Internet Engineering Task Force (IETF)
Request for Comments: 6388
Category: Standards Track
ISSN: 2070-1721

IJ. Wijnands, Ed.
Cisco Systems, Inc.
I. Minei, Ed.
K. Kompella
Juniper Networks
B. Thomas

November 2011

Label Distribution Protocol Extensions for Point-to-Multipoint and Multipoint-to-Multipoint Label Switched Paths

Abstract

This document describes extensions to the Label Distribution Protocol (LDP) for the setup of point-to-multipoint (P2MP) and multipoint-to-multipoint (MP2MP) Label Switched Paths (LSPs) in MPLS networks. These extensions are also referred to as multipoint LDP. Multipoint LDP constructs the P2MP or MP2MP LSPs without interacting with or relying upon any other multicast tree construction protocol. Protocol elements and procedures for this solution are described for building such LSPs in a receiver-initiated manner. There can be various applications for multipoint LSPs, for example IP multicast or support for multicast in BGP/MPLS Layer 3 Virtual Private Networks (L3VPNs). Specification of how such applications can use an LDP signaled multipoint LSP is outside the scope of this document.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

The LDP protocol is described in [RFC5036]. 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. An 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 to an LDP neighbor traversed by the MP LSP. 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 an MP LSP signaled by LDP is also outside the scope of this document.

Related documents that may be of interest include [RFC6348], [L3VPN-MCAST], and [RFC4875].

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 [RFC2119].

All new fields shown as "reserved" in this document MUST be set to zero on transmission and MUST be ignored on receipt.

1.2. Terminology

Some of the following terminology is taken from [RFC6348].

mLDP: Multipoint extensions for LDP.

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 with a distinguished root node that connects a set of nodes, such that traffic sent by any node in the LSP is delivered to all others.

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

Ingress LSR: An Ingress LSR for a particular LSP is an LSR that can send a data packet along the LSP. MP2MP LSPs can have multiple Ingress LSRs, P2MP LSPs have just one, and that node is often referred to as the "root node".

Egress LSR: An Egress LSR for a particular LSP is an LSR that can remove a data packet from that LSP for further processing. P2P and MP2P LSPs have only a single egress node, but P2MP and MP2MP LSPs can have multiple egress nodes.

Transit LSR: An LSR that has reachability to the root of the MP LSP via a directly connected upstream LSR and 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.

Leaf node: A leaf node can be either an Egress or Bud LSR when referred to in the context of a P2MP LSP. In the context of an MP2MP LSP, a leaf is both Ingress and Egress for the same MP2MP LSP and can also be a Bud LSR.

CRC32: This contains a Cyclic Redundancy Check value of the uncompressed data in network byte order computed according to CRC-32 algorithm used in the ISO 3309 standard [ISO3309] and in Section 8.1.1.6.2 of ITU-T recommendation V.42 [ITU.V42.1994].

FEC: Forwarding Equivalence Class

1.3. Manageability

MPLS LSRs can be modeled and managed using the MIB module defined in [RFC3813]. That MIB module is fully capable of handling the one-to-many in-segment to out-segment relationships needed to support P2MP LSPs, and no further changes are required.

[RFC3815] defines managed objects for LDP. The MIB module allows the modeling and management of LDP and LDP speakers for the protocol as defined in [RFC5036]. The protocol extensions defined in this document to support P2MP in LDP may require an additional MIB module or extensions to the modules defined in [RFC3815]. This is for future study, and at the time of this writing, no interest has been expressed in this work.

Future manageability work should pay attention to the protocol extensions defined in this document, and specifically the configurable and variable elements, along with reporting the new protocol fields that identify individual P2MP LSPs.

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 [RFC5561]. 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 (0x0508)  |      Length (= 1)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S| Reserved    |
+-+-+-+-+-+-+-+-+
```

S: As specified in [RFC5561]

The P2MP Capability TLV MUST be advertised 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 label messages using the P2MP FEC Element SHOULD NOT 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 type, 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(0x06)|        Address Family         | Address Length|
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~                       Root Node Address                       ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Opaque Length              |    Opaque Value ...           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+                               +
~                                                               ~
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The type of the P2MP FEC Element is 0x06.

Address Family: Two octet quantity containing a value from IANA's "Address Family Numbers" registry 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 Ingress LSRs and Leaf LSRs, but need not be interpreted by Transit LSRs.

The LDP MP opaque value element basic type 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 < 255    | Length                        | Value ...     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
~                                                               ~
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The Type of the LDP MP opaque value element. IANA maintains a registry of basic types (see Section 11).

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

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

The LDP MP opaque value element extended type 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 = 255    |        Extended Type          | Length (high) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-|
| Length (low)  |                Value                          |
+-+-+-+-+-+-+-+-+                                               |
~                                                               ~
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: Type = 255.

Extended Type: The Extended Type of the LDP MP opaque value element. IANA maintains a registry of extended types (see Section 11).

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 basic type encoded as follows:

Type: 1

Length: 4

Value: A 32-bit 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 is 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 Mapping <X, Y, L>: a Label Mapping message with a FEC TLV with a single P2MP FEC Element <X, Y> and Label TLV with label L. Label L MUST be allocated from the per-platform label space (see [RFC3031], Section 3.14) of the LSR sending the Label Mapping message. The use of the interface label space is outside the scope of this document.

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 that the node plays in the P2MP LSP. Node Z knows that it is a leaf node by a discovery process that 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 Mapping 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 Mapping

The remainder of this section specifies the procedures for originating P2MP Label Mapping messages and for processing received P2MP Label Mapping messages for a particular LSP. The procedures for a particular LSR depend upon the role that LSR plays in the LSP (Ingress, Transit, or Egress).

All labels discussed here are downstream-assigned [RFC5332] except those that are assigned using the procedures of Section 6.

2.4.1.1. Determining One's 'upstream LSR'

Each node that is either an Leaf or Transit LSR of MP LSP needs to use the procedures below to select an upstream LSR. A node Z that wants to join an MP LSP <X, Y> determines the LDP peer U that is Z's next-hop on the best path from Z to the root node X. If there is more than one such LDP peer, only one of them is picked. U is Z's "upstream LSR" for <X, Y>.

When there are several candidate upstream LSRs, the LSR MUST select one upstream LSR. The algorithm used for the LSR selection is a local matter. If the LSR selection is done over a LAN interface and the Section 6 procedures are applied, the following procedure SHOULD be applied to ensure that the same upstream LSR is elected among a set of candidate receivers on that LAN.

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

2. The following hash is performed: H = (CRC32(Opaque Value)) modulo N, where N is the number of upstream LSRs. The 'Opaque Value' is the field identified in the FEC Element right after 'Opaque Length'. The 'Opaque Length' indicates the size of the opaque value used in this calculation.

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

This procedure will ensure that there is a single forwarder over the LAN for a particular LSP.

2.4.1.2. Determining the Forwarding Interface to an LSR

Suppose LSR U receives an MP Label Mapping message from a downstream LSR D, specifying label L. Suppose further that U is connected to D over several LDP enabled interfaces or RSVP-TE Tunnel interfaces. If U needs to transmit to D a data packet whose top label is L, U is free to transmit the packet on any of those interfaces. The algorithm it uses to choose a particular interface and next-hop for a particular such packet is a local matter. For completeness, the following procedure MAY be used. LSR U may do a lookup in the unicast routing table to find the best interface and next-hop to reach LSR D. If the next-hop and interface are also advertised by LSR D via the LDP session, it can be used to transmit the packet to LSR D.

2.4.1.3. 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 Mapping <X, Y, L> to U.

2.4.1.4. Transit Node Operation

Suppose a transit node Z receives a P2MP Label Mapping <X, Y, L> from LSR T. Z checks whether it already has state for <X, Y>. If not, Z determines its upstream LSR U for <X, Y> as per Section 2.4.1.1. Using this Label Mapping to update the label forwarding table MUST NOT be done as long as LSR T is equal to LSR U. If LSR U is different from LSR T, Z will allocate a label L’, and install state to swap L’ with L over interface I associated with LSR T and send a P2MP Label Mapping <X, Y, L’> to LSR U. Interface I is determined via the procedures in Section 2.4.1.2.

If Z already has state for <X, Y>, then Z does not send a Label Mapping message for P2MP LSP <X, Y>. If LSR T is not equal to the upstream LSR of <X, Y> and <I, L> does not already exist as forwarding state, the forwarding state is updated. 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>}. If LSR T is equal to the installed upstream LSR, the Label Mapping from LSR T MUST be retained and MUST NOT update the label forwarding table.

2.4.1.5. Root Node Operation

Suppose the root node Z receives a P2MP Label Mapping <X, Y, L> from LSR 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 next hop, where I is the interface associated with LSR T and determined via the procedures in Section 2.4.1.2.

2.4.2. Label Withdraw

The following section 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 that it has no downstream neighbors in that LSP, and that it has no need to be an Egress LSR for that LSP (by means outside the scope of this document), then 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

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

2.4.3. Upstream LSR Change

Suppose that for a given node Z participating in a P2MP LSP <X, Y>, the upstream LSR changes from U to U' as per Section 2.4.1.1. Z MUST update its forwarding state as follows. It allocates a new label, L', for <X, Y>. The forwarding state for L' is copied from the forwarding state for L, with one exception: if U' was present in the forwarding state of L, it MUST NOT be installed in the forwarding state of L'. Then the forwarding state for L is deleted and the forwarding state for L' is installed. In addition, Z MUST send a Label Mapping <X, Y, L'> to U' and send a Label Withdraw <X, Y, L> to U. Note, if there was a downstream mapping from U that was not installed in the forwarding due to the procedures defined in Section 2.4.1.4, it can now be installed.

While changing the upstream LSR, the following must be taken into consideration. If L' is added before L is removed, there is a potential risk of packet duplication and/or the creation of a transient data-plane forwarding loop. If L is removed before L' is added, packet loss may result. Ideally the change from L to L' is done atomically such that no packet loss or duplication occurs. If that is not possible, the RECOMMENDED default behavior is to remove L before adding L'.

3. 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 an 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 an MP2MP LSP to the receivers is identical to that for a P2MP LSP. Traffic from a downstream node follows the upstream LSP toward the root node and branches downward along the downstream LSP as required to reach other leaf nodes. A packet that is received from a downstream node MUST never be forwarded back out to that same node. 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 an 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 an MP2MP LSP will not traverse a link more than once, with the possible exception of LAN links (see Section 3.3.1), if the procedures of [RFC5331] are not provided.

3.1. Support for MP2MP LSP Setup with LDP

Support for the setup of MP2MP LSPs is advertised using LDP capabilities as defined in [RFC5561]. 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 (0x0509) |      Length (= 1)             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S| Reserved    |
+-+-+-+-+-+-+-+-+
```

S: As specified in [RFC5561]

The MP2MP Capability TLV MUST be advertised 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 label messages using the MP2MP upstream and downstream FEC Elements SHOULD NOT be sent to the peer.

3.2. The MP2MP Downstream and Upstream FEC Elements

For the setup of an 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 (0x08) and MP2MP upstream type (0x07).

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

Note, except when using the procedures of [RFC5331], the MPLS labels used are "downstream-assigned" [RFC5332], even if they are bound to the "upstream FEC Element".

3.3. 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>): an 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-D Label Mapping <X, Y, L>: a Label Mapping message with a FEC TLV with a single MP2MP downstream FEC Element <X, Y> and label TLV with label L. Label L MUST be allocated from the per-platform label space (see [RFC3031], Section 3.14) of the LSR sending the Label Mapping message. The use of the interface label space is outside the scope of this document.

6. MP2MP-U Label Mapping <X, Y, Lu>: a Label Mapping message with a FEC TLV with a single MP2MP upstream FEC Element <X, Y> and label TLV with label Lu. Label Lu MUST be allocated from the per-platform label space (see [RFC3031], Section 3.14) of the LSR sending the Label Mapping message. The use of the interface label space is outside the scope of this document.

7. MP2MP-D Label Withdraw <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-U Label Withdraw <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.

9. MP2MP-D Label Release <X, Y, L>: a Label Release message with a FEC TLV with a single MP2MP downstream FEC Element <X, Y> and Label TLV with label L.

10. MP2MP-U Label Release <X, Y, Lu>: a Label Release 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 that 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 an MP2MP Label Mapping 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 a leaf of the MP2MP LSP.

3.3.1. MP2MP Label Mapping

The remainder of this section specifies the procedures for originating MP2MP Label Mapping messages and for processing received MP2MP Label Mapping messages for a particular LSP. The procedures for a particular LSR depend upon the role that the LSR plays in the LSP (Ingress, Transit, or Egress).

All labels discussed here are downstream-assigned [RFC5332] except those that are assigned using the procedures of Section 6.

3.3.1.1. Determining one’s upstream MP2MP LSR

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

3.3.1.2. Determining One’s Downstream MP2MP LSR

An LDP peer U that receives an MP2MP-D Label Mapping from an LDP peer D will treat D as downstream MP2MP LSR.

3.3.1.3. Installing the Upstream Path of an MP2MP LSP

There are two methods for installing the upstream path of an MP2MP LSP to a downstream neighbor.

1. We can install the upstream MP2MP path (to a downstream neighbor) based on receiving an MP2MP-D Label Mapping from the downstream neighbor. This will install the upstream path on a hop-by-hop basis.

2. We install the upstream MP2MP path (to a downstream neighbor) based on receiving an MP2MP-U Label Mapping from the upstream neighbor. An LSR does not need to wait for the MP2MP-U Label Mapping if it is the root of the MP2MP LSP or if it already received an MP2MP-U Label Mapping from the upstream neighbor. We call this method ordered mode. The typical result of this mode is that the downstream path of the MP2MP is built hop by hop towards the root. Once the root is reached, the root node will trigger an MP2MP-U Label Mapping to the downstream neighbor(s).

For setting up the upstream path of an MP2MP LSP, ordered mode SHOULD be used. Due to ordered mode, the upstream path of the MP2MP LSP is installed at the leaf node once the path to the root has completed. The advantage is that when a leaf starts sending immediately after the upstream path is installed, packets are able to reach the root node without being dropped due to an incomplete LSP. Method 1 is not able to guarantee that the upstream path has completed before the leaf starts sending.

3.3.1.4. MP2MP Leaf Node Operation

A leaf node Z of an MP2MP LSP <X, Y> determines its upstream LSR U for <X, Y> as per Section 3.3.1.1, allocates a label L, and sends an MP2MP-D Label Mapping <X, Y, L> to U.

Leaf node Z expects an MP2MP-U Label Mapping <X, Y, Lu> from node U in response to the MP2MP-D Label Mapping 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.

3.3.1.5. MP2MP Transit Node Operation

Suppose node Z receives an MP2MP-D Label Mapping <X, Y, L> from LSR D. Z checks whether it has forwarding state for downstream <X, Y>. If not, Z determines its upstream LSR U for <X, Y> as per Section 3.3.1.1. Using this Label Mapping to update the label forwarding table MUST NOT be done as long as LSR D is equal to LSR U. If LSR U is different from LSR D, Z will allocate a label L' and install downstream forwarding state to swap label L' with label L over interface I associated with LSR D and send an MP2MP-D Label Mapping <X, Y, L'> to U. Interface I is determined via the procedures in Section 2.4.1.2.

If Z already has forwarding state for downstream <X, Y>, all that Z needs to do in this case is check that LSR D is not equal to the upstream LSR of <X, Y> and 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>}. If the LSR D is equal to the installed upstream LSR, the Label Mapping from LSR D MUST be retained and MUST NOT update the label forwarding table.

Node Z checks if upstream LSR U already assigned a label Lu to <X, Y>. If not, transit node Z waits until it receives an MP2MP-U Label Mapping <X, Y, Lu> from LSR U (see Section 3.3.1.3). Once the MP2MP-U Label Mapping is received from LSR U, 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' with label Lu over interface Iu. Interface Iu is determined via the procedures in Section 2.4.1.2. In addition, it also adds the label swap(s) from the forwarding state downstream <X, Y>, omitting the swap on interface I for node D. The swap on interface I for node D is omitted to prevent a packet originated by D to be forwarded back to D.

Node Z determines the downstream MP2MP LSR as per Section 3.3.1.2, and sends an MP2MP-U Label Mapping <X, Y, Lu'> to node D.

3.3.1.6. MP2MP Root Node Operation

3.3.1.6.1. Root Node Is Also a Leaf

Suppose root/leaf node Z receives an MP2MP-D Label Mapping <X, Y, L> from node D. Z checks whether it already has forwarding state downstream <X, Y>. If not, Z creates downstream forwarding state 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. Interface I is determined via the procedures in Section 2.4.1.2.

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 swap Lu' with the label swap(s) from the forwarding state downstream <X, Y>, except the swap 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 3.3.1.2, and sends an MP2MP-U Label Mapping <X, Y, Lu'> to node D. Since Z is the root of the tree, Z will not send an MP2MP-D Label Mapping and will not receive an MP2MP-U Label Mapping.

3.3.1.6.2. Root Node is Not a Leaf

Suppose the root node Z receives an MP2MP-D Label Mapping <X, Y, L> from node D. 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. Interface I is determined via the procedures in Section 2.4.1.2. 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 from which it was received. Root node Z determines the downstream MP2MP LSR D as per Section 3.3.1.2, and sends an MP2MP-U Label Mapping <X, Y, Lu'> to it. Since Z is the root of the tree, Z will not send an MP2MP-D Label Mapping and will not receive an MP2MP-U Label Mapping.

3.3.2. MP2MP Label Withdraw

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

3.3.2.1. MP2MP Leaf Operation

If a leaf node Z discovers (by means outside the scope of this document) that it has no downstream neighbors in that LSP and that it has no need to be an Egress LSR for that LSP (by means outside the scope of this document), then it SHOULD send an MP2MP-D 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 will also send an unsolicited label release <X, Y, Lu> to U to indicate that the upstream path is no longer used and that label Lu can be removed.

Leaf node Z expects the upstream router U to respond by sending a downstream label release for L.

3.3.2.2. MP2MP Transit Node Operation

If a transit node Z receives an MP2MP-D 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 an MP2MP-D 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>. There is no need to send an MP2MP-U Label Withdraw <X, Y, Lu> to D because node D already removed Lu and sent a label release for Lu to Z.

If deleting L from Z's forwarding state for downstream <X, Y> results in no state remaining for <X, Y>, then Z propagates the MP2MP-D Label Withdraw <X, Y, L> to its upstream node U for <X, Y> and will also send an unsolicited MP2MP-U Label Release <X, Y, Lu> to U to indicate that the upstream path is no longer used and that label Lu can be removed.

3.3.2.3. MP2MP Root Node Operation

When the root node of an MP2MP LSP receives an MP2MP-D Label Withdraw message, the procedure is the same as that for transit nodes, except that the root node will not propagate the Label Withdraw upstream (as it has no upstream).

3.3.3. MP2MP Upstream LSR Change

The procedure for changing the upstream LSR is the same as documented in Section 2.4.3, except it is applied to MP2MP FECs, using the procedures described in Section 3.3.1 through Section 3.3.2.3.

4. Micro-Loops in MP LSPs

Micro-loops created by the unicast routing protocol during convergence may also effect mLDP MP LSPs. Since the tree building logic in mLDP is based on unicast routing, a unicast routing loop may also result in a micro-loop in the MP LSPs. Micro-loops that involve 2 directly connected routers don’t create a loop in mLDP. mLDP is able to prevent this inconsistency by never allowing an upstream LDP neighbor to be added as a downstream LDP neighbor into the Label Forwarding Table (LFT) for the same FEC. Micro-loops that involve more than 2 LSRs are not prevented.

Micro-loops that involve more than 2 LSRs may create a micro-loop in the downstream path of either an MP2MP LSP or P2MP LSP and the upstream path of the MP2MP LSP. The loops are transient and will disappear as soon as the unicast routing protocol converges and mLDP has updated the forwarding state accordingly. Micro-loops that occur in the upstream path of an MP2MP LSP may be detected by including LDP path vector in the MP2MP-U Label Mapping messages. These procedures are currently under investigation and are subjected to further study.

5. The LDP MP Status TLV

An LDP MP capable router MAY use an LDP MP Status TLV to indicate additional status for an 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(0x096F)|            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           Value                               |
~                                                               ~
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LDP MP Status Type: The LDP MP Status (0x096F).

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          | Length                        | Value ...     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+               |
~                                                               ~
|                                                               |
|                               +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Type: The type of the LDP MP Status Value Element. IANA maintains a registry of status value types (see Section 11).

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 TLV may appear either in a Label Mapping message or an 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, as described in [RFC5036]. 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 any 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 a Label TLV may also be present to provide context to the LDP MP Status TLV.

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 [RFC5036] 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 procedures so far discussed would require the upstream LSR to send a copy of the packet to each receiver individually. If there is more than 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 [RFC5331]. Procedures on how to distribute upstream labels using LDP is documented in [RFC6389].

6.1. LDP Multipoint-to-Multipoint on a LAN

The procedure to allocate a context label on a LAN is defined in [RFC5331]. 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 [RFC6389]. 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 an MP2MP LSP is much like a normal P2MP LSP, so we will use the same procedures as those defined in [RFC6389]. A label request for an LSP label is sent 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 in 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

An 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 the MP LSP 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 will be 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 an MP2MP LSP MUST join 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 the LSR that is its next hop to the root of the LSP as its upstream LSR for an MP LSP. When the best path to reach the root changes, the LSR must choose a new upstream LSR. Sections 2.4.3 and 3.3.3 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 previous 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.

The MBB procedures are an optional extension to the MP LSP building procedures described in this document. The procedures in this section offer a make-before-break behavior, except in cases where the new path is part of a transient routing loop involving more than 2 LSRs (also see Section 4).

8.1. MBB Overview

The MBB procedures 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 LSRs 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 LSR 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 or it is the root node for FEC-A.

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 LSR-U, 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 for establishing an MBB MP LSP are different from those for establishing 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 an MBB Status Code to indicate that 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 LSRs 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

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 [RFC5561]. 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          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|S| Reserved    |
+-+-+-+-+-+-+-+-+
```

LDP MP MBB Capability: The MBB Capability Parameter (0x050A)

S: As specified in [RFC5561]

If an LSR has not advertised that it is MBB capable, its LDP peers MUST NOT send it messages that include MBB parameters. If an LSR receives a Label Mapping message with an 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 immediately to LSR-U (see Section 8.4). If this happens, an MBB MP LSP will not be established, but a normal 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 an MBB Notification message to neighbor N with label L.

6. MBB Notification <X, Y, L>: An LDP notification message with an MP LSP <X, Y>, label L, and MBB Status code 2.

7. MBB Label Mapping <X, Y, L>: A P2MP Label Mapping or MP2MP Label Mapping 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 an 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 replacement the active element in the label forwarding database is pending. 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 another inactive accepting element is pending. If that happens, the older inactive accepting element MUST be replaced with a newer inactive element. If an accepting element is removed, a Label Withdraw has to be sent for label L to neighbor N for <X, Y>.

8.4.3. Procedures for Upstream LSR Change

Suppose a node Z has an 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 Mapping <X, Y, L2> to N2 and waits for the new upstream LSR N2 to respond with an 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 another 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 an MBB Notification was received for the inactive element. If a downstream LSR detects that the old upstream LSR went down while waiting for the MBB Notification from the new upstream LSR, the downstream LSR can immediately proceed without waiting for the timer to expire.

8.4.4. Receiving a Label Mapping with MBB Status Code

Suppose node Z has state for an MBB LSP <X, Y> and receives an MBB Label Mapping <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 if node Z is the root of the MBB LSP, an MBB notification <X, Y, L2)> is sent to node N2. If node Z has an inactive accepting element, it marks the Forwarding state as <X, Y, F'(N2, L2)>. If the router Z upstream LSR for <X, Y> happens to be N2, then Z MUST NOT send an MBB notification to N2 at once. Sending the MBB notification to N2 must be done only after Z upstream for <X, Y> stops being N2.

8.4.5. Receiving a Notification with MBB Status Code

Suppose node Z receives an 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 another active accepting element 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 sent to these downstream LSRs. If node Z receives an 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 an MP2MP LSP. The upstream path of the MP2MP is set up as normal without including an MBB Status code. If the MBB procedures apply to an 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. Typed Wildcard for mLDP FEC Element

The format of the mLDP FEC Typed Wildcard FEC is 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Typed Wcard   |     Type      |   Len = 2     |      AFI      ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~               |
+-+-+-+-+-+-+-+-+
```

Typed Wcard: As specified in [RFC5918]

Type: The type of FEC Element Type. Either the P2MP FEC Element or the MP2MP FEC Element using the values defined for those FEC Elements when carried in the FEC TLV as defined in this document.

Len: Len FEC Type Info, two octets (=0x02).

AFI: Address Family, two-octet quantity containing a value from IANA's "Address Family Numbers" registry.

10. Security Considerations

The same security considerations apply as those for the base LDP specification, as described in [RFC5036].

The protocol specified in this document does not provide any authorization mechanism for controlling the set of LSRs that may join a given MP LSP. If such authorization is desirable, additional mechanisms, outside the scope of this document, are needed. Note that authorization policies cannot be implemented and/or configured solely at the root node of the LSP, because the root node does not learn the identities of all the leaf nodes.

11. IANA Considerations

Per this document, IANA has created 3 new registries.

1. "LDP MP Opaque Value Element basic type"

   The range is 0-255, with the following values allocated in this document:

   0: Reserved

   1: Generic LSP identifier

   255: Extended Type field is present in the following two bytes

   The allocation policy for this space is 'Standards Action with Early Allocation'.

2. "LDP MP Opaque Value Element extended type"

   The range is 0-65535, with the following allocation policies:

   0-32767: Standards Action with Early Allocation

   32768-65535: First Come, First Served

3. "LDP MP Status Value Element type"

   The range is 0-255, with the following values allocated in this document:

   0: Reserved

   1: MBB Status

   The allocation policy for this space is 'Standards Action with Early Allocation'.

The code point values listed below have been allocated by IANA through early allocation.

IANA allocated three new code points from the LDP registry "Forwarding Equivalence Class (FEC) Type Name Space". The values are:

P2MP FEC type - requested value 0x06

MP2MP-up FEC type - requested value 0x07

MP2MP-down FEC type - requested value 0x08

IANA assigned three new code points for new Capability Parameter TLVs from the LDP registry "TLV Type Name Space", corresponding to the advertisement of the P2MP, MP2MP, and MBB capabilities. The values are:

P2MP Capability Parameter - 0x0508

MP2MP Capability Parameter - 0x0509

MBB Capability Parameter - 0x050A

IANA assigned an LDP Status Code to indicate that an LDP MP Status TLV is following in the Notification message. The value assigned in the LDP registry "LDP Status Code Name Space" is:

LDP MP status - requested value 0x00000040

IANA assigned a new code point for an LDP MP Status TLV. The value assigned in the LDP registry "LDP TLV Type Name Space" is:

LDP MP Status TLV Type - requested value 0x096F

12. 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, Adrian Farrel, Thomas Morin and Ben Niven-Jenkins.

13. 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

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

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

Bob Thomas
Cisco Systems, Inc.
300 Beaver Brook Road
Boxborough, MA, 01719
EMail: bobthomas@alum.mit.edu

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

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

14. References

14.1. Normative References

[ITU.V42.1994]
International Telecommunications Union, "Error-correcting Procedures for DCEs Using Asynchronous-to-Synchronous Conversion", ITU-T Recommendation V.42, 1994. http://www.itu.int/rec/T-REC-V.42-200203-I

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

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

[RFC5036] Andersson, L., Ed., Minei, I., Ed., and B. Thomas, Ed., "LDP Specification", RFC 5036, October 2007.

[RFC5331] Aggarwal, R., Rekhter, Y., and E. Rosen, "MPLS Upstream Label Assignment and Context-Specific Label Space", RFC 5331, August 2008.

[RFC5561] Thomas, B., Raza, K., Aggarwal, S., Aggarwal, R., and JL. Le Roux, "LDP Capabilities", RFC 5561, July 2009.

[RFC5918] Asati, R., Minei, I., and B. Thomas, "Label Distribution Protocol (LDP) 'Typed Wildcard' Forward Equivalence Class (FEC)", RFC 5918, August 2010.

[RFC6389] Aggarwal, R. and JL. Le Roux, "MPLS Upstream Label Assignment for LDP", RFC 6389, September 2011.

14.2. Informative References

[ISO3309] International Organization for Standardization, "ISO Information Processing Systems - Data Communication - High-Level Data Link Control Procedure - Frame Structure", ISO 3309, 3rd Edition, October 1984.

[L3VPN-MCAST]
Rosen, E., Ed., and R. Aggarwal, Ed., "Multicast in MPLS/BGP IP VPNs", Work in Progress, January 2010.

[RFC3813] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Label Switching Router (LSR) Management Information Base (MIB)", RFC 3813, June 2004.

[RFC3815] Cucchiara, J., Sjostrand, H., and J. Luciani, "Definitions of Managed Objects for the Multiprotocol Label Switching (MPLS), Label Distribution Protocol (LDP)", RFC 3815, June 2004.

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

[RFC5332] Eckert, T., Rosen, E., Ed., Aggarwal, R., and Y. Rekhter, "MPLS Multicast Encapsulations", RFC 5332, August 2008.

[RFC6348] Le Roux, J., Ed., and T. Morin, Ed., "Requirements for Point-to-Multipoint Extensions to the Label Distribution Protocol", RFC 6348, September 2011.

Authors' Addresses

IJsbrand Wijnands (editor)
Cisco Systems, Inc.
De kleetlaan 6a
Diegem 1831
Belgium
EMail: ice@cisco.com

Ina Minei (editor)
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

Bob Thomas
300 Beaver Brook Road
Boxborough 01719
US
EMail: bobthomas@alum.mit.edu