

Network Working Group  
Internet Draft  
Expiration Date: December 2000

Peter Ashwood-Smith  
Yanhe Fan  
Nortel Networks Corp.

Ayan Banerjee  
John Drake  
Jonathan P. Lang  
Calient Networks

Lou Berger  
LabN Consulting, LLC

Greg Bernstein  
Ciena Corporation

Kireeti Kompella  
Juniper Networks, Inc.

Eric Mannie  
GTS Network Services

Bala Rajagopalan  
Debanjan Saha  
Z. Bo Tang  
Tellium, Inc.

Yakov Rekhter  
cisco Systems

Vishal Sharma  
Tellabs

June 2000

## Generalized MPLS - Signaling Functional Description

[draft-ashwood-generalized-mpls-signaling-00.txt](#)

### Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#). 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."

To view the current status of any Internet-Draft, please check the "1id-abstracts.txt" listing contained in an Internet-Drafts Shadow Directory, see <http://www.ietf.org/shadow.html>.

#### Abstract

This document describes extensions to MPLS signaling required to support Generalized MPLS. Generalized MPLS extends MPLS to encompass time-division (e.g. SONET ADMs), wavelength (optical lambdas) and spatial switching (e.g. incoming port or fiber to outgoing port or fiber). This document presents a functional description of the extensions. Protocol specific formats and mechanisms are currently included in this draft but are expected to be split out into separate, per protocol documents.



## Contents

<a href="#">1</a>	Introduction .....	<a href="#">4</a>
<a href="#">2</a>	Overview .....	<a href="#">5</a>
<a href="#">3</a>	Label Related Formats .....	<a href="#">7</a>
<a href="#">3.1</a>	Generalized Label Request .....	<a href="#">7</a>
<a href="#">3.2</a>	Generalized Label .....	<a href="#">11</a>
<a href="#">3.3</a>	Waveband Switching .....	<a href="#">16</a>
<a href="#">3.4</a>	Suggested Label .....	<a href="#">18</a>
<a href="#">3.5</a>	Label Set .....	<a href="#">19</a>
<a href="#">4</a>	Bidirectional LSPs .....	<a href="#">22</a>
<a href="#">4.1</a>	Required Information .....	<a href="#">23</a>
<a href="#">4.2</a>	Procedures .....	<a href="#">23</a>
<a href="#">4.3</a>	Contention Resolution .....	<a href="#">24</a>
<a href="#">5</a>	Notification .....	<a href="#">26</a>
<a href="#">5.1</a>	Notify Message .....	<a href="#">26</a>
<a href="#">5.2</a>	Non-Adjacent Message Bundling .....	<a href="#">27</a>
<a href="#">6</a>	Egress Control .....	<a href="#">28</a>
<a href="#">6.1</a>	Required Information .....	<a href="#">28</a>
<a href="#">6.2</a>	Procedures .....	<a href="#">30</a>
<a href="#">7</a>	Acknowledgments .....	<a href="#">30</a>
<a href="#">8</a>	Security Considerations .....	<a href="#">31</a>
<a href="#">9</a>	References .....	<a href="#">31</a>
<a href="#">10</a>	Authors' Addresses .....	<a href="#">32</a>

Changes from previous version:

- o First revision.

## **1. Introduction**

The Multiprotocol Label Switching (MPLS) architecture [[MPLS-ARCH](#)] has been defined to support the forwarding of data based on a label. In this architecture, Label Switching Routers (LSRs) were assumed to have a forwarding plane that is capable of (a) recognizing either packet or cell boundaries, and (b) being able to process either packet headers (for LSRs capable of recognizing packet boundaries) or cell headers (for LSRs capable of recognizing cell boundaries).

The original architecture has recently been extended to include LSRs whose forwarding plane recognizes neither packet, nor cell boundaries, and therefore, can't forward data based on the information carried in either packet or cell headers. Specifically, such LSRs include devices where the forwarding decision is based on time slots, wavelengths, or physical ports.

Given the above, LSRs, or more precisely interfaces on LSRs, can be subdivided into the following classes:

1. Interfaces that recognize packet/cell boundaries and can forward data based on the content of the packet/cell header. Examples include interfaces on routers that forward data based on the content of the "shim" header, interfaces on ATM-LSRs that forward data based on the ATM VPI/VCI. Such interfaces are referred to as Packet-Switch Capable (PSC).
2. Interfaces that forward data based on the data's time slot in a repeating cycle. An example of such an interface is an interface on a SONET Cross-Connect. Such interfaces are referred to as Time-Division Multiplex Capable (TDM).
3. Interfaces that forward data based on the wavelength on which the data is received. An example of such an interface is an interface on an Optical Cross-Connect that can operate at the level of an individual wavelength. Such interfaces are referred to as Lambda Switch Capable (LSC).
4. Interfaces that forward data based on a position of the data in the real world physical spaces. An example of such an interface is an interface on an Optical Cross-Connect that can operate at the level of a single (or multiple) fibers. Such interfaces are referred to as Fiber-Switch Capable (FSC).



Using the concept of nested LSPs (by using label stack) allows the system to scale by building a forwarding hierarchy. At the top of this hierarchy are FSC interfaces, followed by LSC interfaces, followed by TDM interfaces, followed by PSC interfaces. This way, an LSP that starts and ends on a PSC interface can be nested (together with other LSPs) into an LSP that starts and ends on a TDM interface. This LSP, in turn, can be nested (together with other LSPs) into an LSP that starts and ends on a LSC interface, which in turn can be nested (together with other LSPs) into an LSP that starts and ends on a FSC interface. See [[MPLS-HIERARCHY](#)] for more information on LSP hierarchies.

The establishment of LSPs that span only the first class of interfaces is defined in the [[LDP](#), [CR-LDP](#), [RSVP-TE](#)]. This document presents the extensions needed to support all four classes of interfaces.

This document currently includes data formats for both CR-LDP and RSVP-TE extensions. A future version of this document is expected to move these protocol specific formats to per protocol drafts.

## **2. Overview**

Generalized MPLS differs from traditional MPLS in that it supports multiple types of switching, i.e., the addition of support for TDM, lambda, and fiber (port) switching. The support for the additional types of switching has driven generalized MPLS to extend certain base functions of traditional MPLS and, in some cases, to add functionality. These changes and additions impact basic LSP properties, how labels are requested and communicated, the unidirectional nature of LSPs, how errors are propagated, and information provided for synchronizing the ingress and egress.

In traditional MPLS Traffic Engineering, links traversed by an LSP can include an intermix of links with heterogeneous label encodings. For example, an LSP may span links between routers, links between routers and ATM-LSRs, and links between ATM-LSRs. Generalized MPLS extends this by including links where the label is encoded as a time slot, or a wavelength, or a position in the real world physical space. Just like with traditional MPLS TE, where not all LSRs are capable of recognizing (IP) packet boundaries (e.g., an ATM-LSR) in their forwarding plane, generalized MPLS includes support for LSRs that can't recognize (IP) packet boundaries in their forwarding plane. In traditional MPLS TE an LSP that carries IP has to start and end on a router. Generalized MPLS extends this by requiring an LSP to start and end on similar type of LSRs. Also, in generalized MPLS the type of a payload that can be carried by an LSP is extended





to allow such payloads as SONET/SDH, or 1 or 10Gb Ethernet. These changes from traditional MPLS are reflected in how labels are requested and communicated in generalized MPLS, see Sections [3.1](#) and [3.2](#). A special case of Lambda switching, called Waveband switching is also described in [Section 3.3](#).

Another basic difference between traditional and non-PSC types of generalized MPLS LSPs, is that bandwidth allocation for an LSP can be performed only in discrete units, see [Section 3.1.1](#). There are also likely to be (much) fewer labels on non-PSC links than on PSC links. Note that the use of Forwarding Adjacencies (FA), see [MPLS-HIERARCHY], provides a mechanism that may improve bandwidth utilization, when bandwidth allocation can be performed only in discrete units, as well as a mechanism to aggregate forwarding state, thus allowing the number of required labels to be reduced.

Generalized MPLS allows for a label to be suggested by an upstream node, see [Section 3.4](#). This suggestion may be overridden by a downstream node but, in some cases, at the cost of higher LSP setup time. The suggested label is valuable when establishing LSPs through certain kinds of optical equipment where there may be a lengthy (in electrical terms) delay in configuring the switching fabric. For example micro mirrors may have to be elevated or moved, and this physical motion and subsequent damping takes time. If the labels and hence switching fabric are configured in the reverse direction (the norm) the MAPPING/Resv message may need to be delayed by 10's of milliseconds per hop in order to establish a usable forwarding path.

Generalized MPLS extends on the notion of restricting the range of labels that may be selected by a downstream node, see [Section 3.5](#). In generalized MPLS, an ingress or other upstream node may restrict the labels that may be used by an LSP along either a single hop or along the whole LSP path. This feature is driven from the optical domain where there are cases where wavelengths used by the path must be restricted either to a small subset of possible wavelengths, or to one specific wavelength. This requirement occurs because some equipment may only be able to generate a small set of the wavelengths that intermediate equipment may be able to switch, or because intermediate equipment may not be able to switch a wavelength at all, being only able to redirect it to a different fiber.

While traditional traffic engineered MPLS (and even LDP) are unidirectional, generalized MPLS supports the establishment of bidirectional LSPs, see [Section 4](#). The need for bidirectional LSPs come from non-PSC applications. There are multiple reasons why such LSPs are needed, particularly possible resource contention when allocating reciprocal LSPs via separate signaling sessions, and simplifying failure restoration procedures in the non-PSC case.



Bidirectional LSPs also have the benefit of lower setup latency and lower number of messages required during setup.

Other features supported by generalized MPLS are rapid failure notification, see [Section 5](#), and termination of an LSP on a specific egress port, see [Section 6](#).

### **3. Label Related Formats**

To deal with the widening scope of MPLS into the optical and time domain, several new forms of "label" are required. These new forms of label are collectively referred to as a "generalized label". A generalized label contains enough information to allow the receiving node to program its cross connect, regardless of the type of this cross connect, such as the ingress segments of the path are properly joined. This section defines a generalized label request, a generalized label, support for waveband switching, suggested label and label sets.

Note that since the nodes sending and receiving the new form of label know what kinds of link they are using, the generalized label does not contain a type field, instead the nodes are expected to know from context what type of label to expect.

#### **[3.1. Generalized Label Request](#)**

The Generalized Label Request supports communication of characteristics required to support the LSP being requested. These characteristics include desired link protection, LSP encoding, and LSP payload.

The Generalized Label Request indicates the link protection type desired for the LSP. If a particular protection type, i.e., 1+1, or 1:N, is requested, then a connection request is processed only if the desired protection type can be honored. Note that the protection capabilities of a link may be advertised in routing, see [OMPLS-ISIS, OMPLS-OSPF]. Path computation algorithms may take this information into account when computing paths for setting up LSPs.

The Generalized Label Request also carries an LSP encoding parameter, called LSP Encoding Type. This parameter indicates the encoding type, e.g., SONET/SDH/GigE etc., that will be used with the data associated with the LSP. The LSP Encoding Type represents the nature of the LSP, and not the nature of the links that the LSP traverses. A link may support a set of encoding formats, where support means that a link is able to carry and switch a signal of one or more of

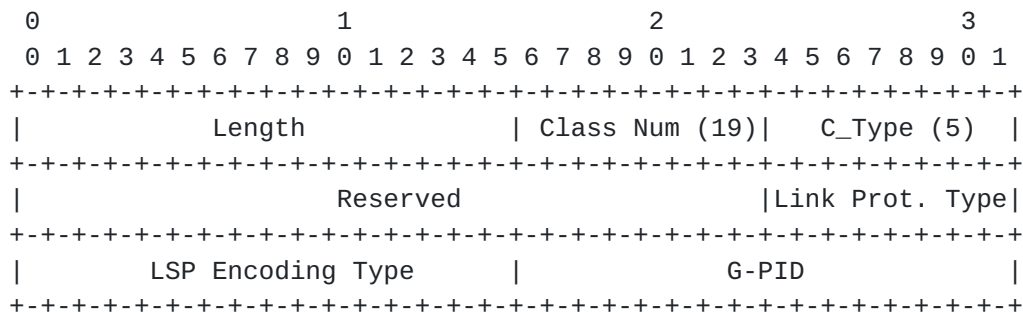


these encoding formats depending on the resource availability and capacity of the link. For example, consider an LSP signaled with "photonic" encoding. It is expected that such an LSP would be supported with no electrical conversion and no knowledge of the modulation and speed by the transit nodes. If the bit rate is known but not the modulation then a Clear encoding suffixed with the rate may be signaled. For example, a bit rate of OC-1 but an encoding type of clear can be signaled. The transit nodes would not look at the frames, but would know the bit rate and as a result can do OEO switching but not OXO switching. All other formats require framing knowledge, and field parameters are broken into the framing type and speed as shown below. A REQUEST/Path message SHOULD contain as specific a LSP Encoding Type as possible to allow the maximum flexibility in switching by transit LSRs.

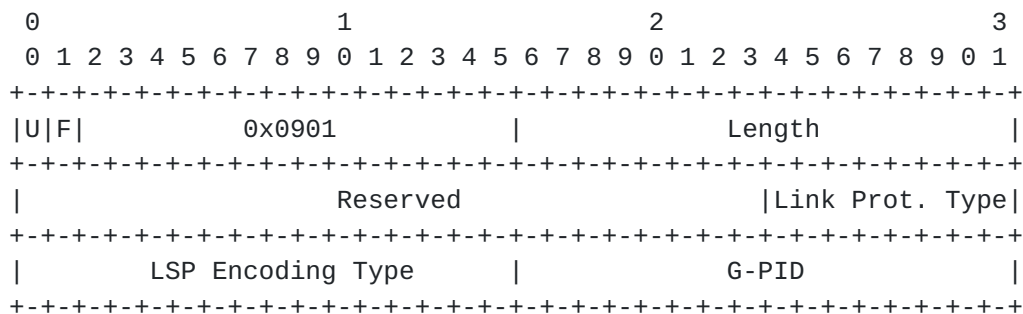
### 3.1.1. Required Information

The Generalized Label Request object/TLV carries the desired information, the format of which is as follows:

The format of a Generalized Label Request (in RSVP) is:



The format of a Generalized Labels (in CR-LDP) is:





## Link Protection Type: 8 bits

Indicates the desired link protection type of the connection setup. A value of 0 implies that this connection does not care about the available protection type. Other values are defined in the [[OMPLS-ISIS](#), [OMPLS-OSPF](#)] draft, which also defines how link protection capabilities may be advertised.

## LSP Encoding Type: 16 bits

Indicates the required encoding. This field is set by the ingress node, transparently passed by transit nodes, and used by the egress node. The following shows permitted values and their meaning:

Value	Bit Rate	Encoding	
-----	-----	-----	
0	N/A	Packets	
<n>	OC-<n>	SONET	1 <= <n> <= 3072
3072 + <n>	STS-<n>	SDH	1 <= <n> <= 3072
6144 + <n>	OC-<n>	Clear	1 <= <n> <= 3072
9217	DS0	DS0	
9218	DS1	DS1	
9219	E1	E1	
9220	DS2	DS2	
9221	E2	E2	
9222	DS3	DS3	
9223	E3	E3	
9224	J3	J3	
9225	DS4	DS4	
9226	E4	E4	
9227	J4	J4	
9228	1Gbps	GigE	
9229	10Gbps	10GigE	
9230	VT-1.5/TU-11;		
9231	VT-2/TU-12;		
9232	VT-3;		
9233	VT-6/TU-2;		
9234	TU-3;		
9235	Photonic Lambda		
9236	Photonic Waveband		





Generalized PID (G-PID): 16 bits

An identifier of the payload carried by an LSP. Standard Ethertype values are used (with new Ethertype values defined as needed). This field is set by the ingress node, transparently passed by transit nodes, and used by the egress node.

### **3.1.2. Procedures**

A node processing the Path/REQUEST message containing the Generalized Label Request must verify that the requested parameters can be satisfied by the node and by the outgoing interface. The node may either directly support the LSP or it may use a tunnel (FA), i.e., another class of switching. In either case, each parameter must be checked.

Note that local node policy dictates when tunnels may be used and when they may be created. Local policy may allow for tunnels to be dynamically established or may be solely administratively controlled. For more information on tunnels and processing of ER hops when using tunnels see [[MPLS-HIERARCHY](#)].

Transit nodes MUST verify that the outgoing interface or tunnel can support the requested Link Protection Type. If it cannot, the node MUST generate a PathErr/NOTIFICATION message, with a "Routing problem/Unsupported Link Protection" indication.

Transit and egress nodes MUST verify that the node itself and, where appropriate, that the outgoing interface or tunnel can support the requested LSP Encoding Type. If encoding cannot be supported, the node MUST generate a PathErr/NOTIFICATION message, with a "Routing problem/Unsupported Encoding" indication.

The G-PID parameter is normally only examined at the egress. If the indicated G-PID cannot be supported then the egress MUST generate a PathErr/NOTIFICATION message, with a "Routing problem/Unsupported GPID" indication. In the case of PSC and when penultimate hop popping (PHP) is requested, the penultimate hop also examines the (stored) G-PID during the processing of the Resv/MAPPING message. In this case if the G-PID is not supported, then the penultimate hop MUST generate a ResvErr/NOTIFICATION message with a "Routing problem/Unacceptable label value" indication.

When an error message is not generated, normal processing occurs. In the transit case this will typically result in a Path/REQUEST message being propagated. In the egress case and PHP special case this will typically result in a Resv/MAPPING message being generated.



### **3.2. Generalized Label**

The Generalized Label extends the traditional Label Object in that it allows the representation of not only labels which travel in-band with associated data packets, but also labels which identify time-slots, wavelengths, or space division multiplexed positions. For example, the Generalized Label may carry a label that represents (a) a single fiber in a bundle, (b) a single waveband within fiber, (c) a single wavelength within a waveband (or fiber), or (d) a set of time-slots within a wavelength (or fiber). It may also carry a label that represents a generic MPLS label, a Frame Relay label, or an ATM label (VCI/VPI).

A Generalized Label does not identify the "class" to which the label belongs. This is implicit in the multiplexing capabilities of the link on which the label is used.

A Generalized Label object only carries a single level of label, i.e., it is non-hierarchical. When multiple levels of label (LSPs within LSPs) are required, each LSP must be established separately, see [[MPLS-HIERARCHY](#)].

The Generalized Label supports link bundling by carrying the identity of the component link. In the presence of link bundling, each Generalized Label indicates label(s) within the context of a specific component link, which is identified by the Link ID (which is carried as part of Generalized Label). The values used to indicate Link ID have local significance between two neighbors. Procedures for determining Link ID are specified elsewhere, see [[MPLS-BUNDLE](#)].

Each Generalized Label object carries a variable length label parameter.



### 3.2.1. Required Information

The format of a Generalized Labels (in RSVP) 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	2	3	4	5	6	7	8	9
Length										Class Num (16)										C_Type (2)																			
Link ID																																							
Label																																							
...																																							

The format of a Generalized Labels (in CR-LDP) 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	2	3	4	5	6	7	8	9
U F		0x0902										Length																											
Link ID																																							
Label																																							
...																																							

Link ID: 32 bits

Indicates link on which label is being request, from the message sender's perspective. Used when bundling several (parallel) links, see [[MPLS-BUNDLE](#)]. MUST be zero when bundling is not being used. Note, values only have significance between two neighbors and the receiver may need to convert the received value into a value that has local significance.

Label: Variable

Carries label information. The semantics of this field depends on the type of the link over which the label is used.



### 3.2.1.1. SDH Labels

For SDH time-slots, the format for the label is given below. If a single label is given, that label is the lowest time-slot of a contiguously concatenated signal; the bandwidth of the LSP request indicates the number of labels to be concatenated to form the SDH signal trail. If there are multiple labels, then the identified labels are virtually concatenated to form the SDH signal trail. The above representation limits virtual concatenation to remain within a single (component) link.

The format of the label for TDM-LSRs 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
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+
|               S               | U   | K   | L   | M   |
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

This is an extension of the numbering scheme defined in G.707 [section 7.3](#), i.e. the (K, L, M) numbering. Each letter indicates a possible branch number starting at the parent node in the naming tree. Branches are considered as numbered starting from the high of the figure, the numbering starts normally at 1 (0 is used to indicate an exception).

1. S is the index of an STS signal in a multiplex. For SDH, S=1 denotes an STM-0 signal; otherwise, S must be a multiple of 3, in which case, S/3 is the index of the STM signal.
2. U indicates a specific VC inside a given STS-1 or STM-1 signal. U=1 indicates a single VC-4, while U=2->4 indicates a specific VC-3 inside the given signal.
3. K indicates possible branches of a VC-4. It is not meaningful for the VC-3 multiplexing and must be 0 in that case. K=1 indicates that the VC-4 is not further sub-divided. K=2->4 indicates a specific TUG-3 inside the given VC-4. A multiplex entry name with K=0 denotes a VC-3 multiplexing of an STM-1 (easy to test and read).
4. L indicates possible branches of either a TUG-3 or a VC-3. In the first case, L=1 indicates that the TUG-3 is not further sub-divided and contains simply a VC-3. L=2->8 indicates a specific TUG-2 inside the given TUG-3. In the second case, L=1 indicates that the VC-3 is not further sub-divided. L=2->8 indicates a specific TUG-2 inside the given VC-3.









The format of a Port and Wavelength label is:

[illegible]

Label: 32 bits

Indicates port/fiber or lambda to be used, from the sender's perspective. Values used in this field only have significance between two neighbors, and the receiver may need to convert the received value into a value that has local significance. Values may be configured or dynamically determined using a protocol such as [LMP](#).

#### 3.2.1.4. Other Labels

Generic MPLS labels and Frame Relay labels are encoded right justified aligned in 32 bits (4 octets). ATM labels are encoded with the VPI right justified in bits 0-15 and the VCI right justified in bits 16-31.

### 3.2.2. Procedures

The Generalized Label travels in the upstream direction in MAPPING/Resv messages.

The presence of both a generalized and normal label object in a Path/REQUEST message is a protocol error and should be treated as a malformed message by the recipient.

If link bundling is not being used, the Link ID MUST be zero on transmission and ignored when received.

When link bundling is being used, the Link ID MUST contain a non zero value which uniquely identifies which link (e.g., fiber, waveband or wavelength) is to contain the label(s). See [\[MPLS-BUNDLE\]](#) for details.

In the case where the the Link ID uniquely identifies the LSP (e.g., wavelength) the label parameter SHOULD be set to zero (0) and MUST be ignored when received.

The recipient of a Resv/MAPPING message containing a Generalized



Label verifies that the values passed are acceptable. If the Link ID is being used and the value is unknown, the recipient MUST generate a ResvErr/NOTIFICATION message with a "Routing problem/Unknown Link ID" indication. If the combination of the Link ID value (when applicable) and label is unacceptable then the recipient MUST generate a ResvErr/NOTIFICATION message with a "Routing problem/MPLS label allocation failure" indication.

### **3.3. Waveband Switching**

A special case of lambda switching is waveband switching. A waveband represents a set of contiguous wavelengths which can be switched together to a new waveband. For optimization reasons it may be desirable for an optical cross connect to optically switch multiple wavelengths as a unit. This may reduce the distortion on the individual wavelengths and may allow tighter separation of the individual wavelengths. The Waveband Label is defined to support this special case.

Waveband switching naturally introduces another level of label hierarchy and as such the waveband is treated the same way all other upper layer labels are treated.

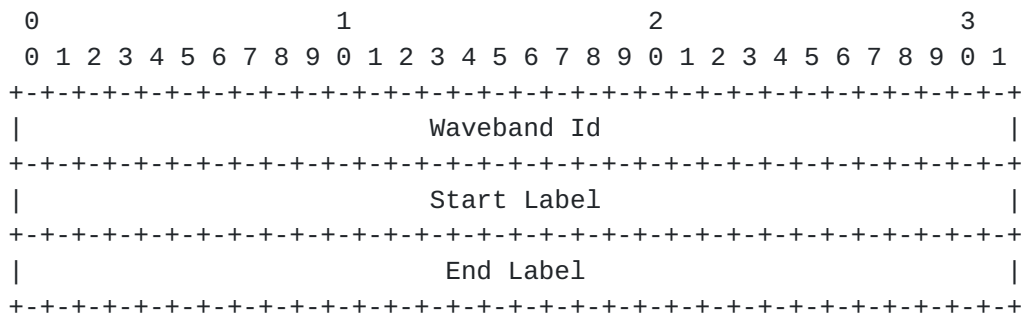
As far as the MPLS protocols are concerned there is little difference between a waveband label and a wavelength label except that semantically the waveband can be subdivided into wavelengths whereas the wavelength can only be subdivided into time or statistically multiplexed labels.

#### **3.3.1. Required information**

Waveband switching uses the same format as the generalized label, see [section 3.2.1](#). For compatibility reasons, a new RSVP c-type and CR-LDP type is assigned for the Waveband Label.



In the context of waveband switching, the generalized label has the following format:



Waveband Id: 32 bits

A waveband identifier. The value is selected by the sender and reused in all subsequent related messages.

Start Label: 32 bits

Indicates the channel identifier, from the sender's perspective, of the lowest value wavelength making up the waveband.

End Label: 32 bits

Indicates the channel identifier, from the sender's perspective, of the highest value wavelength making up the waveband.

Channel identifiers are established either by configuration or by means of a protocol such as LMP [[LMP](#)]. They are normally used in the link id parameter of the Generalized Label Request when bundling is being used or the label parameter for PSC and LSC links when bundling is not being used.

### 3.3.2. Procedures

The procedures defined in [Section 3.2.2](#) apply to waveband switching. This includes generating a ResvErr/NOTIFICATION message with a "Routing problem/MPLS label allocation failure" indication if any of the label fields are unrecognized or unacceptable.

Additionally, when a waveband is switched to another waveband, it is possible that the wavelengths within the waveband will be mirrored about a center frequency. When this type of switching is employed, the start and end label in the waveband label object **MUST** be flipped before forwarding the label object with the new waveband Id. In this





manner an egress/ingress LSR which receives a waveband label which has these values inverted, knows that it must also invert its egress association to pick up the proper wavelengths. Without this mechanism and with an odd number of mirrored switching operations, the egress LSRs will not know that an input wavelength of say L1 will emerge from the waveband tunnel as L100.

This operation **MUST** be performed in both directions when a bidirectional waveband tunnel is being established.

### **3.4. Suggested Label**

The Suggested Label is used to provide a downstream node with the upstream node's label preference. This permits the upstream node to start configuring it's hardware with the proposed label before the label is communicated by the downstream node. Such early configuration is valuable to systems that take non-trivial time to establish a label in hardware. Such early configuration can reduce setup latency, and may be important for restoration purposes where alternate LSPs may need to be rapidly established as a result of network failures.

The use of Suggested Label is only an optimization. If a downstream node passes a different label upstream, an upstream LSR **MUST** reconfigure itself so that it uses the label specified by the downstream node, thereby maintaining the downstream control of a label.

#### **3.4.1. Required Information and Processing**

The format of a suggested label is identical to a generalized label. It is used in Path/REQUEST messages. In RSVP the Suggested Label uses a new class number (TBD of form 10bbbbbb) and the C-type of the label being suggested. In CR-LDP, Suggested Label uses type = 0x904.

Errors in received Suggested Labels **MUST** be ignored. This includes any received inconsistent or unacceptable values.



### **3.5. Label Set**

The Label Set is used to limit the label choices of a downstream node to a set of acceptable labels. This limitation applies on a per hop basis.

There are four cases where a Label Set is useful in the optical domain. The first case is where the end equipment is only capable of transmitting and receiving on a small specific set of wavelengths/bands. The second case is where there are a sequence of interfaces which cannot support wavelength conversion (CI-incapable) and require the same wavelength be used end-to-end over a sequence of hops, or even an entire path. The third case is where it is desirable to limit the amount of wavelength conversion being performed to reduce the distortion on the optical signals. The last case is where two ends of a link support different sets of wavelengths.

Label Set is used to restrict label ranges that may be used for a particular LSP between two peers. The receiver of a Label Set must restrict its choice of labels to one which is in the Label Set. Much like a label, a Label Set may be present across multiple hops. In this case each node generates its own outgoing Label Set, possibly based on the incoming Label Set and the node's hardware capabilities. This case is expected to be the norm for nodes with conversion incapable (CI-incapable) interfaces.

The use of Label Set is optional, if not present, all labels from the valid label range may be used. Conceptually the absence of a Label Set implies a Label Set whose value is {U}, the set of all valid labels.

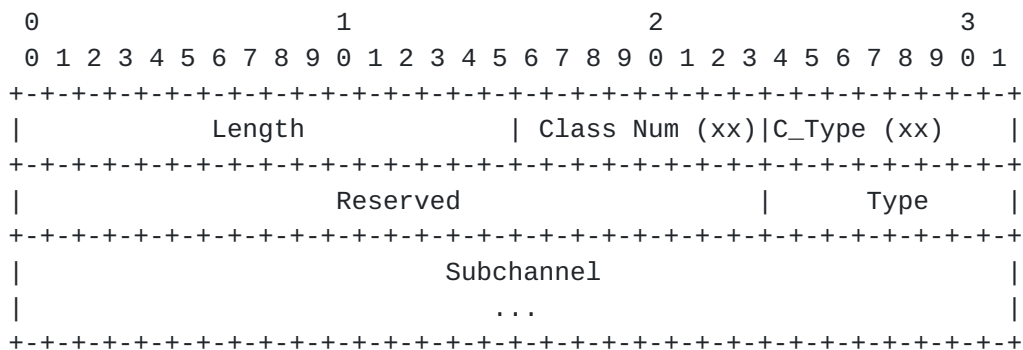
#### **3.5.1. Required Information**

This LabelSet is used in Path/REQUEST messages.

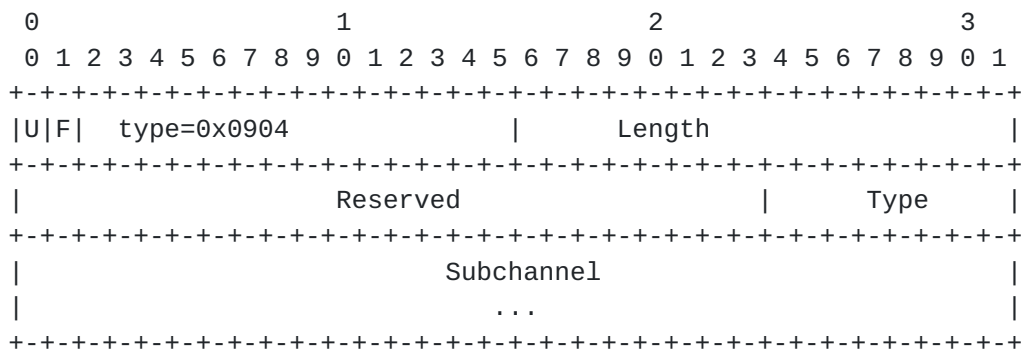
The data required to support the Label Set consists of a variable sized array of labels, or label ranges. These labels are subchannel identifiers and MUST lie within the link identified in Generalized Label Request.



The format of a LabelSet (in RSVP) is:



The format of a LabelSet (in CR-LDP) is:



Type: 2 bits

- 0x00 means subchannel is a single element (inclusive)
- 0x01 means subchannel is a start element (inclusive)
- 0x02 means subchannel is an end element (inclusive)
- 0x03 means subchannel is a single element (exclusive)

Subchannel:

The subchannel represents the label (wavelength, fiber ... ) which is eligible for allocation. This field has the same format as described for labels under [section 3.2](#).

Since subchannel to local channel identifiers (e.g., wavelength) mappings are a local matter, when a Label Set is propagated from one node to the next, the subchannels may have to be remapped to new subchannel values for consistency.

A Label Set can be just a series of single elements (Type=0x00) sorted in increasing order of subchannel value.



A Label Set can be a set of ranges (Type=0x01 followed by Type=0x02). The ranges MUST be sorted. A range which is missing a beginning or an end implies no bound where the bound is missing. A range which contains a Type=0x03 (exclusive) means all subchannels in the range except that subchannel are eligible.

### **3.5.2. Procedures**

The absence of a Label Set implies that all labels are acceptable. A Label Set is included when a node wishes to restrict the label(s) that may be used downstream.

On reception of a Path/REQUEST message a CI-capable interface will restrict its choice of labels to one which is in the Label Set. The CI-capable receiver may also remove the Label Set prior to forwarding the Path/REQUEST message. If the node is unable to pick a label from the Label Set, then the request is terminated and a PathErr/NOTIFICATION message with a "Routing problem/Label Set" indication MUST be generated. It is a local matter if the Label Set is stored for later selection on the RESV/Mapping or if the selection is made immediately for propagation in the RESV/Mapping.

On reception of a Path/REQUEST message for a CI-incapable interface, the Label Set in the message is compared against the set of available labels at the downstream interface and the resulting intersecting Label Set is forwarded in a Path/REQUEST message. When the resulting Label Set is empty, the Path/REQUEST must be terminated, and a PathErr/NOTIFICATION message, and a "Routing problem/Label Set" indication MUST be generated. Note that intersection is based on the physical labels (actual wavelength/band values) which may have different logical values on different links, as a result it is the responsibility of the node to map these values so that they have a consistent physical meaning, or to drop the particular values from the set if no suitable logical label value exists.

On reception of a Resv/MAPPING message at an intermediate node, the label to propagate upstream is selected from within the (stored) Label Set (preferred) or may be preselected from that set to save memory.

Note, on reception of a Resv/MAPPING message for an interface which is CI-incapable it has no other choice than to use the same physical label (wavelength/band) as received in the Resv/MAPPING. In this case, the use and propagation of a Label Set will significantly reduce the chances that this allocation will fail when CI-incapable nodes are traversed.





#### **4. Bidirectional LSPs**

This section defines direct support of bidirectional LSPs. Support is defined for LSPs that have the same traffic engineering requirements including fate sharing, protection and restoration, and resource requirements (e.g., latency and jitter) in each direction. In the remainder of this section, the term "initiator" is used to refer to a node that starts the establishment of an LSP and the term "terminator" is used to refer to the node that is the target of the LSP. Note that for bidirectional LSPs, there is only one "initiator" and one "terminator".

Normally to establish a bidirectional LSP when using [[RSVP-TE](#)] or [[CR-LDP](#)] two unidirectional paths must be independently established. This approach has the following disadvantages:

- \* The latency to establish the bidirectional LSP is equal to one round trip signaling time plus one initiator-terminator signaling transit delay. This not only extends the setup latency for successful LSP establishment, but it extends the worst-case latency for discovering an unsuccessful LSP to as much as two times the initiator-terminator transit delay. These delays are particularly significant for LSPs that are established for restoration purposes.
- \* The control overhead is twice that of a unidirectional LSP. This is because separate control messages (e.g. Path and Resv) must be generated for both segments of the bidirectional LSP.
- \* Because the resources are established in separate segments, route selection is complicated. There is also additional potential race for conditions in assignment of resources, which decreases the overall probability of successfully establishing the bidirectional connection.
- \* It is more difficult to provide a clean interface for SONET equipment that may rely on bidirectional hop-by-hop paths for protection switching. Note that existing SONET gear transmits the control information in-band with the data.
- \* Bidirectional optical LSPs (or lightpaths) are seen as a requirement for many optical networking service providers.

With bidirectional LSPs both the downstream and upstream data paths, i.e., from initiator to terminator and terminator to initiator, are established using a single set of Path/REQUEST and Resv/MAPPING messages. This reduces the setup latency to essentially one initiator-terminator round trip time plus processing time, and limits



the control overhead to the same number of messages as a unidirectional LSP.

#### **4.1. Required Information**

For bidirectional LSPs, two labels must be allocated. Bidirectional LSP setup is indicated by the presence of an Upstream Label in the REQUEST/Path message. An Upstream Label has the same format as the generalized label, see [Section 3.2](#). In RSVP the Upstream Label uses a new class number (TBD of form 0bbbbbbb) and the C-type of the label being suggested. In CR-LDP, Upstream Label uses type=0x0906

#### **4.2. Procedures**

The process of establishing a bidirectional LSP follows the establishment of a unidirectional LSP with some additions. To support bidirectional LSPs an Upstream Label is added to the Path/REQUEST message. The Upstream Label MUST indicate a label that is valid for forwarding at the time the Path/REQUEST message is sent.

When a Path/REQUEST message containing an Upstream Label is received, the receiver first verifies that the upstream label is acceptable. If the label is not acceptable, the receiver MUST issue a PathErr/NOTIFICATION message with a "Routing problem/Unacceptable label value" indication.

An intermediate node must also allocate a label on the outgoing interface and establish internal data paths before filling in an outgoing Upstream Label and propagating the Path/REQUEST message. If an intermediate node is unable to allocate a label or internal resources, then it MUST issue a PathErr/NOTIFICATION message with a "Routing problem/Label allocation failure" indication.

Terminator nodes process Path/REQUEST messages as usual, with the exception that the upstream label can immediately be used to transport associated data upstream to the initiator.

When a bidirectional LSP is removed, both upstream and downstream labels are invalidated and it is no longer valid to send data using the associated labels.



### **4.3. Contention Resolution**

Contention for labels may occur between two bidirectional LSP setup requests traveling in opposite directions. This contention occurs when both sides allocate the same resources (ports) at effectively the same time. If there is no restriction on the ports that can be used for bidirectional LSPs and if there are alternate resources, then both nodes will pass different labels upstream and the contention will be resolved naturally. However, if there is a restriction on the ports that can be used for the bidirectional LSPs (for example, if they must be physically coupled on a single I/O card), or if there are no more resources available, then the contention must be resolved by other means. To resolve contention, the node with the higher node ID will win the contention and it MUST issue a PathErr/NOTIFICATION message with a "Routing problem/Label allocation failure" indication. Upon receipt of such an error, the node SHOULD try to allocate a different Upstream label (and a different Suggested Label if used) to the bidirectional path. However, if no other resources are available, the node must proceed with standard error handling. For the purposes of RSVP contention resolution, the node ID is the IP address used in the RSVP\_HOP object.

To reduce the probability of contention, one may impose a policy that the node with the lower ID never suggests a label in the downstream direction and always accepts a Suggested Label from an upstream node with a higher ID. Furthermore, since the label sets are exchanged using LMP [[LMP](#)], an alternative local policy could further be imposed such that (with respect to the higher numbered node's label set) the higher numbered node could allocate labels from the high end of the label range while the lower numbered node allocates labels from the low end of the label range. This mechanism would augment any close packing algorithms that may be used for bandwidth (or wavelength) optimization.

An example of contention between two nodes (PXC 1 and PXC 2) is shown in Figure 1. In this example PXC 1 assigns an Upstream Label for the channel corresponding to local BCId=2 (local BCId=7 on PXC 2) and sends a Suggested Label for the channel corresponding to local BCId=1 (local BCId=6 on PXC 2). Simultaneously, PXC 2 assigns an Upstream Label for the channel corresponding to its local BCId=6 (local BCId=1 on PXC 1) and sends a Suggested Label for the channel corresponding to its local BCId=7 (local BCId=2 on PXC 1). If there is no restriction on the ports that can be used for bidirectional LSPs and if there are alternate resources available, then both PXC 1 and PXC 2 will pass different labels upstream and the contention is resolved naturally (see Fig. 2). However, if there is a restriction on the ports that can be used for bidirectional LSPs (for example, if they



must be physically coupled on a single I/O card), then the contention must be resolved using the router Id (see Fig. 3).

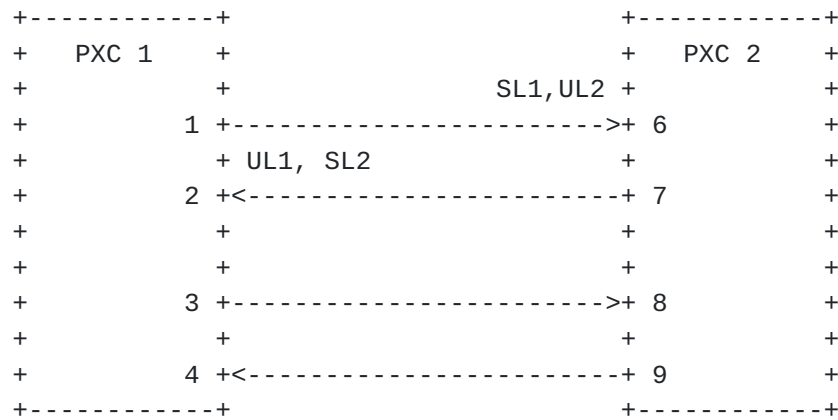


Figure 1. Label Contention

In this example, PXC 1 assigns an Upstream Label using BCId=2 (BCId=7 on PXC 2) and a Suggested Label using BCId=1 (BCId=6 on PXC 2). Simultaneously, PXC 2 assigns an Upstream Label using BCId=6 (BCId=1 on PXC 1) and a Suggested Label using BCId=7 (BCId=2 on PXC 1).

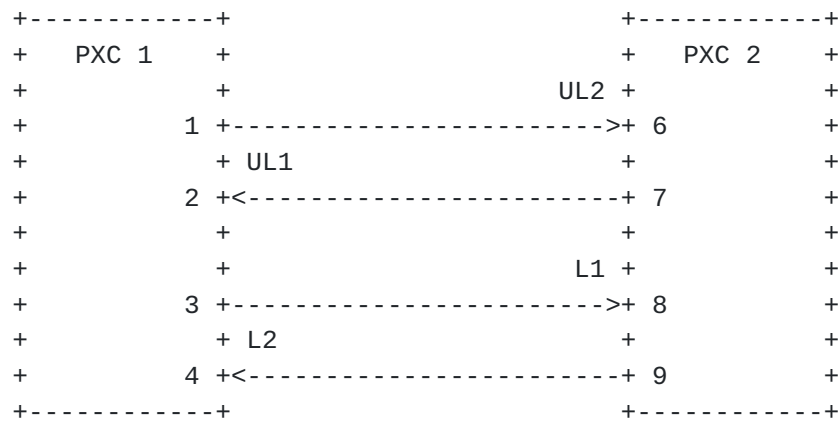


Figure 2. Label Contention Resolution without resource restrictions

In this example, there is no restriction on the ports that can be used by the bidirectional connection and contention is resolved naturally.





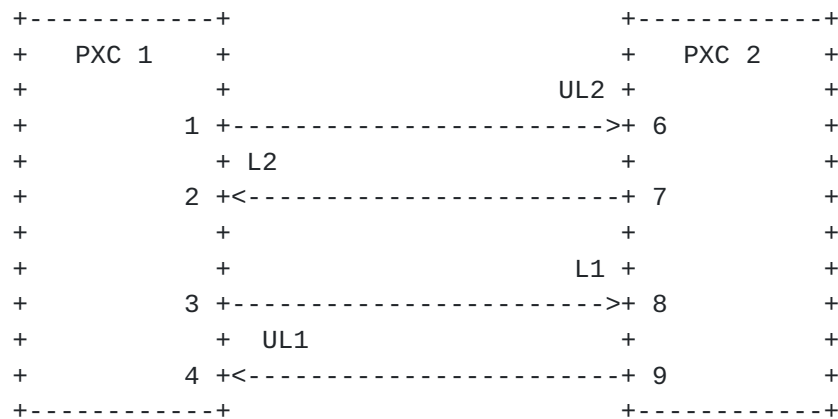


Figure 3. Label Contention Resolution with resource restrictions

In this example, ports 1,2 and 3,4 on PXC 1 (ports 6,7 and 8,9 on PXC 2, respectively) must be used by the same bidirectional connection. Since PXC 2 has a higher node ID, it wins the contention and PXC 1 must use a different set of labels.

## 5. Notification

This section defines two signaling extensions that enable expedited notification of failures and other events to nodes responsible for restoring failed LSPs. The first extension, the Notify message, provides for general event notification. The second allows for the combining of such notifications in a single message. These extensions are RSVP specific.

### 5.1. Notify Message

The Notify message provides a mechanism to inform non-adjacent nodes of LSP related events. This message differs from the currently defined error messages (i.e., PathErr and ResvErr messages of RSVP) in that it can be "targeted" to a node other than the immediate upstream or downstream neighbor and that it is a generalized notification mechanism. The Notify message does not replace existing error messages. The Notify message may be sent either (a) normally, where non-target nodes just forward the Notify message to the target node, similar to ResvConf processing in [RSVP]; or (b) encapsulated in a new IP header whose destination is equal to the target IP address. Regardless of the transmission mechanism, nodes receiving a Notify message not destined to the node forward the message, unmodified, towards the target.

To support reliable delivery of the Notify message, an Ack Message [[RSVP-RR](#)] is used to acknowledge the receipt of a Notify Message. A



node that receives the a Notify message MUST send a Notify\_Ack message confirming receipt of the Notify message.

#### 5.1.1. Required Information

The Notify message is a generalized notification message. The IP destination address is set to the IP address of the intended receiver. The Notify message is sent without the router alert option.

```
<Notify message> ::= <Common Header> [<INTEGRITY>] <MESSAGE_ID>
                        <SESSION> <ERROR_SPEC> [<POLICY_DATA>...]
                        [<sender descriptor>]
```

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                        [<ADSPEC>] [<RECORD_ROUTE>]
```

The ERROR\_SPEC object specifies the error and includes the IP address of either the node that detected the error or the link that has failed. See ERROR\_SPEC definition in [RFC2205](#). The MESSAGE\_ID object is defined in [[RSVP-RR](#)].

Note: for CR-LDP there is not currently a similar mechanism. In CR-LDP, when a failure is detected it will be propagated with RELEASE/WITHDRAW messages radially outward from the point of failure. Resources are to be released in this phase and actual resource information is fed back to the source using the feedback mechanisms of [[FEEDBACK](#)]. In this manner the source will have an accurate view of available resources and can start rerouting much sooner.

#### 5.2. Non-Adjacent Message Bundling

[RSVP-RR] defines the bundle message which can be used to aggregate multiple signaling messages into a single packet. The defined mechanism is limited to adjacent RSVP nodes. This section defines the use of the bundle message between non-adjacent nodes. This is accomplished by setting the IP destination address of the bundle message to the address of the target node. The non-adjacent bundle message may be sent either (a) normally, where non-target nodes just forward the message to the target node (see ResvConf processing in [RSVP] for reference); or (b) encapsulated in a new IP header who's destination is equal to the target IP address. Regardless of the transmission mechanism, nodes receiving a bundle message not destined to the node just forward the message, unmodified, towards the target. The motivation for supporting non-adjacent message bundling is to support the bundling of Notify Messages.



Non-adjacent bundle messages can only be sent to RSVP nodes that support bundling. Currently, the only method for discovering such information about a non-adjacent node is through manual configuration.

#### **5.2.1. Required Information**

The RSVP bundle message is defined in [[RSVP-RR](#)]. The only modification from what is specified in [[RSVP-RR](#)] is that the IP destination address does not have to be an immediate upstream/downstream neighbor.

### **6. Egress Control**

The LSR at the head-end of an LSP can control the termination of the LSP by using the ERO. To terminate an LSP on a particular outgoing interface of the egress LSR, the head-end may specify the IP address of that interface as the last element in the ERO, provided that that interface has an associated IP address.

There are cases where the use of IP address doesn't provide enough information to uniquely identify the egress termination. One case is when the outgoing interface on the egress LSR is a component link of a link bundle. Another case is when it is desirable to "splice" two LSPs together, i.e., where the tail of the first LSP would be "spliced" into the head of the second LSP. This last case is more likely to be used in the non-PSC classes of links.

#### **6.1. Required Information**

To handle scenarios described in the previous paragraph, the ERO subobject Egress Label type is introduced.

For RSVP, this ERO subobject - Egress Label is defined 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	2	3	4	5	6	7	8	9																														
L										Type																				Length																				Reserved																			
																				Link ID																																																	
																				Label																																																	
																				...																																																	

For CR-LDP the Egress Label ER-Hop is defined as follows:

										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   0		0x901										Length																											
+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +									
L		Reserved										Link ID																											
+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +									
		Link ID (continued)										Label																											
+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +									
		Label (continued)																																					
												...																											
+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +										+ + + + + + + + + +									

L

This bit must be set to 0.

Type (RSVP Only)

3 Egress Label (To be assigned)

Length

The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always divisible by 4.

Link ID

Same definition as in Generalized Label. See [Section 3.2.1](#).





## Label

When an LSP has to be spliced into another LSP, this field identifies the label that the tail of the first LSP has to use in order to slice it into the head of the second LSP. In such cases the format of this field is identical to the one used by the Label field in the Generalized Label Object. In all other cases this field is omitted.

## 6.2. Procedures

The Egress Label subobject may appear only as the last subobject in the ERO/ER. Appearance of this subobject anywhere else in the ERO/ER is treated as a "Bad strict node" error.

During an LSP setup, when a node processing the ERO/RR performs Next Hop selection finds that the second subobject is an Egress Label Subobject, the node uses the information carried in this subobject to determine the handling of the data received over that LSP. Specifically, if the Link ID field of the subobject is non zero, then this field identifies a specific (outgoing) link of the node that should be used for sending all the data received over the LSP. If the Label field of the subobject is not Implicit NULL label, this field specifies the label that should be used as an outgoing label on the data received over the LSP.

Procedures by which an LSR at the head-end of an LSP obtains the information needed to construct the Egress Label subobject are outside the scope of this document.

## 7. Acknowledgments

This draft is the work of numerous authors and consists of a composition of a number of previous drafts in this area. A list of the drafts from which material and ideas were incorporated follows:

[draft-saha-rsvp-optical-signaling-00.txt](#)

[draft-lang-mpls-rsvp-oxc-00.txt](#)

[draft-kompella-mpls-optical-00.txt](#)

[draft-fan-mpls-lambda-signaling-00.txt](#)



## 8. Security Considerations

This draft does not introduce any new security considerations to either [\[CR-LDP\]](#) or [\[RSVP-TE\]](#).

## 9. References

- [CR-LDP] Jamoussi et al., "Constraint-Based LSP Setup using LDP", [draft-ietf-mpls-cr-ldp-04.txt](#), July, 2000.
- [LDP] Andersson et al., "LDP Specification", [draft-ietf-mpls-ldp-08.txt](#), June 2000.
- [LMP] Lang, J.P., Mitra, K., Drake, J., Kompella, K., Rekhter, Y., Saha, D., Berger, L., Basak, D., "Link Management Protocol", Internet Draft, [draft-lang-mpls-lmp-01.txt](#), July 2000.
- [MPLS-ARCH] Rosen et al., "Multiprotocol label switching Architecture", Internet Draft, [draft-ietf-mpls-arch-06.txt](#), August 1999.
- [MPLS-BUNDLE] Kompella, K., Rekhter, Y., and Berger, L., "Link Bundling in MPLS Traffic Engineering", Internet Draft, [draft-kompella-mpls-bundle-02.txt](#), July 2000.
- [MPLS-HIERARCHY] Kompella, K., and Rekhter, Y., "LSP Hierarchy with MPLS TE", Internet Draft, [draft-ietf-mpls-lsp-hierarchy-00.txt](#), July 2000.
- [OMPLS-ISIS] Kompella, K., Rekhter, Y., Banerjee, A., Drake, J., Bernstein, G., Fedyk, D., Mannie, E., Saha, D., and Sharma, V., "IS-IS Extensions in Support of MPL(ambda)S", Internet Draft, [draft-ompls-isis-extensions-00.txt](#), July 2000.
- [OMPLS-OSPF] Kompella, K., Rekhter, Y., Banerjee, A., Drake, J., Bernstein, G., Fedyk, D., Mannie, E., Saha, D., and Sharma, V., "OSPF Extensions in Support of MPL(ambda)S", Internet Draft, [draft-ompls-ospf-extensions-00.txt](#), July 2000.
- [RSVP-TE] Awduche, D.O., Berger, L., Gan, D.-H., Li, T., Swallow, G., and Srinivasan, V., "RSVP-TE: Extensions to RSVP for LSP Tunnels," Internet Draft, [draft-ietf-mpls-rsvp-lsp-tunnel-06.txt](#), July 2000.
- [RSVP-RR] Berger L., Gan D., Swallow G., Pan P., Tommasi F.,



Molendini S., "RSVP Refresh Overhead Reduction Extensions",  
[draft-ietf-rsvp-refresh-reduct-05.txt](#), June 2000.

[FEEDBACK] P. Ashwood-Smith, B. Jamoussi, D. Fedyk, D. Skalecki,  
"Improving Topology Data Base Accuracy With LSP Feedback  
via CR-LDP", Internet Draft, [draft-ietf-mpls-te-feed-00.txt](#).

## **10. Authors' Addresses**

Peter Ashwood-Smith  
Nortel Networks Corp.  
P.O. Box 3511 Station C,  
Ottawa, ON K1Y 4H7  
Canada  
Phone: +1 613 763 4534  
Email: [petera@nortelnetworks.com](mailto:petera@nortelnetworks.com)

Ayan Banerjee  
Calient Networks  
5853 Rue Ferrari  
San Jose, CA 95138  
Phone: +1 408 972-3645  
Email: [abanerjee@calient.net](mailto:abanerjee@calient.net)

Lou Berger  
LabN Consulting, LLC  
Phone: +1 301 468 9228  
Email: [lberger@labn.net](mailto:lberger@labn.net)

Greg Bernstein  
Ciena Corporation  
10480 Ridgeview Court  
Cupertino, CA 94014  
Phone: +1 408 366 4713  
Email: [greg@ciena.com](mailto:greg@ciena.com)

John Drake  
Calient Networks  
5853 Rue Ferrari  
San Jose, CA 95138  
Phone: +1 408 972 3720  
Email: [jdrake@calient.net](mailto:jdrake@calient.net)



Yanhe Fan  
Nortel Networks  
PO Box 3511 Station C  
Ottawa, ON, K1Y 4H7  
Canada  
Phone: +1 613 765 2315  
Email: yanhe@nortelnetworks.com

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

Jonathan P. Lang  
Calient Networks  
25 Castilian  
Goleta, CA 93117  
Email: jplang@calient.net

Eric Mannie  
Optical Networking  
Network R&D  
GTS Network Services  
Terhulpesteenweg 6A  
1560 Hoeilaart - Belgium  
Phone: +32 2 658 56 52  
Mobile: +32 496 58 56 52  
Fax: +32 2 658 51 18  
Email: eric.mannie@gtsgroup.com

Bala Rajagopalan  
Tellium, Inc.  
2 Crescent Place  
P.O. Box 901  
Oceanport, NJ 07757-0901  
Phone: +1 732 923 4237  
Fax: +1 732 923 9804  
Email: braja@tellium.com

Yakov Rekhter  
cisco Systems  
Email: yakov@cisco.com





Debanjan Saha  
Tellium Optical Systems  
2 Crescent Place  
Oceanport, NJ 07757-0901  
Phone: +1 732 923 4264  
Fax: +1 732 923 9804  
Email: dsaha@tellium.com

Vishal Sharma  
Tellabs Research Center  
One Kendall Square  
Bldg. 100, Ste. 121  
Cambridge, MA 02139-1562  
Phone: +1 617 577 8760  
Email: Vishal.Sharma@tellabs.com

Z. Bo Tang  
Tellium, Inc.  
2 Crescent Place  
P.O. Box 901  
Oceanport, NJ 07757-0901  
Phone: +1 732 923 4231  
Fax: +1 732 923 9804  
Email: btang@tellium.com



Generated on: Fri Jul 14 13:13:19 EDT 2000