Internet-Draft
Expiration Date: May 1998

Nancy Feldman
IBM Corp.

Paul Doolan
Ennovate Networks

Andre Fredette
Bay Networks Inc

Loa Andersson
Ericsson

November 1997

# LDP Specification

<draft-feldman-ldp-spec-00.txt>

## Status of This Memo



## Abstract

An overview of Multi Protocol Label Switching (MPLS) is provided in [FRAMEWORK] and a proposed architecture in [ARCH]. A fundamental

concept in MPLS is that two Label Switching Routers (LSRs) must agree on the meaning of the labels used to forward traffic between and through them. This common understanding is achieved by using the Label Distribution Protocol (LDP) referenced in [FRAMEWORK] and [ARCH]. This document defines the LDP protocol.

Table of Contents

## 1. Protocol Overview

LDP is the set of procedures and messages by which one LSR informs another of the mappings between labels and Streams that it has made. Two LSRs which use an LDP to exchange label/Stream mapping information are known as "LDP Peers" with respect to that information and we speak of there being an "LDP Adjacency" between them. A single LDP adjacency allows each peer to learn the other's label mappings ie the protocol is bidirectional.

LDP provides a mechanism whereby LSRs continually indicate their presence in a network using advertisements which are sent as UDP packets to the LDP port at the 'all routers' group multicast address. When, perhaps in response to hearing an advertisement, one LSR decides to establish an adjacency with another it uses the initialization procedure of LDP. On succesful completion of this initialization procedure the two LSRs are LDP peers and may exchange label mappings.

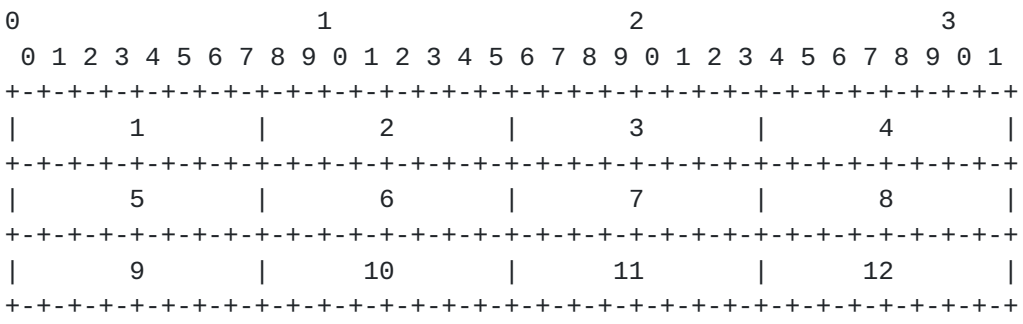
Note that this document is written with respect to unicast routing only. Multicast will be addressed in a future revision.

LDP messages are broken into two classes: those required to establish and maintain neighbor adjacencies, and those which deal with the advertisement of label mappings.

Correct operation of the label forwarding paradigm requires that forwarding peers agree on the 'meaning' of labels. This imposes certain requirements on the LDP including reliable and in order delivery of mappings (although there are circumstances when this second requirement could be relaxed). To satisfy these requirements LDP uses the TCP transport.

The convention used in this document is the same as that used in the documentation of the internet protocols [rfc1700] ie to express numbers in decimal and to picture data in "big-endian" order. Fields are described left to right, with the most significant octet on the left and the least significant octet on the right.

The order of transmission of the header and data described in this document is resolved to the octet level. Whenever a diagram shows a group of octets, the order of transmission of those octets is the normal order in which they are read in English. For example, in the following diagram the octets are transmitted in the order they are numbered.




Transmission Order of Bytes

Whenever an octet represents a numeric quantity the left most bit in the diagram is the high order or most significant bit. That is, the bit labeled 0 is the most significant bit.
Similarly, whenever a multi-octet field represents a numeric quantity the left most bit of the whole field is the most significant bit. When a multi-octet quantity is transmitted the most significant octet is transmitted first.

## 2. Local and Egress Control

Each LSR must be configured for local or egress control, to determine the behavior for the initial setup of LSPs.

When LSP control is done locally, each node may at any time pass label mappings to its neighbors for each stream recognized by that node. When the neighboring nodes recognize the same streams, they map incoming labels to received outgoing labels.

When LSP control is done via egress control, then only the egress node may initiate the transmission of label mappings. Non-egress nodes wait until they get a label from downstream for a recognized stream before mapping the stream and passing a corresponding label for the stream to upstream peers.

An LSR behaves as an egress (that is, initiating mappings) on a per stream basis. Thus, an LSR may be considered an egress for a particular set of streams, and a non-egress for others. An LSR is an egress LSR, with respect to a particular stream, under any of the following conditions:

1. The stream refers to the LSR itself (including one of its

directly

attached interfaces).

2. The stream is reachable via a next hop router that is outside the LSR switching infrastructure.

3. The stream is reachable by crossing a routing domain boundary, such as another area for OSPF summary net-works, or another autonomous system for OSPF AS externals and BGP routes [rfc1583] [rfc1771].

## 3. Selecting Streams

A stream is an aggregate of one or more flows, treated as one for the purpose of forwarding; that is, all aggregate flows use a single label.

Following are the currently defined streams. New streams types may be added as needed:

a) IPv4 address
This stream is a single a IP prefix. This identifier is used for host or CIDR prefixes [rfc1519]. This type results in each IP destination prefix sustaining its own LSP tree. It is recommended in environments where no aggregation information is provided by the routing protocols (such as RIP), or in networks where the number of destination prefixes is limited.

b) BGP Next Hop
This stream is the value in the BGP NEXT_HOP attribute. It may be the IP address of a BGP border router (enabling one LSP tree for all destinations reachable through the same egress point), or the address of an external BGP peer (enabling one LSP tree for all routes destinated to the same external peer). This identifier provides the maximum obtainable aggregation.

c) OSPF Router ID
This stream is the OSPF Router ID of the router that initiated the link state advertisement. This type allows aggregation of traffic on behalf of multiple datagram

protocols routed by OSPF.

d) OSPF Area Border Router
This stream is the OSPF Router ID of the border router. This identifier is used in OSPF external link advertisement with a non-zero forwarding address.

e) Explicit Path
This stream is an explicitly defined source-routed path. This information may be provided via configuration, or may be computed via a Dijkstra calculation for a certain metric (e.g. QoS, Tos), and may be used for point-to-point, point-to-multipoint, or multipoint-to-point LSPs. This type of stream may be initiated by an ingress or egress LSR.

f) Aggregate group
This stream is a list of prefixes that are to share a common egress point. This type is configured, and may be used when additional aggregation not provided by the routing protocols is required.

g) Flow
This stream contains information pertaining to a constant set of datagram information, such as port, dest-addr, src-addr, etc. This feature provides the user with the ability to use MPLS with no aggregation. This type of stream may be initiated by an ingress or egress LSR.

h) Multicast (S,G)
This stream is a unique (Source, Group) multicast pair. It creates one LSP tree per (S,G) pair. It is used by DVMRP and PIM-DM.

i) Multicast (G)
This stream is a unique multicast group on a multicast tree. It creates one switched path tree per group. It is used by PIM-SM.

# 4. LDP Messaging

## 4.1. Hello Message

The hello message is periodically transmitted to autodiscover LSR peers. The are transmitted as UDP packets to the LDP port at the 'all routers' group multicast address.

## 4.2. Protocol Message Overview

The protocol messages are constructed from a common header followed by one or more message 'objects'. The message object structure is of the form Type, Length, Value.

### 4.2.1. Adjacency Class Messages

Initialization
Transmitted after neighbor discovery, to exchange LSR characteristics, and establish a neighbor adjacency.

KeepAlive
Periodically transmitted to maintain a neighbor adjacency. It need be sent only when no other MPLS messages are transmitted within the KeepAlive timer interval.

Shutdown
Transmitted to ACTIVE neighbors, to terminate a neighbor adjacency.

### 4.2.2. Advertisement Class Messages

Mapping
Transmitted to establish an LSP by transmitting a mapping between a stream and a label, with associated characteristics.

Request
Transmitted to request a label mapping for a stream.

Withdraw
Transmitted to withdraw a previously allocated label.

Release
Transmitted to indicate a previously received label is no longer in use.

Query
Transmitted when diffusion is in use, to verify that a new routed path to the stream is loop free.

Acknowledge
An error transmitted on the receipt of an Advertisement Class message, or as a response to a query message

### 4.3. Advertisement Message

The Advertisement message is used by a downstream LSR distribute a label mapping for a stream to its LDP peers. If an LSR must distribute a mapping for a stream to multiple MPLS peers, it is a local matter whether it maps a single label to the stream, and distributes that mapping to all its peers, or whether it uses a different mapping for each of its peers.

It is the responsibility of the downstream LSR to keep track of the mappings which it has distributed, and to ensure that the upstream peer always has these mappings.

#### 4.3.1. Local Control

If an LSR is configured for local control, an advertisement is transmitted by an LSR to upstream peers upon any of the following conditions:

1. The LSR recognizes a new stream via the forwarding table.

2. The LSR receives a Request Advertisement from an upstream peer for a stream present in the LSR's forwarding table.

3. The next hop for a stream changes.

4. The next hop for a stream changes to another LDP peer.

5. The receipt of a mapping from the downstream next hop.

### 4.3.2. Egress Control

If an LSR is configured for egress control, an advertisement is transmitted by downstream LSRs upon any of the following conditions:

1. The LSR recognizes a new stream via the forwarding table, and is the egress for the stream.

2. The LSR receives a Request Advertisement from an upstream peer for a stream present in the LSR's forwarding table, and the LSR is the egress for that stream OR has a downstream mapping for that stream.

3. The next hop for a stream changes to another LDP peer.

4. The attributes of a mapping change.

5. The receipt of a mapping from the downstream next hop.

### 4.3.3. Label Use

An upstream LSR uses a mapping received from the downstream peer if that peer is the next hop for the stream, and a routed path loop is not detected via the LSR-path-vector object (see section 8.6.7). If diffusion is configured, a diffusion algorith may need to be performed before the label is used (see section 6). If, at the time the mapping is received, the downstream peer is NOT the LSR's Next Hop for the stream, or an object is determined to be in a loop, the LSR will not use of the mapping at that time.

## 4.4. Request Message

The Request message is used by the upstream LSR to explicitly request that the downstream LSR map and advertise a label for a stream.

An LSR transmits a Request message under any of the following conditions:

1. The LSR recognizes a new stream via the forwarding table, and the next hop is an ACTIVE MPLS peer.

2. The next hop to the stream changes, and one doesn't already have a mapping from that next hop for the given stream.

If a request cannot be satisfied by the downstream LSR, the requesting LSR may optionally choose to request again at a later time, or may wait for the mapping, assuming that the the downstream LSR will provide the mapping automatically when it is available.

### 4.5. Withdraw Message

A downstream LSR distributes a Withdraw message to upstream peers when it decides to break the mapping between a stream and a label. Note that if a downstream LSR peer becomes non-ACTIVE, all labels received from that peer are to be considered withdrawn.

An LSR transmits a Withdraw message under the following condition:

1. The LSR no longer recognizes a previously known stream.

2. Optionally, the LSR has unspliced an upstream label from the downstream label.

Note that a withdrawn label MAY NOT be reused until the upstream peer has acknowledged the withdraw via a Release message.

### 4.6. Release Message

An LSR transmits a Release message to a downstream peer when it is not using a label previously received from that peer.

An LSR transmits a Release message under any of the following conditions:

1. The downstream LSR which sent the label mapping is not the next hop for the mapped stream.

2. The downstream LSR which sent the label has ceased to be the next hop for a stream.

3. The LSR has received a Withdraw message for a previously received label.

Note that if an LSR is configured for "liberal mode", a release message will never be transmitted. In this case, the upstream LSR keeps each unused label, so that it can immediately be used later if the downstream peer becomes the next hop for the stream.

### 4.7. Acknowledge Message

An LSR transmits an acknowledge message if it cannot process a received Advertisement message, or in response to the receipt of a diffusion query message.

### 4.8. Query Message

The Query message is used with the diffusion algorithm to verify that new paths are loop-free before creating an LSP on the new path. See diffusion section ???

## 5. Loop Detection

All LSRs perform loop detection via the LSR-path-vector object contained within each label advertisement. Upon receiving such a message, the LSR performs loop detection by verifying that its unique router-id is not already present in the list. If a loop is detected, the LSR must transmit a NAK message to the sending node, and not install the mapping or propagate the message any further. In addition, if there is an upstream label spliced to the downstream label for the stream, the LSR must unsplice the labels. On those messages in which no loop is detected, the LSR must concatenate itself to the LSR-path-vector before propagating.

## 6. Loop Prevention via Diffusion

LSR diffusion support is a configurable option, which permits an LSR to verify that a new routed path is loop free before installing an LSP on that path. An LSR which supports diffusion does not splice an upstream label a new downstream label until it ensures that concatenation of the upstream path with the new downstream path will be loop free.

A LSR which detects a new nexthop for a stream transmits a query message containing its unique router id to each of its upstream peers. A node that receives such a query processes the query as follows:

- If the sending LSR not the correct next hop for the given stream, the receiving LSR responds with a positive acknowledge message, indicating that the sending LSR may change to the new

path.

- If the sending LSR is the correct next hop for the given stream, the receiving LSR performs loop detection via the LSR-path-vector.

- If a loop is detected, the receiving LSR responds with a negative "prune" acknowledgment, and unsplices all connections to the sending node, thereby pruning itself off of the tree.

- If a loop is not detected, the receiving node concatenates its unique router-id to the LSR-path-vector, and propagates the query message to its upstream neighbors.

- Each LSR which receives a query acknowledgement from its upstream neighbor in turn forwards the acknowledgement to the downstream LSR which sent the query advertisement.

- If an LSR doesn't receive an acknowledgment within a "reasonable" period of time, it "unsplices" its unsplice the upstream neighbor that has not responded, and responds with a negative "prune" acknowledgement.

- An LSR which receives a new query advertisement for a stream before it has received responses from all of its upstream neighbors for a previous query advertisement must concatentated the old and the new LSR-path-vector within the new query advertisement before propagating.

- The diffusion computation continues until each upstream path responds with an acknowledgment. An LSR that does not have any upstream MPLS neighbors must acknowledge the query advertisement.

The LSR which began the diffusion may splice its upstream label to the new downstream label only after receiving an acknowledge message from the upstream peer.

As LSR diffusion support is a configurable option, an LSRs which do not support diffusion will never originate a query advertisement. However, these LSRs must still recognize and process the query message, as described above.

## 7. Merging

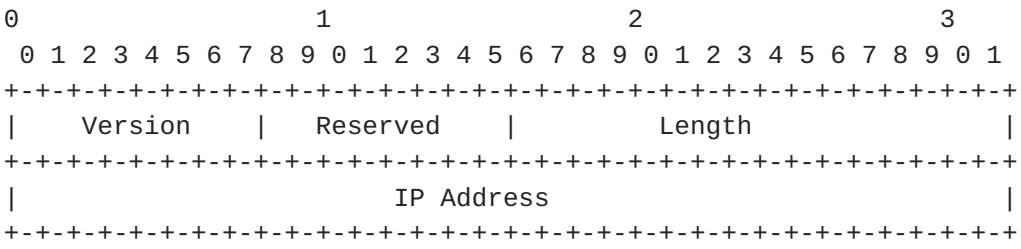
VC/VP merging, non-merging, and interoperability will be addressed in a future revision.

## 8. Specification

The hello mechanism is described first. Following that we define the structure of the common header and of the objects. We then provide definitions of the adjacency and advertisement messages and follow that with definitions for the objects that constitute those messages.

### 8.1. LDP Hello mechanism

Hello message




Version:
This one octet unsigned integer contains the version number of the protocol. This version of the specification specifies protocol Version = 0x01.

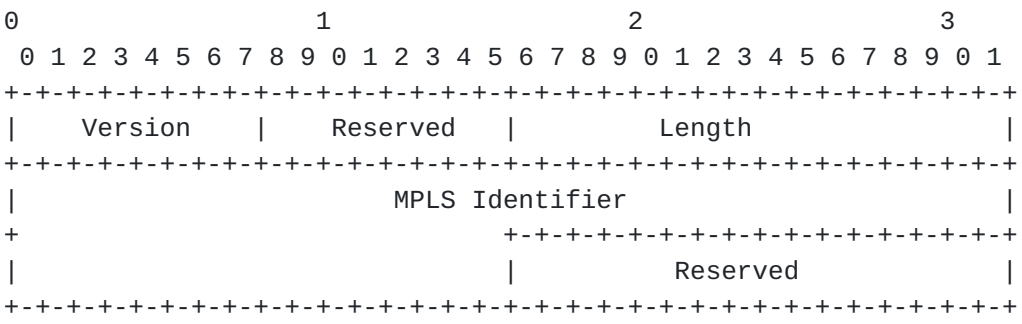
Reserved
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

Length:
This two octet integer specifies the total length of this PDU in bytes.

Network Address
This 4 octet integer contains the address of the LSR which originated the discovery message.

### 8.2. Common Header

All LDP PDUs, with the exception of the hello message, must begin with the following common header:




Version:
One octet unsigned integer containing the version number of the protocol. This version of the specification specifies protocol Version = 0x01.

Length:
Two octet integer specifying the total length of this PDU in bytes, including the common header.

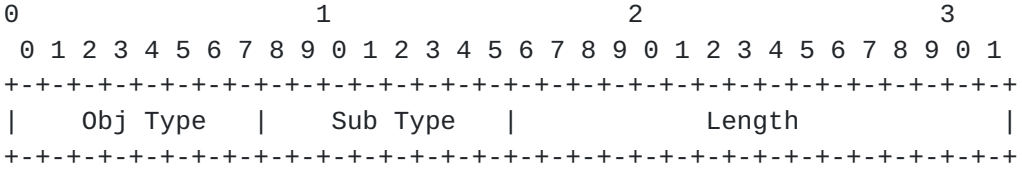
MsgClass
One octet integer defining the class of the MPLS message. This version of the specification defines:
Adjacency Class = 1
Advertisement Class = 2

MPLS Identifier:
Six octet unsigned integer containing a unique identifier for the LSR that generated the PDU. The value of this Identifier is determined on startup. The first four octets encode an IP address assigned to the LSR. The last two octets represent the 'instance' of MPLS on the LSR. A LSR with only one active MPLS session would supply the value zero in this field.

Res:
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

### 8.3. Object Header

All objects in an MPLS message must begin with the following object header. Objects must be placed back-to-back within the message, and must be padded to a word boundary.




Object Type

This one octet field specifies the type of the object following.

This version of the specification defines the following objects for adjacency class messages:

```
MSGTYPE_OBJECT          = 1
LSR_OBJECT              = 2
LABELRANGE_OBJECT       = 3
```

This version of the specification defines the following objects for advertisement class messages:

```
MSGTYPE_OBJECT          = 1
SMD_OBJECT              = 2
LABEL_OBJECT            = 3
COS_OBJECT              = 4
LSR_PATH_VECTOR_OBJECT  = 5
HOPCOUNT_OBJECT         = 6
MTU_OBJECT              = 7
STACK_OBJECT            = 8
ERROR_OBJECT            = 9
```

Sub Type

This one octet field specifies the subtype of this object. See each of the object definitions for a definition of the subtypes.

Length

This two octet unsigned integer specifies the length of the object, including this object header.

### 8.4. Adjacency Class Messages

The following notations show the objects that are valid with each Advertisement Class Message. Only one message may be contained within a single Adjacency Class PDU.

```
 Initialization Message:
   <Common Header>
   <MSGTYPE_OBJECT>
   <LSR_OBJECT>
   <LABELRANGE_OBJECT>

 KeepAlive Message:
   <Common Header>
   <MSGTYPE_OBJECT>

Shutdown Message:
   <Common Header>
   <MSGTYPE_OBJECT>
```

### 8.5. Advertisement Class

All Advertisement Class PDUs begin with a common header:
<Common Header>

The following notations show the objects that are valid with each Advertisement Class Message. Multiple messages may be contained within a single Advertisement Class PDU.

```
Mapping Message:
   <MSGTYPE_OBJECT>
   <SMD>
      [ <SMD> ] ...

   <SMD> := <SMD_OBJECT> <LABEL> [MTU_OBJECT] [STACK_OBJECT]
   <LABEL> := <LABEL_OBJECT> <LSR_PATH_VECTOR_OBJECT> [COS_OBJECT]
                      [HOPCOUNT_OBJECT]

Request Message:
   <MSGTYPE_OBJECT>
   <SMD_OBJECT>
      [SMD_OBJECT] ...

Withdraw Message:
   <MSGTYPE_OBJECT>
```

```
      <SMD_OBJECT>
      <LABEL_OBJECT>
         [<SMD_OBJECT>
          <LABEL_OBJECT>] ...

   Release Message:
      <MSGTYPE_OBJECT>
      <SMD_OBJECT>
      <LABEL_OBJECT>
         [<SMD_OBJECT>
          <LABEL_OBJECT>] ...

   Query Message:
      <MSGTYPE_OBJECT>
      <SMD_OBJECT>
      <LSR_PATH_VECTOR_OBJECT>

   Acknowledge Message:
      <MSGTYPE_OBJECT>
      <SMD_OBJECT>
      [<LABEL_OBJECT>]
      <ERROR_OBJECT>
         [<SMD_OBJECT>
          [<LABEL_OBJECT>]
          <ERROR_OBJECT>] ...
```

## 8.6. LDP Object definitions

### 8.6.1. MsgType Object

**MsgClass = Adjacency or Advertisement**

ObjType = 1

```
   SubType = 1  Initialization Message
             2  KeepAlive Message
             3  ShutDown Message
             4  Mapping Message
             5  Withdraw Message
             6  Request Message
             7  Release Message
             8  Query Message
             9  Acknowledge Message
```

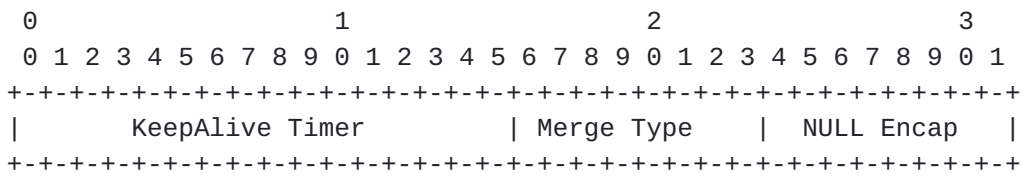
The MsgType object identifies the type of message to be processed. This object has no additional fields. The length is a fixed 4 bytes.

### 8.6.2. LSR Object

MsgClass = Adjacency
ObjType = 2

The LSR object exchanges LSR characteristics with peers at initialization.

SubType = 1  ATM




KeepAlive Timer
Two octet unsigned non zero integer that indicates the number of seconds that the peer initiating the connection proposes for the value of the KeepAlive Interval. The receiving LSR MUST MUST calculate the value of the KeepAlive Timer by using the smaller of its configured KEEPALIVE_TIME and the KEEPALIVE_TIME received in the PDU. The value chosen for KEEPALIVE_TIME indicates the maximum number of seconds that may elapse between the receipt of successive PDUs from the LSR peer. The Hold Timer is reset each time a PDU arrives.

MergeType
One octet integer specifying the switch merge capabilities. The following values are supported in this version of the specification:

Non Merge = 0
VP Merge = 1
VC Merge = 2
VP & VC Merge = 3

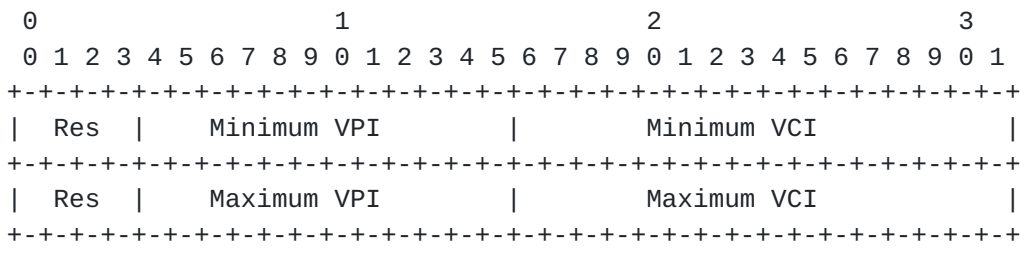
NULL Encap
One octet parameter set to non-zero when the LSR supports the null encapsulation of [rfc1483] for its data VCs. In this case IP packets are carried directly inside AAL5 frames.

### 8.6.3. LabelRange Object

MsgClass = Adjacency
ObjType = 3

The LabelRange object contains the label range supported by the transmitting LSR. A receiving LSR MUST calculate the intersection between the received range and its own supported label range. The intersection is the range in which the LSR may allocate and accept labels. LSRs may NOT establish an adjacency with neighbors whose intersection range is NULL.

SubType = 1 Default




Res
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

Minimum VPI (12 bits)
This 12 bit field specifies the lower bound of a block of Virtual Path Identifiers that is supported on the originating switch. If the VPI is less than 12-bits it should be right justified in this field and preceding bits should be set to 0.

Minimum VCI (16 bits)
This 16 bit field specifies the lower bound of a block of Virtual Connection Identifiers that is supported on the originating switch. If the VCI is less than 16-bits it should be right justified in this field and preceding bits should be set to 0.

Maximum VPI (12 bits)
This 12 bit field specifies the upper bound of a block of Virtual Path Identifiers that is supported on the originating switch. If the VPI is less than 12-bits it should be right justified in this field and preceding bits should be set to 0.

Maximum VCI (16 bits)

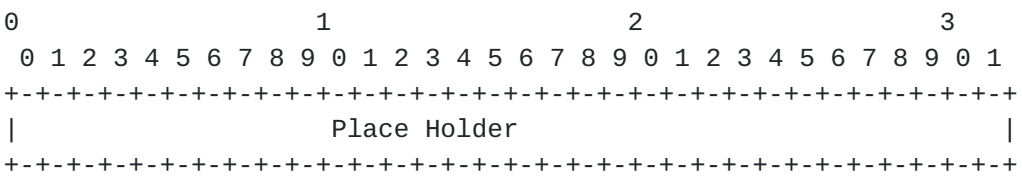
This 16 bit field specifies the upper bound of a block of Virtual Connection Identifiers that is supported on the originating switch. If the VCI is less than 16-bits it should be right justified in this field and preceding bits should be set to 0.

### 8.6.4. Stream Member Descriptor (SMD) Object

MsgClass = Advertisement
ObjType = 2

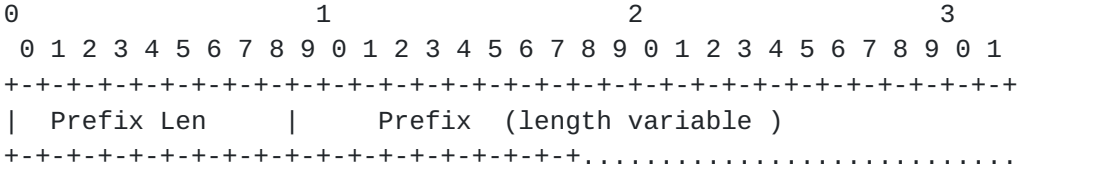
This mandatory object specifies the streams for which LSPs are created. All objects following an SMD_OBJECT are associated with the SMD, until the next SMD_OBJECT or MSGTYPE_OBJECT.

SubType = 1 Wild Card




Place Holder
Four octet integer set to all 0s. This field is used in association with a Label Object, to indicate ALL SMDs associated with the given label are to be processed.

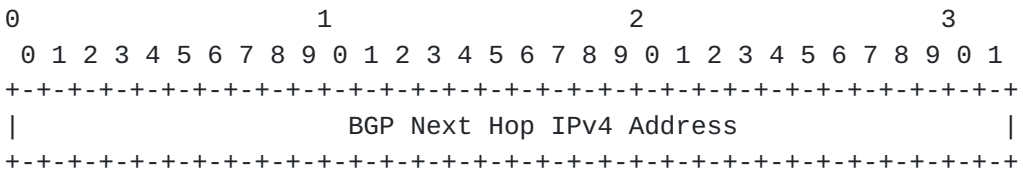
SubType = 2 Network Address




Prefix Len
One octet unsigned integer containing the length in bits of the address prefix that follows.

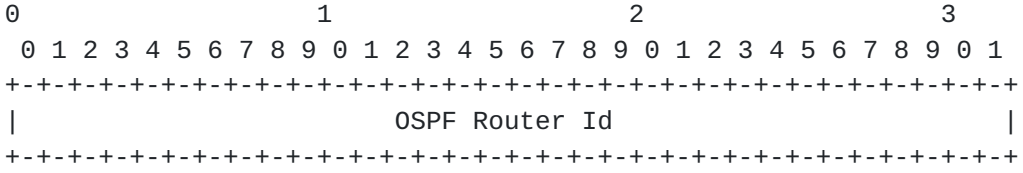
Prefix:
A variable length field containing an address prefix whose length, in bits, was specified in the previous (Prefix Len) field. A Prefix must be padded with sufficient trailing zero bits to cause the end of the field to fall on a word boundary.

SubType = 3 BGP Next Hop



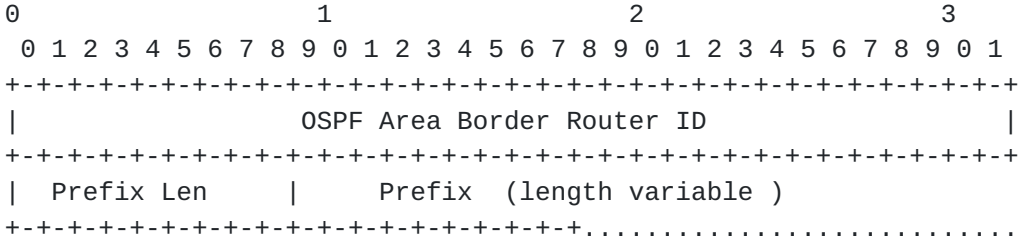

BGP Next Hop
   Four octet IPv4 address of the BGP Next Hop router.

SubType = 4  OSPF Router Id




OSPF Router ID
   Four octet router identifier of an OSPF node.

SubType = 5  OSPF Area Border Router

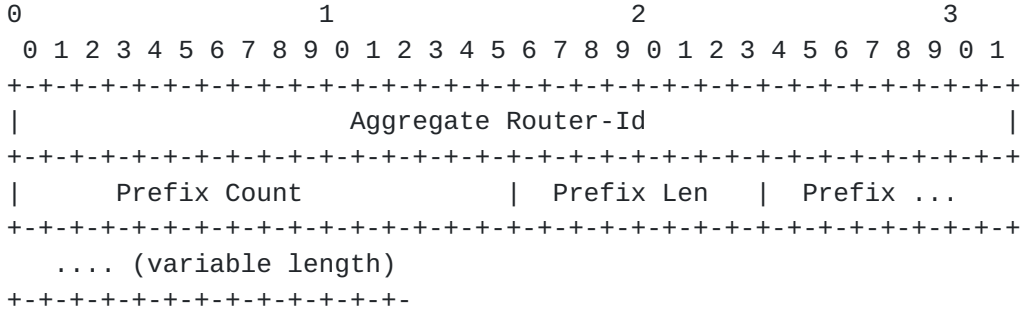



OSPF Area Border Router ID
   Four octet router identifier of the OSPF ABR node.

Prefix Len
   One octet unsigned integer containing the length in bits of the
   address prefix that follows.

Prefix:
   A variable length field containing an address prefix whose length,
   in bits, was specified in the previous (Prefix Len) field. A Prefix
   must be padded with sufficient trailing zero bits  to cause the end
   of the field to fall on a word boundary.

SubType = 6  Aggregation list



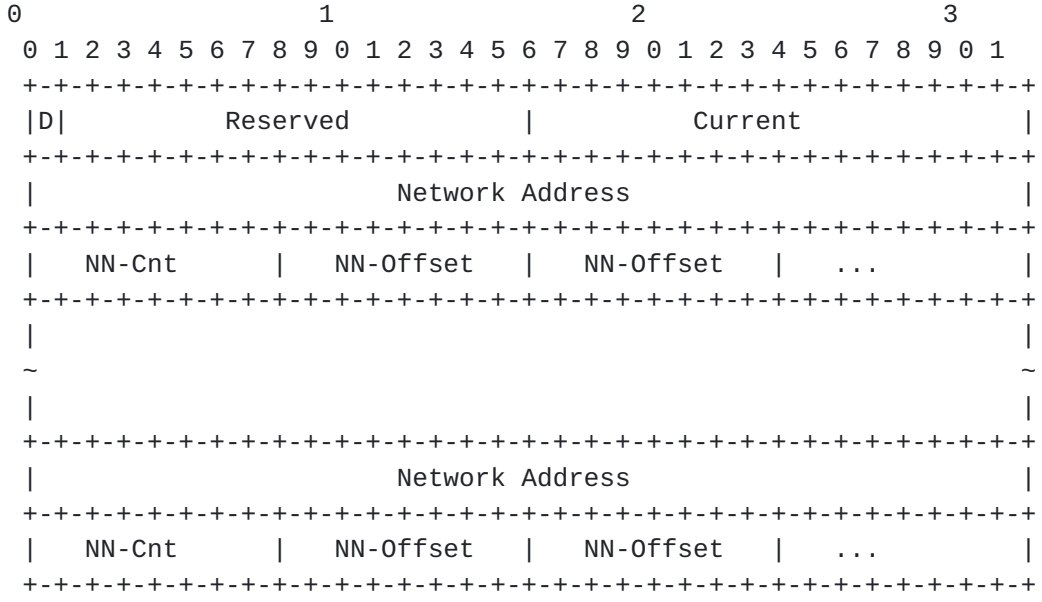

Aggregate Router-Id
Four octet unique identifier on an LSR which is initiating the aggregation.

Prefix Count
Two octet number of address prefixes in this object, which comprise the set of prefixes that are to be aggregated together.

Prefix Len
One octet unsigned integer containing the length in bits of the address prefix that follows.

Prefix:
A variable length field containing an address prefix whose length, in bits, was specified in the previous (Prefix Len) field. The last prefix in the list must be padded with sufficient trailing zero bits to cause the end of the field to fall on a word boundary.

SubType = 7  Explicit Route




D-bit (Direction)
One-bit field indicating the direction of the LSP. Field is set to 0 when initiated by an ingress LSR. Field is set to 1 when initiated by an egress LSR. All explicit routes must have an end-to-end acknowledgment. In the case of ingress initiated explicit, the egress LSR is responsible for initiating the assignment of labels with a mapping advertisement. In the case of egress initiated explicit, the ingress LSR is responsible for initiating an acknowledge message back to the egress. Note that each LSR receiving an explicit object must keep explicit path state, in order to process the label mapping or acknowledgment when it arrives.

Reserved
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

Current
Two-octet pointer to the current network address location in the explicit path.

Network Address
Four-octet network address of a node in the LSP. Note this value may not be word aligned.

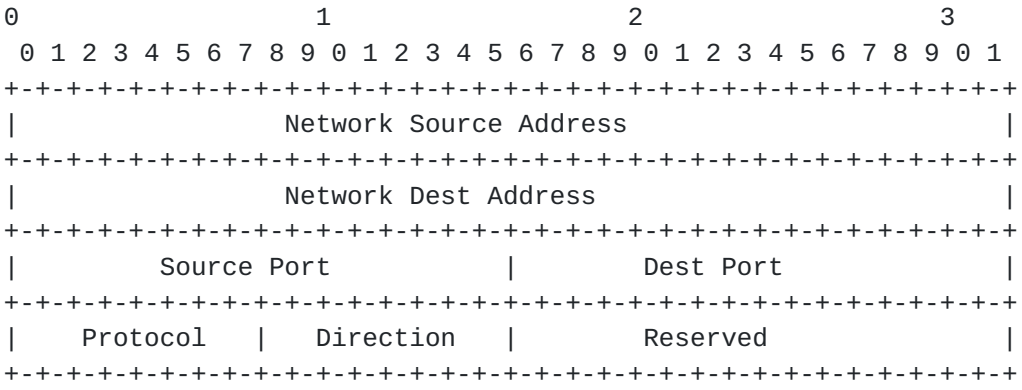
NN-Cnt (Next-Node Count)
One-octet field containing the number of next-nodes corresponding to the network address, for which the explicit message need be forwarded to. If this value is greater than 1, a mpt-to-pt or

pt-to-mpt explicit path is created. A value of 0 indicates the end of the list.

NN-Offset (Next-Node Offset)
One-octet relative offset from the corresponding network address to the location within the object of a next-hop network address.

SubType = 8 Hop-by-Hop Tunnel

SubType = 9 Flow




Network Source Address
Four octet source Network address.

Network Destination Address
Four octet destination Network address.

Source Port
Two octet source port.

Destination Port
Two octet destination port.

Protocol
Protocol type.

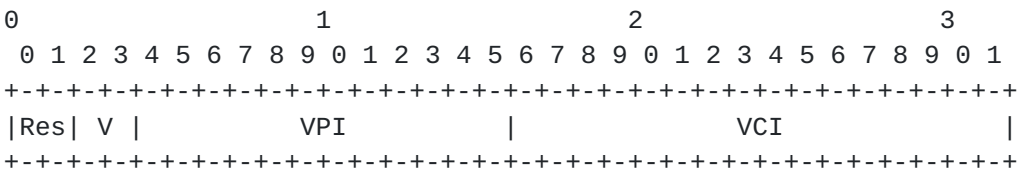
Direction
One octet indicating the direction of the LSP. Field is set to 1 on Downstream; field is set to 2 on Upstream.

### 8.6.5. Label Object

MsgClass = Advertisement
ObjType = 3

This object specifies a link-layer label associated with an SMD.

SubType = 1 ATM




Res
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

V-bits
Two-bit switching indicator. If V-bits is 00, both the VPI and VCI are significant. If V-bits is 01, only the VPI field is significant. If V-bit is 10, only the VCI is significant.

VPI (12 bits)
Virtual Path Identifier. If VPI is less than 12-bits it should be right justified in this field and preceding bits should be set to 0. If both the VPI and the VCI are 0, the receiver allocates the label.

VCI (16 bits)
Virtual Connection Identifier. If the VCI is less than 16-bits, it should be right justified in the field and the preceding bits must be set to 0. If Virtual Path switching is indicated in the V-bits field, then this field must be ignored by the receiver and set to 0 by the sender. If both the VPI and the VCI are 0, the receiver allocates the label.

SubType = 2 Shim

### 8.6.6. Class-of-Service Object

MsgClass = Advertisement
ObjType = 4

This object specifies a class of service that is to be associated with a label.

SubType = 1  Default

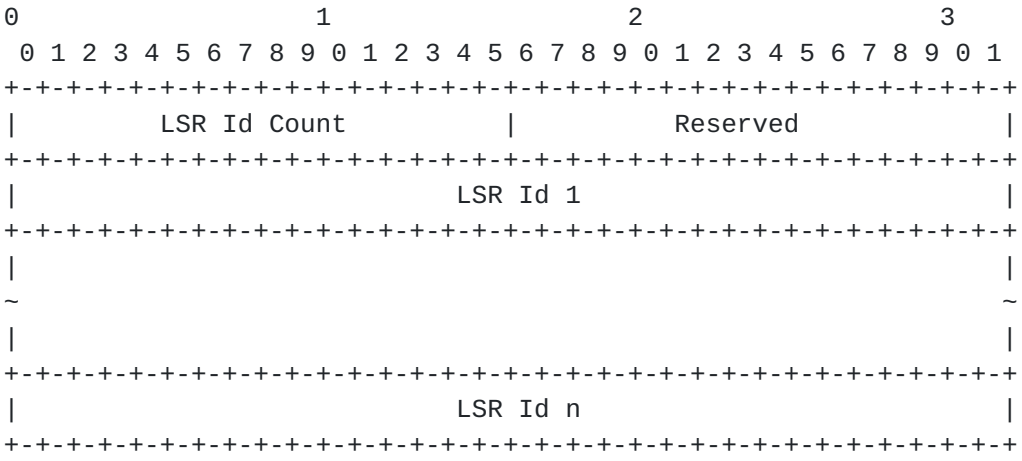
This object to be specified in a future revision.

### 8.6.7. LSR Path Vector Object

MsgClass = Advertisement
ObjType = 5

This mandatory object contains the LSR path the advertisement has traversed. Any LSR that finds its own unique LSR Id in the received LSR Path Vector is determined to be in a loop.

SubType = 1  Default




Reserved
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

Router Id Count
Two octet number of Router Identifiers in this object.

Router Id 1 to n-1
A series of LSR Identifiers indicating the path that the mapping message has traversed. Each LSR Id must be unique within the LSR network.

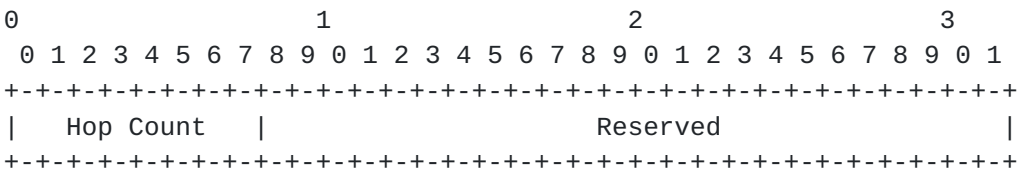
Router Id n
LSR Identifier of the router that sent the current message. Each LSR receiving this object MUST append its own unique LSR Id to the object before forwarding the object.

### 8.6.8. Hop Count Object

MsgClass = Advertisement
ObjType = 6

This object calculates the number of LSR hops along a LSP.

SubType = 1 Default




Hop Count
The number of LSR hops along the LSP. It is incremented by one at each LSR forwarding the object.

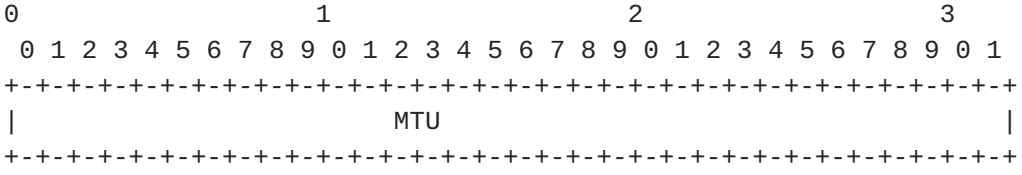
Reserved
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

### 8.6.9. MTU Object

MsgClass = Advertisement
ObjType = 7

This object identifies the MTU along an LSP. An LSR initiating data along the LSP path may not transmit data larger than the given MTU.

SubType = 1 Default



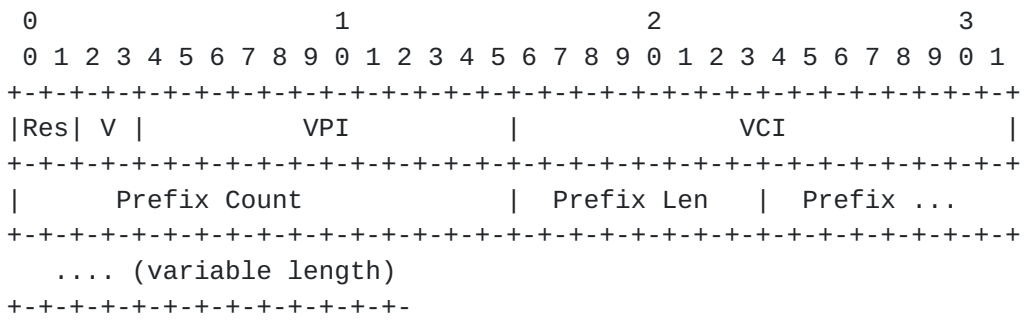

MTU
The maximum transmission unit for a given LSP. Any LSR receiving the MTU object must compare the given value with its own MTU on the receiving link. The minimum of these values is the MTU to be associated with the LSP and forwarded in the object.

### 8.6.10. Stack Object

MsgClass = Advertisement
ObjType = 8

This object contains a stacked label and a list of prefixes which are to use the stacked label. This enables a deaggregating LSR to avoid L3 forwarding, by removing an incoming L2 header with the label given in a LABEL_OBJECT, and then continuing switching on the given stack label.

SubType = 1 ATM




Res
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

V-bits
Two-bit switching indicator. If V-bits is 00, both the VPI and VCI are significant. If V-bits is 01, only the VPI field is significant. If V-bit is 10, only the VCI is significant.

VPI (12 bits)
Virtual Path Identifier. If VPI is less than 12-bits it should be right justified in this field and preceding bits should be set to 0. If both the VPI and the VCI are 0, the receiver allocates the label.

VCI (16 bits)
Virtual Connection Identifier. If the VCI is less than 16-bits, it should be right justified in the field and the preceding bits must be set to 0. If Virtual Path switching is indicated in the V-bits field, then this field must be ignored by the receiver and set to 0 by the sender. If both the VPI and the VCI are 0, the receiver allocates the label.

Prefix Count
Two octet number of address prefixes in this object, which

comprise the set of prefixes that are use the given label.

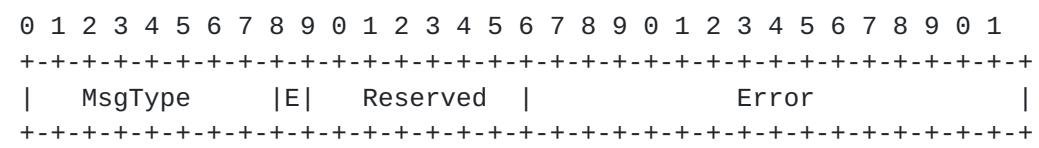
Prefix Len
One octet unsigned integer containing the length in bits of the address prefix that follows.

Prefix:
A variable length field containing an address prefix whose length, in bits, was specified in the previous (Prefix Len) field. The last prefix in the list must be padded with sufficient trailing zero bits to cause the end of the field to fall on a word boundary.

### 8.6.11. Error Object

MsgClass = Advertisement
ObjType = 9

This object contains an error code associated with an SMD.




MsgType
Two-octet value indicating the Advertisement message type being acknowledged.

E-bit (End-to-End Ack)
One-bit value indicating whether the error is to be forwarded to the associated SMDs next hop, or next node in the case of explicit paths. A value of 0 indicates DO NOT FORWARD; A value of 1 indicates FORWARD.

Reserved
This field is reserved. It must be set to zero on transmission and must be ignored on receipt.

Error
An error code. A value of 0 indicates no error.
Error values TBA.

## 9. Intellectual Property Considerations

## 10. Acknowledgments

The ideas and text in this document have been collected from a number of sources. We would like to thank Rick Boivie, Ross Callon, Eric Rosen, Yakov Rekhter, and Arun Viswanathan.

## 11. References

[FRAMEWORK] Callon et al, "A Framework for Multiprotocol Label Switching" draft-ietf-mpls-framework-01.txt, July 1997

[ARCH] Rosen et al, "A Proposed Architecture for MPLS" draft-ietf-mpls-arch-00.txt, August 1997

[rfc1700] J. Reynolds, J. Postel, "Assigned Numbers", RFC 1700, ISI, October 1994

[rfc1583] J. Moy, "OSPF Version 2", RFC 1583, Proteon Inc, March 1994

[rfc1771] Y. Rekhter, T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, IBM Corp, Cisco Systems, March 1995

[rfc1519] V. Fuller, T. Li, J. Yu, K. Varadhan, "Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy", RFC 1519, BARRNET, Cisco Systems, MERIT, OARnet, September, 1993

[rfc1483] J. Heinanen, "Multiprotocol Encapsulation over ATM Adaptation Layer 5", RFC 1483, Telecom Finland, July 1993

## 12. Author Information

Loa Andersson
Ericsson
Phone: + 46 8 719 52 67
email: loa.andersson@etx.ericsson.se

Paul Doolan
Ennovate Networks
330 Codman Hill Rd
Marlborough MA 01719
Phone: 978 263-2002
email: pdoolan@cisco.com

Nancy Feldman
IBM Corp.
17 Skyline Drive
Hawthorne NY 10532
Phone: 914-784-3254
email: nkf@vnet.ibm.com

Andre Fredette
Bay Networks Inc
3 Federal Street
Billerica, MA 01821
Phone: 508-916-8524
email: fredette@baynetworks.com