

Network Working Group  
Internet-Draft  
Expires: April 17, 2006

I. Minei (Editor)  
K. Kompella  
Juniper Networks  
I. Wijnands (Editor)  
B. Thomas  
Cisco Systems, Inc.  
October 14, 2005

**Label Distribution Protocol Extensions for Point-to-Multipoint and  
Multipoint-to-Multipoint Label Switched Paths  
draft-minei-wijnands-mpls-ldp-p2mp-00**

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

This Internet-Draft will expire on April 17, 2006.

Copyright Notice

Copyright (C) The Internet Society (2005).

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 . . . . .](#) [3](#)
- [1.1. Conventions used in this document . . . . .](#) [3](#)
- [1.2. Terminology . . . . .](#) [3](#)
- [2. Setting up P2MP LSPs with LDP . . . . .](#) [4](#)
- [2.1. The P2MP FEC Element . . . . .](#) [4](#)
- [2.2. The LDP MP Opaque Value Element . . . . .](#) [6](#)
- [2.3. Using the P2MP FEC Element . . . . .](#) [6](#)
- [2.3.1. Label Map . . . . .](#) [7](#)
- [2.3.2. Label Withdraw . . . . .](#) [8](#)
- [3. Shared Trees . . . . .](#) [9](#)
- [4. Setting up MP2MP LSPs with LDP . . . . .](#) [10](#)
- [4.1. The MP2MP downstream and upstream FEC elements. . . . .](#) [10](#)
- [4.2. Using the MP2MP FEC elements . . . . .](#) [11](#)
- [4.2.1. MP2MP Label Map upstream and downstream . . . . .](#) [12](#)
- [4.2.2. MP2MP Label Withdraw . . . . .](#) [14](#)
- [5. Security Considerations . . . . .](#) [15](#)
- [6. IANA considerations . . . . .](#) [15](#)
- [7. Acknowledgments . . . . .](#) [15](#)
- [8. Contributing authors . . . . .](#) [16](#)
- [9. References . . . . .](#) [17](#)
- [9.1. Normative References . . . . .](#) [17](#)
- [9.2. Informative References . . . . .](#) [18](#)
- [Authors' Addresses . . . . .](#) [19](#)
- [Intellectual Property and Copyright Statements . . . . .](#) [20](#)



## **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 2.3.1](#)). 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 requirement draft [4] and other documents [6] and [7].

### **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 [4].

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.

MP2MP LSP: A LSP that has one or more Ingress LSRs and one unique Egress LSR.



MP2MP LSP: A 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.

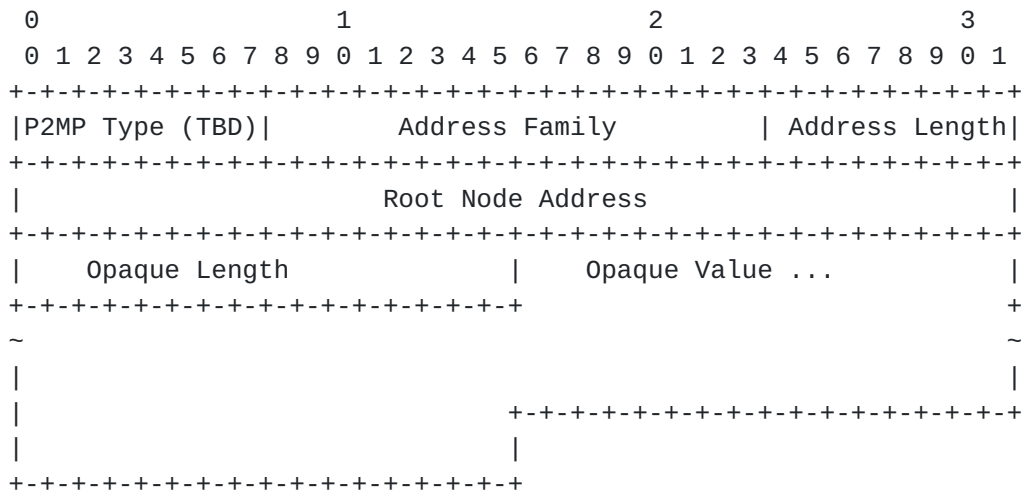
For the setup of a P2MP LSP with LDP, we define one new protocol entity, the P2MP FEC Element to be used in the FEC TLV. The description of the P2MP FEC Element follows.

### **2.1. The P2MP FEC Element**

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:



Type: The type of the P2MP FEC element is to be assigned by IANA, such that the U-bit is set (=1) and the F-bit is clear (=0). This ensures that an LSR which cannot process the P2MP FEC element, silently ignores it.

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.





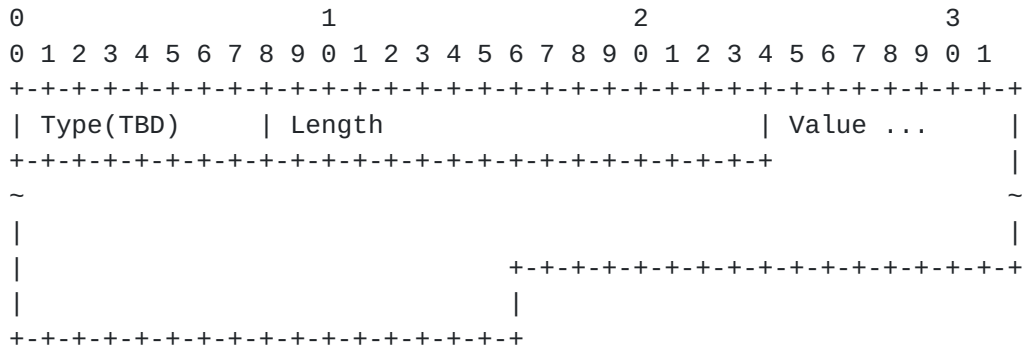
If the Address Length doesn't match the defined length for the Address Family, the receiver SHOULD abort processing the message containing the FEC Element, and send an "Unknown FEC" Notification message to its LDP peer signaling an error.

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

**2.2. 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:



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. 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  $L' \rightarrow \{ \langle I1, L1 \rangle \langle I2, L2 \rangle \dots, \langle In, Ln \rangle \}$  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.

### **2.3.1. 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, if there are several receivers for a P2MP LSP on a LAN, packets are replicated over the LAN. This may not be optimal; optimizing this case is for further study, see [8].

#### **2.3.1.1. 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>.



### **2.3.1.2. Leaf Operation**

A leaf node Z of P2MP LSP <X, Y> determines its upstream LSR U for <X, Y> as per [Section 2.3.1.1](#), allocates a label L, and sends a P2MP Label Map <X, Y, L> to U.

### **2.3.1.3. Transit Node operation**

Suppose a transit node Z receives a P2MP Label Map <X, Y, L> over interface I. 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. Z also determines its upstream LSR U for <X, Y> as per [Section 2.3.1.1](#), 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>}.

### **2.3.1.4. Root Node Operation**

Suppose the root node Z receives a P2MP Label Map <X, Y, L> over interface I. 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.

## **2.3.2. 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.

### **2.3.2.1. 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>.

### **2.3.2.2. 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 <X, Y, L> to its upstream for <X, Y>.

#### **2.3.2.3. 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).

#### **2.3.2.4. 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) [5], 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 4.2.1](#)

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 in the FEC TLV. The description of the MP2MP elements follow.

##### **4.1. The MP2MP downstream and upstream FEC elements.**

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 2.1](#). 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.

#### **[4.2.](#) 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  $\langle X, Y \rangle$  (or simply downstream  $\langle X, Y \rangle$ ): an MP2MP LSP downstream path with root node address X and opaque value Y.
2. MP2MP upstream LSP  $\langle X, Y, D \rangle$  (or simply upstream  $\langle X, Y, D \rangle$ ): a MP2MP LSP upstream path for downstream node D with root node address X and opaque value Y.
3. MP2MP downstream FEC element  $\langle X, Y \rangle$ : a FEC element with root node address X and opaque value Y used for a downstream MP2MP LSP.
4. MP2MP upstream FEC element  $\langle X, Y \rangle$ : a FEC element with root node address X and opaque value Y used for an upstream MP2MP LSP.
5. MP2MP Label Map downstream  $\langle X, Y, L \rangle$ : A Label Map message with a FEC TLV with a single MP2MP downstream FEC element  $\langle X, Y \rangle$  and label TLV with label L.
6. MP2MP Label Map upstream  $\langle X, Y, Lu \rangle$ : A Label Map message with a FEC TLV with a single MP2MP upstream FEC element  $\langle X, Y \rangle$  and label TLV with label Lu.
7. MP2MP Label Withdraw downstream  $\langle X, Y, L \rangle$ : a Label Withdraw message with a FEC TLV with a single MP2MP downstream FEC element  $\langle X, Y \rangle$  and label TLV with label L.
8. MP2MP Label Withdraw upstream  $\langle X, Y, Lu \rangle$ : 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 than 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.

#### **4.2.1. 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 [8].

##### **4.2.1.1. 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 2.3.1.1](#).

##### **4.2.1.2. 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.

##### **4.2.1.3. 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 4.2.1.1](#), 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.



#### **4.2.1.4. MP2MP transit node operation**

When node Z receives a MP2MP Label Map downstream  $\langle X, Y, L \rangle$  over interface I from node D it checks whether it has forwarding state for downstream  $\langle X, Y \rangle$ . 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  $\langle X, Y \rangle$  as per [Section 4.2.1.1](#), and sends a MP2MP Label Map downstream  $\langle X, Y, L' \rangle$  to U.

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

Node Z checks whether it already has forwarding state upstream  $\langle X, Y, D \rangle$ . 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  $\langle X, Y \rangle$ , 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  $\langle X, Y, L \rangle$ . Node Z determines the downstream MP2MP LSR as per [Section 4.2.1.2](#), and sends a MP2MP Label Map upstream  $\langle X, Y, Lu \rangle$  to node D.

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

#### **4.2.1.5. MP2MP root node operation**

##### **4.2.1.5.1. Root node is also a leaf**

Suppose root/leaf node Z receives a MP2MP Label Map downstream  $\langle X, Y, L \rangle$  over interface I from node D. Z checks whether it already has forwarding state downstream  $\langle X, Y \rangle$ . 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  $\langle X, Y \rangle$ , then Z will add the label push for L over interface I to it.

Node Z checks if it has forwarding state for upstream  $\langle X, Y, D \rangle$ . 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 4.2.1.2](#), 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.

#### **[4.2.1.5.2](#). Root node is not a leaf**

Suppose the root node Z receives a MP2MP Label Map downstream <X, Y, L> over interface I 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. 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 it was received from. Root node Z determines the downstream MP2MP LSR D as per [Section 4.2.1.2](#), 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.

#### **[4.2.2](#). 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.

##### **[4.2.2.1](#). 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.



#### **4.2.2.2. 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>.

#### **4.2.2.3. 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).

#### **4.2.2.4. MP2MP Upstream LSR change**

The procedure for changing the upstream LSR is the same as documented in [Section 2.3.2.4](#), except it is applied to MP2MP FECs, using the procedures described in [Section 4.2.1](#) through [Section 4.2.2.3](#).

### **5. Security Considerations**

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

### **6. IANA considerations**

This document creates a new name space (the LDP MP Opaque Value Element type) that is to be managed by IANA. Also, this document requires allocation of three new LDP FEC element types: the P2MP type, the MP2MP-up and the MP2MP-down types.

### **7. 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 and George Swallow.



## **8. 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  
AT&T  
200 Laurel Avenue, Room C2-3B35  
Middletown, NJ 07748  
US  
Email: luyuanfang@att.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: rthomas@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

## **9. References**

### **9.1. 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. and J. Postel, "Assigned Numbers", [RFC 1700](#), October 1994.
- [4] Roux, J., "Requirements for point-to-multipoint extensions to the Label Distribution Protocol", [draft-leroux-mpls-mp-ldp-reqs-01](#) (work in progress), July 2005.

## **9.2. Informative References**

- [5] Andersson, L. and E. Rosen, "Framework for Layer 2 Virtual Private Networks (L2VPNs)", [draft-ietf-l2vpn-l2-framework-05](#) (work in progress), June 2004.
- [6] Aggarwal, R., "Extensions to RSVP-TE for Point to Multipoint TE LSPs", [draft-ietf-mpls-rsvp-te-p2mp-02](#) (work in progress), July 2005.
- [7] Rosen, E. and R. Aggarwal, "Multicast in MPLS/BGP IP VPNs", [draft-ietf-l3vpn-2547bis-mcast-00](#) (work in progress), June 2005.
- [8] Aggarwal, R., "MPLS Upstream Label Assignment and Context Specific Label Space", [draft-raggarwa-mpls-upstream-label-00](#) (work in progress), April 2005.



Authors' Addresses

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

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

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

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

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

Email: [ice@cisco.com](mailto:ice@cisco.com)

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

Email: [rhthomas@cisco.com](mailto:rhthomas@cisco.com)



## Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

## Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.

