Network Working Group
Internet Draft
Expiration Date: April 2001

Peter Ashwood-Smith (Nortel Networks Corp.)
Ayan Banerjee (Calient Networks)
Lou Berger (Movaz Networks)
Greg Bernstein (Ciena Corporation)
John Drake (Calient Networks)
Yanhe Fan (Axiowave Networks)
Kireeti Kompella (Juniper Networks, Inc.)
Eric Mannie (GTS Network Services)
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.)

October 2000

Generalized MPLS - Signaling Functional Description

draft-ietf-mpls-generalized-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

Changes from previous version:

- Removed Link ID in labels, redundant with [MPLS-BUNDLE].
- Revised Generalized Label Request and SDH/SONET encodings
- Added Notify Request object
- Added multiple sender notifies per Notify message
- Added ability to remove path state on sending PathErr
- Removed Non-Adjacent Message Bundling
- Enabled label specification via ERO/ER
- Added Section 7
- 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 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 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 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.

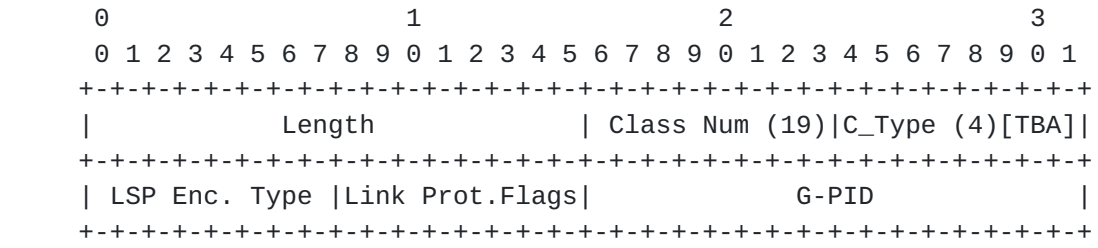
The Generalized Label Request indicates the link protection type desired for the LSP. If a particular protection type, i.e., 1+1, or 1:N, is requested, then a connection request is processed only if the desired protection type can be honored. Note that the protection capabilities of a link may be advertised in routing, see [GMPLS-ISIS,

GMPLS-OSPF]. Path computation algorithms may take this information into account when computing paths for setting up LSPs.

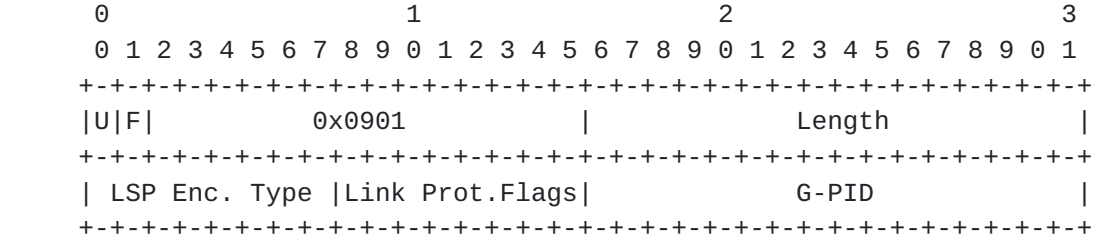
The Generalized Label Request also carries an LSP encoding parameter, called LSP Encoding Type. This parameter indicates the encoding type, e.g., SONET/SDH/GigE etc., that will be used with the data associated with the LSP. The LSP Encoding Type represents the nature of the LSP, and not the nature of 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. A REQUEST/Path message SHOULD contain as specific a LSP Encoding Type as possible to allow the maximum flexibility in switching by transit LSRs. A Generalized Label Request object/TLV is set by the ingress node, transparently passed by transit nodes, and used by the egress node.

3.1.1. Generalized Label Request Format

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



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



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:

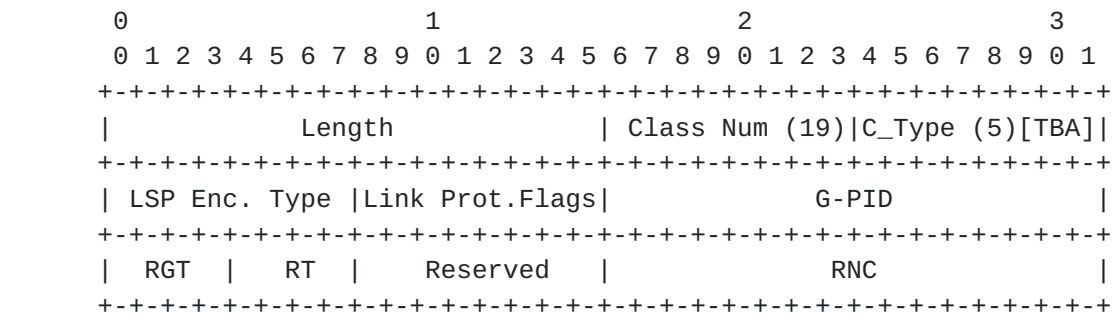
Value	Type	Technology
0	Unknown	All
1	DS1 SF	ANSI-PDH
2	DS1 ESF	ANSI-PDH
3	DS3 M23	ANSI-PDH
4	DS3 C-Bit Parity	ANSI-PDH
5	Asynchronous mapping of E4	SDH
6	Asynchronous mapping of DS3/T3	SDH
7	Asynchronous mapping of E3	SDH
8	Bit synchronous mapping of E3	SDH
9	Byte synchronous mapping of E3	SDH
10	Asynchronous 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

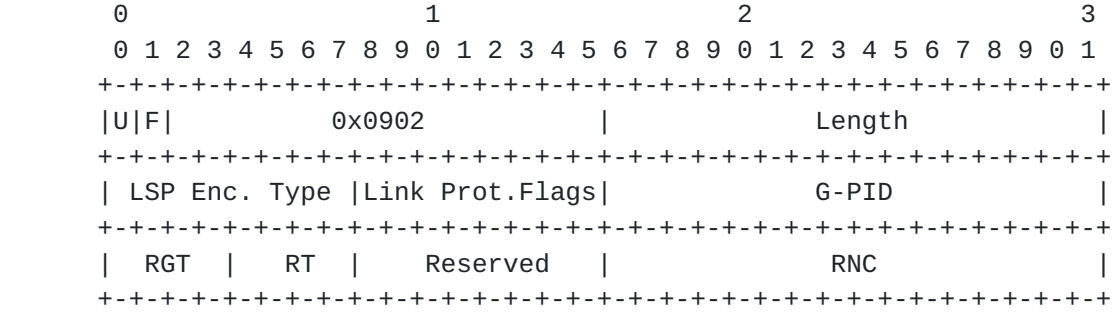
The Generalized Label Request with SONET/SDH Label Range object/TLV is used to represent specific characteristics related to the two TDM technologies. If the RGT, RT, and RNC, fields are all set to zero, it means that no concatenation, bundling or transparency is requested. If the requested LSP is itself 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 format of a Generalized Label Request with SONET/SDH Label Range (in RSVP) is:



The format of a Generalized Label Request with SONET/SDH label range (in CR-LDP) is:



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	Inverse Multiplexing

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. Procedures

A node processing the Path/REQUEST message containing the Generalized Label Request must verify that the requested parameters can be satisfied by the incoming interface, 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 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.

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

3.1.4. Bandwidth Encoding

Bandwidth encodings are carried in the SENDER_TSPEC and FLOWSPEC objects in RSVP and in the Traffic Parameters TLV in CR-LDP. They are carried 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. These values are set in the Peak Data Rate field of Int-Serv objects carried in RSVP messages and in the Peak and Committed Data Rate fields of the Traffic Parameters TLV carried in CR-LDP messages. Other bandwidth/service related parameters in the object/TLV are ignored and carried transparently. Some typical values for the requested bandwidth are enumerated below. Additional values will be defined as needed.

Signal Type	(Bit-rate)	Value
DS0	(0.064 Mbps)	0x477A0000
DS1	(1.544 Mbps)	0x49BC7A00
E1	(2.048 Mbps)	0x49FA0000
DS2	(6.312 Mbps)	0x4AC0A080
E2	(8.448 Mbps)	0x4B00E800
Ethernet	(10.00 Mbps)	0x4B189680
E3	(34.368 Mbps)	0x4C031A80
DS3	(44.736 Mbps)	0x4C2AA780
STS-1	(51.84 Mbps)	0x4C45C100
Fast Ethernet	(100.00 Mbps)	0x4CBEBC20
E4	(139.264 Mbps)	0x4D04D000
OC-3/STM-1	(155.52 Mbps)	0x4D1450C0
OC-12/STM-4	(622.08 Mbps)	0x4E1450C0
GigE	(1000.00 Mbps)	0x4E6E6B28
OC-48	(2488.32 Mbps)	0x4F1450C0
OC-192	(9953.28 Mbps)	0x501450C0
10GigE-LAN	(10000.00 Mbps)	0x501502F9

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).

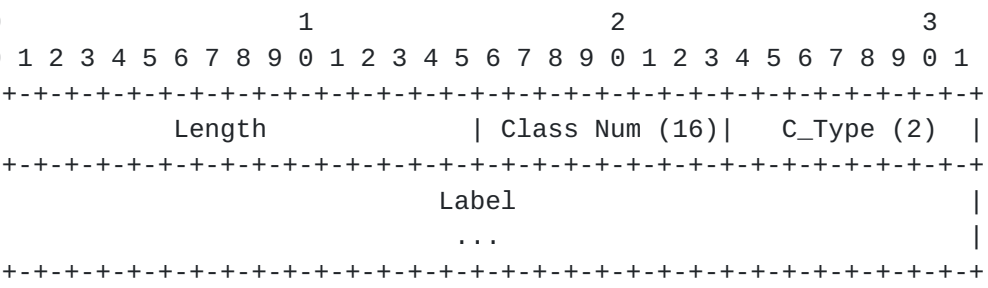
A Generalized Label does not identify the "class" to which the label belongs. This is implicit in the multiplexing capabilities of the link on which the label is used.

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].

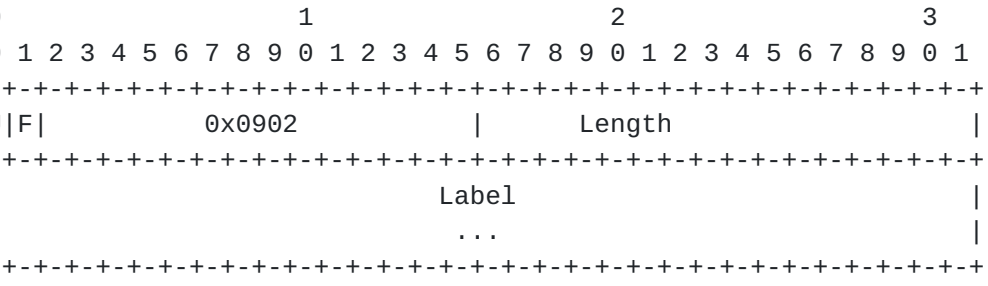
Each Generalized Label object carries a variable length label parameter.

3.2.1. Required Information

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



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



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 define each 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 directly the distinction between SDH and SONET.

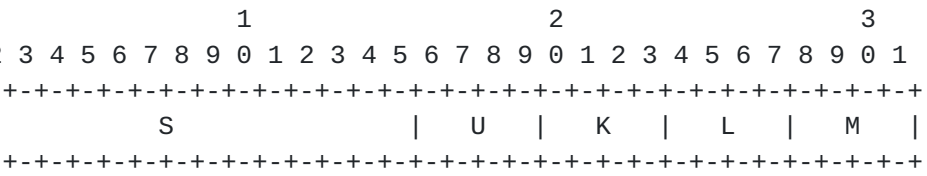
In case of signal concatenation or inverse multiplexing, a list of labels may appear in the Label field of a Generalized Label.

In case of virtual or arbitrary 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 inverse multiplexing, the explicit list of all signals that take part into the inverse multiplexing is given. Inverse multiplexing is useful when a higher order signal need 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:



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 the 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.

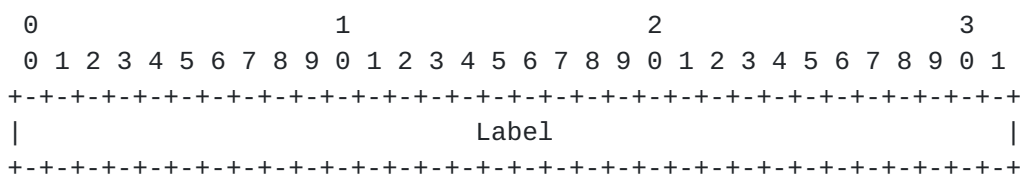
Example 4: S>0, U=0, K=0, L>1, M=1
Denotes the VT-6 in the Lth-1 VT Group in the Sth STS-1.

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 format of a Port and Wavelength label is:



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.

3.2.2. 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 Resv/MAPPING message is a protocol error and should treated as a malformed message by the recipient.

The recipient of a Resv/MAPPING message containing a Generalized Label verifies that the values passed are acceptable. If the 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.

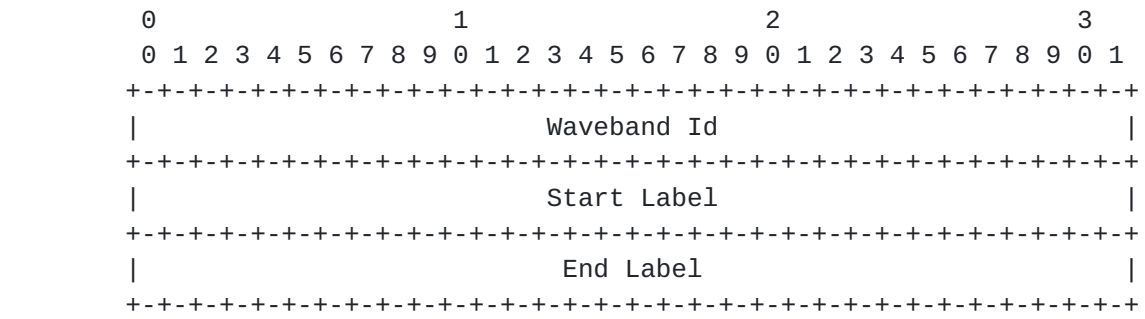
Waveband switching naturally introduces another level of label hierarchy and as such the waveband is treated the same way all other upper layer labels are treated.

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 label parameter of the Generalized Label one PSC and LSC.

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 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.

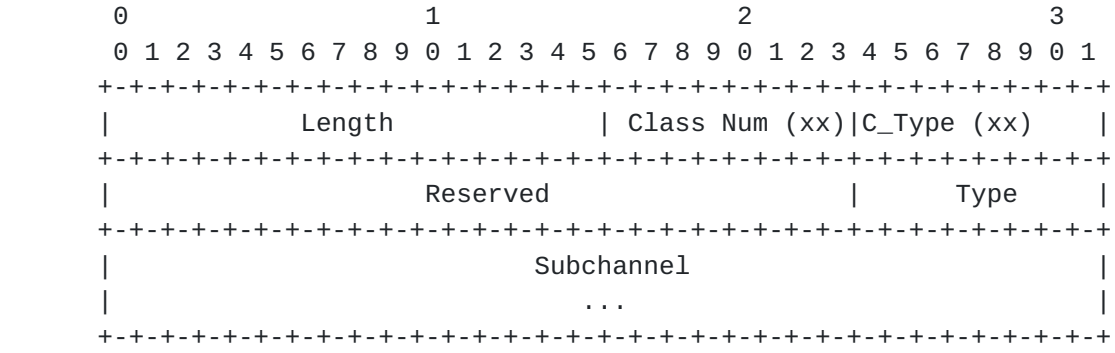
The use of Label Set is optional, if not present, all labels from the valid label range may be used. Conceptually the absence of a Label Set implies a Label Set whose value is {U}, the set of all valid labels.

3.5.1. Required Information

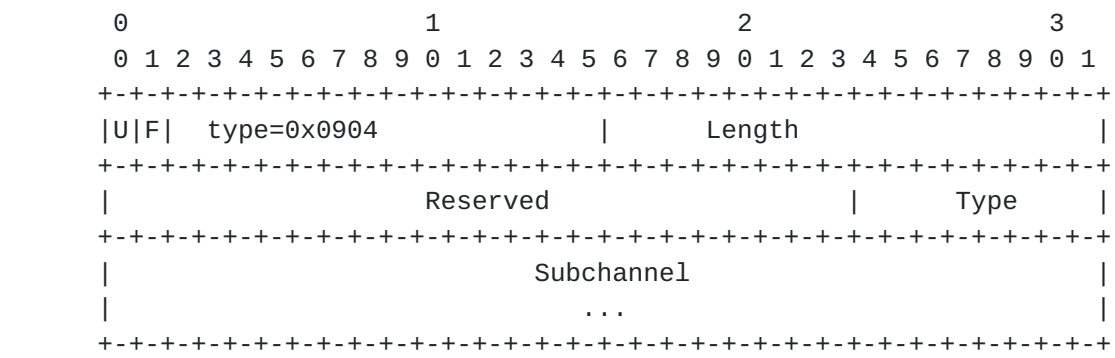
This Label_Set is used in Path/REQUEST messages.

The data required to support the Label Set consists of a variable sized array of labels, or label ranges. These labels are subchannel identifiers.

The format of a Label_Set (in RSVP) is:



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



Type: 2 bits

0 means subchannel is a single element (inclusive)
1 means subchannel is a start element (inclusive)
2 means subchannel is an end element (inclusive)
3 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=0) sorted in increasing order of subchannel value.

A Label Set can be a set of ranges (Type=1 followed by Type=2). 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=3 (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 data traffic associated with the LSP upstream towards 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).

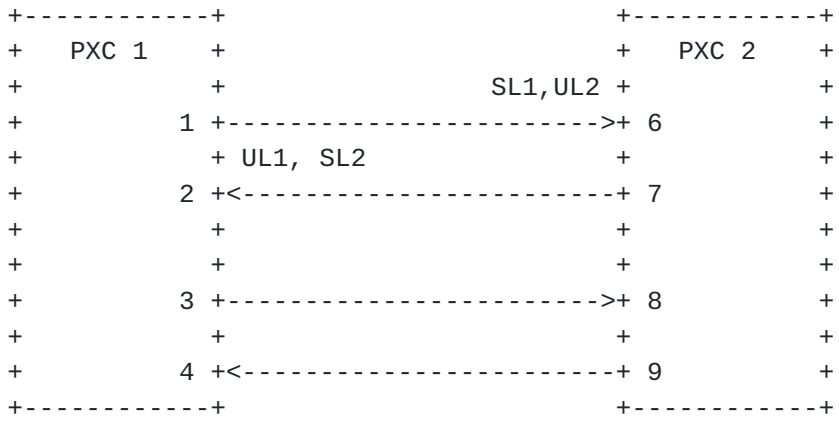


Figure 1. Label Contention

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).

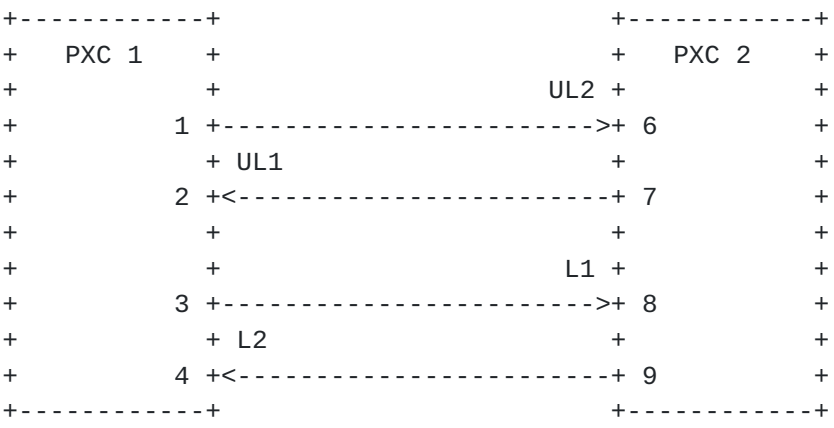


Figure 2. Label Contention Resolution without resource restrictions

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

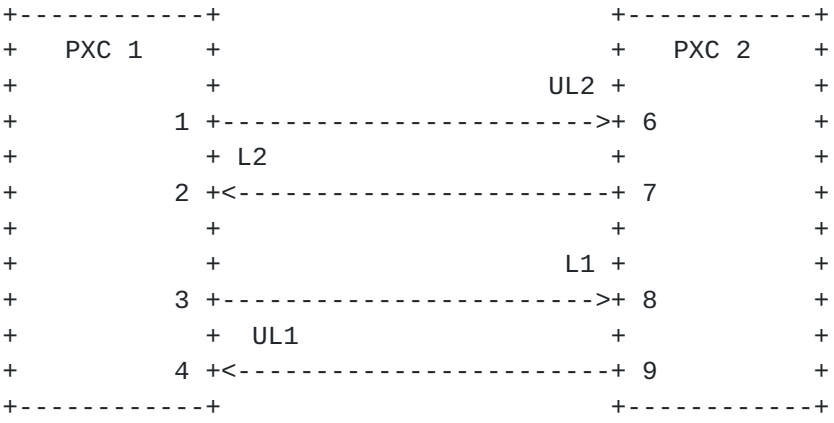


Figure 3. Label Contention Resolution with resource restrictions

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

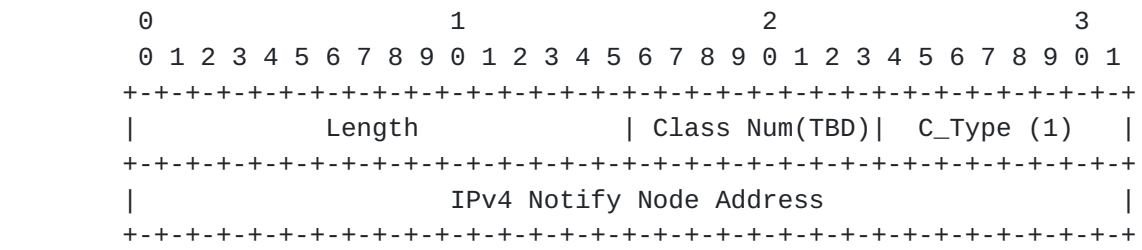
This section defines three signaling extensions that modify error handling, enable expedited notification of failures and other events to nodes responsible for restoring failed LSPs. The first extension, the Notify Request object, identifies where event notifications are to be sent. The second, the Notify message, provides for general event notification. The final extension allows for the removal of Path state on handling of PathErr messages.

5.1. Notify Request Object

Notifications may be sent via the Notify message defined below. The Notify Request object is used to request the generation of notifications. Notifications, i.e., the sending of a Notify message, may be requested in both the upstream and downstream directions.

5.1.1. Required Information

The Notify Request Object may be carried in Path or Resv Messages, see Section 7. The NOTIFY_REQUEST class number is TBA (of form 11bbbbbb). The format of a Notify Request is:



IPv4 Notify Node Address: 32 bits

The IP address of the node that should be notified when generating an error message.

5.1.2. Procedures

A Notify Request object may be inserted in Path or Resv messages to indicate the address of a node that should be notified of an LSP failure. As previously mentioned, notifications may be requested in both the upstream and downstream directions. Upstream notification is indicated via the inclusion of a Notify Target Object in the corresponding Path message. Downstream notification is indicated via the inclusion of a Notify Target Object in the corresponding Resv message.

A node receiving a message containing a Notify Request object SHOULD store the Notify Node Address in the corresponding state block. If the node is a transit node, it SHOULD also included a Notify Request object in the outgoing Path or Resv message. The outgoing Notify Node Address MAY be updated based on local policy.

Note that the inclusion of a Notify Request object does not guarantee that a Notify message will be generated.

5.2. Notify Message

The Notify message provides a mechanism to inform non-adjacent nodes of LSP related events. Notify messages are only generated after a Notify Request object has been received. The Notify 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 who's 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. See [RSVP-RR] for details on reliable RSVP message delivery.

5.2.1. 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>
                     <ERROR_SPEC> <notify session list>

<notify session list> :== [<notify session list>] <notify session>

<notify session> :== <SESSION> [<POLICY_DATA>...]
                     <sender descriptor>
```

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.

5.2.2. Procedures

Notify messages are generated at nodes that detect an error that will trigger the generation of a PathErr or ResvErr message. If a PathErr message is to be generated and a Notify Request object has been received in the corresponding Path message, then a Notify message destined to the recorded node SHOULD be generated. If a ResvErr message is to be generated and a Notify Request object has been received in the corresponding Resv message, then a Notify message destined to the recorded node SHOULD be generated. As previously mentioned, a single error may generate a Notify message in both the upstream and downstream directions. Note a Notify message MUST NOT be generated unless an appropriate Notify Request object has been received.

When generating Notify messages, a node SHOULD attempt to combine notifications being sent to the same Notify Node and that share the same ERROR_SPEC into a single Notify message. The means by which a node determines which information may be combined is implementation dependent. Implementations may use event, timer based or other approaches. If using a timer based approach, the implementation SHOULD allow the user to configure the interval over which notifications are combined. When using a timer based approach, a default "notification interval" of 1 ms SHOULD be used. Notify messages SHOULD be delivered using the reliable message delivery mechanisms defined in [RSVP-RR]

Upon receiving a Notify message, the Notify Node SHOULD send a corresponding Ack message.

5.3. Removing State with a PathErr message

The PathErr message as currently defined in [RFC2205] is sent hop-by-hop to the source of the associated Path message. Intermediate nodes may inspect this message, but take no action upon it. In an environment where Path messages are routed according to an IGP and that route may change dynamically, this behavior is a fine design choice.

However, when RSVP is used with explicit routes, it is often the case that errors can only be corrected at the source node or some other node further upstream. In order to clean up resources, the source must receive the PathErr and then either send a PathTear (or wait for the messages to timeout). This causes idle resources to be held

longer than necessary increases control message load. In a situation where the control plane is attempting to recover from a serious outage, both the message load and the delay in freeing resources hamper the ability to rapidly reconverge.

The situation can be greatly improved by allowing state to be removed by intermediate nodes on certain error conditions. To facilitate this a new flag is defined in the ERROR_SPEC object. The two currently defined ERROR_SPEC objects (IPv4 and IPv6 error spec objects) each contain a one byte flag field. Within that field two flags are defined. This specification defines a third flag, 0x04, Path_State_Removed.

The semantics of the Path_State_Removed flag are simply that the node forwarding the error message has removed the Path state associated with the PathErr. By default, the Path_State_Removed flag is always set to zero when generating or forwarding a PathErr message. A node which encounters an error MAY set this flag if the error results in the associated Path state being discarded. If the node setting the flag is not the session endpoint, the node SHOULD generate a corresponding PathTear. A node receiving a PathErr message containing an ERROR_SPEC object with the Path_State_Removed flag set MAY also remove the associated Path state. If the Path state is removed the Path_State_Removed flag SHOULD be set in the outgoing PathErr message. A node which does not remove the associated Path state MUST NOT set the Path_State_Removed flag. A node that receives an error with the Path_State_Removed flag set to zero MUST NOT set this flag unless it also generates a corresponding PathTear message.

Note that the use of this flag does not result in any interoperability incompatibilities.

6. 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.

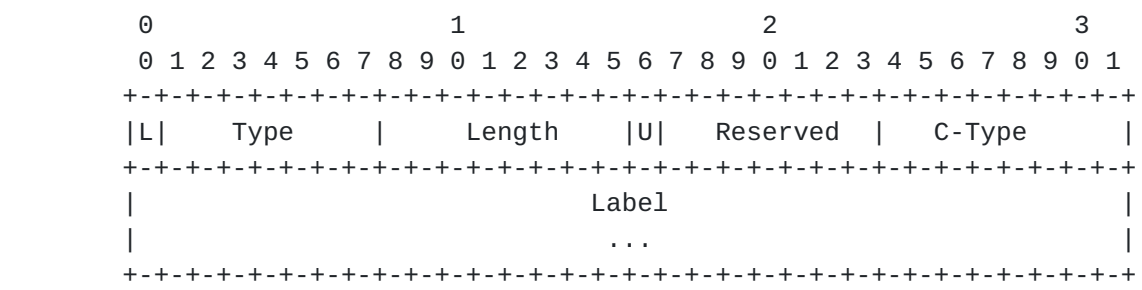
There are cases where the existing explicit route semantics do not provide enough information to control the LSP to the degree desired. This occurs case when the LSP initiator wishes to select a label used on a link. An example of this is when it is desirable to "splice"

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.

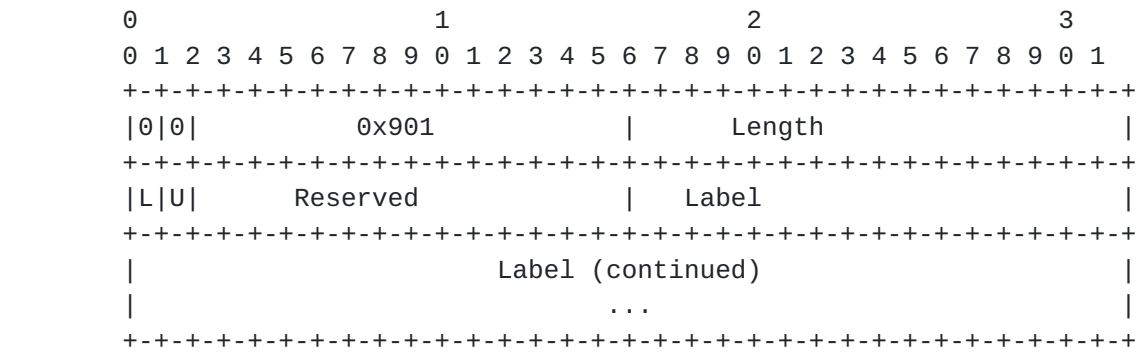
To to cover this case, the Label ERO subobject is introduced.

6.1. Required Information

For RSVP, this ERO subobject - Label is defined as follows:



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



L

This bit must be set to 0.

Type (RSVP Only)

3 Label

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.

U

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.

C-Type (RSVP Only)

The C-Type of the included Label Object. Copied from the Label Object.

Label

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

6.2. Procedures

The Label subobject follows a subobject containing the IP address, or the interface identifier [MPLS-UNNUM], associated with the link on which it is to be used. The preceding subobject must be a strict object. Up to two label subobjects may be present, one for the downstream label and one for the upstream label. The following SHOULD result in "Bad EXPLICIT_ROUTE object" errors:

- For a label subobject to follow a subobject that has the L-bit set
- On unidirectional LSP setup, for there to be a label subobject with the U-bit set
- For there to be two label subobjects with the same U-bit values

To support the label subobject, a node must check to see if the subobject following it's associate address/interface is a label subobject. If it is, one subobject is examined for unidirectional LSPs and two subobjects for bidirectional LSPs. If the U-bit of the subobject being examined is clear (0), then value of the label is copied into a new Label_Set object. This Label_Set object MUST be included on the corresponding outgoing Path/Mapping message.

If the U-bit of the subobject being examined is set (1), then value of the label is label to be used for upstream traffic associated with the bidirectional LSP. If this label is not acceptable, a "Bad EXPLICIT_ROUTE object" error SHOULD be generated. If the label is acceptable, the label is copied into a new Upstream Label object. This Upsteam Label object MUST be included on the corresponding outgoing Path/Mapping message.

After processing, the label subobjects are removed from the ERO/ER.

Note an implication of the above procedures is that the label subobject should never be the first subobject in a newly received message. If the label subobject is the the first subobject an a received ERO/ER, then it SHOULD be treated as a "Bad strict node" error.

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

7. RSVP Message Formats

This section presents the RSVP message related formats as modified by this document. Where they differ, formats for unidirection LSPs are presented separately from bidirectional LSPs. Unmodified formats are not listed.

The format of a Path message is as follows:

```
<Path Message> ::=      <Common Header> [ <INTEGRITY> ]
                        <SESSION> <RSVP_HOP>
                        <TIME_VALUES>
                        [ <EXPLICIT_ROUTE> ]
                        <LABEL_REQUEST>
                        <LABEL_SET>
                        [ <SESSION_ATTRIBUTE> ]
                        [ <NOTIFY_REQUEST> ]
                        [ <POLICY_DATA> ... ]
                        <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

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

The format of the sender description for bidirectional LSPs is:

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

The format of a Resv message is as follows:

```
<Resv Message> ::=      <Common Header> [ <INTEGRITY> ]
                        <SESSION>  <RSVP_HOP>
                        <TIME_VALUES>
                        [ <RESV_CONFIRM> ]  [ <SCOPE> ]
                        [ <NOTIFY_REQUEST> ]
                        [ <POLICY_DATA> ... ]
                        <STYLE> <flow descriptor list>
```

<flow descriptor list> is not modified by this document.

8. 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, notably Adrian Farrel. Portions of Section 5 are based on suggestions and text proposed by Adrian.

9. Security Considerations

Non-adjacent bundle messages, and the transmission of notify messages using IP in IP, break RSVP's hop-by-hop integrity and authentication model. Fortunately, such usage mirrors the IP end-to-end model. In the case where RSVP is generating end-to-end messages and integrity and/or authentication are desired, the standard IPSEC based integrity and authentication methods SHOULD be used.

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

10. 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-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.

[RFC2205] Braden, R. Ed. et al, "Resource ReserVation Protocol -- Version 1 Functional Specification", RFC 2205, September 1997.

[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.

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

11. 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
Optical Networking
Network R&D
GTS Network Services
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@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

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 Oct 20 11:15:15 EDT 2000