Network Working Group Peter Ashwood-Smith (Nortel Networks Corp.)
Internet Draft Ayan Banerjee (Calient Networks)
Expiration Date: May 2001 Lou Berger (Movaz Networks)
Greg Bernstein (Ciena Corporation)
John Drake (Calient Networks)
Yanhe Fan (Axiowave Networks)
Kireeti Kompella (Juniper Networks, Inc.)
Eric Mannie (GTS)
Jonathan P. Lang (Calient Networks)
Bala Rajagopalan (Tellium, Inc.)
Yakov Rekhter (Cisco Systems)
Debanjan Saha (Tellium, Inc.)
Vishal Sharma (Tellabs)
George Swallow (Cisco Systems)
Z. Bo Tang (Tellium, Inc.)

November 2000

Generalized MPLS - Signaling Functional Description

draft-ietf-mpls-generalized-signaling-01.txt

Status of this Memo

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 specified in [GMPLS-RSVP] and [GMPLS-LDP].

Contents

Changes from previous version:

- o Moved protocol specific details into two documents, one for RSVP-TE and one for CR-LDP.
- o Clarified Label Set
- o Fixed bandwidth encodings
- o Minor text cleanup

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 an 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 a functional description of the extensions needed to support each of the four classes of interfaces. Only signaling protocol independent formats and definitions are provided in this document. Protocol specific formats are defined in [GMPLS-RSVP] and [GMPLS-LDP].

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 3.1.3. 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 comes 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.

Generalized MPLS also supports the termination of an LSP on a specific egress port, see Section 5. [GMPLS-RSVP] also supports an RSVP specific mechanism for rapid failure notification.

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 that 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 [GMPLS-ISIS, GMPLS-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. All other formats require framing knowledge, and field parameters are broken into the framing type and speed as shown below.

3.1.1. Generalized Label Request Information

The information carried in a Generalized Label Request 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| LSP Enc. Type |Link Prot.Flags|             G-PID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LSP Encoding Type: 8 bits

Indicates the encoding of the LSP being requested. The following shows permitted values and their meaning:

Value	Type
1	Packet
2	Ethernet
3	ANSI PDH
4	ETSI PDH
5	SDH
6	SONET
7	Digital Wrapper
8	Lambda (photonic)
9	Fiber

The ANSI PDH and ETSI PDH types designate these respective networking technologies. DS1 and DS3 are examples of ANSI PDH LSPs. An E1 LSP would be ETSI PDH. The Lambda encoding type refers to the switching of wavelengths. The Fiber encoding type refers to switching at the fiber port level.

Link Protection Flags: 8 bits

Link Protection Flags indicate the desired protection level(s) for each link along the LSP. Note that the flags are distinct from MPLS-level LSP protection, see [RECOVERY]. A value of 0 implies that this connection does not care about which, if any, link protection is used. More than one bit may be set to indicate when multiple protection types are acceptable. When multiple bits are set and multiple protection types are available, the choice of protection type is a local (policy) decision.

The following flags are defined:

0x01 Unprotected

Indicates that the LSP should not use any link layer protection.

0x02 Shared

Indicates that a shared link layer protection scheme, such as 1:N protection, should be used to support the LSP.

0x04 Dedicated 1:1

Indicates that a dedicated link layer protection scheme, i.e., 1:1 protection, should be used to support the LSP.

0x08 Dedicated 1+1

Indicates that a dedicated link layer protection scheme, i.e., 1+1 protection, should be used to support the LSP.

0x10 Enhanced

Indicates that a protection scheme that is more reliable than Dedicated 1+1 should be used, e.g., 4 fiber BLSR/MS-SPRING.

Generalized PID (G-PID): 16 bits

An identifier of the payload carried by an LSP, i.e. an identifier of the client layer of that LSP. This must be interpreted according to the technology encoding type of the LSP and is used by the nodes at the endpoints of the LSP. Standard Ethertype values are used for packet and Ethernet LSPs; other values are:

Value	Type	Technology
0	Unknown	All
1	DS1 SF	ANSI-PDH
2	DS1 ESF	ANSI-PDH
3	DS3 M23	ANSI-PDH
4	DS3 C-Bit Parity	ANSI-PDH
5	Asynchronous mapping of E4	SDH
6	Asynchronous mapping of DS3/T3	SDH
7	Asynchronous mapping of E3	SDH
8	Bit synchronous mapping of E3	SDH
9	Byte synchronous mapping of E3	SDH
10	Asynchronous mapping of DS2/T2	SDH
11	Bit synchronous mapping of DS2/T2	SDH
12	Byte synchronous mapping of DS2/T2	SDH
13	Asynchronous mapping of E1	SDH
14	Byte synchronous mapping of E1	SDH
15	Byte synchronous mapping of 31 * DS0	SDH
16	Asynchronous mapping of DS1/T1	SDH
17	Bit synchronous mapping of DS1/T1	SDH
18	Byte synchronous mapping of DS1/T1	SDH
19	Same as 12 but in a VC-12	SDH
20	Same as 13 but in a VC-12	SDH
21	Same as 14 but in a VC-12	SDH
22	ATM mapping	SDH, SONET
22	DS1 SF Asynchronous	SONET
23	DS1 ESF Asynchronous	SONET
24	DS3 M23 Asynchronous	SONET
25	DS3 C-Bit Parity Asynchronous	SONET
26	VT	SONET
27	POS	SONET
28	STS	SONET
29	Ethernet	Lambda, Fiber
30	SDH	Lambda, Fiber
31	SONET	Lambda, Fiber
32	Digital Wrapper	Lambda, Fiber
33	Lambda	Fiber

3.1.2. Generalized Label Request with SONET/SDH Label Range

The Generalized Label Request with SONET/SDH Label Range object/TLV is used to represent specific characteristics related to the two TDM technologies. If the RGT, RT, and RNC, fields are all set to zero, it means that no concatenation, bundling or transparency is requested. If the requested LSP is itself a grouping of several components (e.g. a SONET concatenation), it is assumed that all components have the same characteristics. Note that the bandwidth carried in the signaling messages, see Section 3.1.4, are the aggregate bandwidth; in the instance where multiple components are signaled for, the individual component bandwidth is obtained by dividing this aggregated value by the requested number of components.

The information carried in a Generalized Label Request with SONET/SDH Label Range 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| LSP Enc. Type |Link Prot.Flags|             G-PID             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  RGT   |   RT   |    Reserved   |              RNC             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

LSP Encoding Type: 8 bits

See Section 3.1.1.

Link Protection Flags: 8 bits

See Section 3.1.1.

Generalized PID (G-PID): 16 bits

See Section 3.1.1.

Requested Grouping Type (RGT): 4 bits

This field indicates the SDH/SONET type of grouping requested for the LSP, it is used to constraint the type of concatenation. The values are defined in the following table:

Value	Grouping type
0	(Implies no concatenation/bundling when RNC = RT = 0)
1	Virtual concatenation
2	Contiguous standard concatenation
3	Contiguous arbitrary concatenation
4	Bundle (group of individual signals)

Requested Transparency (RT): 4 bits

This field indicates the type of SDH/SONET transparency ("emulation") requested for that LSP. The values are defined in the following table:

Value	Requested transparency
0	(Implies no concatenation/bundling when RNC = RGT = 0)
1	SDH Regenerator Section/SONET Section transparency
2	SDH Multiplex Section/SONET Line transparency
3	SDH Path/SONET Path transparency

Requested Number of Components (RNC): 16 bits

This field indicates the number of identical SDH/SONET signal types that are requested to be concatenated or inverse multiplexed in that LSP, as specified in the previous field. In these cases, the bandwidth of each component of that concatenation/bundling is obtained by dividing the aggregate bandwidth by the number of components requested. It is assumed that all these components have identical characteristics. This field is set to zero if non concatenation or bundling is requested.

3.1.3. Bandwidth Encoding

Bandwidth encodings are carried in in 32 bit number in IEEE floating point format (the unit is bytes per second). For non-packet LSPs, it is useful to define discrete values to identify the bandwidth of the LSP. Some typical values for the requested bandwidth are enumerated below. Additional values will be defined as needed. Bandwidth encoding values are carried in a per protocol specific manner, see

[GMPLS-RSVP] and [GMPLS-LDP].

Signal Type	(Bit-rate)	Value (Bytes/Sec) (IEEE Floating point)
DS0	(0.064 Mbps)	0x45FA0000
DS1	(1.544 Mbps)	0x483C7A00
E1	(2.048 Mbps)	0x487A0000
DS2	(6.312 Mbps)	0x4940A080
E2	(8.448 Mbps)	0x4980E800
Ethernet	(10.00 Mbps)	0x49989680
E3	(34.368 Mbps)	0x4A831A80
DS3	(44.736 Mbps)	0x4AAAA780
STS-1	(51.84 Mbps)	0x4AC5C100
Fast Ethernet	(100.00 Mbps)	0x4B3EBC20
E4	(139.264 Mbps)	0x4B84D000
OC-3/STM-1	(155.52 Mbps)	0x4B9450C0
OC-12/STM-4	(622.08 Mbps)	0x4C9450C0
GigE	(1000.00 Mbps)	0x4CEE6B28
OC-48	(2488.32 Mbps)	0x4D9450C0
OC-192	(9953.28 Mbps)	0x4E9450C0
10GigE-LAN	(10000.00 Mbps)	0x4E9502F9

3.2. Generalized Label

The Generalized Label extends the traditional label by allowing 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 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].

Each Generalized Label object carries a variable length label parameter.

3.2.1. Required Information

The information carried in a Generalized Label 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 and SONET Labels

SDH and SONET each define a multiplexing structure. These two structures are trees whose roots are respectively an STM-N or an STS-N; and whose leaves are the signals (time-slots) that can be transported and switched, i.e. a VC-x or a VT-x. A label will identify the type of a particular signal and its exact position in a multiplexing structure (both are related).

These multiplexing structures will be used as naming trees to create unique multiplex entry names or labels. Since the SONET multiplexing structure may be seen as a subset of the SDH multiplexing structure, the same format of label is used for SDH and SONET. As explained before (section 3.2), a label does not identify the "class" to which the label belongs. This is implicitly determined by the link on which the label is used. However, the encoding specified hereafter makes the direct distinction between SDH and SONET.

In case of signal concatenation or bundling, a list of labels may appear in the Label field of a Generalized Label.

In case of virtual concatenation, the explicit list of all signals in the concatenation is given. The signals identified by these labels are virtually concatenated to form the SDH or SONET signal trail. The above representation limits virtual concatenation to remain within a single (component) link.

In case of any type of contiguous concatenation (e.g. standard or arbitrary SONET concatenation), only one label appears in the Label field. That label is the lowest signal of the contiguously

concatenated signal. The bandwidth of the LSP request indicates the number of labels to be concatenated to form the SDH or SONET signal trail. By lowest signal we mean the one having the lowest label when compared as integer values, i.e. the first component signal of the concatenated signal encountered when descending the tree.

In case of bundling, the explicit list of all signals that take part in the bundling is given. An example of bundling is inverse multiplexing, it is useful when a higher order signal needs to be transported over a number of lower order signals, e.g. when a 10Gbps signal must be transported over four 2.5Gbps signals. In that case, the lower order signals must follow exactly the same path, and be treated in the same way, in order to achieve the same characteristics (e.g. delay). To support inverse multiplexing, a request is made to open in parallel and in one single operation several LSPs at the same time.

The format of the label for SDH and/or SONET TDM-LSR link 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   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For SDH, this is an extension of the numbering scheme defined in G.707 section 7.3, i.e. the (K, L, M) numbering. For SONET, the U and K fields are not significant and must be set to zero. Only the S, L and M fields are significant for SONET and have a similar meaning as for SDH.

Each letter indicates a possible branch number starting at the parent node in the multiplex structure. Branches are considered as numbered in increasing order, starting from the top of the multiplexing structure. The numbering starts at 1, zero is used to indicate a non-significant field.

When a field is not significant in a particular context it MUST be set to zero when transmitted, and MUST be ignored when received. This simple rule allows distinguishing very easily between an SDH label and an SONET label. A label with U=0 will always indicate a SONET label. This is a nice feature for debugging purposes. Note that it is easier to test U and K together, rather than only the U field alone, since they fit exactly in the third octet of the label.

1. S is the index of a particular STM-1/STS-1 signal. S=1->N indicates a specific STM-1/STS-1 inside an STM-N/STS-N multiplex. For example, S=1 indicates the first STM-1/STS-1, and S=N indicates the last STM-1/STS-1 of this multiplex. S=0 is invalid.

2. U is only significant for SDH and must be ignored for SONET. It indicates a specific VC inside a given STM-1. U=1 indicates a single VC-4, while U=2->4 indicates a specific VC-3 inside the given STM-1.

3. K is only significant for SDH and must be ignored for SONET. It indicates a specific branch of a VC-4. K=1 indicates that the VC-4 is not further sub- divided and contains a C-4. K=2->4 indicates a specific TUG-3 inside the VC-4. K is not significant when the STM-1 is divided into VC-3s (easy to read and test).

4. L indicates a specific branch of a TUG-3, VC-3 or STS-1 SPE. It is not significant for an unstructured VC-4. L=1 indicates that the TUG-3/VC-3/STS-1 SPE is not further sub-divided and contains a VC-3/C-3 in SDH or the equivalent in SONET. L=2->8 indicates a specific TUG-2/VT Group inside the corresponding higher order signal.

5. M indicates a specific branch of a TUG-2/VT Group. It is not significant for an unstructured VC-4, TUG-3, VC-3 or STS-1 SPE. M=1 indicates that the TUG-2/VT Group is not further sub-divided and contains a VC-2/VT-6. M=2->3 indicates a specific VT-3 inside the corresponding VT Group, these values MUST NOT be used for SDH since there is no equivalent of VT-3 with SDH. M=4->6 indicates a specific VC-12/VT-2 inside the corresponding TUG-2/VT Group. M=7->10 indicates a specific VC-11/VT-1.5 inside the corresponding TUG-2/VT Group. Note that M=0 denotes an unstructured VC-4, VC-3 or STS-1 SPE (easy for debugging).

The M encoding is summarized in the following table:

M	SDH	SONET
0	unstructured VC-4/VC-3	unstructured STS-1 SPE
1	VC-2	VT-6
2	-	1st VT-3
3	-	2nd VT-3
4	1st VC-12	1st VT-2
5	2nd VC-12	2nd VT-2
6	3rd VC-12	3rd VT-2
7	1st VC-11	1st VT-1.5
8	2nd VC-11	2nd VT-1.5
9	3rd VC-11	3rd VT-1.5
10	4th VC-11	4th VT-1.5

For instance,

Example 1: S>0, U=1, K=1, L=0, M=0
Denotes the unstructured VC-4 of the Sth STM-1.

Example 2: S>0, U=1, K>1, L=1, M=0
Denotes the unstructured VC-3 of the Kth-1 TUG-3 of the Sth STM-1.

Example 3: S>0, U=0, K=0, L=0, M=0)
Denotes the unstructured STS-1 SPE of the Sth STS-1.

Example 4: S>0, U=0, K=0, L>1, M=1
Denotes the VT-6 in the Lth-1 VT Group in the Sth STS-1.

Example 5: S>0, U=0, K=0, L>1, M=9
Denotes the 3rd VT-1.5 in the Lth-1 VT Group in the Sth STS-1.

3.2.1.2. Port and Wavelength Labels

Some configurations of fiber switching (FSC) and lambda switching (LSC) use multiple data channels/links controlled by a single control channel. In such cases the label indicates the data channel/link to be used for the LSP. Note that this case is not the same as when [MPLS-BUNDLING] is being used.

The information carried in a Port and Wavelength label 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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].

3.2.1.3. 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.

3.3. 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.

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

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Waveband Id                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                          Start Label                          |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                           End Label                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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 label parameter of the Generalized Label one PSC and LSC.

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.

The information carried in a suggested label is identical to a generalized label.

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 is 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 it's 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

A label set is composed of one or more Label_Set objects/TLVs. Each object/TLV contains one or more elements of the Label Set. Each element is referred to as a subchannel identifier and has the same format as a label.

The information carried in a Label_Set 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Reserved           |  Label Type   |    Action     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Subchannel 1                         |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
:                               :                               :
:                               :                               :
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Subchannel N                         |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Label Type: 8 bits

Indicates the type and format of the labels carried in the object/TLV. Values are signaling protocol specific.

Action: 8 bits

0 - Inclusive List

Indicates that the object/TLV contains one or more subchannel elements that are included in the Label Set.

1 - Exclusive List

Indicates that the object/TLV contains one or more subchannel elements that are excluded from the Label Set.

2 - Inclusive Range

Indicates that the object/TLV contains a range of labels. The object/TLV contains two subchannel elements. The first element indicates the start of the range. The second element indicates the end of the range. A value of zero indicates that there is no bound on the corresponding portion of the range.

3 - Exclusive Range

Indicates that the object/TLV contains a range of labels that are excluded from the Label Set. The object/TLV contains two subchannel elements. The first element indicates the start of the range. The second element indicates the end of the range. A value of zero indicates that there is no bound on the corresponding portion of the range.

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.

Note that subchannel to local channel identifiers (e.g., wavelength) mappings are a local matter.

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, LSRs, 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 signaling 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 object/TLV in the appropriate signaling message. An Upstream Label has the same format as the generalized label, see Section 3.2.

4.2. 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.

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. One special case that should be noted when using RSVP and supporting this approach is that the neighbor's node ID might not be known when sending an initial Path message. When this case occurs, a node should suggest a label choosen at random from the available label space.

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).

```
          +------------+                         +------------+
          +   PXC 1    +                         +   PXC 2    +
          +            +                 SL1,UL2 +            +
          +          1 +------------------------>+ 6          +
          +            + UL1, SL2                +            +
          +          2 +<------------------------+ 7          +
          +            +                         +            +
          +            +                         +            +
          +          3 +------------------------>+ 8          +
          +            +                         +            +
          +          4 +<------------------------+ 9          +
          +------------+                         +------------+
                       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).

```
          +------------+                         +------------+
          +   PXC 1    +                         +   PXC 2    +
          +            +                     UL2 +            +
          +          1 +------------------------>+ 6          +
          +            + UL1                     +            +
          +          2 +<------------------------+ 7          +
          +            +                         +            +
          +            +                      L1 +            +
          +          3 +------------------------>+ 8          +
          +            + L2                      +            +
          +          4 +<------------------------+ 9          +
          +------------+                         +------------+
     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.

```
           +------------+                          +------------+
           +   PXC 1    +                          +   PXC 2    +
           +            +                      UL2 +            +
           +          1 +------------------------->+ 6          +
           +            + L2                       +            +
           +          2 +<-------------------------+ 7          +
           +            +                          +            +
           +            +                       L1 +            +
           +          3 +------------------------->+ 8          +
           +            +  UL1                     +            +
           +          4 +<-------------------------+ 9          +
           +------------+                          +------------+
```

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. Explicit Label Control

The LSR at the initiator of an LSP can control nodes used by an LSP and the termination of the LSP by using an explicit route, i.e., ERO or ER-Hop. To require the usage of a particular node, that node is included in the explicit route. To terminate an LSP on a particular outgoing interface of the egress LSR, the head-end may specify the IP address or the interface identifier [MPLS-UNNUM] of that interface as the last element in the explicit route, provided that that interface has an associated IP address.

There are cases where the existing explicit route semantics do not provide enough information to control the LSP to the degree desired. This occurs in the case when the LSP initiator wishes to select a label used on a link. An example of this 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.

To to cover this case, the Label ERO subobject / ER Hop is introduced.

5.1. Required Information

The Label Explicit Route contains:

L: 1 bit

This bit must be set to 0.

U: 1 bit

This bit indicates the direction of the label. It is 0 for the downstream label. It is set to 1 for the upstream label and is only used on bidirectional LSPs.

Label: Variable

This field identifies the label to be used. The format of this field is identical to the one used by the Label field in Generalized Label, see Section 3.2.1.

Placement and ordering of these parameters are signaling protocol specific.

6. 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

Valuable comments and input were received from a number of people, including Igor Bryskin and Adrian Farrel.

7. Security Considerations

This draft introduce no new security considerations to either [CR-LDP] or [RSVP-TE].

8. 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-11.txt, August 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.

[GMPLS-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 Generalized MPLS", Internet Draft, draft-ietf-isis-gmpls-extensions-00.txt, July 2000.

[GMPLS-LDP] Ashwood-Smith, P. et al, "Generalized MPLS Signaling - CR-LDP Extensions", Internet Draft, draft-ietf-mpls-generalized-cr-ldp-00.txt, November 2000.

[GMPLS-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 MPLambdaS", Internet Draft, draft-ompls-ospf-extensions-00.txt, July 2000.

[GMPLS-RSVP] Ashwood-Smith, P. et al, "Generalized MPLS Signaling - RSVP-TE Extensions", Internet Draft, draft-ietf-mpls-generalized-rsvp-te-00.txt, November 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.

[RECOVERY] Makam, et al "A Framework for MPLS-based Recovery," draft-ieft-mpls-recovery-frmwrk-00.txt, August 2000.

9. 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

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

Lou Berger
Movaz Networks
Phone: +1 301 468 9228
Email: lberger@movaz.com

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

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

Yanhe Fan
Axiowave Networks, Inc.
100 Nickerson Road
Marlborough, MA 01752
Phone: +1 508 460 6969 Ext. 627
Email: yfan@axiowave.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
GTS
Terhulpsesteenweg 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@gts.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

George Swallow
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA 01824
Voice: +1 978 244 8143
Email: swallow@cisco.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 Nov 17 11:56:41 EST 2000