement as defined in [6]. The LDP MP MBB capability has the following format:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0| LDP MP MBB Capability     |          Length = 1           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1| Reserved    |
+-+-+-+-+-+-+-+-+
```

Note: LDP MP MBB Capability (Pending IANA assignment)


```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1|0| LDP MP MBB Capability     |          Length = 1           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|1| Reserved    |
+-+-+-+-+-+-+-+-+
```


If an LSR has not advertised that it is MBB capable, its LDP peers MUST NOT send it messages which include MBB parameters. If an LSR receives a Label Mapping message with a MBB parameter from downstream LSR-D and its upstream LSR-U has not advertised that it is MBB capable, the LSR MUST send an MBB notification immediatly to LSR-U (see Section 8.4). If this happens an MBB MP LSP will not be established, but normal a MP LSP will be the result.

8.4. The MBB procedures

8.4.1. Terminology

1. MBB LSP <X, Y>: A P2MP or MP2MP Make Before Break (MBB) LSP entry with Root Node Address X and Opaque Value Y.

2. A(N, L): An Accepting element that consists of an upstream Neighbor N and Local label L. This LSR assigned label L to

neighbor N for a specific MBB LSP. For an active element the corresponding Label is stored in the label forwarding database.

3. iA(N, L): An inactive Accepting element that consists of an upstream neighbor N and local Label L. This LSR assigned label L to neighbor N for a specific MBB LSP. For an inactive element the corresponding Label is not stored in the label forwarding database.

4. F(N, L): A Forwarding state that consists of downstream Neighbor N and Label L. This LSR is sending label packets with label L to neighbor N for a specific FEC.

5. F'(N, L): A Forwarding state that has been marked for sending a MBB Notification message to Neighbor N with Label L.

6. MBB Notification <X, Y, L>: A LDP notification message with a MP LSP <X, Y>, Label L and MBB Status code 2.

7. MBB Label Map <X, Y, L>: A P2MP Label Map or MP2MP Label Map downstream with a FEC element <X, Y>, Label L and MBB Status code 1.

8.4.2. Accepting elements

An accepting element represents a specific label value L that has been advertised to a neighbor N for a MBB LSP <X, Y> and is a candidate for accepting labels switched packets on. An LSR can have two accepting elements for a specific MBB LSP <X, Y> LSP, only one of them MUST be active. An active element is the element for which the label value has been installed in the label forwarding database. An inactive accepting element is created after a new upstream LSR is chosen and is pending to replace the active element in the label forwarding database. Inactive elements only exist temporarily while switching to a new upstream LSR. Once the switch has been completed only one active element remains. During network convergence it is possible that an inactive accepting element is created while an other inactive accepting element is pending. If that happens the older inactive accepting element MUST be replaced with an newer inactive element. If an accepting element is removed a Label Withdraw has to be send for label L to neighbor N for <X, Y>.

8.4.3. Procedures for upstream LSR change

Suppose a node Z has a MBB LSP <X, Y> with an active accepting element A(N1, L1). Due to a routing change it detects a new best path for root X and selects a new upstream LSR N2. Node Z allocates a new local label L2 and creates an inactive accepting element iA(N2,

L2). Node Z sends MBB Label Map <X, Y, L2>to N2 and waits for the new upstream LSR N2 to respond with a MBB Notification for <X, Y, L2>. During this transition phase there are two accepting elements, the element A(N1, L1) still accepting packets from N1 over label L1 and the new inactive element iA(N2, L2).

While waiting for the MBB Notification from upstream LSR N2, it is possible that an other transition occurs due to a routing change. Suppose the new upstream LSR is N3. An inactive element iA(N3, L3) is created and the old inactive element iA(N2, L2) MUST be removed. A label withdraw MUST be sent to N2 for <X, Y, L2>. The MBB Notification for <X, Y, L2> from N2 will be ignored because the inactive element is removed.

It is possible that the MBB Notification from upstream LSR is never received due to link or node failure. To prevent waiting indefinitely for the MBB Notification a timeout SHOULD be applied. As soon as the timer expires, the procedures in Section 8.4.5 are applied as if a MBB Notification was received for the inactive element.

8.4.4. Receiving a Label Map with MBB status code

Suppose node Z has state for a MBB LSP <X, Y> and receives a MBB Label Map <X, Y, L2> from N2. A new forwarding state F(N2, L2) will be added to the MP LSP if it did not already exist. If this MBB LSP has an active accepting element or node Z is the root of the MBB LSP a MBB notification <X, Y, L2)> is send to node N2. If node Z has an inactive accepting element it marks the Forwarding state as <X, Y, F'(N2, L2)>.

8.4.5. Receiving a Notification with MBB status code

Suppose node Z receives a MBB Notification <X, Y, L> from N. If node Z has state for MBB LSP <X, Y> and an inactive accepting element iA(N, L) that matches with N and L, we activate this accepting element and install label L in the label forwarding database. If an other active accepting was present it will be removed from the label forwarding database.

If this MBB LSP <X, Y> also has Forwarding states marked for sending MBB Notifications, like <X, Y, F'(N2, L2)>, MBB Notifications are send to these downstream LSRs. If node Z receives a MBB Notification for an accepting element that is not inactive or does not match the Label value and Neighbor address, the MBB notification is ignored.

8.4.6. Node operation for MP2MP LSPs

The procedures described above apply to the downstream path of a MP2MP LSP. The upstream path of the MP2MP is setup as normal without including a MBB Status code. If the MBB procedures apply to a MP2MP downstream FEC element, the upstream path to a node N is only installed in the label forwarding database if node N is part of the active accepting element. If node N is part of an inactive accepting element, the upstream path is installed when this inactive accepting element is activated.

9. Security Considerations

The same security considerations apply as for the base LDP specification, as described in [1].

10. IANA considerations

This document creates a new name space (the LDP MP Opaque Value Element type) that is to be managed by IANA, and the allocation of the value 1 for the type of Generic LSP Identifier.

This document requires allocation of three new LDP FEC Element types:

1. the P2MP FEC type - requested value 0x04

2. the MP2MP-up FEC type - requested value 0x05

3. the MP2MP-down FEC type - requested value 0x06

This document requires the assignment of three new code points for three new Capability Parameter TLVs, corresponding to the advertisement of the P2MP, MP2MP and MBB capabilities. The values requested are:

P2MP Capability Parameter - requested value 0x0508

MP2MP Capability Parameter - requested value 0x0509

MBB Capability Parameter - requested value 0x050A

This document requires the assignment of a LDP Status TLV code to indicate a LDP MP Status TLV is following in the Notification message. The value requested is:

LDP MP status - requested value 0x0000002C

This document requires the assigment of a new code point for a LDP MP Status TLV. The value requested is:

LDP MP Status TLV Type - requested value 0x096D

This document creates a new name space (the LDP MP Status Value Element type) that is to be managed by IANA, and the allocation of the value 1 for the type of MBB Status.

11. Acknowledgments

The authors would like to thank the following individuals for their review and contribution: Nischal Sheth, Yakov Rekhter, Rahul Aggarwal, Arjen Boers, Eric Rosen, Nidhi Bhaskar, Toerless Eckert, George Swallow, Jin Lizhong, Vanson Lim and Adrian Farrel.

12. Contributing authors

Below is a list of the contributing authors in alphabetical order:

Shane Amante
Level 3 Communications, LLC
1025 Eldorado Blvd
Broomfield, CO 80021
US
Email: Shane.Amante@Level3.com

Luyuan Fang
Cisco Systems
300 Beaver Brook Road
Boxborough, MA 01719
US
Email: lufang@cisco.com

Hitoshi Fukuda
NTT Communications Corporation
1-1-6, Uchisaiwai-cho, Chiyoda-ku
Tokyo 100-8019,
Japan
Email: hitoshi.fukuda@ntt.com

Yuji Kamite
NTT Communications Corporation
Tokyo Opera City Tower
3-20-2 Nishi Shinjuku, Shinjuku-ku,
Tokyo 163-1421,
Japan
Email: y.kamite@ntt.com

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US
Email: kireeti@juniper.net

Ina Minei
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US
Email: ina@juniper.net

Jean-Louis Le Roux
France Telecom
2, avenue Pierre-Marzin
Lannion, Cedex 22307
France
Email: jeanlouis.leroux@francetelecom.com

Bob Thomas
Cisco Systems, Inc.
300 Beaver Brook Road
Boxborough, MA, 01719
E-mail: rhthomas@cisco.com

Lei Wang
Telenor
Snaroyveien 30
Fornebu 1331
Norway
Email: lei.wang@telenor.com

IJsbrand Wijnands
Cisco Systems, Inc.
De kleetlaan 6a
1831 Diegem
Belgium
E-mail: ice@cisco.com

13. References

13.1. Normative References

[1] Andersson, L., Doolan, P., Feldman, N., Fredette, A., and B. Thomas, "LDP Specification", RFC 3036, January 2001.

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

[3] Reynolds, J., "Assigned Numbers: RFC 1700 is Replaced by an On-line Database", RFC 3232, January 2002.

[4] Aggarwal, R., "MPLS Upstream Label Assignment and Context Specific Label Space", draft-ietf-mpls-upstream-label-02 (work in progress), March 2007.

[5] Aggarwal, R. and J. Roux, "MPLS Upstream Label Assignment for LDP", draft-ietf-mpls-ldp-upstream-01 (work in progress), March 2007.

[6] Thomas, B., "LDP Capabilities", draft-ietf-mpls-ldp-capabilities-00 (work in progress), May 2007.

13.2. Informative References

[7] Andersson, L. and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", RFC 4664, September 2006.

[8] 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.

[9] Roux, J., "Requirements for Point-To-Multipoint Extensions to the Label Distribution Protocol", draft-ietf-mpls-mp-ldp-reqs-02 (work in progress), March 2007.

[10] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs",

draft-ietf-l3vpn-2547bis-mcast-00 (work in progress),
June 2005.

Authors' Addresses

Ina Minei
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: ina@juniper.net

Kireeti Kompella
Juniper Networks
1194 N. Mathilda Ave.
Sunnyvale, CA 94089
US

Email: kireeti@juniper.net

IJsbrand Wijnands
Cisco Systems, Inc.
De kleetlaan 6a
Diegem 1831
Belgium

Email: ice@cisco.com

Bob Thomas
Cisco Systems, Inc.
300 Beaver Brook Road
Boxborough 01719
US

Email: rhthomas@cisco.com

Full Copyright Statement

Intellectual Property

Acknowledgment

Funding for the RFC Editor function is provided by the IETF Administrative Support Activity (IASA).