Contribution Number: OIF2000.125.3

Working Group: Architecture, OAM&P, PLL, & Signaling Working Groups

TITLE: User Network Interface (UNI) 1.0 Signaling Specification

DATE: December, 8, 2000

Document Status:
Project Name: Multiple OIF Projects
Project Number:

For additional information contact:
The Optical Networking Forum, 39355 California Street,
Suite 307, Fremont, CA 94538
510-608-5990 phone ✦ info@oiforum.com

List of Contributors

Osama Abul-Magd, Nortel
Stefan Ansorge Alcatel
K. Arvind, Tenor Networks
Krishna Bala, Tellium (Chair, Signaling WG)
Ayan Banerjee, Calient
Rick Barry, Sycamore (Chair, Architecture WG)
Debashis Basak, Accelight Networks
Greg Bernstein, Ciena
Curtis Brownmiller, Worldcom (Chair, OAMP WG)
Yang Cao, Sycamore
John Drake, Calient
William Goodson, Lucent
Gert Grammel, Alcatel
Eric Gray, Brightwave
Riad Hartini, Caspian Networks
Raj Jain, Nayna Networks
LiangYu Jia, ONI Systems
Jim Jones, Alcatel
Suresh Katukam, Cisco
Prabhu Kavi, Tenor Networks
Nooshin Komaee, Tellium
Kireeti Kompella, Juniper
Jonathan P. Lang, Calient
Monica Lazer, AT&T
Fong Liaw, Zaffire
Larry McAdams, Cisco
Yassi Moghaddam, Avici
Dimitiri Papadimitriou, Alcatel
Dimitrios Pendarakis, Tellium
Kavi Prabhu, Tenor Networks
Bala Rajagopalan, Tellium (Editor)
Rajiv Ramaswamy, Nortel
Anil Rao, ONI Systems
Robert Rennison, Laurel Networks
Yakov Rekhter, Cisco
Debanjan Saha, Tellium
Arnold Sodder, Tenor Networks
John Strand, AT&T (Carrier Group Rep.)
George Swallow, Cisco
Yangguang Xu, Lucent
Tao Yang, Sycamore
Jennifer Yates, AT&T
John Z. Yu, Zaffire
Alex Zinin, Cisco
Zhensheng Zhang, Sorrento

Table of Contents

1 Document Summary

1.1 Working Group project(s)

1.2 Working Group(s)

Architecture Working Group
OAM&P Working Group
PLL Working Group
Signaling Working Group

1.3 Problem Statement

This document defines a set of services, physical transport technologies and signaling capabilities that may be implemented by manufacturers and services providers to support users. This document is scoped to allow an early implementation of an open optical network layer that enables dynamic connectivity to client layers like IP, ATM, SONET and others.

1.4 Scope

The scope of this implementation agreement is to define a set of services, physical transport technologies and signaling capabilities that may be implemented by manufactures and services providers to support users. This document is scoped to allow an early implementation based on current and newly available technologies and capabilities. This implementation agreement is not intended to restrict future additions. UNI is defined as the interface between the service provider network and user equipment (such as IP routers and other end systems), as illustrated in Figure 1-1. Figure 1-1 distinguishes between the transport links and the signaling link at the UNI. Furthermore, the figure illustrates two types of UNI interfaces covered in this specification. The left user domain illustrates UNI signaling between client and network entities that are not co-located with the corresponding network elements – also referred to as third party signaling. The right user domain illustrates a UNI over which direct signaling occurs between client and optical network elements over the transport links – also referred to as channel associated signaling and in-band signaling.

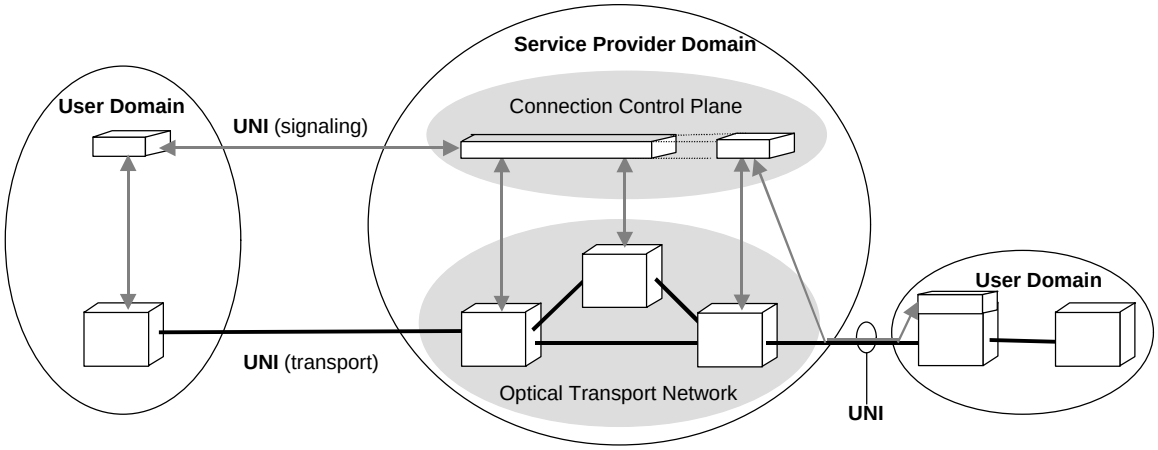


Figure 1-1 - UNI

The UNI 1.0 specification focuses on SONET/SDH framed signals at the client layer.

1.5 Expected Outcome

This document specifies an implementable set of services, interfaces, and signaling capabilities in support of UNI 1.0.

1.6 Merits to OIF

The UNI 1.0 specification is a key step towards the implementation of an open optical network layer that allows dynamic interconnection of client layers like IP, ATM, SONET and others. This activity supports the overall mission of the OIF.

1.7 Relationship to other Standards Bodies

This document, to the maximum extent possible, utilizes standards and specifications already available from other organizations, most notably, the IETF. This document also specifies certain new capabilities. It is expected that the OIF would liaison this information to the appropriate organization for formalization.

1.8 Unique Viewpoint

This document is unique in that it consolidates many aspects with respect to the work being done at other standards bodies. In particular, it addresses the optical internetworking requirements for

- Services associated with data clients;
- Rapid provisioning of SONET/SDH framed circuits through an optical internetwork; and
- Client to optical network signaling

2 Terminology and Abbreviations

2.1 Terminology

Client/Server	Generic terms used to differentiate the initiator in a relationship (the client) from the responder and service provider (the server). The term "client" and "server" are multi-used and so should not be used in isolation, without qualification or context.
Client-Layer	A layer acting as a client with regard to transport services provided by a server layer.
Client-Layer Address	An address used in client-layer protocols
Connection Identifier	A identifier for the connection that is unique within the scope of the service provider network.
Connection Termination Point	The client side of an adaptation function. Refer to M.3100 for complete definition of connection termination point.
Contract identifier	An identifier which refers to the service contract under which a connection is established. This identifier is created by the service provider.
Channel Identifier	When multiple channels are multiplexed onto the same port, this identifier is used to uniquely identify a channel.
In-Band Signaling	In-band signaling refers to the transport of signaling traffic and data traffic over the same connection in the same layer.
In-Fiber Signaling	In-fiber signaling refers to the transport of signaling traffic and data traffic over the same fiber, although not necessarily in the same layer.
Layer Network	For SONET Layers refer to ANSI T1.105 For SDH Layers refer to ITU-T G.803 For OTN Layers refer to ITU-T G.872
Layer-Network	A set of potentially connectable Access-Points (AP) all of the same Characteristic-Information (CI). A layer network can be either or both a client-layer-network and/or a server-layer-network. (It can be isolated, or it can be at the top or bottom of a stack, or it can be in the middle of a stack of layer-networks.) Refer to G.805 for complete definition of layer-network.
Link	Represents the available transport capacity between two sub-networks. Refer to G.805 for complete definition of link.
Link Connection	An entity that represents the smallest granularity capacity that can be allocated on a link. Refer to G.805 for complete definition of link connection.
Network Connection	Entity that transports information transparently across a particular network layer without any check on the quality of the transport. Refer to G.805 for complete definition of network connection.
Out-of-Band Signaling	Out-of-band signaling refers to the transport of signaling traffic and data traffic over separate connections, and possibly in different layers.
Out-of-Fiber Signaling	Out-of-fiber signaling refers to the transport of signaling traffic and data traffic over a separate fiber.
Port Index	A locally unique identifier within an optical network element for a switch port.

Server Layer	The server layer provides transparent transport for the client layer. Refer to G.805 for complete definition of server layer.
Service Level	A parameter in UNI signaling messages indicating the level of service desired. The service level is interpreted by the service provider based on contracted services.
Signal Type Identifier	A SDH/SONET name, such as STS-1.
Sub-Channel Identifier	When multiple sub-channels are multiplexed onto a channel, this identifier is used to uniquely identify a sub-channel.
Sub-network	In the context of the connection domain, an entity that represents flexible connectivity; there is no notion of distance being traversed. Refer to G.805 for complete definition of sub-network.
Sub-network Connection	A transport entity that transfers information across a sub-network. It is formed by the flexible association of ports on the boundary of the sub-network. Refer to G.805 for complete definition of sub-network connection.
Sub-network Points	A pair of addressable connection points in the client(s) network between which a link connection is established using UNI signaling.
Trail	An end-to-end connection across a particular network layer, and the entity required to provide an automatic means to check the quality of transport Refer to G.805 for complete definition of trail.
Trail Termination Point	End points of a network connection (which coincides with the ends of a trail in the information model). Refer to M.3100 for complete definition of trail termination point.
UNI	The user-network interface refers to the signaling and transport interfaces between a user and a service provider's domain.
UNI Signaling Agent – Client (UNI-C)	This is the logical entity on the user side that originates and terminates UNI signaling.
UNI Signaling Agent – Network (UNI-N)	This is the logical entity on the service provider side that originates and terminates UNI signaling.
UNI Signaling Channel	This is the logical communication channel over which UNI signaling messages are sent.
User group identifier	A network-wide unique identifier that refers to a collection of users. The user group identifier may be used to impose policies on connectivity, etc.

2.3 ITU Terminology

ITU recommendations make use of functional and informational models to provide an abstraction of a network and the elements within the network. Many of the terms list above are used in these models. The following clarifies these terms and their context.

2.3.1 Functional Model Terminology

The functional model, illustrated in Figure 2-1, represents all network functions by "atomic" or "elementary" functions: connection, termination and adaptation. Note that the link-connection, trail and network connection are defined above, and shown in bold in Figure 2-1.

.

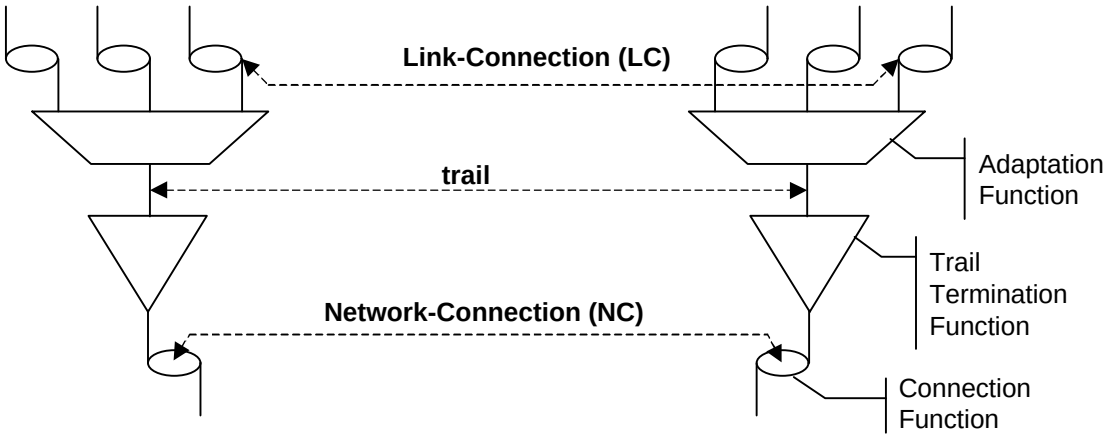


Figure 2-1 - Functional Model

The adaptation function changes the information from the client layer to a form suitable for the server layer and vice-versa. The trail termination function provides the capability to monitor the integrity of the signal, typically by adding, deleting and using overhead. The connection function provides the means to communicate the information (or items of information) between groups of atomic functions within the same layer.

2.3.2 Informational Model Terminology

The informational model, illustrated in , presents data specified in atomic functions to a management system. Note that the link-connection and network connection are common with the functional model and are defined above. In addition, the sub-network, sub-network-connection, CTP and TTP are defined above and shown in bold in .

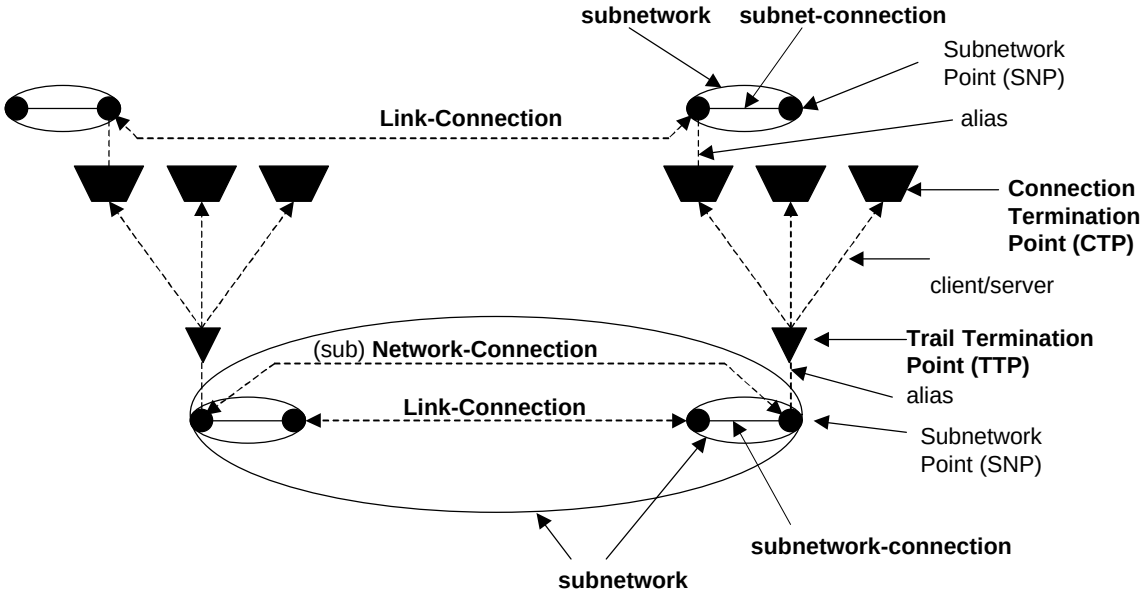


Figure 2-3 – Informational Model

The information model follows from the functional model. They are, however, not identical because they have different emphasis. The functional model is concerned with the transformation of payload signals, while the informational model is concerned with the connectivity of signals.

The mapping of the functional model to the informational model is illustrated in Figure 2-4.

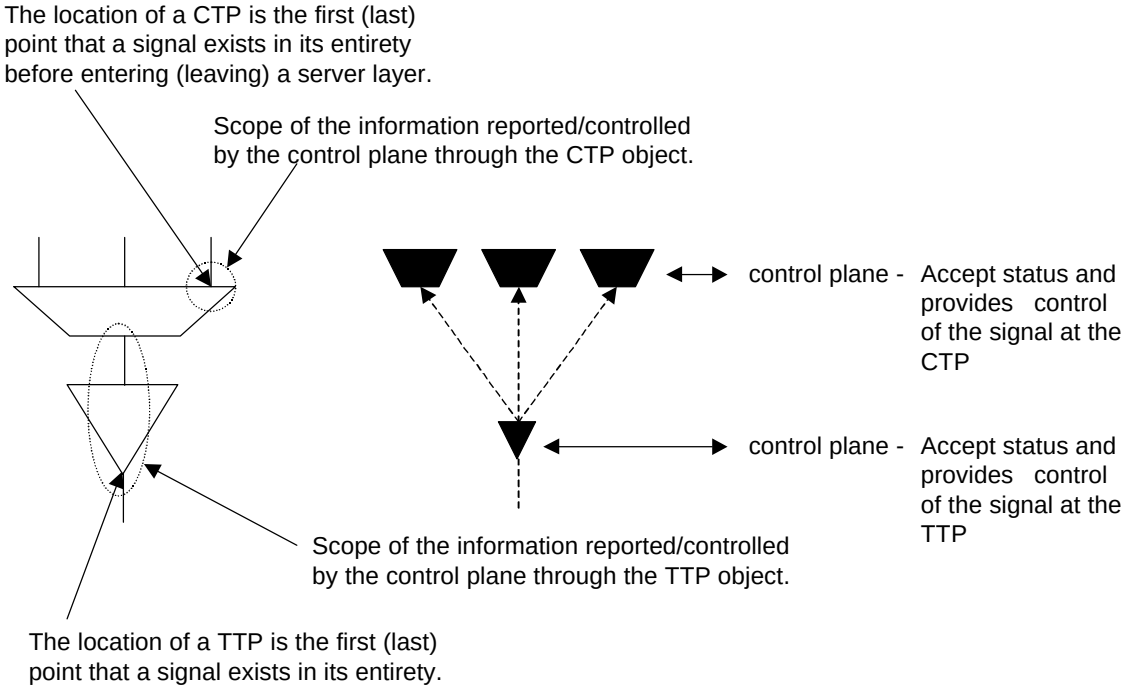


Figure 2-4 - Mapping, Functional Model-to-Informational Model

2.4 Abbreviations

CTP	Connection Termination Point
IPCC	IP Control Channel
ISI	Internal Signaling Interface
LC	Link Connection
LDP	Label Distribution Protocol
LMP	Link Management Protocol
LTE	Line Terminating Equipment
MPLS	Multi-Protocol Label Switching
GMPLS	Generalized Multi-Protocol Label Switching
NC	Network Connection
ND	Neighbor Discovery
OC-N	Optical Carrier level N
ONA	Optical-Network Assigned
ONE	Optical Network Element
PLR	Physical Layer Regenerator
RSVP	Resource reSerVation Protocol
STE	Section Terminating Equipment
STM-M	Synchronous Transport Module level M
STS-N	Synchronous Transport Signal level N
TLV	Type-Length-Value encoding
TTP	Trail Termination Point

UNI	User Network Interface
UNI- N	UNI Signaling Agent – Network
UNI-C	UNI Signaling Agent – Client
VPN	Virtual Private Network

3 Introduction

With the development of optical multiplexing using technologies such as Dense Wave Division Multiplexing (DWDM), a new layer has begun to evolve in fiber optics based telecommunications networks. This is the optical network layer that provides transport services to interconnect clients such as IP routers, ATM switches, etc. In its initial form, the optical network uses SONET/SDH as the framing structure, but the optical network may not be limited to these clients in the future. The User-Network Interface (UNI) 1.0 specified in this document is the signaling interface between the optical network and equipment in user's networks (also called *clients*). Signaling over the UNI is used to invoke services that the optical network offers to clients. This document defines the services offered over the UNI, the manner in which these services may be invoked, and the signaling mechanism used for invoking the services.

By definition, the Optical Internetworking Forum's primary focus is to develop implementation agreements that result in interoperability between manufactures. To effectively serve the telecommunications community, the OIF implementation agreements must reflect the needs of service providers who are building networks and the users that are using those networks. In addition, manufactures must be able to build cost-effective elements to support both the users' and service providers' needs. This document defines a unique set of services, physical transport technologies and signaling capabilities that may be implemented by manufacturers and deployed by service providers to support users. This document is scoped to allow an early implementation.

3.1 UNI Activities and Roles

3.1.1 Activities

Three activities are supported across the UNI:

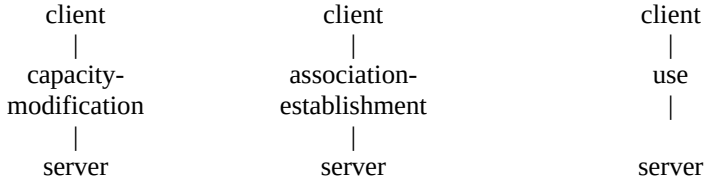
1. Association-establishment (signaling)
2. Capacity-modification (signaling)
3. Use (traffic)

The result of association-establishment is the creation of a link. The result of capacity-modification is an increase or decrease in the number of link-connections in a link, or the destruction of a link. The result of "use" is the exchange of user information over a link-connection, i.e. this is the traffic exchange.

3.1.2 Roles

For each activity there is a client and a server role. In all cases the server role is provided by (equipment that is) an agent of the server-layer-network, and the client role is provided by an agent of the client-layer-network.

Thus we have 6 roles:



The three client roles need not necessarily be performed by the same agent, i.e. they need not necessarily be co-located in the same equipment. Similarly the three server roles need not necessarily be performed by the same agent.

The three client/server interactions need not necessarily take place over the same facility (this is the in-band/out-of-band discussion).

3.2 Outline of the Specification

This specification deals with the following topics:

- Definition of UNI signaling reference configurations: Three reference configurations covering direct and third-party signaling are defined.
- Definition of services offered over the UNI.
- Definition of the neighbor discovery procedure. This allows an optical network element and a directly attached client device to discovery each other automatically and verify the consistency of configured parameters.
- Definition of different signaling channel configurations for in-band and out-of-band signaling.
- Definition of procedures for bootstrapping and maintaining the signaling control channel under various configurations.
- Definition of the service discovery procedure. This allows a client device to determine the particulars of the UNI services offered and negotiate certain parameters.
- Definition of UNI signaling messages and attributes, and
- Definition of UNI signaling protocols. UNI 1.0 signaling is based on the adaptations of two MPLS signaling protocols: LDP [1] and RSVP-TE [2].

3.3 Document Organization

This document is organized as follows:

- Section 4 defines the services offered over the UNI (Release 1.0)
- Section 5 describes the UNI reference configurations
- Section 6 describes the addressing scheme assumed in the UNI 1.0 specification
- Section 7 describes the signaling transport mechanisms
- Section 8 describes neighbor discovery procedures
- Section 9 describes the service discovery procedure
- Section 10 defines the UNI abstract messages and attributes
- Section 11 describes the LDP-based UNI signaling protocol
- Section 12 describes the RSVP-based UNI signaling protocol
- Section 13 describes policy and security capabilities supported under UNI 1.0
- Section 14 contains the references.
- Appendix A describes the relation of this specification to external standards
- Appendix B describes Multi-Layer Neighbor Discovery
- Appendix C describes COPS usage under UNI 1.0
- Appendix D is the list of companies belonging to the OIF when this document was approved.

3.4 Keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in IETF RFC-2119 [3].

4 Services Offered over the UNI (Version 1.0)

The optical network primarily offers high bandwidth connectivity in the form of optical layer *connections*, where a connection is defined to be a fixed bandwidth circuit between two user network elements (also called *clients*), established via the optical network. For UNI 1.0 this definition is restricted to consider a connection as being a SONET/SDH service of bandwidth STS-1/STM-1 or higher (with optical interfaces). The properties of the connection are defined by the attributes specified during connection establishment. UNI 1.0 signaling allows a client to establish a single connection at a time.

4.1 UNI 1.0 Signaling Actions

UNI 1.0 signaling is used to invoke the following actions:

1. *Connection creation*: This action allows a connection with the specified attributes to be created between a pair of client points of attachment to the optical network (Section 6). Connection creation may be subject to network-defined policies (e.g., user group connectivity restrictions) and security procedures.
2. *Connection deletion*: This action allows an existing connection to be deleted.
3. *Connection status enquiry*: This action allows the status of certain parameters of the connection to be queried.

Connection modification, which allows parameters of an already established connection to be modified, is not supported under UNI 1.0.

4.2 Supporting Procedures

4.2.1 UNI Neighbor Discovery

The neighbor discovery procedure aids in verifying local port connectivity between the optical and client devices. It also allows the UNI signaling control channel to be brought up. Neighbor discovery procedures for different signaling control reference configurations are described in Section 8.

4.2.2 Signaling Control Channel Maintenance

UNI signaling requires a control channel between the client-side and the network-side signaling entities. Different control channel configurations are possible, as defined in Section 7. UNI 1.0 supports procedures for maintenance of the control channel under all these configurations, as described in Section 8.

4.2.3 Address Resolution

This service allows a client to obtain the optical network points of attachment of other clients, subject to user group and policy restrictions. Addressing is described in Section 6.

5 UNI Service Invocation Reference Configurations

The reference configurations described in this section indicate the different ways in which optical network services may be invoked. Three reference configurations are supported under UNI 1.0. In each of these configurations, the client-side and network-side UNI signaling agents are referred to as UNI-C and UNI-N, respectively. UNI signaling messages between the UNI-C and the UNI-N are transported over an IP control channel, as described in Sections 7 and 8. UNI signaling protocols are based on adapting RSVP-TE and LDP, as defined in Sections 11 and 12.

5.1 Reference Configuration 1: Direct Service Invocation (Client to ONE)

Figure 5-1 illustrates the first reference configuration. Here, the clients are assumed to invoke optical network services directly over the UNI. The UNI-C functionality is thus present in the client and the UNI-N functionality is implemented in the Optical Network Element (ONE). The UNI neighbor discovery function may be available over each interface between a client and an ONE. The IP control channel between the UNI-C and the UNI-N can be in-band or out-of-band, as described in Section 7.

The trigger for the client to invoke optical network services may come from a management system in the client network or via traffic engineering decisions made by the client itself. Within the optical network, the services requested over the UNI may be provided through a management system or by the use of distributed protocols. These aspects are not considered in the definition of the UNI.

5.2 Reference Configuration 2: Direct Service Invocation (Client to Network Agent)

Figure 5-2 illustrates the second reference configuration. As in the first reference configuration, the clients directly invoke optical network services, but through an agent in the optical network external to the ONEs. Thus, the UNI-C functionality is present in the client but the UNI-N functionality is external to the ONE. The IP control channel between UNI-C and UNI-N is out-of-fiber/out-of-band, as described in Section 7. As before, the UNI neighbor discovery function may be available over each interface between a client and an ONE. An internal signaling interface within the optical network is used to carry out connection provisioning, as shown. The details of this internal interface is not relevant to the specification of the UNI.

5.3 Reference Config. 3: Third-Party Service Invocation (Client Agent to Network Agent)

Figure 5-3 illustrates the reference configuration for third-party service invocation using proxy signaling. In this figure, an entity called the Proxy UNI-C performs UNI signaling functions on behalf of one or more clients. The clients are not required to be co-located with the Proxy UNI-C as shown. Under this configuration,

1. The client does not implement the automatic neighbor discovery procedure (Section 8). The UNI-N and the proxy UNI-C must be manually configured with the neighbor information.
2. The Proxy UNI-C performs service discovery, address resolution and signaling on behalf of the clients it represents.
3. The Proxy UNI-C and a client it represents may communicate using a vendor specific or proprietary interface. Such interfaces are beyond the scope of this specification.

The Proxy UNI-C and UNI-N run the RSVP and LDP UNI signaling protocols defined in this specification (Sections 11 and 12). The IP control channel between the Proxy UNI-C and the UNI-N is out-of-fiber/out-of-band, as described in Section 7.

The parameters that must be configured in the Proxy UNI-C for each client it represents include:

- The client-layer address, which can be of any type described in Section 6.
- The node ID and port ID of the client endpoints connected to the optical network.
- The service capabilities of the client (e.g. SONET/SDH capabilities).

A Proxy UNI-C must be able to support multiple clients. Furthermore, the two endpoints of an optical connection may be handled by any of the following combinations:

- Proxy UNI-C and UNI-C
- Two separate Proxy UNI-Cs, one for each endpoint
- The same Proxy UNI-C for both endpoints

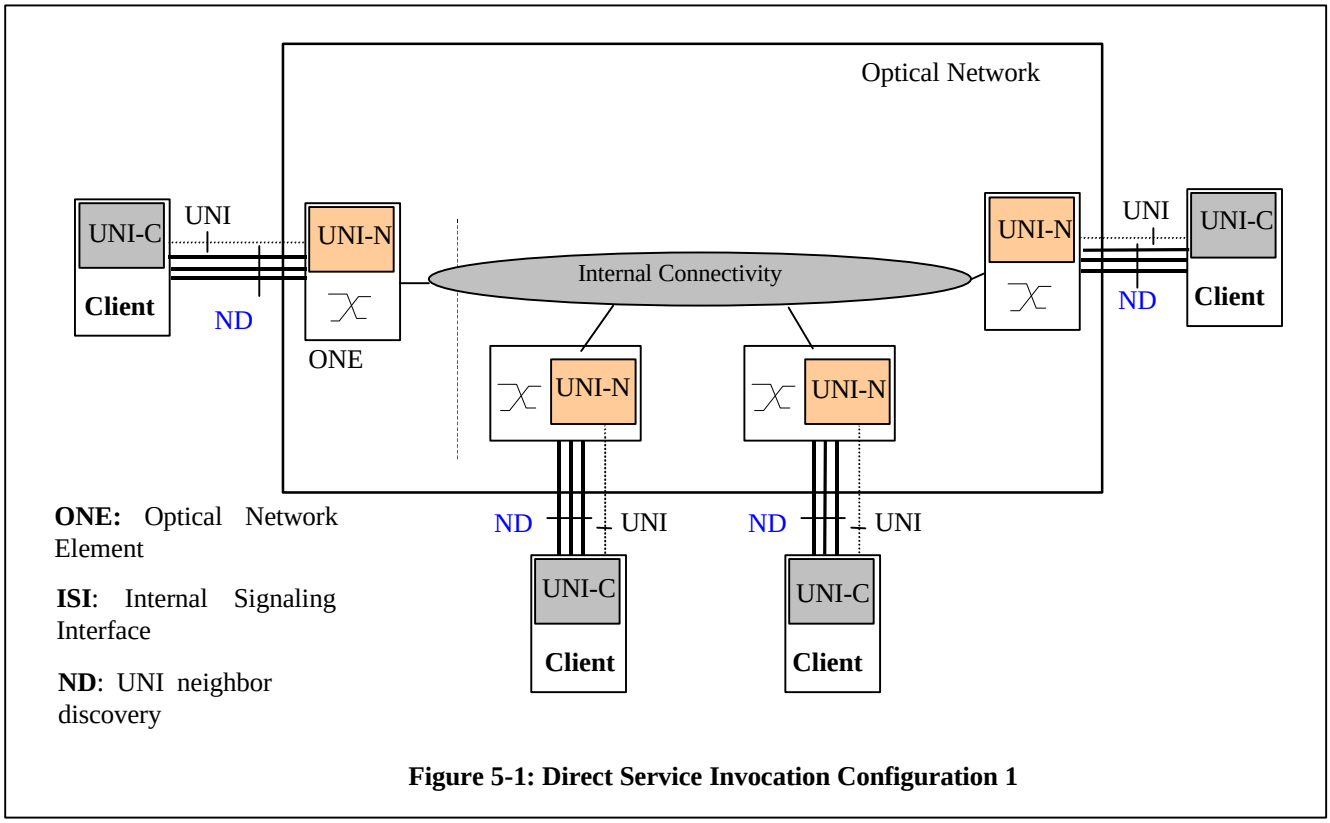


Figure 5-1: Direct Service Invocation Configuration 1

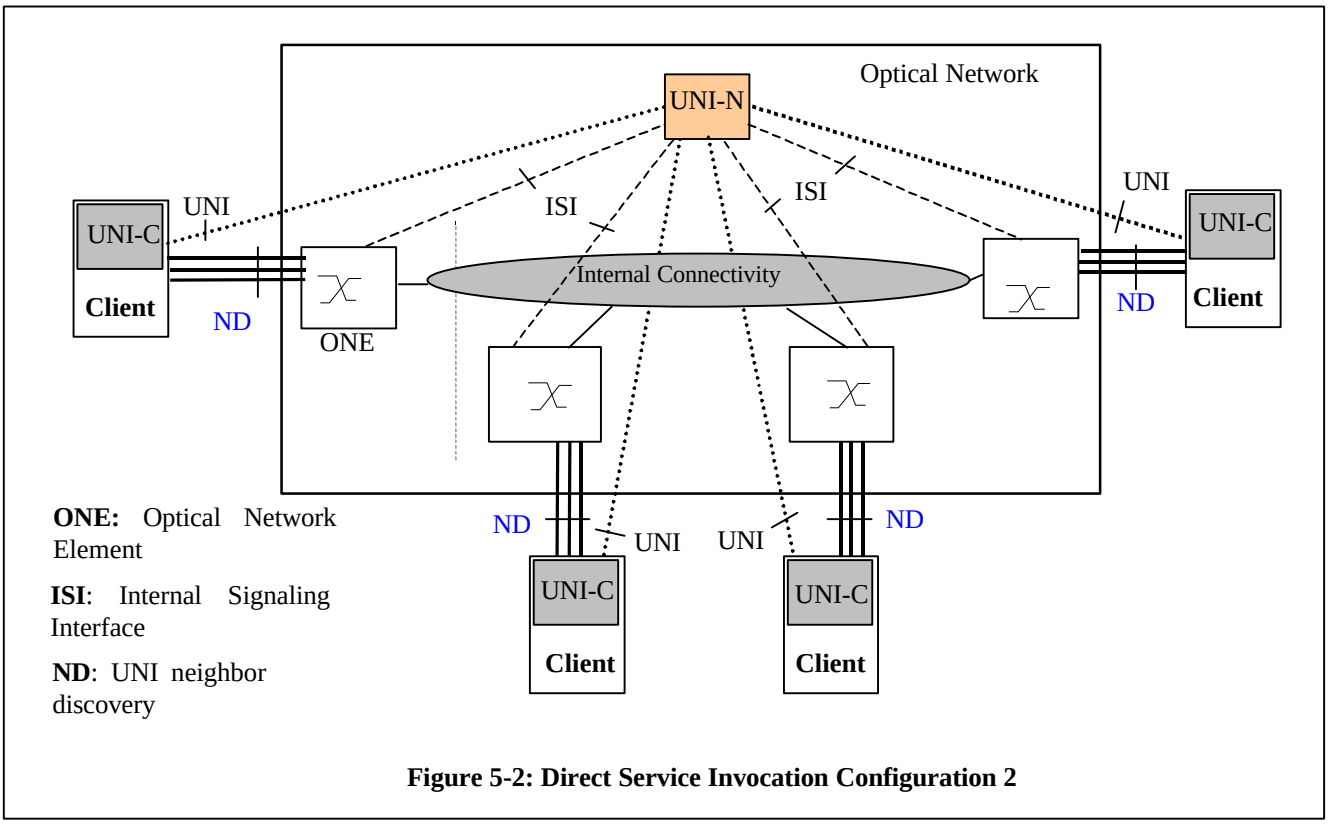


Figure 5-2: Direct Service Invocation Configuration 2

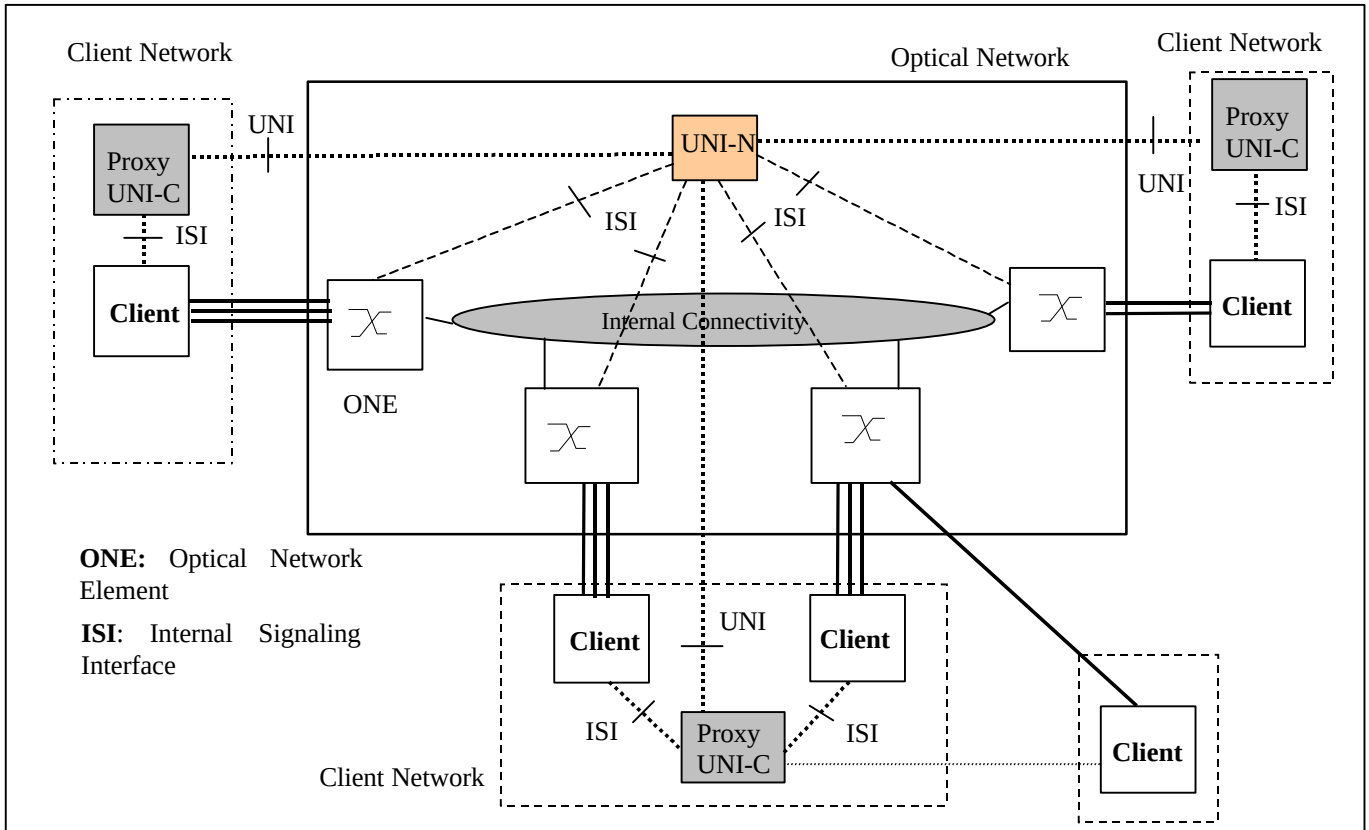


Figure 5-3: Third-Party Signaling Configuration

6 Addressing

6.1 Client vs Optical Network Address Spaces

The UNI 1.0 specification realizes a separation between client and optical network address spaces. The architectural model followed in this specification is the "overlay" model whereby the optical network provides a common set of services to different client layers, e.g., IP, ATM, etc. Under this model, the optical network provides dynamic physical connectivity between client devices that are part of a client network. The addressing and control procedures in client networks are therefore independent of the procedures in the optical network. Indeed, once an optical layer connection is established between a pair of client devices, they may be considered topologically adjacent from the perspective of client network control procedures.

6.2 Optical Network Points of Attachment

Figure 6-1 illustrates the case where an optical network services IP and ATM clients. The points of attachment of client devices to the optical network have been marked A - E in this figure. These points correspond to physical interfaces in ONEs. Under this specification, the client points of attachment to the optical network are identified by either

1. an IPv4 address administered internally in the optical network, called the *optical-network-administered* (*ONA*) *IP address*, or
2. an ONA IP address along with an index that indicates a port in the ONE that supports the point of attachment.

Thus, every client point of attachment to the optical network has an associated ONA IP address. The second choice above permits the assignment of the same ONA IP address to multiple client points of attachment, with the port index as the discriminator. This allows the use of a small number of optical network IP addresses to identify a large number of client points of attachment. Because the ONA IP

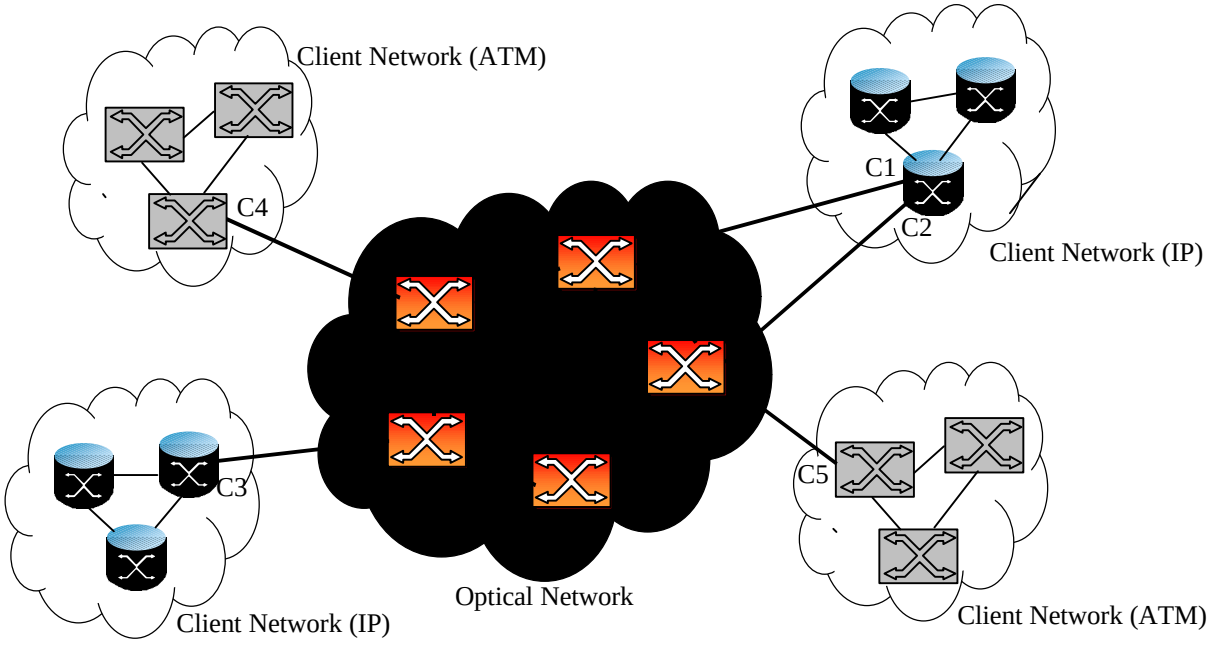


Figure 6-1: Optical Network Servicing Multiple Client Network Types

addresses, with or without port indices, are used to identify specific client points of attachment, they may also be considered as *optical network assigned client addresses*.

ONA IP addresses must be unique within the optical network. The port index must be unique with reference to the associated ONA IP address. It is expected that the path computation procedure used within the optical network will be able to use the ONA IP address directly to determine the ingress or egress point of an optical layer connection. Thus, these addresses may be used in UNI signaling messages to identify connection termination points (Section 6.3).

An alternative way to identify connection termination points is to utilize client-layer addresses directly. These addresses could then be used in UNI signaling messages. But this scheme has the following requirements:

1. UNI connection establishment messages must carry multiple address types. This requires major changes to RSVP/LDP specifications used for UNI signaling.
2. Client-layer addresses must be translated to routable optical network layer addresses for path computation. This requires an address translation mechanism built into the connection establishment process. Such a procedure disrupts the normal flow of control in existing implementations of RSVP/LDP signaling thereby imposing a major change.

Thus, the approach followed in this specification is to utilize ONA IP addresses to identify client points of attachment. An *address resolution* service, however, is specified whereby client-layer addresses may be dynamically associated with optical network points of attachment (Section 6.4). This service is specified to function independently of connection establishment signaling.

6.3 Connection Termination Points

UNI signaling for connection establishment requires the identification of the termination points of the connection. The termination points are *client-side* logical interfaces at which a connection terminates. As shown in Figure 6-2, a termination point may be identified

1. By identifying an optical network point of attachment;
2. By identifying an optical network point of attachment along with a *channel*;
3. By identifying an optical network point of attachment identifier along with a channel and a *sub-channel*.

The detail required depends on the granularity of the service. For instance, if the connection bandwidth is

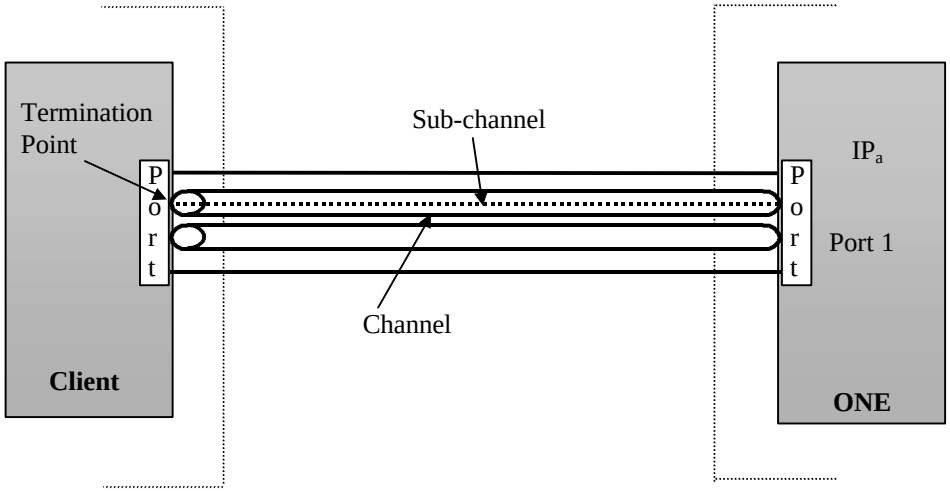


Figure 6-2: Connection Termination Points

same as the port speed (i.e., there is no multiplexing), the termination point can be identified by the point of attachment (in the example shown in Figure 6-2, the point of attachment identifier would be (IP_a, Port 1)). On the other hand, a connection that is derived from a multiplexed stream requires another level of identification. The channel and sub-channel indices provide two levels of de-multiplexing below the port level. The channel index must be unique with reference to the chosen point of attachment. The sub-channel index must be unique with reference to the chosen channel.

Under UNI signaling, the selection of a channel and a sub-channel for a connection (when required) is a local matter between a ONE and a client. That is, the channel and sub-channel are selected on the source-side by the source UNI-C or UNI-N. Similarly, the channel and the sub-channel are selected on the destination-side by the destination UNI-C or UNI-N. There is no need for the source UNI-C to specify the destination-side channel or sub-channel. The selection of the channel and sub-channel are in essence incorporated in GMPLS "label" selection during signaling, which is a local matter as defined in Sections 11 and 12. Thus, a source UNI-C need specify only the client point of attachment for the destination, which is either an ONA IP address alone, or an ONA IP address along with a port index.

In summary, the connection termination point is identified in UNI 1.0 signaling messages by:

- an optical-network-administered IP address identifying the client point of attachment (required), and
- a port index (if necessary)

UNI signaling messages, as defined in Section 10, do provide a place for *carrying* channel and sub-channel identification from the UNI-C to the UNI-N, and from the UNI-N to the UNI-C. This allows either the UNI-C or the UNI-N to select these values and inform the other.

Suppose a connection establishment request has been sent by a UNI-C with a certain destination termination point identifier. At the destination UNI-N, this is mapped to a specific port (and perhaps a channel and a sub-channel) through which the connection must be established to the client. When the destination UNI-N informs the UNI-C of the incoming connection, it must identify the termination point to the UNI-C. The issue that arises here whether the UNI-C has knowledge of the point of attachment identifier that the optical network has assigned for its interface. It is possible for the UNI-C to be informed of this point of attachment identifier during neighbor discovery. On the other hand, if the UNI-C does not have this knowledge, the UNI-N can substitute the corresponding client node and port identification learnt using neighbor discovery in UNI signaling messages. This specification requires that UNI signaling implementations follow the latter approach.

6.4 Address Resolution

In addition to client point of attachments in the optical network, A-E, Figure 6-1 also illustrates the corresponding interfaces in the client networks, C1-C5. These interfaces may be identified in protocols that run in the client layer using client-network-administered addresses. For instance, C1-C3 may be referred to by IP addresses administered in the client IP networks, and C4 and C5 by ATM NSAPs in the client ATM networks in the respective network protocols.

Under UNI 1.0, an address resolution service is specified whereby

- a client device can request one or more client-layer address(es) to be associated with an optical network point of attachment (this procedure is called *registration*)
- a client can supply a (remote) client-layer address and obtain the associated optical network point of attachment identifier(s) (this procedure is called *query*)

The following client layer addresses are permissible under UNI 1.0. During the registration and query procedures, the type of address being registered or queried must be indicated:

- IPv4 address (32 bits)
- IPv6 address (128 bits)

- ITU-T E.164 ATM End System Address (AESA) (160 bits)
- British Standards Institute ICD AESA (160 bits)
- ANSI DCC AESA (160 bits)
- NSAP address (160-bits)

The definition of the registration service allows a client to register multiple client-layer addresses through the same optical network point of attachment. Also, as per this definition, the same client-layer address can be registered with multiple optical network points of attachment. It is, however, expected that the client layer addresses associated with a given optical network point of attachment will be of the same type (e.g., IP, ATM, etc.). The registration mechanism is implemented using the same protocol used for service discovery/negotiation, as described in Section 9. The query procedure is implemented using a UNI signaling message as described in Sections 11 and 12.

6.5 User Group Identification

Client devices may provide user group identification when registering client layer addresses with the optical network. This would allow client devices in different client networks to register the same client-layer addresses. Furthermore, policies based on user group membership may be used to restrict clients from querying information about other clients. User group identifiers under UNI 1.0 are same as IP Virtual Private Network (VPN) identifiers [4].

7 Signaling Transport Configurations

As per the UNI service invocation reference configurations described in Section 5, a signaling control channel is required between the UNI-C and the UNI-N to transport signaling messages. Under UNI 1.0, the following types of signaling transport configurations are allowed:

In-fiber, In-band: The signaling messages are carried over a communication channel embedded in the data-carrying optical link between the client and the ONE. Under UNI 1.0, in-fiber, in-band signaling is specified only for SONET/SDH links. This type of signaling applies only to the direct service invocation model depicted in Figure 5-1.

In-fiber, Out-of-band: The signaling messages are carried over a separate communication channel that shares the physical link with the data channels. Under UNI 1.0, in-fiber, out-of-band signaling is specified for both SONET/SDH links and transparent links with SONET/SDH framing. This type of signaling also applies only to the direct service invocation model shown in Figure 5-1.

Out-of-fiber, Out-of-band: The signaling messages are carried over a dedicated communication link between the client and the ONE, separate from the data-bearing optical links. Under UNI 1.0, out-of-fiber, out-of-band signaling is allowed regardless of the type of the transport links (SONET/SDH or transparent). This type of signaling applies to the direct service invocation model shown in Figure 5-2, as well as the third party service invocation model shown in Figure 5-3.

In all these cases, the signaling channel must support the transport of IP packets. UNI signaling messages themselves are carried over IP. For example, the UNI signaling protocol could be based on RSVP, in which case IP packets carrying RSVP messages will flow over the signaling channel.

The details of the signaling transport configurations recognized under UNI 1.0 are as follows.

7.1 In-fiber, In-band Signaling over SONET/SDH Line or Section DCC Bytes

This configuration is shown in Figure 7-1. Here, the client and the ONE are connected by one or more pairs of unidirectional SONET/SDH links. Each pair of SONET/SDH links consist of a (unidirectional) link from the client to the ONE, and a (unidirectional) link from the ONE to the client. Together, a link pair may be treated as a bi-directonal data link. Within each such link, the overhead bytes may be used to define a (unidirectional) transport link for IP packets. The UNI signaling messages are then sent over this channel,

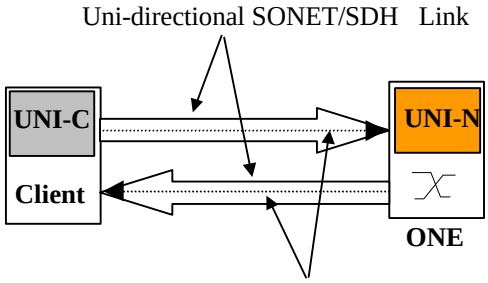


Figure 7-1: In-Fiber, In-Band Signaling Channels

called the IP Control Channel (IPCC).

The functional components of the control path between a client and the ONE are shown in Figure 7-2. Here, an IP packet generated by a client or ONE is first passed to a *Channel Manager*. This entity is responsible for selecting one of the possibly many physical control channels available to its peer. A driver prepares the packet for transmission, and the transmission hardware is responsible for framing the packet, serializing it and physically transmitting it over the selected overhead bytes to the peer. At the destination, the transmitted frames are extracted from the overhead bytes, the IP packet recovered and delivered to the IP destination. Thus, an in-band control channel is logically equivalent to a unidirectional point-to-point link. The channel manager essentially maintains a collection of such point-to-point links, monitors their status and determines which link to use when sending a packet to a neighbor. An essential feature of the above organization is that an IP packet transmitted over any of the available physical channels is received by the destination. This means that there is no coordination required apriori between the transmitter and receiver to determine which of the many physical channels should be used for sending a given packet.

Under this specification, the IPCC must be realized using SONET/SDH line or section DCC (Data Communication Channel) bytes:

- **Section DCC:** This consists of D1, D2, and D3 section overhead bytes. Used as a transparent sequence of bytes, these three bytes provide a 192Kbps message channel.
- **Line DCC:** This consists of D4–D12 line overhead bytes. Overall available bandwidth is 576Kbps.

Furthermore, IP packets sent over these channels are required to be encapsulated using PPP in HDLC like framing as per IETF RFC1662 [5], with bit stuffed framing. The automatic discovery by the client and ONE of the other's IP address (to send UNI signaling packets), the selection and maintenance of the IPCC, etc., are described in Section 8.

7.2 In-Fiber, Out-of-Band Signaling over a Separate Circuit

This case corresponds to a WDM link with multiple wavelengths, each of which may contain multiple TDM sub-channels. A TDM sub-channel or an entire wavelength can be designated as the IPCC, over which signaling messages are sent. As before, a WDM link is unidirectional, and a pair of such links may be used to realize a bi-directional IPCC (and hence a bi-directional signaling channel). This is shown in Figure 7-3. Under this specification, information about which sub-channels or wavelengths are used for realizing an IPCC must be configured in the client and the ONE. The selection of a "primary" IPCC, and the switch-over to a "back-up" link may require coordination between the client and the ONE, as described in Section 8. This specification requires that IP packets sent over an IPCC realized using a dedicated

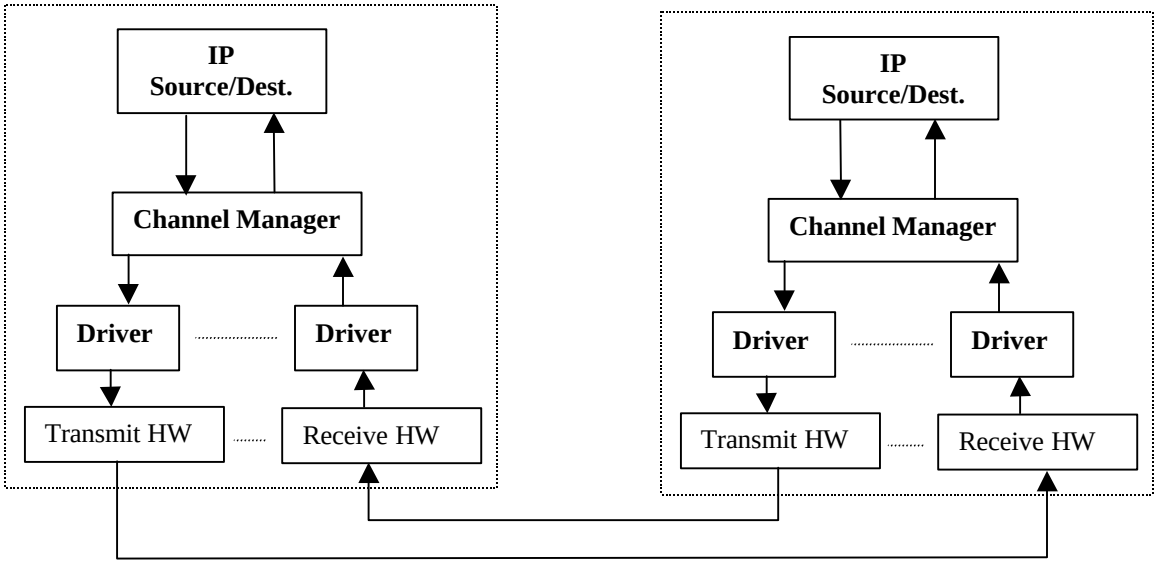


Figure 7-2: IP Transport Channel Functional Components

wavelength should use PPP in HDLC-like framing, as defined in IETF RFC 2615 [6] (Packet over SONET). When the IPCC is realized over a TDM sub-channel, this specification requires the usage of PPP in HDLC as defined in IETF RFC 1662 [5] (PPP over other circuit types). The functional components of the IPCC are as depicted in Figure 7-2. The auto discovery of IP addresses to be used in control packets is described in Section 8.

7.3 Out-of-fiber, Out-of-Band Signaling

In this case, the UNI signaling is over an out-of-band IP transport network. This is shown in Figure 7-4 for both direct and third-party service invocation (Figures 5-1 and 5-3). Here, the client (or Proxy) and the ONE (or UNI-N) have points of attachment to two distinct IP transport networks. The nature of these networks is not specified in this document. The IP interface addresses of the client (Proxy) and the ONE (UNI-N) to each of these networks is also shown. As in the co-located case, control messages may be sent over either network without coordination. But each device (client and ONE, or Proxy UNI-C and UNI-N) must determine the IP address of the other to send packets. These addresses may be configured in each device or determined using procedures defined in Section 8. Under UNI 1.0, it is assumed that the IP networks through which the IPCC is realized are secure. That is, no additional procedures are specified to ensure secure signaling message transfer over the out-of-band control networks.

7.4 Signaling Transport Realization

Although three distinct signaling transport configurations are recognized, this specification does not preclude the implementation of more than one type of transport link between a client and an ONE. For instance, a client and an ONE could have both in-fiber, in-band IPCCs, and an out-of-fiber, out-of-band IPCC. Similarly, an ONE could have different types of signaling transport configurations with different clients. Thus, configuration is required in clients and ONEs to indicate which type of IPCC is implemented for which sets of data links. The configuration of such “control interfaces” and the bootstrapping of the IPCC are described in Section 8. Also described in Section 8 are procedures to maintain the UNI signaling channel when failures affect one or more IPCCs between a client and an ONE.

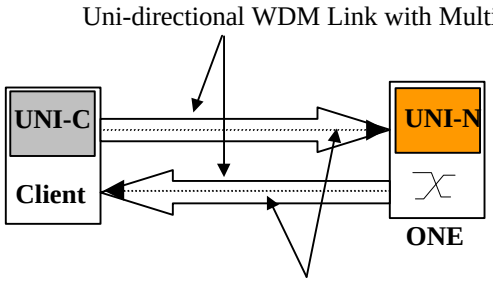


Figure 7-3: In-Fiber, Out-of-Band Signaling over a Dedicated Wavelength

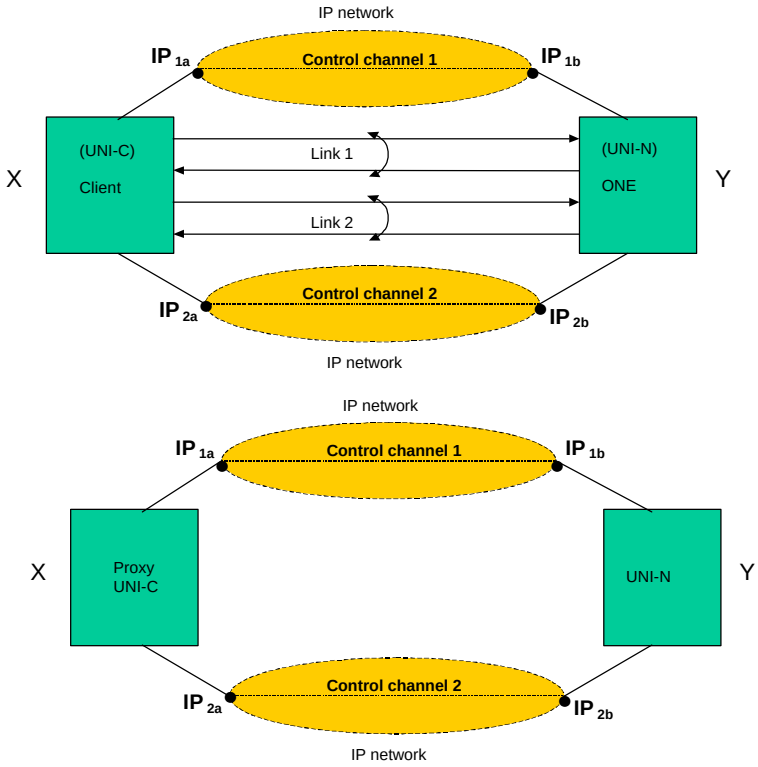


Figure 7-4: Out-of-Fiber, Out-of-Band Signaling Configuration

8 Neighbor Discovery and IP Control Channel Maintenance

8.1 Overview

UNI neighbor discovery is an *optional* but an important feature of the UNI 1.0 signaling specification. The procedures defined in this section allows ONEs and client devices to:

- *Bootstrap the IP control channel (IPCC)*: ONEs and clients can automatically obtain information from their peer to bring up the IPCC.
- *Establish basic configuration*: ONEs and clients can determine their neighbor identities and establish the basic configuration parameters for the control channel.
- *Discover port connectivity and verify configured parameters*: ONEs and clients can determine the identities of remote ports to which their local ports are connected. During this process, nodes can discover "mis-wiring", i.e., the case where the transmit and receive sides of a port are connected to two different remote ports. Furthermore, the configured information on the two sides can be verified for consistency.

It is recognized in this specification that all of the above functionality may be realized via manual configuration. For this reason, the implementation of procedures defined in this section are optional for supporting UNI 1.0 signaling. The implementation of these procedures, however, would avoid the need for manual configuration and eliminate the potential errors that may arise from relying on such configuration. Also, they provide a means to detect inconsistencies in the physical wiring, which is a useful feature.

8.2 Scope of UNI 1.0 Neighbor Discovery

Figure 8-1 illustrates peers at four SONET layers, Physical Layer Regenerators (PLR), Section Terminating Equipment (STE), Line Terminating Equipment (LTE) and Path Terminating Equipment (PTE). The neighbor discovery procedures defined in this section are targeted for LTE peers. These are also the UNI signaling peers.

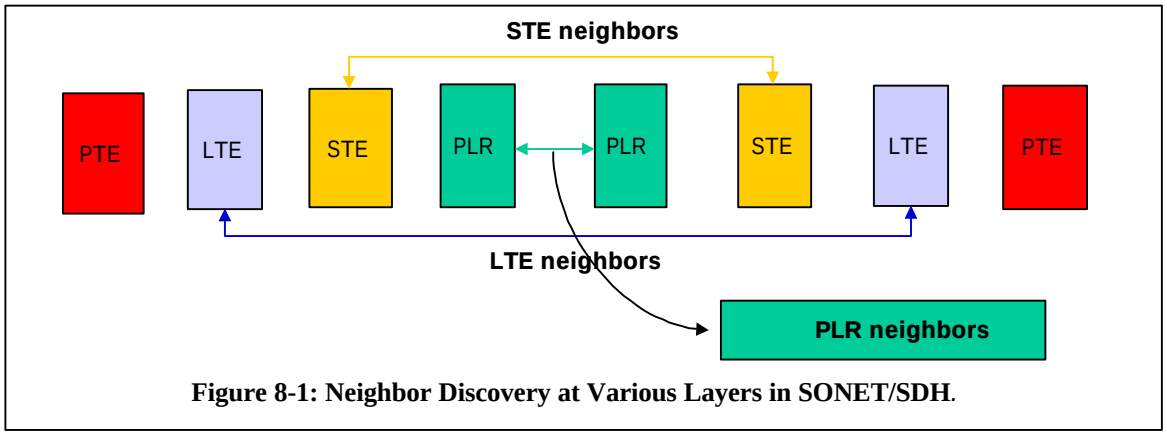


Figure 8-1: Neighbor Discovery at Various Layers in SONET/SDH.

8.3 Outline

An ONE may be connected to multiple clients and vice versa, but for the definition of neighbor discovery and IPCC maintenance procedures, it is adequate to consider a single <ONE, client> pair. Such a pair may be connected by a number of links, which may be logically grouped into *bundles*. Individual links within a bundle are referred to as *component* links. Under UNI 1.0, all the links between a given ONE and a client

are considered as part of a single bundle. These may be SONET/SDH links or transparent links with SONET/SDH framing for data. In either case, the procedures defined in this section require the ability in ONEs and clients to send and receive certain messages in-band on these links. These procedures support all of the control channel configurations defined in Section 7.

There are four basic steps defined for neighbor discovery and IPCC maintenance. These are:

1. *IPCC Bootstrap*: The client and the ONE exchange certain basic identifiers *in-fiber* to bring up one or more IPCCs between themselves. (The same information may also be manually configured, in which case this step need not be executed).
2. *Configuration*: The basic parameters of the IPCC are configured.
3. *Hello Initiation*: A Hello protocol is initiated over all the IPCCs brought up in Step 1. This protocol serves to maintain the control channel connectivity between the neighbors.
4. *Link Verification*: Following the successful initiation of the Hello protocol, a link verification procedure is initiated to determine port connectivity and to correlate configured link parameters.

The order of execution of these steps is defined slightly differently for different signaling control configurations. Specifically,

- In-fiber, In-band (IF/IB): In this case, an IPCC is realized on every component link (over SONET Line or Section DCC bytes). Control channel bootstrapping, configuration and explicit link verification are not required. Neighbor discovery for this case therefore consists of step 3 only.
- In-fiber, out-of-band (IF/OB): In this case, an IPCC is realized on each of a subset of component links (over section or line DCC bytes, or using a separate wavelength). Neighbor discovery procedure for this case consists of executing all the four steps in sequence.
- Out-of-fiber, out-of-band (OF/OB)—In this case, the IPCC is realized independent of the data links between the client and the ONE. As before, the neighbor discovery procedure consists of executing all the four steps in sequence.

In the following, the neighbor discovery and IPCC maintenance procedures for the in-fiber/in-band case is defined first, followed by the procedures for out-of-band cases.

8.4 In-Fiber/In-Band Neighbor Discovery

8.4.1 Overview

The neighbor discovery procedure builds and maintain a table that lists the set of ports that are locally available for sending IP packets along with other parameters of interest. A “port” here refers to the transmit/receive pair that constitute a bi-directional SONET link. This table also lists the corresponding information for the remote side. Figure 8-2 illustrates the table, which is referred to as the *port-state table*. The local port IDs and “other information” is configured when a port is provisioned. The local port status is determined automatically. All of the remote side information (shaded) is also determined automatically. The status of a port could be

- Up: The port is available for transmitting IP packets.
- Down: The port is not available for transmitting IP packets.

Port ID	Port Status	Other Information (e.g., port speed, SRLG, etc.)	Remote Node ID	Remote Port	Other Info.
1	Up		IP address	Port ID	
2	Up		IP address	Port ID	
3	Down		Unknown	Unknown	

Figure 8-2: Port State Table

The Neighbor Discovery Protocol (NDP) is essentially a Hello protocol. There are two types of NDP messages -- Hello and Hello-ACK. When a device is unaware of node and the port ID of the neighboring device across a link, it sends NDP Hello messages. A device sends NDP Hello-ACK messages in response to NDP Hello or Hello-ACK messages from its neighbor. The Hello-ACK messages are transmitted along the same link over which the corresponding Hello (or Hello-ACK) was received. NDP messages are carried over IP (with PPP/HDLC framing as defined in Section 7). The following sequence of actions are performed as part of NDP:

1. By default Hello messages are sent every 2 seconds. This interval is configurable by setting the NDP HelloInterval in the client and the ONE.
    - The Hello message is addressed to the All-Node-Multicast address (224.0.0.1).
    - The Hello message includes the node ID of originating device and the port ID and other port attributes of the port on which it is sent.

2. If a device has received a Hello or Hello-ACK message on a port within the last NDP DeadInterval (also configured), it sends Hello-ACK messages instead of Hello messages. By default, Hello-ACK messages are sent every two seconds. This Interval is configured by setting the NDP HelloInterval.
    - The Hello-ACK message is addressed to the unicast address of the neighboring device learnt from the Hello message or the Hello-ACK message.
    - The Hello-ACK message also echoes the port ID and port attributes of the neighbor contained in the Hello or the Hello-ACK message received from the neighbor.
    - The Hello-ACK message contains the node ID of the originating device and and port ID and port attributes of the port on the originating device.

3. Detecting mis-wiring: Because a Hello-ACK is received on the same port that a Hello is sent, the reflected information in the Hello-ACK allows a node detect mis-wiring of transmit/receive port pairs. For instance, if a node receives a Hello-ACK with reflected information that is not same as what was sent in its own Hello message, mis-wiring is detected.

4. Detecting mis-configuration: If a received Hello-ACK reflects the port id correctly, but has different information regarding other configured parameters then inconsistent configuration is detected.

5. Detection of control channel availability: A device expects to receive Hello or Hello–ACK messages from its neighbor at least once every NDP DeadInterval. If a device does not receive such messages from its neighbor within the NDP DeadInterval (default value 6 seconds) over a certain port, the neighbor at the remote end of the port is assumed to be either unavailable or incapable of recognizing IP packets over the link in question. In this case, the status of the port is set to “Down”. (It is possible that a bi-directional SONET link fails in one direction only. In this case, a device may not receive any messages over a link but may be able to successfully transmit messages in the other direction. In this case, the sender must anyway set the port status to “Down”. This simplifies the protocol operation).

8.4.2 NDP Finite State Machine (FSM)

8.4.2.1 Data Structures

The NDP FSM uses the following data structures.

- NDP HelloInterval timer: Default value 2 seconds.
- NDP DeadInterval timer: Default value 6 seconds.

8.4.2.2 Protocol States

- **State 1**: This is the initial state of the NDP state machine.
- **State 2**: In this state NDP is active and is sending Hello messages to its neighbor, but neighbor is inactive and is not sending Hello or Hell-ACKs.
- **State 3**: In this state the neighbor is active and is sending Hello messages, but not Hello-ACKs.
- **State 4**: In this state the device is receiving Hello-ACKs from the neighbor, but the node, port or other information received does not mach the expected values.
- **State 5**: In this state the device is receiving Hello-ACKs from the neighbor, and the node, port and other information received mach the expected values.

8.4.2.3 Events

- **Event 1**: Port activated. This event occurs when a functioning port hardware is detected (e.g., power up or recovery from failure).
- **Event 2**: Port deactivated: This event occurs when no functioning hardware is detected (i.e., power down, TR failure).
- **Event 3**: Hello received from the neighbor.
- **Event 4**. Hello-ACK with expected node, port, and other values received from the neighbor.
- **Event 5**: Hello-ACK with unexpected node, port, and other values received from the neighbor.
- **Event 6**: DeadInterval timer expired.
- **Event 7**: HelloInterval timer expired.

Figure 8-3 shows the NDP FSM. Figure 8-4 indicates the events, states, state transitions and the actions taken during state transitions as per the NDP FSM specification. A cell corresponding to event *i* and state *j* indicates the action taken when the FSM is in state *j* and event *i* occurs, along with the identity of the next state to which the FSM transitions.

8.4.2.4 Actions

- **Action1**: Start/reset HelloInterval timer.
- **Action 2**: Stop HelloInterval timer.
- **Action 3**: Start/reset DeadInterval timer.
- **Action 4**: Stop DeadInterval timer.
- **Action 5**: Send Hello message.
- **Action 6**: Send Hello-ACK message.
- **Action 7**: Update the remote port and node ID fields of the port in the port-state table. Set port status to "Up"
- **Action 8**: Flag mis-configuration to the management system.
- **Action 9**: Set port status to "Down" in the port-state table.
- **Error**: FSM error. Should not occur.

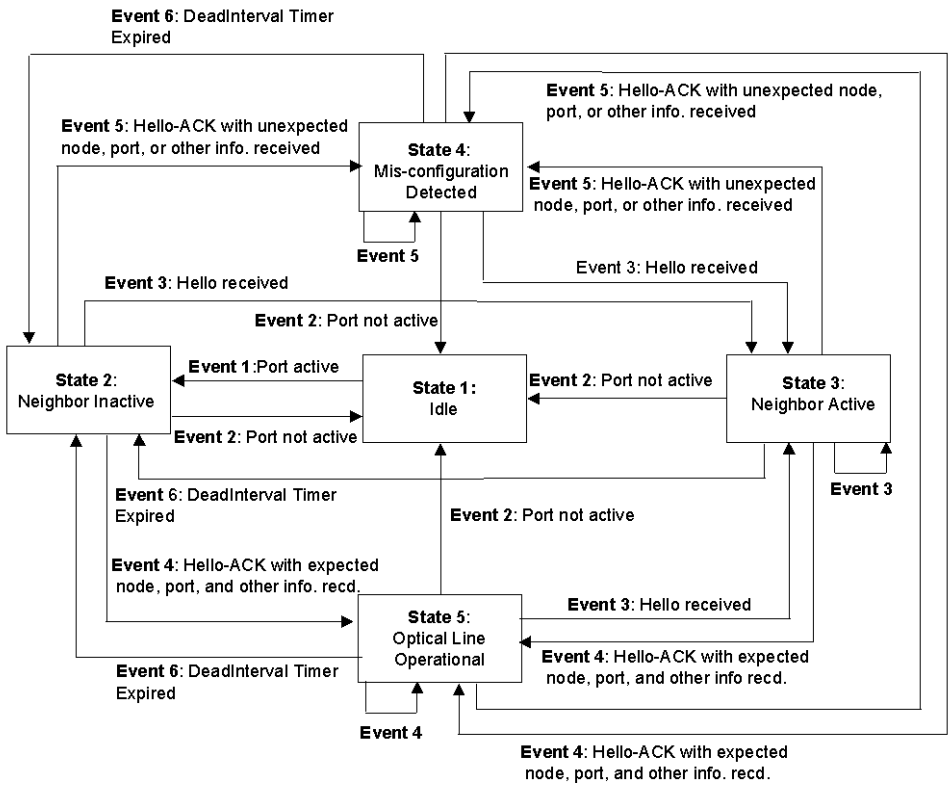


Figure 8-3: NDP Finite State Machine

	State 1	State 2	State 3	State 4	State 5
Event 1	State 2 Action 1	Error	Error	Error	Error
Event 2	Error	State 1 Action 2	State 1 Action 2,4	State 1 Action 2,4,9	State 1 Action 2,4
Event 3	Error	State 3 Action 3	State 3 Action 3	State 3 Action 3,9	State 3 Action 3
Event 4	Error	State 4 Action 3,7	State 4 Action 3,7	State 4 Action 3	State 5 Action 3,7
Event 5	Error	State 5 Action 3,8	State 5 Action 3,8	State 4 Action 3,9	State 5 Action 3
Event 6	Error	Error	State 2 Action 4	State 2 Action 4,9	State 2 Action 4
Event 7	Error	State 2 Action 5	State 3 Action 6	State 4 Action 6	State 5 State 6

Figure 8-4: NDP FSM Details

8.4.3 Message Format

NDP messages consist of a fixed header followed by an optional variable list of TLV (type-length-value) coded information elements. The header format is shown in Figure 8-5.

8.4.3.1 Header Format

NDP header has the following fields.

Version: Current version is 1.
Flags: No flags are currently assigned
Message Type: Currently only two types of messages are defined.
Hello 01
Hello-ACK 02
Total Length: Length of the NDP message including header.
Source Address: IP address of the source device
Source Port ID: Port ID at the source.
Destination Address: IP address of the destination device. If the destination device address is not known, unspecified address (0.0.0.0) is used.
Destination Port ID: Port ID at the destination. When an unspecified destination address is used, this field does not have a valid interpretation.

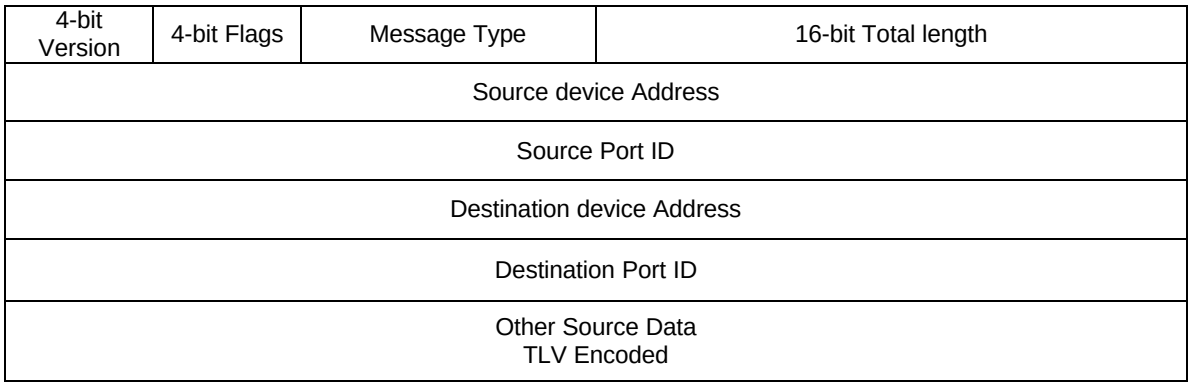


Figure 8-5: NDP Message Format

8.4.3.2 TLVs

Other data that must be exchanged for checking configuration consistency are carried in TLV encoded form. Such information is left for specification in the futures.

8.4.4 NDP and SONET Failure Detection Mechanisms

While NDP messaging helps determine the Up/Down status of ports, it is possible to update the port status using information received from SONET alarms and signal quality information. Indeed, the latter may allow failures to be detected quickly compared to NDP. NDP Hellos, however, are still essential since they determine whether an IP control channel can be realized over a link.

8.4.5 Selection of a Physical Channel for IP Control

The port-state table maintained by NDP allows the IP channel manager (Figure 7-2) to determine the set of available ports at any given instant. There may be many ports with “Up” status to a given neighbor. To send an IP packet to that neighbor, the channel manager must first match the IP destination address with a neighbor’s address, and select a port whose status is “Up” to send the packet. The algorithm by which a

port is selected is not specified in this contribution. It is required, however, that the selection of the port does not result in mis-ordering of IP packets (belonging to the same application) sent between a pair of neighbors. Because the IP control channel is unidirectional, there need not be any coordination between a pair of neighbors about which of the many available channels can be used to send an IP packet. (Note: The only restriction is on sending NDP packets. NDP Hello-ACKs must be sent over the same port over which the corresponding Hello or Hello-ACK message was received).

The channel manager at a node may choose a control channel to a neighbor and keep using it until its status changes. In such an event, the channel manager may choose any other control channel to the neighbor whose port status is "Up". If there is no such port then an IP control channel to that neighbor cannot be maintained. In this case, the channel manager begins to passively monitor the port state table for changes in status of entries pertaining to the neighbor in question. (A network management event may also be generated to note the absence of a control channel).

8.5 Neighbor Discovery and Control Channel Maintenance with Out-of-Band IPCC

In both the in-fiber/out-of-band (IF/OB) and out-of-fiber/out-of-band (OF/OB) cases, neighbor discovery and IPCC maintenance procedures are based on utilizing the IETF Link Management Protocol (LMP) for executing the four steps outlined in Section 8.3: Control channel bootstrapping, Configuration, Hello initiation and Verification. These steps are described in detail below.

8.5.1 Bootstrapping the IPCC

8.5.1.1 Overview

Bootstrapping refers to the procedure by which the client and the ONE determine which type of IP control channel should be used (IF/OB, OF/OB), and the remote (detination) IP address to which control packets must be sent.

The bootstrap procedure defined in this is optional, i.e., the bootstrap information may be manually configured in the client and the ONE. When automatic bootstrapping is implemented, certain local identifiers must be configured in the client and the ONE prior to the initiation of the bootstrap procedure. These identifiers are as follows:

- *Node Identifier* (NodeID): This is an IPv4 address that identifies the node (ONE or the client). For example, the NodeID of a router client could be its Router ID. The NodeID must be unique within the scope of the optical network.
- *Port Identifier* (PortID): This is an index that identifies the ports or links within a node. The PortID must be unique with reference to a given NodeID.
- *Bundle Identifier* (BundleID): This number identifies a bundled link supported by the node. The BundleID must be unique with reference to a given NodeID. The special value, BundleID=0, indicates that all the links between the given client, ONE pair belong to the same bundle.
- *Control Interface (CI) Address*: This is the IP address associated with the node to which control packets will be sent by its peer. The CI address must be unique within the scope of the control network over which the client and the ONE communicate. More than one CI address may be configured, and used during the bootstrap procedure.
- *Control Channel ID* (*CCID*): For every control interface, there can be multiple CCIDs configured. The CCIDs are 32-bit numbers that must be unique within the scope of a NodeID. The CCIDs provide a way to logically group control messages so that they may be processed separately.

These identifiers are exchanged during the bootstrap procedure. Additionally, the following information is exchanged:

- The type of signaling transport implementation—IF/OB, or OF/OB.
- The identification of the port that must be used for the control interface in case of IF/OB signaling transport implementation.

The in-fiber bootstrap messaging mechanism and message formats are as follows.

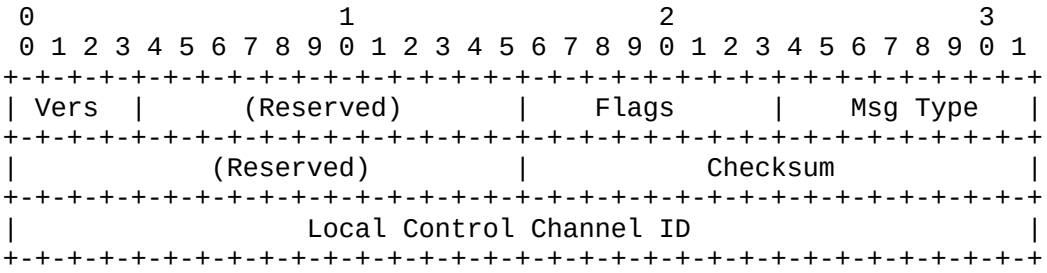
8.5.1.2 In-fiber Transport Mechanism

Certain messages must be sent in-fiber to implement the bootstrapping procedure. Under UNI 1.0, one of the following sets of SONET/SDH overhead bytes may be used for this:

- **Section DCC:** This consists of D1, D2, and D3 section overhead bytes. Used as a transparent sequence of bytes, these three bytes provide a 192Kbps message channel. It is required that PPP over HDLC-like framing be used as defined in Section 8.1. Control messages are encapsulated in IP packets and then sent over this link.
- **Line DCC:** This consists of D4–D12 line overhead bytes. Overall available bandwidth is 576Kbps. As before, PPP over HDLC-like framing be used as defined in Section 8.1. Control messages are encapsulated in IP packets and then sent over this link.
- **Section Trace Byte J0**: This consists of a 16-byte SONET section trace string. Of these 16 bytes, only 15 are available for sending neighbor discovery messages (the first byte of the string is used as the frame start marker /CRC-7 code). In addition, due to the frame start procedure, bit 0 of the usable bytess are required to be set to 0, thereby resulting in a 7-bit field in each byte.

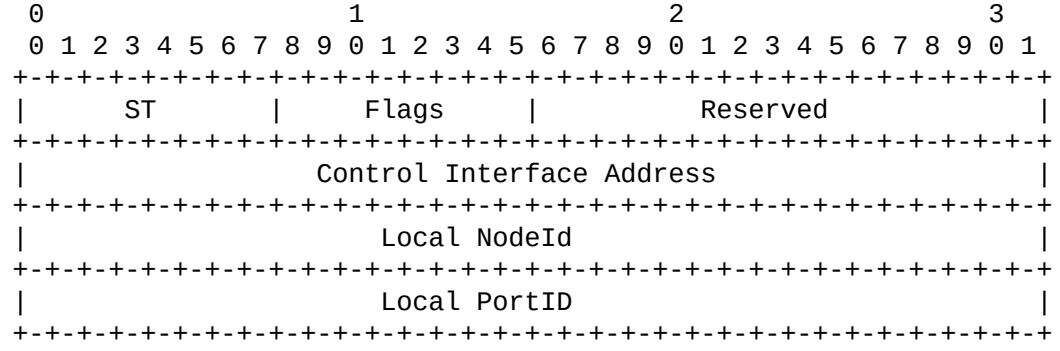
8.5.1.3 Bootstrap Message Format over Section and Line DCC Bytes

Bootstrapping messages are LMP messages, sent over IP. The LMP header preceding the bootstrap message has the following format:



The Message Type for bootstrap message is 20. The Flags and Local Control Channel ID fields are not relevant for the bootstrap message.

The format of the bootstrap message that follows the header is as shown below:



The fields in the message are:

ST (8 bits)
Signaling Transport Type. The following values are defined:
0x00—In-fiber, Out-of-band
0x01—Out-of-fiber, out-of-band
All other values are reserved.

Flags (8 bits)
Different flags are set by turning on individual bits in this field. The following values are defined:

0x01: C flag. This bit is used only in case of IF/OB signaling transport and indicates that the component link over which the message is sent should be used as the control interface.

Reserved (16 bits)
This field is unused. It should be set to 0 when the message is sent and ignored when the message is received.

Control Interface Address (32 bits)
IPv4 address for sending the control traffic to this node.

Local NodeID (32 bits)
NodeID of the sending node. This is an IPv4 address.

Local Port ID (32 bits)
ID of the port at the sending node over which the message is being sent.

8.5.1.4 Bootstrap Message Format over J0 Bytes

The J0 byte sequence allows 15 data bytes to be sent repeatedly (with 7-bits usable in each byte). The bootstrap message is not sent in IP packets, but directly over the J0 bytes. The format of the message is represented below. The entire J0 string is considered to be a 105-bit field (7 bits times 15 bytes), with the usable bits numbered 0 through 104, left to right:

<table>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>8</td><td>9</td><td>10</td><td>11</td><td>12</td><td>13</td><td>14</td><td>15</td></tr>
<tr><td>Flags</td><td colspan="5">CI Address</td><td colspan="5">NodeID</td><td colspan="5">PortID</td></tr>
</table>

The description of the fields are as follows:

Flags (10 bits, Bits 0 - 9)
Different flags are set by turning on individual bits in this field. The bits are numbered 0 to 6, from left to right. The following flags are defined:

T (Bit 0)**: T** = 0 indicates that the J0/J1 byte sequence contains a control channel bootstrapping message**.**

ST (Bit 1)**:** This bit identifies the signaling transport mechanism to be used. The following values are defined:

- **ST=0:** In-fiber, Out-of-band
- **ST=1:** Out-of-fiber, Out-of-band

C (Bit 2): This bit is used only in case of IF/OB signaling transport and indicates the component link to be used for the control channel.

Reserved (Bits 3-5): These bits are reserved for possible future use in multi-layer neighbor discovery, see Appendix X.

Reserved (Bits 6-9): Reserved for future use.

CI Address (32 bits, Bits 10-41)
IP address for sending control traffic to this node.

NodeID (32 bits, Bits 42-73)
NodeID of the sending node.

Port ID (32 bits, Bits 74-105)
ID of the port at the sending node over which the message is being sent.

8.5.1.5 Bootstrap Procedure

Consider an ONE or a client with multiple outgoing links. To run the bootstrap procedure, a node must be configured with the identifiers described in Section 8.5.1. In addition, the node must be configured with the identities of all links over which neighbor discovery must be performed. This set of links is referred to as the "neighbor discovery set". By default, bootstrap messages will be sent over all these links. The node may also be configured explicitly with the identities of one or more links in the neighbor discovery set over which the bootstrap messages should be sent. If multiple control interface addresses are available, the particular CI address to be sent over each link must be configured.

A node must send the bootstrap message periodically over each link it is configured to send the message. If sent over line or section DCC bytes as an LMP message, a parameter called "Bootstrap Send Interval" must be configured to enforce the periodic transmission. The default value for this parameter is 1 second. If sent over J0 bytes, the bootstrap message is repeated continuously. A node must stop sending bootstrap messages over a link when the identity of the neighbor across the link has been determined. Neighbor discovery is completed as per procedures defined in Section 8.5.2-8.5.4.

If the bootstrapping procedure is not run over all links in the neighbor discovery set, it is possible that neighbor discovery may not be completed for some links in the set. Thus, if bootstrapping messages are not being sent on any link while there are still some links for which the identities of the neighbors are not known, a management action must be initiated to complete neighbor discovery.

8.5.2 Configuration

8.5.2.1 Control Channel and Basic Parameter Configuration: Overview

Consider a node (a client device or an ONE) that receives an in-fiber bootstrap message from a neighbor, or has been configured to use in-fiber/in-band IPCCs on all links. The node must then establish the IP control channel(s) with the neighbor, if this has not already been done. After the control channel is established, the node can begin exchanging basic LMP configuration parameters. Port connectivity verification is defined as a separate step in Section 8.5.4 for out-of-band control. The configuration procedure and parameters are defined in this section.

Suppose a node has determined the control interface address corresponding to an IPCC. It must then attempt to establish the IP control channel with the neighbor. This is accomplished by using three LMP messages, the Config, ConfigAck and ConfigNack. In essence, the node must send a Config message to the control interface address over the corresponding configured physical IP control interface on its side. The Config message may contain a number of parameters encoded in the Type-Length-Value (TLV) format.

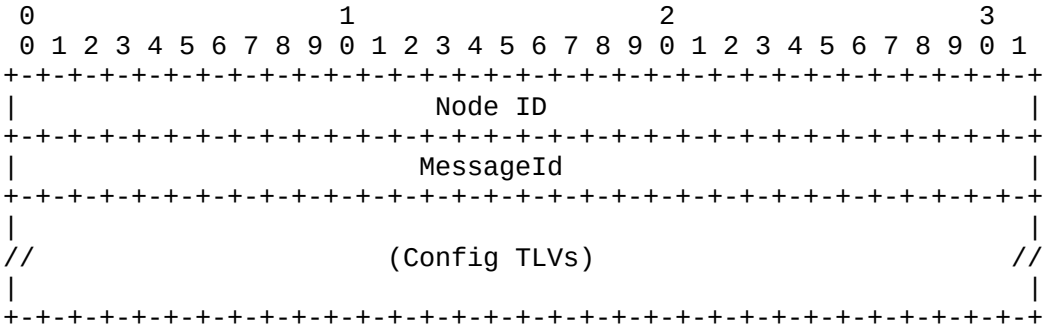
These parameters could be negotiable or non-negotiable. In response to a Config message, a node must return a ConfigAck or ConfigNack indicating acceptance or rejection of parameters in the Config message. The reception of a ConfigAck completes the configuration process. The reception of a ConfigNack may terminate the configuration process unsuccessfully, or may result in the transmission of another Config message with revised parameters.

LMP messages related to configuration are sent in IP packets over the control channel, using the CI address received in the bootstrap message as the destination address.

All LMP messages are preceded by a common header described in Section 8.5.1.3.

8.5.2.2 The Config Message

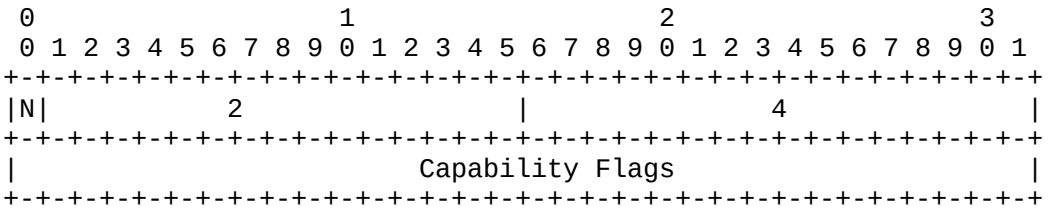
The Config message has the following format:



The NodeID indicates the ID of the node sending the Config message. The Message ID is a 32-bit number unique within the scope of the NodeID. The Config TLVs that should be included for neighbor discovery are as follows.

8.5.2.2.1 LMP Capability TLV

This indicates the capabilities supported. It has the following format:



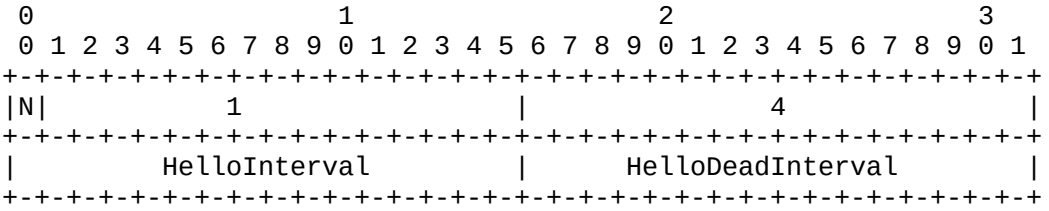
where the bit N must be set to 0, and Capabilities Flags must be set to one of the following values:

Bit 0: Only the base set of LMP procedures are supported (Link verification feature not supported)

Bit 1: Link verification feature supported.
All other bits must be set to 0 for UNI 1.0 compatibility.

8.5.2.2.2 HelloConfig TLV:

This TLV has the following format:



The N bit must be set to 0 if the HelloInterval and HelloDeadInterval are non-negotiable. Otherwise, the bit must be set to 1. The HelloInterval indicates how frequently the Hello packets will be sent and it is specified in milliseconds. If no Hello packets are received within the HelloDeadInterval, the control channel is assumed to have failed and this value is also specified in milliseconds.

8.5.2.3 ConfigAck Message

The ConfigAck message is used as defined in LMP [7].

8.5.2.4 ConfigNack Message

The ConfigNack message should be used as defined in LMP [7] with the following directives:

- The Config TLVs as defined in Section 8.5.2.2 are used.

8.5.3 The Hello Protocol and IPCC Maintenance

The LMP Hello protocol is a lightweight keep-alive mechanism that detects control channel failures rapidly. With out-of-band control, the Hello protocol runs on each IPCC designated as a "primary" control channel.

Consider an IPCC over which the Hello protocol is run. Each node (client or ONE) executing the Hello protocol must periodically send a uni-cast "Hello" message over the IPCC. This is an LMP message, sent as an IP packet. The IP destination address must be the control interface address obtained during the bootstrap procedure, or during the configuration procedure. The periodicity of the Hello message is governed by the HelloInterval established during the configuration phase (Section 8.5.2). Each LMP "Hello" message contains two 32-bit sequence numbers: a SendSeqNum, which is the sequence number of this "Hello" message and a RecSeqNum, which is the sequence number of the last "Hello" message received by the sender. The reserved sequence numbers 1 and 0 are used to indicate that a node (client or ONE) has rebooted and that a Hello message has not yet been received by the sender, respectively.

If a node is sending Hellos but does not receive any during the HelloDeadInterval period, the corresponding IPCC must be declared unusable. From a UNI signaling perspective, a new IPCC to the same neighbor must be selected (if possible). The procedure for this is based on LMP procedures for control channel switchovers, as defined in [7].

The Hello protocol details are defined as specified in LMP [7].

8.5.4 Link Verification and Link Property Correlation

8.5.4.1 Link Verification

This phase deals with determining the usability of data links between the ONE and the client, the port mappings and the consistency of configured link parameters. With out-of-band control, this has to be done explicitly. The LMP link verification procedure is used for this. The use of this procedure is negotiated as part of the configuration exchange using the Verification Procedure flag of the LMP Capability TLV (Section 8.5.2.2).

Link verification is initiated by one of the nodes (client or ONE) by sending an LMP BeginVerify message to the other node over the control channel. In response to the BeginVerify message, the remote node may send either a BeginVerifyAck or BeginVerifyNack message to accept or reject the process. If a BeginVerifyAck message is sent, the remote node provides a VerifyID to the local node in the BeginVerifyAck message. The VerifyID is then used in all corresponding Test messages to differentiate them from different LMP peers. If the testing procedure is acknowledged by the receiver, link verification (Test) messages are then sent in-fiber by the initiating node over each data link. If the transport mechanism is line or section DCC, LMP Test messages are sent over IP. If the transport is over J0 bytes, a 15-byte link verification message is sent. The format of this message is shown below.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Flags	**VerifyId**							**PortId**				**BundleId**		

This message has 105 bits (Seven usable bits in each byte), numbered 0 to 104 from left to right. The contents of this message are:

Flags (9 bits, Bits 0-8)
There are currently 3 Flags defined.

T (Bit 0)
T = 1 indicates that the J0 stream is used with link bundling and the end system discovery using LMP.

Ic (Bit 1)
$\mathbf{I_c}$ = 0 indicates that the control channel interface is unnumbered; $\mathbf{I_c}$ = 1 indicates that the control channel interface is numbered;

$\mathbf{I_b}$ (Bit 2)
$\mathbf{I_b}$ = 0 indicates that the bundle ID is unnumbered; $\mathbf{I_b}$ = 1 indicates that the bundle ID is numbered.

Reserved (Bits 3-8)

VerifyID (Bits 9-40)
The control channel ID at the sending node. The sending node is identified by the NodeID field in the LMP BeginTest message received over the IPCC. The CCID is a 32-bit number unique within the scope of the NodeID of the sending node.

PortID (Bits 41-72)
This is the 32-bit number identifying the port at the sending node over which this message was sent.

BundleID (Bits 73-104)
This identifies the associated bundle for the Port. Under UNI 1.0, the BundleId MUST be 0x00.

The verification procedure is as follows. After a client and ONE agree to begin testing the component links, the initiating node sends a "Test" message over each unidirectional fiber link. Upon receipt of the "Test" message, the receiver correlates the Test message to a particular LMP session using the VerifyId and maps the received **PortId** and **BundleId** to the corresponding local values. The receiver then sends its local **PortId** back to the end system in a TestStatusSuccess message over the control channel to the initiating node.

8.5.4.2 Link Property Correlation

As part of LMP, a link property correlation exchange is defined using the LinkSummary, LinkSummaryAck, and LinkSummaryNack messages. The LinkSummary message must be transmitted in

order to add data-bearing links to a link bundle, change Port IDs, or change a link's protection mechanism. In addition, the LinkSummary message can be exchanged at any time a link is UP and not in the Verification process.

The LinkSummary message contains the local/remote Bundle Id (set to 0 by default under UNI 1.0) and the drop-side protection for the bundle, followed by a list of primary and secondary channel subobjects. The LinkSummary message supports unprotected, dedicated (1:1,1+1), shared, and enhanced protection. The Number of Secondary Entities MUST be zero when summarizing an unprotected link bundle. The Number of Primary and Secondary Entities MUST be equal when summarizing a dedicated (1:1 or 1+1) link bundle. The primary/secondary channel subobjects contain the local/remote Port Ids.

If the LinkSummary message is received from a remote node and the Port ID mappings match those that are stored locally, then the two nodes have agreement on the Verify process. If the verification procedure is not used, the LinkSummary message can be used to verify manual configuration. Furthermore, any protection definitions that are included in the LinkSummary message must be accepted or rejected by the local node. To signal agreement on the Port Interface Id mappings and protection definitions, a LinkSummaryAck message is transmitted. Otherwise, a LinkSummaryNack message will be transmitted, indicating which channels are not correct and/or which protection definitions are not accepted. If a LinkSummaryNack message indicates that the Port Id mappings are not correct and the link verification procedure is enabled, the link verification process should be repeated for all mismatched free data-bearing links; if an allocated data-bearing link has a mapping mismatch, it should be flagged and verified when it becomes free.

8.5.5 Example of Neighbor Discovery with Out-of-Band IPCC

8.5.5.1 Neighbor Discovery

Consider a client and an ONE connected by multiple component links, with one or more out-of-band IPCCs between them. As per the procedures defined in this section,

1. The client and the ONE start sending the control channel bootstrapping messages over the component links. This information is used to start the control channel.
2. As a part of the control channel connection establishment, the devices exchange their NodeIDs, CCIDs and IPCC configuration parameters. Then the link verification and bundling part may be initiated.
3. The client sends a “BeginVerify” message (over the control channel) to indicate that the client will begin sending “Test” messages using an in-band transport over each of the component links between the client and the network equipment. The “BeginVerify” message contains the number of component links (if more than one) that are to be verified;
4. When the network equipment receives a “BeginVerify” message and it is ready to receive “Test” messages, it sends a “BeginVerifyAck” message (over the control channel) back to the client;
5. When the client receives a “BeginVerifyAck” message from the network equipment, it will begin transmitting periodic “Test” messages toward the network equipment direction, including the **(PortId,BundleId)**, over the in-band transport of the first data-bearing link;
6. Upon receiving the “Test” message, the network equipment will record the received **(PortId,BundleId)**, and then return a “TestStatusSuccess” message (over the control channel), including the client **(PortId,BundleId)** and network equipment **PortId** over the link, in response for each link to indicate that the physical connectivity of the link has been verified and the client has been registered for the link; if, however, the “Test” message is not detected at the network equipment within a period (specified by a timeout value), the network equipment will send a “TestStatusFailure” message (over the control channel) to indicate that the client discovery process of the link has failed;
7. Upon receiving the “TestStatusSuccess” message over the control channel, the client will be able to detect misconfiguration. If no misconfiguration is identified, the client will mark the component link as “Discovered” and records the network equipment **(PortId,BundleId**);

8. When all the links on the list have been tested, the client will send an "EndVerify" message (over the control channel) to indicate that the testing has been completed. Subsequently, an "EndVerifyAck" is sent (over the control channel) as a response from the network equipment to indicate that the client discovery is done.

After the end system discovery process is finished, the end system and the network equipment should be notified and OH bytes used for in-band transport (in case of SONET/SDH) should go back to the normal operation. (The scheme is essentially non-intrusive, and allows for bit transparent operation with SONET/SDN signals).

With the above process, both end system and the network equipment maintain the complete list of link identifier mappings, received **PortID** and Local **PortID**. As part of LMP, there is also a "LinkSummary" message that can be exchanged at any time the nodes are not in the Test procedure.

1. The end system sends a "LinkSummary" message over the control channel, including all the ports for the links over the UNI;
2. Upon receiving the "LinkSummary" message, the network equipment checks if the receiving **(PortId,BundleId)** match the local **(PortId,BundleId)** associations; if yes, it sends back a "LinkSummaryAck" message over the control channel; otherwise, it sends back a "LinkSummaryNack" message, indicating which ID associations are not correct;
3. Upon receiving the "LinkSummaryAck" message, the link verification and end system discovery procedure is completed; if a "LinkSummaryNack" message is received, the process should be repeated for all mismatched links.

8.5.5.2 Fiber Link Miss-wire Diagnosis

The scheme presented above for the end system discovery can also be used to detect and correct the fiber miss-wiring. Figure 8-6 illustrates a situation where a fiber link is mis-wired. Using the neighbor discovery procedure defined in this section:

1. A Test message is transmitted from Node1 over f1 using in-band transport. The Test message includes the local **CCId**, **BundleId**, and **PortId** ("Port1");
2. Node2 receives the Test message on Port3 and records the parameters. A TestStatusSuccess message, including local **CCId** and **PortId** ("Port3"), is transmitted over the control channel;
3. Node1 sends a Test message over f2 using in-band transport. The Test message contains the local **CCId**, **BundleId**, and **PortId** ("Port2");
4. Node2 receives the Test message on Port4 and records the parameters. A TestStatusSuccess message, including local **CCId** and **PortId** ("Port4"), is transmitted over the control channel.

This finishes the ES to NE transmission. Similarly, the following procedure is initiated by the NE:

1. Node2 sends a Test message over b2 using in-band transport. The Test message contains the local **CCId**, **BundleId**, and **PortId** ("Port3");
2. Node1 receives the Test message on Port2 and records the parameters. A TestStatusSuccess message, including the local **CCId** and **PortId** ("Port2"), is transmitted over the control channel;
3. Node2 sends a Test message over b1 using in-band transport. The Test message contains the local **CCId**, **BundleId**, and **PortId** ("Port4");
4. Node1 receives the Test message on Port1 and records the parameters. A TestStatusSuccess message, including the local **CCId** and **PortId** ("Port2"), is transmitted over the control channel.

By now, fiber link miss-wire is detected since Port2 is connected to Port3 and Port4 simultaneously. An alarm may be raised and Port1, Port2, Port3, or Port4 are not announced in the LinkSummary message.

9 Service Discovery and Address Registration

9.1 Overview

Service discovery is the procedure by which the client obtains information concerning services from the optical network. Address registration allows the client to register client-layer addresses and user group identifiers with the optical network.

Service discovery and address registration are closely related to neighbor discovery. These mechanisms complement each other and they do not depend on the establishment and use of signaling channels between the two parties.

9.2 Service Attributes

The following service attributes are defined:

1. Supported signaling protocol: Whether RSVP or LDP-based UNI signaling is supported.
2. Support for address resolution: Whether address resolution is supported.
3. Port level service granularity: The STS-n and STS-m servies supported over each port in the case of SONET/SDH, respectively.
4. Transparency Support: Whether PLR (Physical Layer Regeneration), STE (Section Terminating Equipment), LTE (Line Terminating Equipment) are supported.

Additionally, it is recognized that some of the signaling attributes (e.g., diverse routing) may not be supported by all networks. The support for these, however, may be determined during connection signaling.

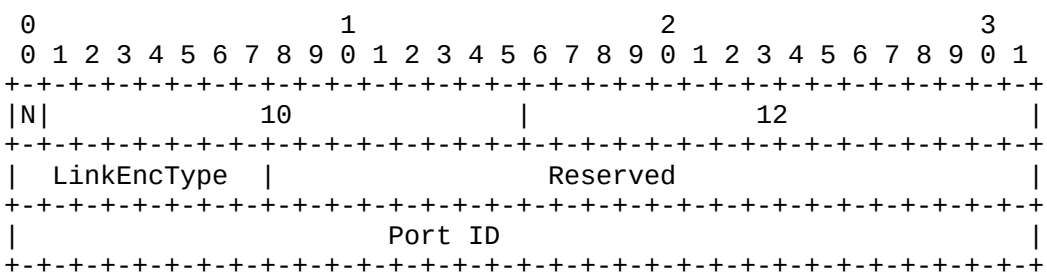
9.3 Service Discovery Procedure

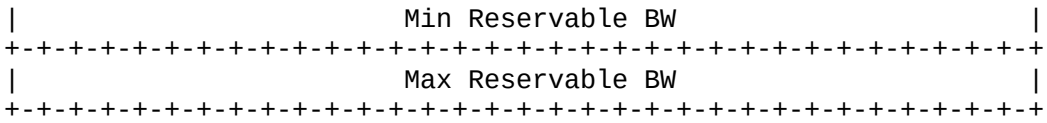
The service discovery procedure is invoked after neighbor discovery is complete, and it is run over the IPCC. This procedure consists of the client and ONE exchanging three messages: ServiceConfig, ServiceConfig Ack and ServiceConfig Nack. These messages are defined as LMP messages and their format is identical to the Config, ConfigAck and ConfigNack messages, respectively, described in Section 8.

To initiate service discovery, a client sends a ServiceConfig message to the ONE. This message is preceded by the LMP header (with the proposed MsgType = 21). Service attributes are coded as ServiceConfig TLVs and sent in the ServiceConfig message. In return, the ONE sends a ServiceConfigAck or ServiceConfigNack message (with proposed MsgTypes 22 and 23, respectively). The procedure for this exchange are as defined for LMP Config message exchange [7]. The following ServiceConfig TLVs are defined:

9.3.1.1 The Link Descriptor TLV

The TLV indicates the service granularity at each port. It is sent from the client to the ONE and its format is as follows:





The Link Descriptor TLV contains the following fields:

N Bit

This bit must be set to 0 to indicate non-negotiability of the values.

LinkEncType (8 bits)

Link Encoding type values must be one of the following:

Value	Link Encoding Type
1	Standard SONET
2	Arbitrary SONET
3	Standard SDH
4	Arbitrary SDH

PortID

The 32-bit Port ID of the ONE port whose capabilities that this TLV refers to.

Min Reservable BW

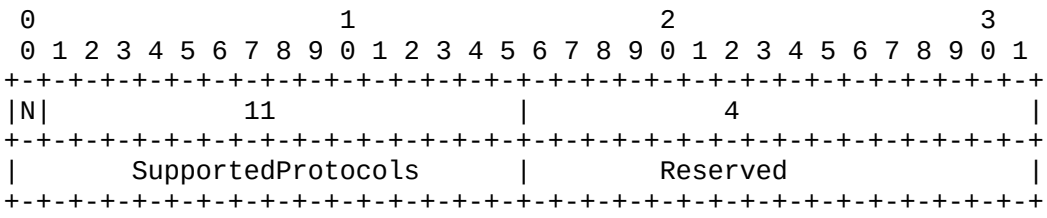
This specifies the minimum reservable bandwidth (in IEEE floating point format, the unit being bytes per second) for this link encoding type.

Max Reservable BW

This specifies the maximum reservable bandwidth (in IEEE floating point format, the unit being bytes per second) for this link encoding type.

9.3.1.2 Supported Protocols TLV

This TLV indicates the supported protocols at the UNI. This TLV is sent from the ONE to the client. Its format is as follows :



The TLV contains the following fields.

N Bit

Must be set to 0.

SupportedProtocols

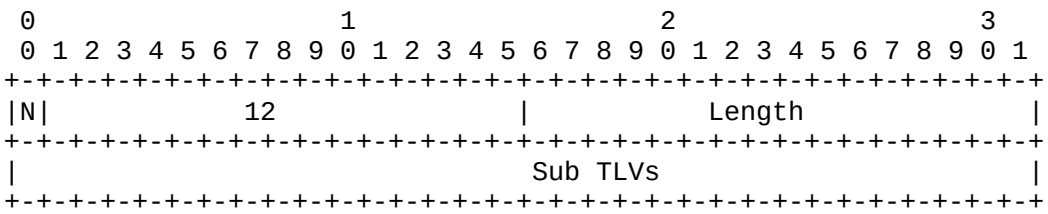
A 16-bit field that contains flags identifying the control-plane protocols supported by the device. The following values are defined.

0x01: RSVP-TE - This indicates that the device supports RSVP-TE-based UNI signaling protocol.

0x02: LDP - This indicates that the device supports LDP-based UNI signaling protocol.

9.3.1.3 Supported Services TLV

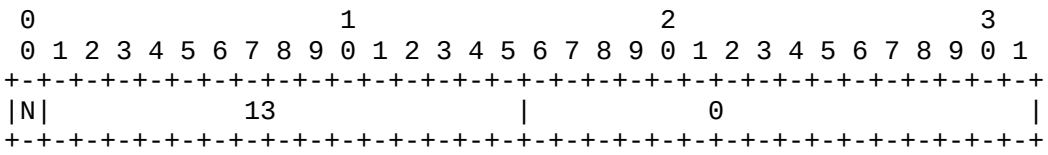
This TLV has the following format:



The following sub-TLVs are defined:

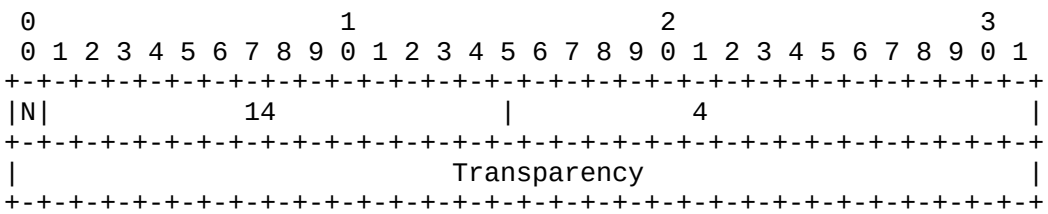
9.3.1.3.1 Address Resolution

This TLV is present if address resolution is supported by the network. The format of this TLV is:



9.3.1.3.2 Transparency

This TLV indicates the supported transparency. Its format is:



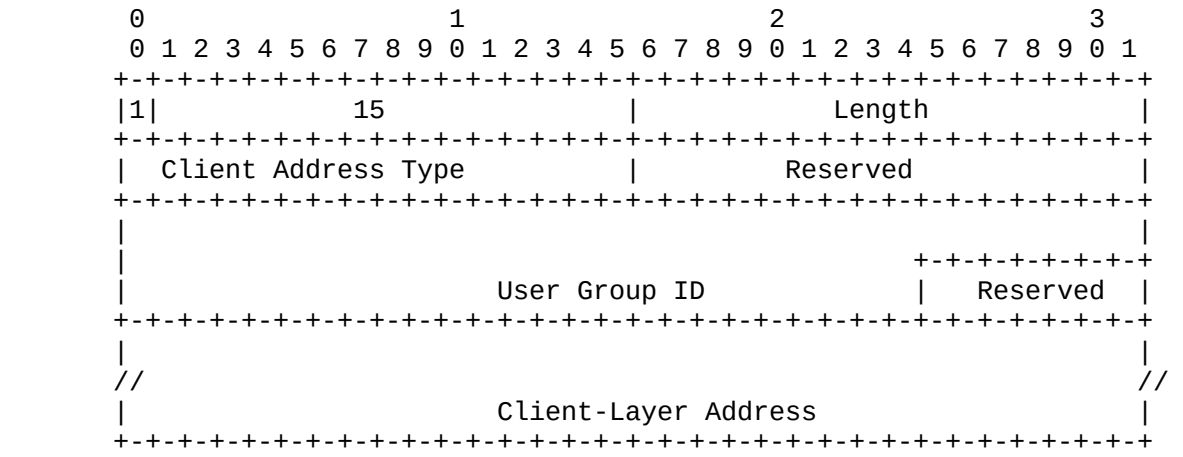
Transparency (Bits 0-2):
Bit 0 is set if PLR service is supported; Bit 1 is set if STE service is supported; Bit 2 is set if LTE service is supported.

9.4 Address Registration

Address registration is also done as part of the ServiceConfig procedure. The client sends the following TLVs to the ONE to perform address registration:

9.4.1.1 The Address Registration TLV

This TLV indicates an address registration request. This TLV can only be present in a ServiceConfig message from the client to the ONE. The format of this TLV is as follows:



Client Address Type

This is a 16-bit field carrying the type of the client-layer address. Currently defined values are:

Type	Value
0x960	IPv4
0x961	IPv6
0x962	ATM NSAP
0x963	E.164
0x964	ICD

User Group ID

This is a 7-byte VPN ID value as per RFC 2685 [4].

Client-Layer Address

This is a variable length field whose value is based on the address type (see Section 6).

If address registration is successful, a ServiceConfigAck is returned in the same manner as the Config Ack. [7]. Otherwise, a ServiceConfigNack is returned. The determination of why the address resolution failed is not part of this specification.

10 .UNI Abstract Messages

The UNI signaling messages are described in this section. These messages are denoted "abstract" since the actual realization depends on the signaling protocol used. Sections 11 and 12 describe RSVP and LDP signaling messages corresponding to the abstract messages defined here. In the following description, the terms "initiating UNI-C" and "terminating UNI-C" are used to identify the entities at two ends of a connection that initiate and terminate signaling actions. Under service invocation configurations 1 and 2 (Figures 5-1, 5-2), a UNI-C entity at either end of a connection can initiate a signaling action. The UNI-C entity at the other end then becomes the terminating client. Under configuration 3 (Figure 5-3), the initiating and terminating UNI-C could be the same entity.

The following abstract messages (Table 10-1) are currently defined:

Message No.	Abstract Message Description	Message Direction
1	Connection Create Request	UNI-C→UNI-N & UNI-N→UNI-C
2	Connection Create Response	UNI-N→UNI-C & UNI-C→UNI-N
3	Connection Delete Request	UNI-C→UNI-N & UNI-N→UNI-C
4	Connection Delete Response	UNI-N→UNI-C & UNI-C→UNI-N
5	Connection Status Enquiry	UNI-C→UNI-N & UNI-N→UNI-C
6	Connection Status Response	UNI-N→UNI-C & UNI-N→UNI-C
7	Address Resolution Query	UNI-C→UNI-N
8	Address Resolution Response	UNI-N →UNI-C
9	Notification	UNI-N →UNI-C

Table 10-1: UNI Messages

In connection creation requests, a list of attributes for the connections can be specified. Some of these attributes are required while others are optional. A network may not support all of the requested attributes. In such cases, the network shall send appropriate indications to the client in response messages. The encoding of these attributes is not defined in this section since this too depends on the particular signaling protocol used. Refer to Sections 11 and 12 for attribute encoding under RSVP and LDP signaling protocols.

The manner in which the UNI abstract messages are mapped to actions within the optical network, and the signaling protocol used within the optical network to realize the actions are outside the scope of this specification. Furthermore, the resolution of conflicts when UNI signaling is concurrently invoked on both sides of a connection to perform certain actions is outside the scope of this specification.

The UNI abstract messages are described in detail below.

10.1 Connection Create Request

This message is sent from

- the initiating UNI-C to UNI-N to request the creation of a connection;
- the UNI-N to the terminating UNI-C to indicate an incoming connection request.

The mandatory (M) and optional (O) attributes carried in this message are shown below. The section in which each attribute is described is also indicated.

Attributes	Reference
Source Optical-Network-Administered IP Address (M)	Section 10.10.1.1
Source Port Index (O)	Section 10.10.1.2
Source Channel Index (O)	Section 10.10.1.3
Source Sub-Channel Index (O)	Section 10.10.1.4
Destination Optical-Network-Administered IP Address (M)	Section 10.10.1.1
Destination Port Index (O)	Section 10.10.1.2
Destination Channel Index (O)	Section 10.10.1.3
Destination Sub-Channel Index (O)	Section 10.10.1.4
Connection ID (M)	Section 10.10.1.6
Contract ID (M)	Section 10.10.4.1
Framing Type (M)	Section 10.10.2.1
Overhead Termination Type (M)	Section 10.10.2.2
Bandwidth (M)	Section 10.10.2.3
Directionality (O)	Section 10.10.2.4
Service Level (O)	Section 10.10.2.5
Diversity (O)	Section 10.10.3.1

10.2 Connection Create Response

This message is sent from

- the terminating UNI-C to UNI-N to accept an incoming connection create request;
- the UNI-N to the initiating UNI-C to indicate the successful creation of (or failure to create) a connection requested previously

The attributes carried in this message are listed below:

Attributes	Reference
Source Optical-Network-Administered IP Address (M)	Section 10.10.1.1
Source Port Index (O)	Section 10.10.1.2
Source Channel Index (O)	Section 10.10.1.3
Source Sub-Channel Index (O)	Section 10.10.1.4
Destination Optical Network Administered IP Address(M)	Section 10.10.1.1
Destination Port Index (O)	Section 10.10.1.2
Destination Channel Index (O)	Section 10.10.1.3
Destination Sub-Channel Index (O)	Section 10.10.1.4
Connection ID (M)	Section 10.10.1.6
Connection Status (M)	Section 10.10.5.1

The source and destination termination point identifiers in this message merely serve to identify the corresponding connection create request. A specific signaling implementation may use other methods (e.g., LDP Message ID) to correlate a response with a request.

10.3 Connection Delete Request

This message is sent from

- the initiating UNI-C to UNI-N to delete a connection;
- the UNI-N to a UNI-C to indicate the deletion of a connection by the network.

The attribute carried this message is listed below:

Attribute	Reference
Connection ID (M)	Section 10.10.1.6

10.4 Connection Delete Response

This message is sent from

- the terminating UNI-C to UNI-N to acknowledge an incoming connection delete request.
- the UNI-N to the initiating UNI-C to indicate the successful deletion of the connection as requested previously.

The attributes carried this message are listed below:

Attribute	Reference
Connection ID (M)	Section 10.10.1.6
Connection Status (M)	Section 10.10.5.1

10.5 Connection Status Enquiry

This message is sent from

- the initiating UNI-C to UNI-N to enquire about the status and/or the attributes of one or more connections owned by the UNI-C.
- the UNI-N to either UNI-C to enquire about the status of the attributes of one or more connections owned by the UNI-C.

The attribute carried in this message is listed below:

Connection ID (O)	Section 10.10.1.6

If the connection ID is not present then the attributes of all connections owned by the UNI-C is returned. Otherwise, the status of the indicated connection is returned

10.6 Connection Status Response

This message is sent from

- the UNI-N to the UNI-C to indicate the status of connection attributes as requested previously;
- the UNI-C to UNI-N to indicate the status of connection attributes as requested previously.

The attributes carried this message are listed below. If the status of multiple connections is being returned, the contents shown must repeat for each connection.

Connection ID (M)	Section 10.10.1.6
Connection Status (M)	Section 10.10.5.1
Source Optical-Network-Administered IP Address (O)	Section 10.10.1.1
Source Port Index (O)	Section 10.10.1.2
Source Channel Index (O)	Section 10.10.1.3
Source Sub-Channel Index (O)	Section 10.10.1.4
Destination Optical-Network-Administered IP Address (O)	Section 10.10.1.1
Destination Port Index (O)	Section 10.10.1.2
Destination Channel Index (O)	Section 10.10.1.3
Destination Sub-Channel Index (O)	Section 10.10.1.4
Contract ID (O)	Section 10.10.4.1
Framing Type (O)	Section 10.10.2.1
Overhead Transparency Type (O)	Section 10.10.2.2
Bandwidth (O)	Section 10.10.2.3
Directionality (O)	Section 10.10.2.4
Service Level (O)	Section 10.10.2.5
Diversity (O)	Section 10.10.3.1

10.7 Address Resolution Query

This message is sent from the initiating UNI-C to UNI-N to resolve a client-layer address into an optical network point of attachment identifier (see Section 6).

The attributes carried in this message are listed below:

Attributes	Reference
Remote Client Address Type (M)	Section 10.10.1.7
Remote Client Address (M)	Section 10.10.1.8
Remote Client User Group ID (M)	Section 10.10.1.5

10.8 Address Resolution Reply

This message is sent from the UNI-N to the initiating UNI-C to indicate a an optical network point of attachment identifier corresponding to a client layer address (see Section 6).

The attributes carried in this message are listed below:

Attributes	Reference
Client Address Type (M)	Section 10.10.1.7
Client Address (M)	Section 10.10.1.8
Client User Group ID (M)	Section 10.10.1.5
Optical-Network-Administered IP Address (M)	Section 10.10.1.1
Port Index (O)	Section 10.10.1.2

10.9 Notification

This message is sent autonomously by UNI-N to either UNI-C to indicate a change in the status of the connection (e.g., unrestorable connection failure). The attributes carried in this message are listed below:

Connection ID	Section 10.10.1.6
Connection Status	Section 10.10.5.1

10.10 Description of Attributes

The attributes are classified into identification-related, service-related, routing-related, policy-related and miscellaneous. The encoding of these attributes would depend on the signaling protocol used and are described in Sections 12 and 13. In this section, the attributes are described in a general manner.

10.10.1 Identification-Related Attributes

10.10.1.1 Optical-Network-Administered IP Address

This is the IPv4 address associated with a client point of attachment to the optical network.

10.10.1.2 Port Index

This is an integer that indicates a port in an ONE. The range desired for this attribute is from 0 to 2^{16}-1.

10.10.1.3 Channel Index

This is an integer that indicates a channel with respect to the specified port ID. The range desired for this attribute is from 0 to 2^{8}-1.

10.10.1.4 Sub-Channel ID

This is an integer that indicates a sub-channel with respect to the specified channel ID. The range desired for this attribute is from 0 to 2^{8}-1.

10.10.1.5 User Group ID

This identifier logically distinguishes between different client networks that are autonomously operated. In essence, it identifies a specific virtual private network when different client networks are interconnected over the optical network. The user group ID is the 7-octet structure as described in IETF RFC 2685 [4].

10.10.1.6 Connection ID

This identifier uniquely identifies the connection within the optical network. The connection ID is created by the optical network in response to a connection create request and conveyed to the initiating UNI-C in "Connection Create Response" message. It is conveyed to the terminating UNI-C in the "Connection Create Request" message. This identifier is set to a NULL value when sent from the initiating UNI-C to UNI-N in the "Connection Create Request" message. The connection ID is not interpreted by clients, and hence treated as a string of bytes in UNI signaling messages.

10.10.1.7 Client Address Type

This attribute indicates the type of client address. The address types encoded are

- IPv4

- IPv6
- ITU-T E.164 ATM End System Address (AESA)
- British Standards Institute ICD AESA
- ANSI DCC AESA
- NSAP address

10.10.1.8 Client Layer Address

This attribute indicates a specific client layer address. The address must correspond to the indicated address type. The following addresses are permitted:

- IPv4 address (32 bits)
- IPv6 address (128 bits)
- ITU-T E.164 ATM End System Address (AESA) (160 bits)
- British Standards Institute ICD AESA (160 bits)
- ANSI DCC AESA (160 bits)
- NSAP address (160-bits)

10.10.2 Service-Related Attributes

10.10.2.1 Framing Type

This is an integer that specifies the framing format of the signal to be transported across the UNI. The framing options specified are:

- SONET T1.105
- SDH G.707

10.10.2.2 Overhead Termination Type

This integer field is framing specific. For SONET and SDH framing this field specifies to what degree the framing overhead bytes are terminated. Consistent with ITU definitions, the following values are supported:

- **Regenerator Section (RS):** Signal without termination of any overhead.

- **Multiplex Section (MS):** Signal with the possible termination of RS overhead.

- **Virtual Concatenation (VC):** Signal with the possible termination of RS, MS and Tandom Connection Monitoring (TCM) overhead.

10.10.2.3 Bandwidth

This attribute specifies the bandwidth of the service and it is interpreted with respect to the framing. For SONET the permitted values are STS-1 through STS-768. For SDH, the permitted values are STM-1 through STM-256.

10.10.2.4 Directionality

This is an integer that indicates whether the connection is uni-directional or bi-directional. Default is bi-directional.

10.10.2.5 Service Level

This integer attribute indicates a class of service. A carrier may specify a range of different classes of service (e.g. gold, silver, bronze) with predefined characteristics (e.g. restoration plans). The pre-defined service types correspond to different types of restoration (e.g. no restoration, 1+1 protection), connection set-up and hold priorities, reversion strategies for the connection after failures have been repaired, and retention strategies. The range desired for this attribute is from 0 to 2^{16}-1. Some values (e.g., 0-255) should be reserved for future use. The remaining values are provider specific. Default is set by the provider.

10.10.3 Routing-Related Attributes

10.10.3.1 Diversity

For a new connection being created, this attribute indicates a list of *n* existing connections with which diversity is desired. This attribute contains *n* items of the form <*diversity type*, connection ID>, where

- diversity type indicates whether node, link, or SRLG diversity is desired, and
- connection ID identifies the connection with which the new connection must be diverse in the sense indicated by diversity type.

10.10.4 Policy-Related Attributes

10.10.4.1 Contract ID

This identifier is assigned by the service provider and configured in clients. This is not interpreted by the clients and treated as a string of characters.

10.10.5 Miscellaneous Attributes

10.10.5.1 Connection Status

This is an integer that indicates the status of a connection. The range desired for this integer is from 0 2^8-1. The following values are defined:

- Connection active
- Connection does not exist
- Connection primary failed
- Connection under restoration

10.10.5.2 Error Code

This is an integer that indicates any errors resulting from connection actions. The particular errors indicated and the encoding of the error code are left up to specific signaling protocol definitions (see Sections 12 and 13).

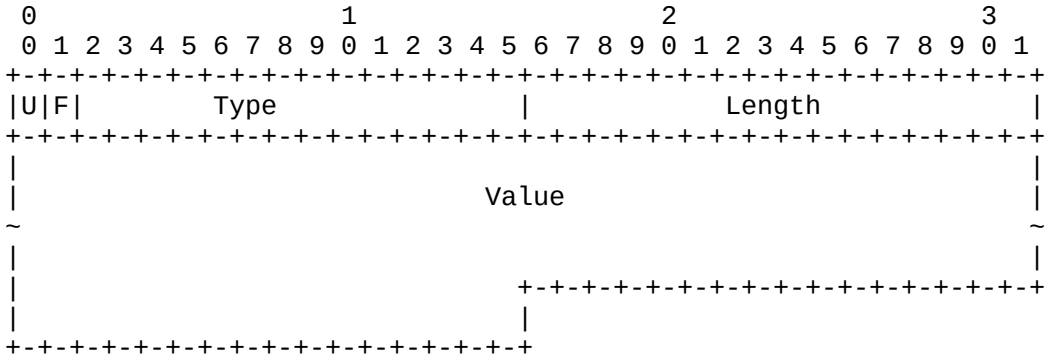
11 LDP Extensions for UNI Signaling

11.1 Overview

The Label Distribution Protocol (LDP) [1] has been defined for distributing labels among Label Switched Routers (LSRs) in a Multi-Protocol Label Switching (MPLS) network. Two LSRs which directly communicate using LDP to exchange labels are known as “LDP Peers”. An “LDP Session” (realized over a TCP connection) is required between peer LSRs. LDP procedures permit LSRs to establish Label Switched Paths (LSPs) through an MPLS network by mapping network-layer routing information directly to data-link layer switched paths.

LDP associates a Forwarding Equivalence Class (FEC) with each LSP it creates. The FEC associated with an LSP specifies which packets are “mapped” to that LSP at the ingress LSR.

All LDP messages have a common structure that uses a Type-Length-Value (TLV) encoding scheme as shown in Figure 12-1. The number of bits allocated for each field is as shown. The Value part of a TLV-encoded object, or TLV for short, may itself contain one or more TLVs.



There are four categories of LDP messages:

- Discovery messages, used to announce and maintain the presence of an LSR in a network.
- Session messages, used to establish, maintain, and terminate sessions between LDP peers.
- Advertisement messages, used to create, change, and delete label mappings for FECs.
- Notification messages, used to provide advisory information and to signal error information.

Discovery messages provide a mechanism whereby LSRs indicate their presence in a network by sending a Hello message periodically. This is transmitted as a UDP packet to the LDP port at the `all routers on this subnet' group multicast address. When an LSR chooses to establish a session with another LSR learned via the Hello message, it uses the LDP initialization procedure over TCP transport. Upon successful completion of the initialization procedure, the two LSRs are LDP peers, and may exchange advertisement messages.

LDP uses the TCP transport for session, advertisement and notification messages; i.e., for everything but the UDP-based discovery mechanism.

The LDP Messages defined in [1] are:

Message Name	Function
Notification Message	LSR notification of advisory or error information
Hello Message	Peer discovery
Initialization Message	LDP session establishment
KeepAlive Message	Monitors the integrity of the LDP session transport connection
Address Message	Advertise interface addresses
Address Withdraw Message	Withdraw previously advertised addresses
Label Mapping Message	Advertise FEC-label binding
Label Request Message	Request a binding (mapping) for a FEC
Label Abort Request Message	Abort an outstanding request
Label Withdraw Message	Breaks up the mapping between the FEC and the labels
Label Release Message	Signals the no need for specific FEC-label mapping.

LDP can operate in a number of modes depending on label distribution mode (*independent* or *ordered*), label retention mode (*conservative* or *liberal*), and label advertisement mode (*downstream on demand* or *downstream unsolicited*). These are defined in [1].

11.2 Use of LDP for UNI Signaling

The LDP extensions for UNI signaling defined in this section

- Utilize LDP message formats and messages to carry UNI signaling information;
- Make use of LDP session management and control procedures;
- With a few additions, utilize the LDP procedures already specified for notification of errors;
- Reuse the LDP security mechanism; and
- Utilize Generalized MPLS (GMPLS) signaling object formats [8] to encode UNI attributes wherever possible.

The following LDP mechanisms and messages defined in [1] are adapted for UNI signaling:

- Basic and/or extended discovery mechanisms.
- Label Request Message in downstream on demand label advertisement mode with ordered control.
- Label Mapping Message in downstream on demand label advertisement mode with ordered control.
- Notification Message.
- Withdraw and Release Messages.

Additional messages are defined to support the propagation of connection status information.

11.3 UNI Session Management and Control

LDP messages that relevant to the UNI session management and control are the Hello message, the Initialization message, and the KeepAlive message.

11.3.1 Hello Message

The use of the LDP Hello message for discovery is *optional* at the UNI. If used, the LDP Hello Message formats and procedures for UNI signaling are as defined in Section 3.5.2. of [1].

11.3.2 KeepAlive Message

The LDP KeepAlive message format and procedures for UNI signaling are as defined in Section 3.5.4 of [1].

11.3.3 Initilaization Message

The Initilaization Message is as defined in section 3.5.3 of [1] with the following modifications:

- The Label Advertisement Discipline (the "A" bit) is always set at 1 to indicate Downstream on Demand label distribution mode. Downstream on Demand is the only label distribution mode supported at the UNI. The assignment A=0 should result in generating a Notification Message with the appropriate error code.
- Loop Detection is always disabled, with the "D" bit set to 0. The assignment D=1 should result in generating a Notification Message with the appropriate error code.

11.4 LDP Messages for UNI Signaling

Section 10.1 defines the UNI abstract messages. Each of these messages is realized using an appropriate LDP message. In processing these messages,

- The LDP FEC TLV, if present, is ignored since it has no significance, and
- The LDP Message ID semantics is preserved.

The UNI abstract messages and the corresponding LDP messages are as follows:

1. *Connection Create Request*: The LDP Label Request Message is used to realize this abstract message. The Generalized Label Request TLV defined in [9] is used to convey some of the connection attributes to the network side.
2. *Connection Create Response*: The LDP Label Mapping Message is used to realize this abstract message. The Generalized Label TLV defined in [10] is used to convey some of the attributes. The Label Mapping procedures are limited to downstream on demand, ordered control mode with conservative label retention mode [1].
3. *Connection Delete Request*: The LDP Label Release Request Message is used to realize this abstract message. This message can be sent from the client or the network at any time after the establishment of the connection to delete it.
4. *Connection Delete Response*: The LDP Label Withdraw Message is used to realize this abstract message. This message is sent from the client or the network in response to a Label Release Request.
5. *Connection Status Enquiry*: A new LDP message, called Status Enquiry, is used.
6. *Connection Status Response*: A new LDP message, called Status Response, is used.
7. *Notification*: The CR-LDP Notification Message [11] is used to realize this abstract message.
8. *Address Resolution Query*: A new LDP message, called AR Query, is used.
9. *Address Resolution Response*: A new LDP message, called AR Response, is used.

The UNI 1.0 attributes and the corresponding LDP parameters are shown in the following table:

UNI Attribute	LDP Parameter
Bandwidth, Framing Type, Overhead Termination Type	Generalized Label Request [9]
Contract ID	Policy TLV (New)

Service Level, Directionality,	Service TLV (New)
Diversity	Diversity TLV (New)
Connection ID	Connection_ID TLV (New)
Error code	Status TLV (New)
Connection Status	Connection Status TLV (New)
Termination Point	Termination Point TLV (New)
User Group ID	User Group ID

The LDP messages and procedures are described in detail below.

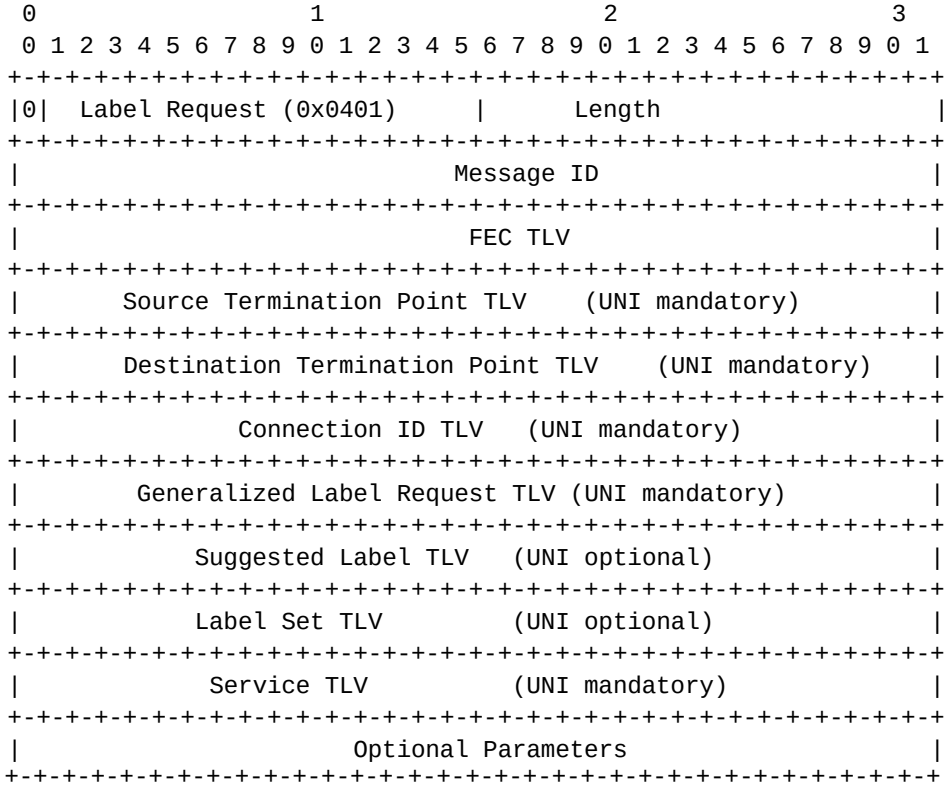
11.5 LDP Message Extensions

11.5.1 Label Request Message

The LDP Label Request message is used for UNI signaling as defined in 3.5.8. of [1] with the following modifications:

- The FEC TLV is ignored at the UNI
- The procedures to handle the Label Request Message are augmented by the procedures for processing of the UNI TLVs, as defined in this section

The encoding for the UNI LDP Label Request Message is as shown below. The TLVs and other parameters are described in Section 11.6.



11.5.1.1 Procedure

The Label Request Message is sent from

- the initiating UNI-C to UNI-N to request the creation of a connection;
- the UNI-N to the terminating UNI-C to indicate an incoming connection request.

In the Label Request Message, the initiating UNI-C identifies the two connection termination points (Source Termination Point and Destination Termination Point TLVs). For initial set-up, the ActFlg is set to 0. The UNI-N is expected to assign a Connection ID that is unique within the optical network. The Connection Id is passed to the terminating UNI-C in the Label Request Message from the corresponding UNI-N.

Upon the reception of the Label Request Message, the UNI-N should verify that the signaled attributes (including the validity of the Source and the Destination Termination Points) can be supported. Failure to support one or more of the connection attributes triggers the generation of the Notification Message with the appropriate error code.

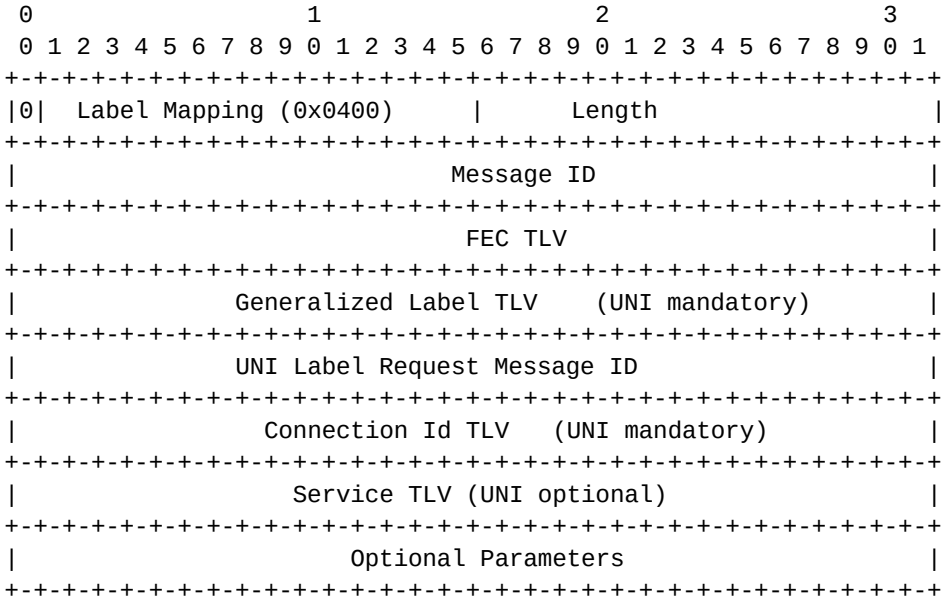
A Label Request Message with the same content as sent by the initiating UNI-C should be sent to the terminating UNI-C by the corresponding UNI-N entity. How this is accomplished within the optical network is outside the scope of this specification.

11.5.2 Label Mapping Message

The Label Mapping Message is used as defined in 3.5.7 of [1] with the following modification:

- The Label Mapping Message procedures are limited to downstream on demand ordered control mode.

The encoding of the UNI Label Mapping Message is as follows. The TLVs and other parameters are described in Section 11.6:



11.5.2.1 Procedure

The UNI Label Mapping Message flows between

- the terminating UNI-C to UNI-N in response to a Label Request message;
- the UNI-N to the initiating UNI-C to indicate the successful establishment of a connection requested previously

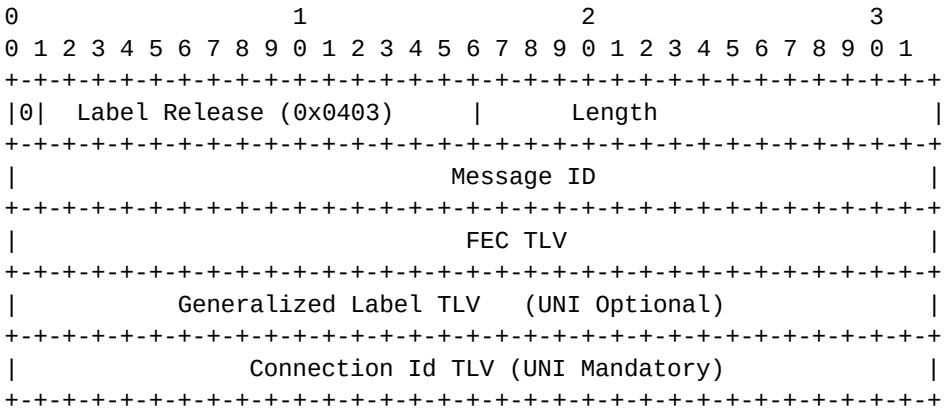
The network transports the assigned Connection Id to the calling client in the Label Mapping Message. This Connection Id value is used by the client and the network for the exchange of connection status information.

The UNI Label Mapping Message also includes a Generalized Label TLV. Its purpose is to indicate to the client label value, e.g. which wavelength, to be used.

The UNI Label Mapping Message optionally includes a Service TLV that summarizes the level of service extended from the optical network to its client. The Service TLV must be included for the cases where reserved connection attributes, e.g. its bandwidth, are different from those requested by the customer.

11.5.3 The Label Release Message

The format of the UNI Label Release Message is as follows:



11.5.3.1 Procedure

The UNI Label Release Message is sent by

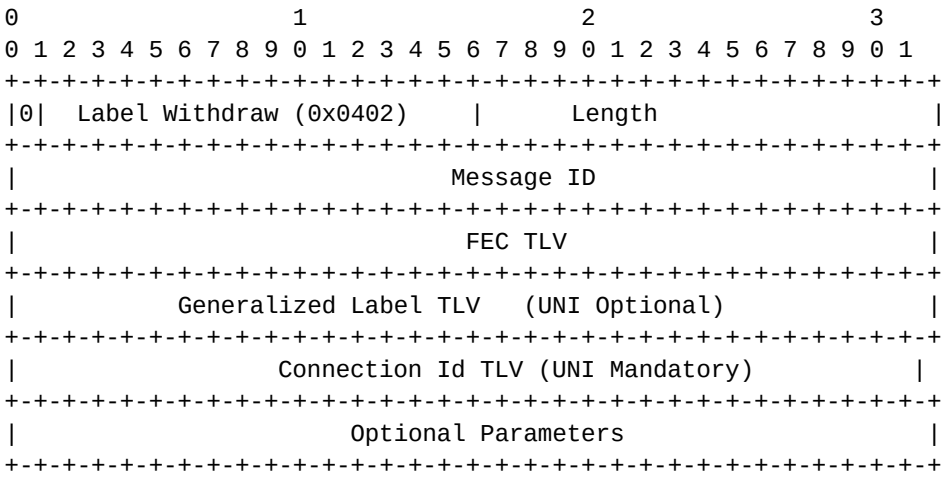
- the initiating UNI-C to UNI-N to delete a connection;

- the UNI-N to a UNI-C to indicate the deletion of a connection by the network.

The procedure for the UNI Label Release Message is as described in section 3.5.11. of [1]. The UNI Label Release Message carries a mandatory Connection Id to indicate which connection should be terminated. Optionally, the label assigned earlier to the connection may be included.

11.5.4 The Label Withdraw Message

The format for the UNI Label Withdraw Message is as follows:



11.5.4.1 Procedure

The Label Withdraw Message is sent by

- the terminating UNI-C to UNI-N to acknowledge an incoming connection delete request.
- the UNI-N to the initiating UNI-C to indicate the successful deletion of the connection as requested previously.

The procedure for the Label Withdraw Message follows that defined in section 3.5.10 of [1]. The Label withdraw Message for UNI carries a mandatory Connection Id. The reception of the Label Withdraw Message acts as an indication to the client or the network that the connection defined by its Connection Id has been terminated.

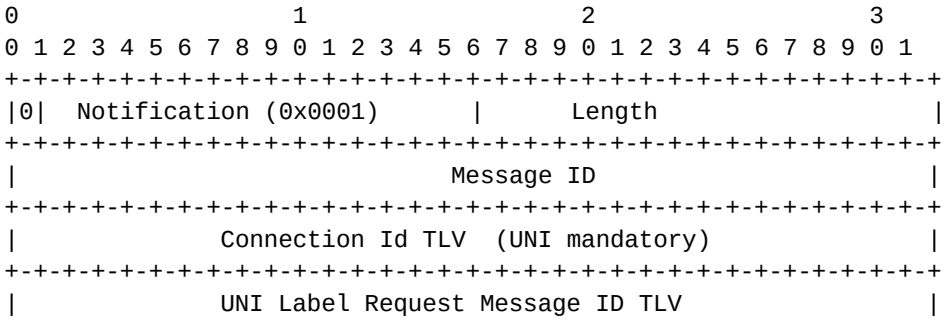
The message ID should match the message ID in the label release request, right? If so, need to add this here.

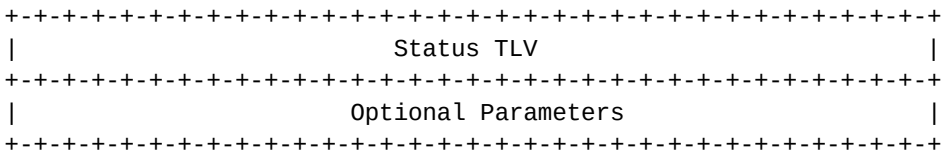
No, at least according to the LDP. Actually I think what is needed to be added the the generalized label TLV. Could please do this? Thanks.

11.5.5 The Notification Message

The Notification Message is as defined in section 3.5.1. of [1] with the following modifications:

- The UNI Notification Message is sent autonomously by UNI-N to the corresponding UNI-C to indicate the status of the connection request.
- The UNI Notification Message includes a mandatory Connection Id TLV





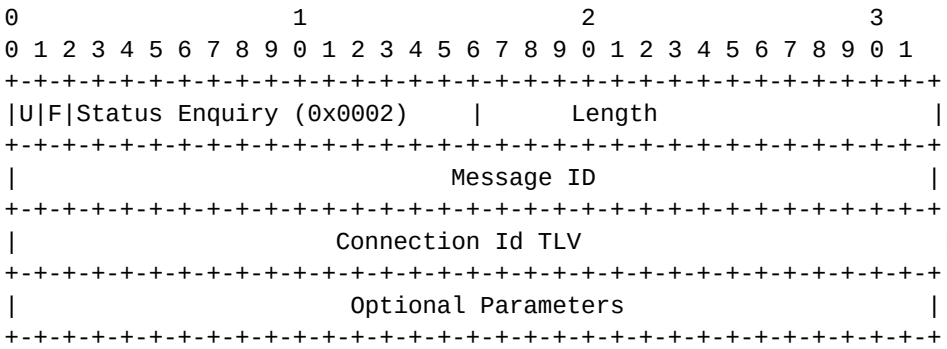
11.5.5.1 Procedure

The UNI Notification Message is used by the optical network to signal to its clients failure condition during or after the connection establishment phase.

If it has been already set, the Notification Messages includes the Connection Id TLV. If not set, e.g. for initial set up, the Connection Id TLV is set to 0. If the Connection Id is not set, the Notification Message must include an UNI Label Request Message ID TLV as defined in section 6.2.2 in [11] .

11.5.6 The Status Enquiry Message

The Status Enquiry Message is a new LDP message. The encoding for the Status Enquiry Message is:

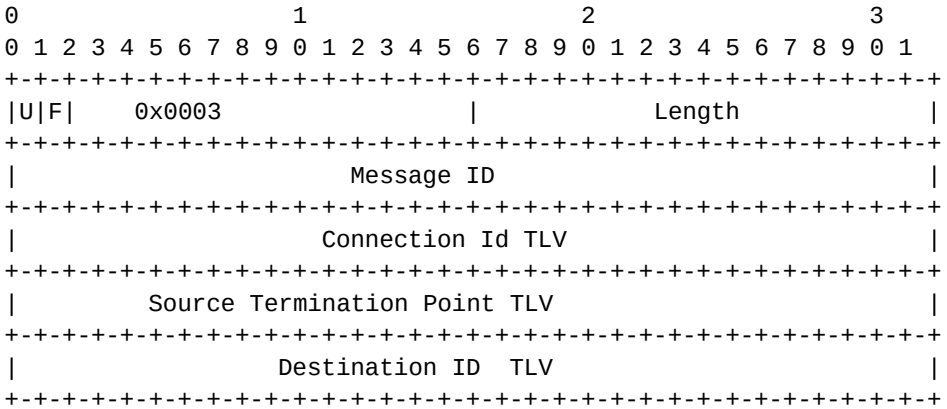


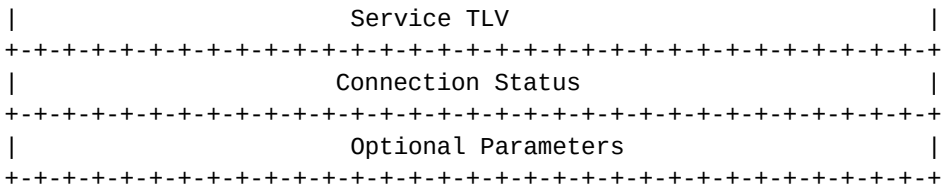
11.5.6.1 Procedure

The Status Enquiry message is sent by UNI-C or UNI-N at any time to solicit a Status Response message from its peer. The connection under consideration is identified by the Connection Id TLV.

11.5.7 The Status Response Message

The Status Response message is a new LDP message. The encoding for the Status Response message is:





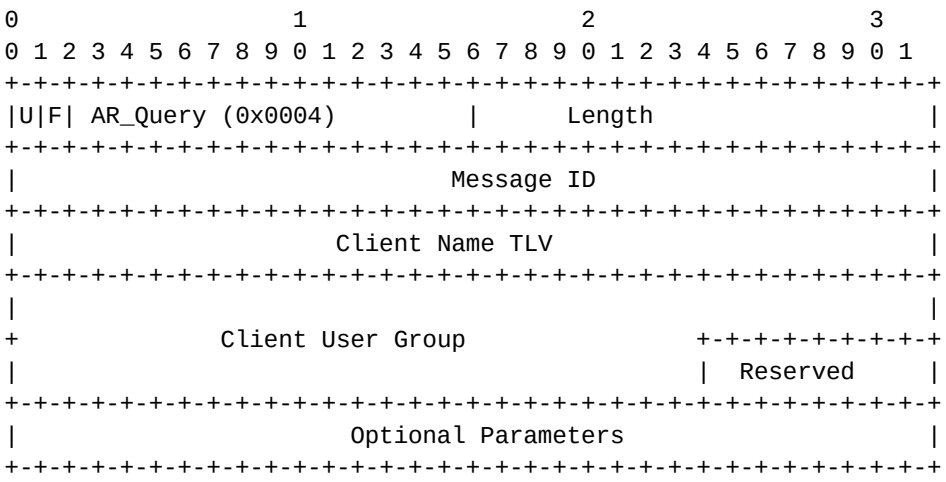
11.5.7.1 Procedure

The Status Response message is sent by UNI-C or UNI-N in response to a Status Enquiry message. The Status TLV carries information that describes the current status of a connection as defined by the Connection Id TLV. The status of the connection is encoded using the LDP Status TLV.

The Status Response Message could optionally include connection attributes as defined by Source Termination Point, Destination Termination Point, and the level of service.

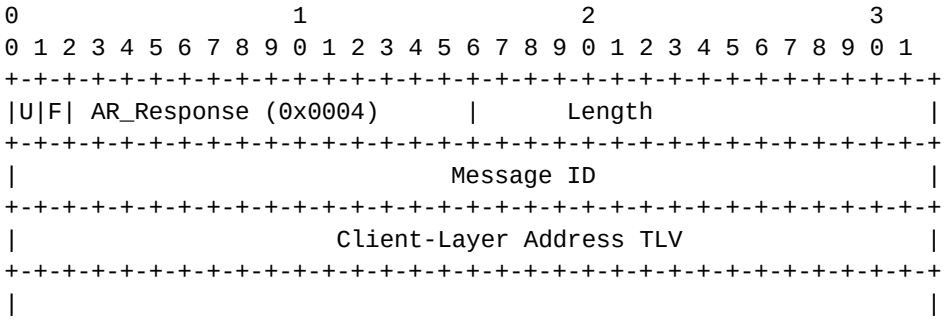
11.5.8 Address Resolution Query Message

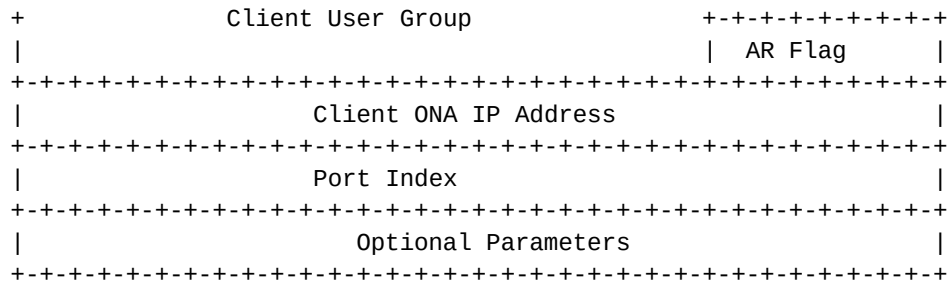
Address resolution query is a new message added to LDP. The encoding of the address resolution query (AR_Query) Message is:



11.5.9 Address Resolution Response Message

The encoding for the address resolution response (AR_Response) Message is:





11.5.9.1 Procedure

The initiating UNI-C can issue an address resolution query at any time by sending the LDP AR_Query Message to the corresponding UNI-N. The UNI-C inserts the appropriate client-layer address in the Client-Layer AddressTLV.

The UNI-N responds to the AR_Query message by sending an AR_Response Message back to the UNI-C. The AR Flag field indicates whether address resolution was successful. The AR_Response Message contains the ONA IPv4 address to be used in signaling messages, if address resolution was successful.

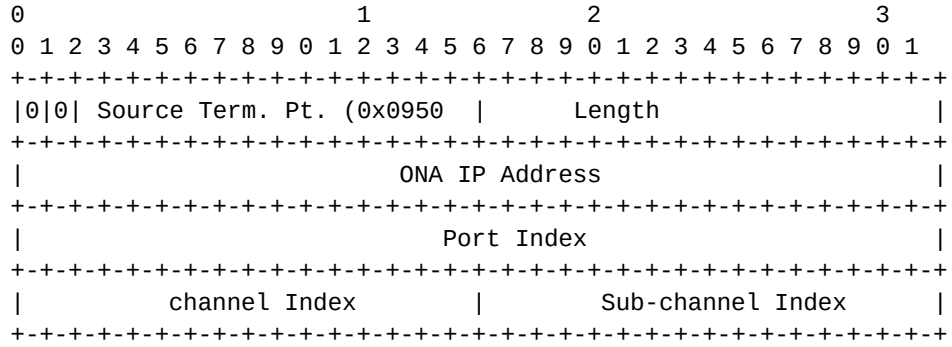
11.6 Definition of TLVs and other Parameters used in UNI Signaling

11.6.1 Message ID

A 32-bit value identifying the message. The procedure for assigning the message ID, as defined in [1], should be used.

11.6.2 Source Termination Point TLV

The Source Termination Point TLV identifies the connection termination point (Section 6.3) on the initiating client side. The encoding of the Source Termination Point TLV is:



ONA IP Address:

This the optical-network-assigned (ONA) IPv4 address associated with the connection termination point (Section 6.2).

Port Index:

Port Index is a four-octet unsigned integer indicating the port number in an ONE (Section 6.2). Since port index is optional, a NULL value of 0xFFFFFFFF (all 1s) is defined to indicate an unspecified port. A NULL port index shall be ignored by the UNI-N.

Channel Index:

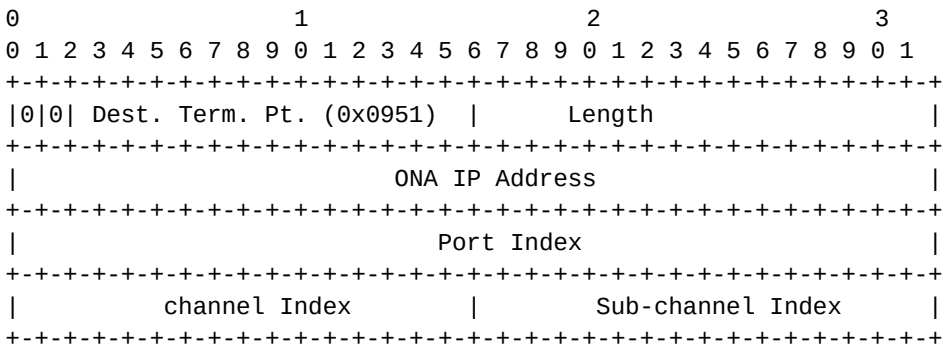
Channel Index is a two-octet unsigned integer indicating a channel with respect to the specified port (Section 6.3). Since channel index is optional, a NULL value of 0xFFFF (all 1s) is defined to indicate an unspecified channel. A NULL channel index shall be ignored by the UNI-N.

Sub-Channel Index:

Sub-Channel Index is a two-octet unsigned integer indicating a sub-channel with respect to the specified channel (Section 6.3). Since sub-channel index is optional, a NULL value of 0xFFFF (all 1s) is defined to indicate an unspecified channel. A NULL channel index shall be ignored by the UNI-N.

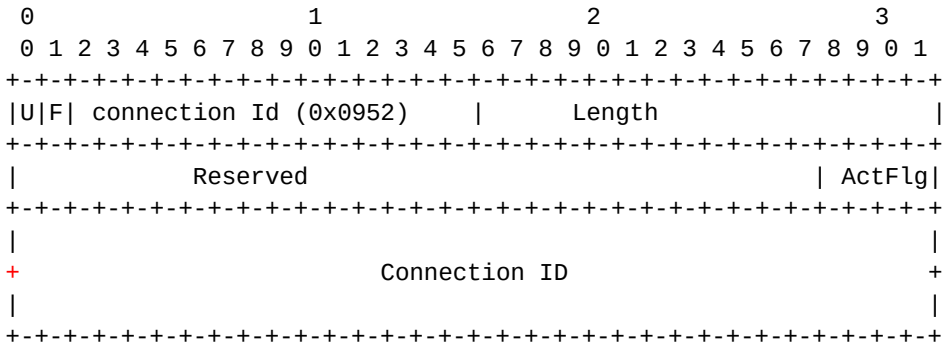
11.6.3 Destination Termination Point TLV:

The Destination Termination Point TLV identifies the connection termination point (Section 6.3) on the destination client side. It has the same structure as the Source Termination Point TLV. The format of the Destination Termination Point TLV is:



11.6.4 Connection Id TLV

The format of the Connection Id TLV is as follows:



ActFlag:

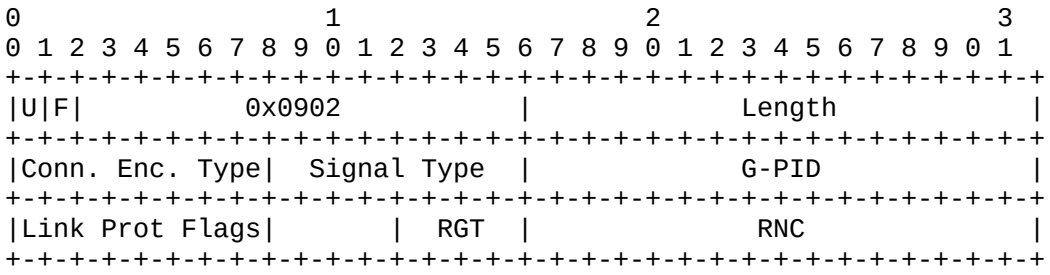
A 4-bit field that explicitly indicates the action that should be taken on an already existing connection. The code points are

0x0 = initial connection setup
0x1 = modify connection (for future inclusion)

Connection ID:
A network-assigned 64-bit identifier.

11.6.5 Generalized Label Request TLV

This TLV encodes the Framing Type, OH Termination Type and Bandwidth abstract parameters (Section 10). The format of a Generalized Label Request TLV is [9]:



Connection Encoding Type
This 8-bit unsigned integer indicates the framing type. The following are permitted values under UNI 1.0 signaling:

Value	Type
5	SDH
6	SONET

Signal Type
This 8 bit unsigned integer indicates the overhead termination type and bandwidth of the signal. This parameter is interpreted in relation to the LSP Encoding Type.

Permitted signal type values for SDH are:

Value	Type
8	STM-1
9	STM-1 MS
10	STM-1 RS
12	STM-4
13	STM-4 MS
14	STM-4 RS
16	STM-16
17	STM-16 MS
18	STM-16 RS
20	STM-64
21	STM-64 MS
22	STM-64 RS
24	STM-256
25	STM-256 MS
26	STM-256 RS

The “STM-N MS” and “STM-N RS” signal types represent transparent STM Multiplex Section and Regenerator Section LSPs respectively. Simply, “STM-N” signifies path layer transparency,

that is, the set of AUs contained within the STM-N taken as a group (equivalent to AUG-N). These are defined for the standard values of N (1, 4, 16, 64, 256).

Permitted values for SONET are:

Value	Type
6	STS-1
7	OC-1 Line
8	OC-1 Section
10	OC-3 Path Group
11	OC-3 Line
12	OC-3 Section
14	OC-12 Path Group
15	OC-12 Line
16	OC-12 Section
18	OC-48 Path Group
19	OC-48 Line
20	OC-48 Section
22	OC-192 Path Group
23	OC-192 Line
24	OC-192 Section
26	OC-768 Path Group
27	OC-768 Line
28	OC-768 Section

SONET group (OC-n Path Group) and OC-N transparent line/section Signal Types are defined in the same way as their SDH counterparts above.

Generalized PID (G-PID):

This is a 16-bit unsigned integer that identifies the payload carried by an LSP. This must be interpreted according to LSP Encoding Type and is used by the nodes at the endpoints of the LSP.

Permitted GPID values for SDH are:

Value	Client Type
0	Unknown
1	Asynchronous mapping of E4
2	Asynchronous mapping of DS3
3	Asynchronous mapping of E3
4	Bit synchronous mapping of E3
5	Byte synchronous mapping of E3
6	Asynchronous mapping of DS2
7	Bit synchronous mapping of DS2
8	Byte synchronous mapping of DS2
9	Asynchronous mapping of E1
10	Byte synchronous mapping of E1
11	Byte synchronous mapping of 31 * DS0
12	Asynchronous mapping of DS1
13	Bit synchronous mapping of DS1
14	Byte synchronous mapping of DS1
15	ATM mapping

Permitted GPID values for SONET are:

Value	Client Type
0	Unknown
1	DS1 SF Asynchronous
2	DS1 ESF Asynchronous
3	DS3 M23 Asynchronous
4	DS3 C-Bit Parity Asynchronous
5	VT
6	STS
7	ATM
8	POS

Link Protection Flags: 8 bits.

Ignored in UNI 1.0 signaling.

Requested Grouping Type (**RGT**): 4 bits

Ignored in UNI 1.0 signaling.

Requested Number of Components (**RNC**): 16 bits

Ignored in UNI 1.0 signaling.

11.6.6 Suggested Label TLV

The Suggested Label TLV format and procedure are as defined in section 3.4. of [10].

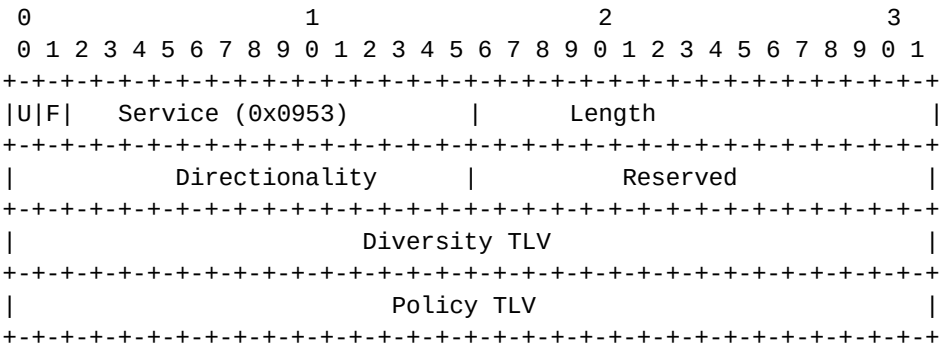
The Suggested Label is used to provide a downstream node with the upstream node's label preference. This permits the upstream node to start configuring it's hardware with the proposed label before the label is communicated by the downstream node. This is included for compatibility with GMPLS signaling.

11.6.7 Label Set TLV

The format and the procedure of the Label Set TLV is as described in section 3.5. of [10]. This is included for compatibility with GMLS signaling.

11.6.8 Service TLV

The Service TLV defines the service attributes requested by the network client. The encoding of the Service TLV is as follows:



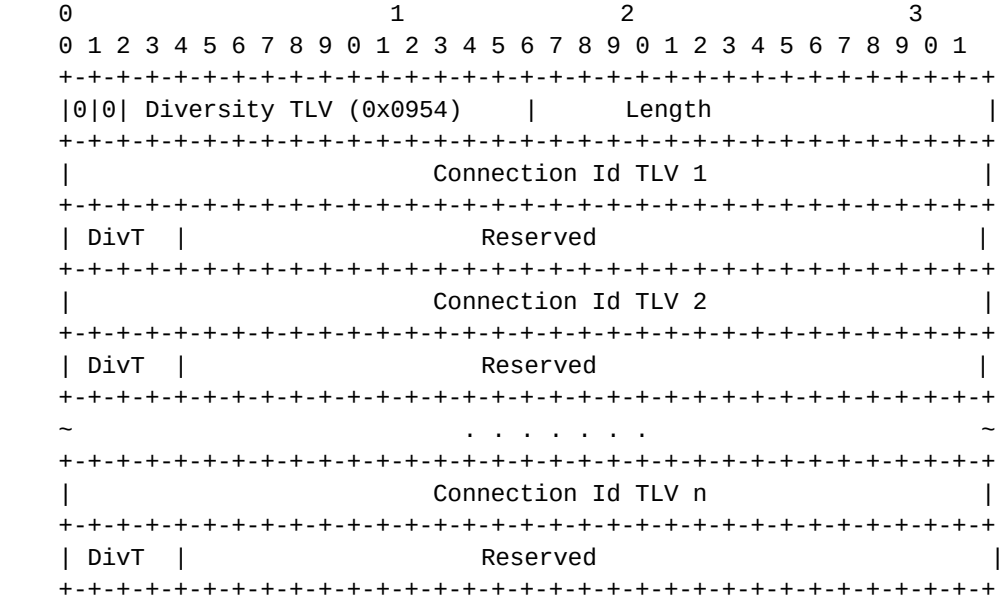
Directionality:

Directionality is 16-bit field that specifies the directionality of the requested connection. The allowed values are:

0x0000 = Uni-directional
0x0001 = Bi-directional

Diversity TLV:

Diversity TLV lists all the other connections from which the requested connection MUST be diverse. It also specifies the type of diversity. Diversity is only valid within a single routing domain. The encoding of the Diversity TLV is as follows:



Connection Id TLV *n:*

This is the Connection Id of the LSP from which the requested ligthpath must be diverse.

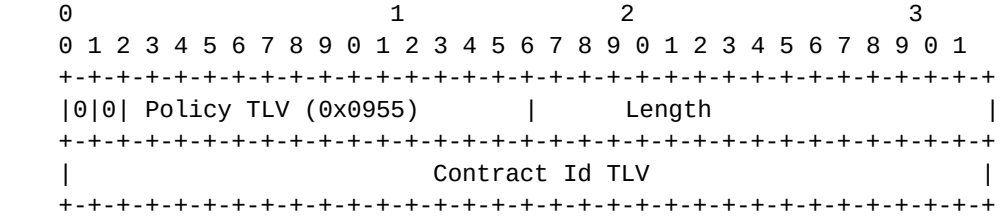
DivT (**Diversity Type**):

DivT specifies the manner by which the requested connection should be diverse. The allowed values are:

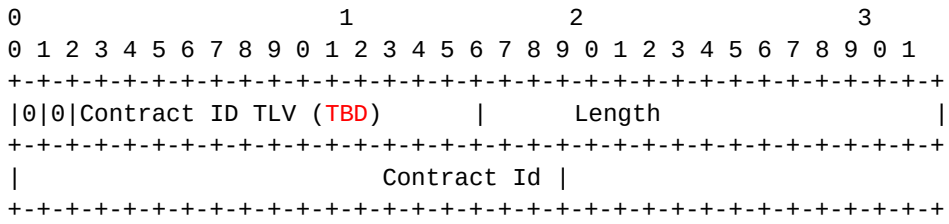
0x0 = Link diverse
0x1 = Node diverse
0x2 = Shared Risk Link Group (SRLG) diverse

Policy TLV:

The format of the Policy TLV is as follows. For UNI 1.0, this TLV encodes only the Contract ID. Other policy-related information may be encoded in the future.



Contract ID TLV:



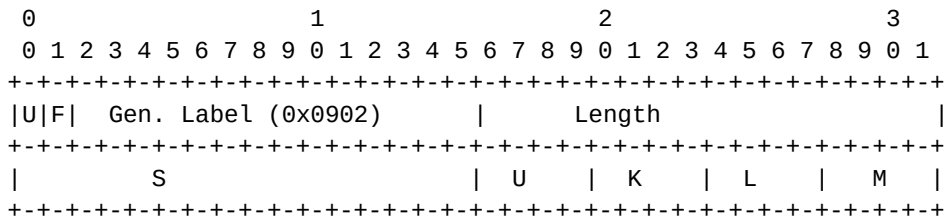
This is a variable-length string of characters. Contract ID is assigned by the provider and carried without interpretation across the UNI.

11.6.9 Optional Parameters:

Variable length set of message optional parameters, e.g. vendor specific capabilities. When included, optional parameters may appear in any order.

11.6.10 Generalized Label TLV

The format of the Generalized Label TLV is as follows [10]:



S is the index of a particular STM-1/STS-1 signal. S = 1 to 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.

The label components U, K, L and M must be ignored in UNI 1.0 signaling.

The procedures for setting these parameters is as defined for GMPLS signaling [8].

11.6.11 UNI Label Request Message ID

This is the 32-bit message ID received in the corresponding Label Request Message.

11.6.12 Status TLV

The Status TLV is as defined in section 3.4.6. of [1]. New Status codes relevant to UNI signaling are:

0x00001000 = not able to connect to destination
0x00001001 = invalid destination address
0x00001002 = invalid port Index
0x00001003 = invalid channel index
0x00001004 = invalid sub-channel index

0x00001005 = bandwidth unavailable
0x00001006 = protection mode unavailable
0x00001007 = routing directive unavailable
0x00001008 = failure to create connection
0x00001009 = failure to modify connection
0x0000100A = Failure to delete connection
0x0000100B = Encoding unavailable

11.6.13 User Group

This is a 7-octet VPN identifier as defined in [4].

11.6.14 Connection Status

Connection Status is a 32-bit unsigned integer parameter. At UNI-C the connection states are:

- *Null*: No connection exists
- *Connection Initiated*: The UNI-C has sent a Label Request Message, but has not yet received the Label Mapping Message from the corresponding UNI-N.
- *Connection Present*: The UNI-C has received a Label Request Message from the UNI-N, but hasn't yet responded to it
- *Active*: The terminating UNI-C has sent a Label Mapping Message to the corresponding UNI-N. The state also exists when the initiating UNI-C receives the Label Mapping Message from the corresponding UNI-N.
- *Release Requested*: The UNI-C has sent a Label Release Message to the UNI-N.
- *Release Indicated*: The UNI-C has received a Label Release Message.
- *Dropped*: This state exists when a connection was made and successfully established but was dropped by the provider.

Similar connection states exist at the UNI-N.

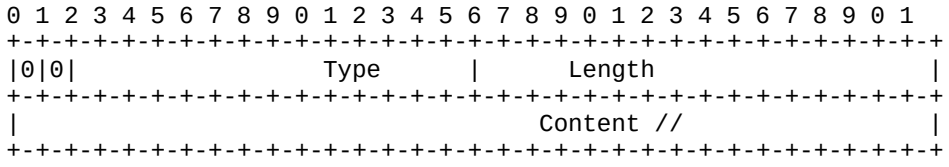
The Connection Status codes for the connection states are:

0x0000100C = Null
0x0000100D = Call Initiated
0x0000100E = Call Present
0x0000100F = Active
0x00001010 = Release Requested
0x00001011 = Release Indicated
0x00001100 = Dropped

The connection states defined here form a subset of the connection states defined in [12].

11.6.15 Client-Layer Address TLV

This TLV is the destination name that needs to be resolved to obtain the destination IP address used for signaling. The encoding of this TLV is:



Type is a 14-bit field carrying the type of the client-layer address. Currently defined values are:

Type	Value
0x960	IPv4
0x961	IPv6
0x962	ATM NSAP
0x963	E.164
0x964	ICD

The Content field indicates the appropriate client-layer address value.

11.6.16 AR Flag

This is an 8-bit unsigned integer which indicates whether the address resolution attempt was successful. The values for this field are:

0x00 : Address Resolution Successful
0x01: Address Resolution Unsuccessful.

12 RSVP Extensions for UNI Signaling

12.1 Overview

The Resource reSerVation Protocol (RSVP) is an IETF-defined protocol for establishing network resources for IP sessions (or "flows") [13]. The RSVP definition consists of basic procedures, message and object formats for signaling in an IP network. RSVP with Traffic Engineering extensions (RSVP-TE) [2] has been defined for establishing LSPs subject to routing constraints in an MPLS network. The RSVP-TE definition includes additional procedures, message and object formats over the base RSVP definition. Generalized MPLS (GMPLS) signaling [8, 14] extends basic MPLS signaling procedures and abstract messages to cover different types of switching applications such as circuit switching, wavelength switching, etc.

In this section, UNI signaling based on adapting RSVP, RSVP-TE, and GMPLS procedures, messages and objects is defined. This definition leverages existing specifications to the maximum extent possible. A few new message and objects are defined to indicate connection attributes that are unique to UNI 1.0.

In this section, we define the directional terms "source" vs. "destination", "originating" vs. "terminating", "upstream" vs. "downstream", "previous hop" vs. "next hop", and "incoming interface" vs. "outgoing interface" with respect to the direction of connection as in [2, 8, 13].

12.2 Basic RSVP Protocol Operation

There are two fundamental RSVP message types: Path and Resv [13]. A node originates a connection establishment request by transmitting an RSVP Path message addressed to the connection destination. In response to receiving a Path message, the destination node sends a reservation request (Resv) message upstream towards the connection source. Resv messages create "reservation state" in each optical nodes at the source and destination UNI. A connection is established when the source user node receives a Resv message for the connection.

The Path and Resv state can be explicitly removed using the PathTear and ResvTear messages. The PathTear message is sent from the source to the destination and removes both the Path and Resv state for the associated connection. The ResvTear message is sent from the destination to the source and only removes the associated Resv state.

For each RSVP message type, there is a set of rules for the permissible choice of object types. These rules are specified using Backus-Naur Form (BNF). The BNF implies an order for the objects in a message. However, in many (but not all) cases, object order makes no logical difference. An implementation should create messages with the objects in the order shown for each message, and accept the objects in any permissible order.

12.3 UNI 1.0 Signaling Messages and RSVP Objects

The UNI 1.0 abstract messages were described in Section 10. Most of these may be directly supported by re-using existing procedures, messages, and objects defined in RSVP-TE [2] and in Generalized MPLS Signaling [8].

Section 10 also defines the set of attributes to be signaled. The following table summarizes those attributes and the corresponding RSVP objects. Specific UNI-related object formats and usage are described in Section 12.3.

UNI Attribute	RSVP Object
Bandwidth	Sender_Tspec [14]
Contract ID	Policy Data [15]
Destination ONA IP Address	Session [2]

Destination Port/Channe/SubChannel	ERO, Explicit Label Control [2]
Directionality	Upstream Label [14]
Diversity	Diversity (new)
Framing Type	Generalized Label Request [14]
Connection ID	Connection_ID (New)
Error code	Error Spec [13]
Service Type	Session Attribute (new CType)
Source ONA IP Address	Sender Template [2]
Source Port/Channel/Sub-Channel Indices	Label Set/Suggest Label [14]
Overhead Termination Type	Generalized Label Request [14]
User Group ID	Address Resolution Request/Response (new)
Address Resolution Request	Address Resolution Request (new)
Address Resolution Response	Address Resolution Response (new)

12.4 Use of RSVP-TE and Generalized MPLS signaling for UNI

The RSVP protocol definitions in this section apply only for UNI signaling. There is no implied requirement that RSVP-based signaling be supported within the optical network. It is, however, required that the network should provide for coordination of signaling information at the initiating and terminating side of the connection.

12.4.1 UNI Interfaces, Control Channels, and Addressing

RSVP messages are exchanged over the UNI signaling control channel. The determination of UNI control interfaces and the maintenance of the control channel are described in Section 8. The identification of connection endpoints is described in Section 6.

12.4.2 Sending UNI RSVP Messages

When an RSVP message is sent over a control channel that directly connects UNI-C and UNI-N (signaling reference configurations 1-3, Section 7), the message format defined in [13] (section 3.3) is used, i.e. messages are sent as “raw” IP datagrams with protocol number 46. The IP destination address of all O-UNI RSVP messages SHALL be set to the adjacent optical node's control channel interface identifier and the source address to its own control channel's interface identifier.

In addition, RSVP Path, PathTear, and ResvConf messages may be sent with an “IP router alert” option, but is not required.

The Notify message MAY be generated at any time to allow expedited notification of a change in the status of a connection. The IP destination address is set to the IP address of the intended receiver. The Notify message is sent without the router alert option.

The Query message MAY be generated at any time by an optical node to request and receive requested information from its adjacent node.

RSVP messages sent on a control channel that is realized over a multi-hop IP network (signaling reference configuration 3, Section 7) shall be encapsulated in IP-in-IP header before transmission.

12.4.3 Receiving UNI RSVP Messages

UNI-C or UNI-N SHALL verify a received RSVP message as specified in [2, 14]. In addition, if a Path message is received, the receiver SHALL verify that the IP address in the RSVP_HOP object matches an IPCC (Section 8) address advertised to a neighbor. If a match does not exist, the error code “routing

problem" SHALL be reported in a PathErr message. Otherwise, the message is associated with the control interface over which it was received.

12.4.4 Reliable Messaging

To support reliable messaging across the UNI, the Message_ID object and "Ack Desired" flag defined in [16] MUST be used in every RSVP message.

Message identification and acknowledgment are done on a per-hop basis. Each Message_ID object contains a message identifier. This identifier MUST be unique with respect to the object generator (as identified by its IP address). No more than one Message_ID object may be included in an RSVP message. Each message containing a Message_ID object may be acknowledged via a Message_ID_Ack object, when so indicated.

Message_ID_Ack and Message_ID_Nack objects may be sent piggybacked in unrelated RSVP messages or in RSVP Ack messages. RSVP messages carrying any of the three object types may be included in an RSVP *bundled* message. When included, each object is treated as if it were contained in a standard, non-bundled, RSVP message.

If a Message_ID_Ack object of an RSVP message cannot be piggybacked in an RSVP message immediately, the node SHALL generate an Ack message including the Message_ID_Ack object to its adjacent RSVP node. This allows faster confirmation of the message.

12.4.5 Reservation Style

RSVP reservation "styles" are defined in [13]. For UNI signaling, the Fixed Format (FF) style MUST be used.

12.4.6 Connection Identification

A new Connection_ID object is introduced to carry the optical-network-assigned unique connection identification. The Connection_ID MUST be carried in all subsequent RSVP messages once it is assigned by the network.

12.4.7 Address resolution

If a network does not support a client-layer addressing scheme, a user node MAY request the optical network point of attachment identifier corresponding to a client-layer address by sending a Query message.

Specifically, before a UNI-C signals for a connection, it may send a Query message (the Address Resolution Query) containing one or more client-layer addresses to the UNI-N. When the UNI-N receives such a Query message, it sends back a Query message (the Address Resolution Query Response) with the optical network point of attachment identifier corresponding to each client address. Upon receiving the Query message back from the UNI-N, the UNI-C can initiate connection requests specifying the obtained (destination) address(es).

12.4.8 Connection Creation

To create a connection, the client UNI node creates an RSVP Path message, with the Session Type set to LSP_TUNNEL_IPv4, and inserts a Generalized Label_Request object into the Path message. This object indicates that a label binding is requested for this connection and also provides an indication of the characteristics, such as encoding type, requested transparency, etc.

To create a bi-directional connection, an Upstream Label Object MUST be inserted in the Path Message. Therefore, if the encoding type in a Generalized Label Request object indicates a bi-directional connection

(default), an Upstream Label object MUST be inserted in the Path message; if no such an Upstream Label object is inserted, an error code “incompatible” SHALL be reported in PathErr Message. A terminating user node SHALL insert a RESV_CONFIRM object in Resv Message and waits for the ResvConf message before start transmission on a bi-directional connection.

12.4.9 Connection Deletion

There are three scenarios for connection deletion: deletion by the UNI-C that originated the connection establishment (source UNI-C), deletion by the destination UNI-C, and deletion by UNI-N.

The source UNI-C SHALL send a PathTear message to the corresponding UNI-N to delete a connection.

The destination UNI-C SHALL send a ResvTear message to the corresponding UNI-N to delete a connection. If the source UNI-C receives a ResvTear message for a connection, it MUST respond with a PathTear message to complete connection deletion.

A UNI-N MAY send a PathErr message with error code set to “Network Initiated Deletion, Normal” to request the source UNI-C to delete the connection. If the path state is autonomously removed in the network then the Path_State_Removed flag MUST be set in the PathErr message. Furthermore, a PathTear message MUST be sent to the destination UNI-C.

12.4.10 Connection Status Enquiry And Response

The purpose of connection status enquiry and response, as defined in Section 10, is to allow a UNI-C and the corresponding UNI-N to re-synchronize their connection states when necessary, for example, after a link or node failure. There could potentially be a large number of connections whose states have to be re-synchronized. Therefore, the procedure must allow the re-synchronization to occur efficiently. This specification uses the Srefresh message [16] for efficiency.

When a UNI-C or a UNI-N decides that it is necessary to resynchronize its connection state with its peer, it SHALL send Srefresh messages to refresh the Message_IDs of some or all active Resv and Path Messages. If a connection state has already been deleted by the peer, it MUST respond with a Message_Id_Nack Object for the deleted connection. Once a UNI-C or a UNI-N receives a Message_Id_Nack, it SHALL send a Resv or Path message to its peer. This would trigger the re-establishment of the connection.

The above procedure may change the reservation states in a UNI-C or a UNI-N. The need for a procedure that do not affect state is for further study.

12.4.11 Node Failure Detection

RSVP HELLO procedure [2] MAY be used when there are no other procedures available to detect the failure of UNI-C or UNI-N entities. The RSVP Hello extension enables a UNI-C or a UNI-N to detect the unreachability of its signaling peer. The mechanism also helps detect node failure detection.

When such a failure is detected it is handled much the same as a link layer communication failure. This mechanism is intended to be used when the link layer failure notification is not available, or when the failure detection mechanisms provided by the link layer are not sufficient for timely node failure detection.

The procedure is optional. When it is implemented, it should be configurable to be on or off when other node failure detection methods are available.

12.5 RSVP Messages And Objects For UNI Signaling

The following sections describe messages, procedures, and objects defined in [2, 13, 14, 15] that are related to UNI signaling. If this document does not explicitly specify some aspects of messages, procedures, and objects related to UNI signaling, the procedures defined in [8, 14], [2], [16], [15], and [13] SHALL apply in the order listed.

The UNI-C and UNI-N entities MUST support the following RSVP messages:

Path, PathTear, PathErr, Resv, ResvTear, ResvConf, ResvErr, Ack, Srefresh, Notify and Query.

The UNI-C and UNI-N entities MAY support the following RSVP messages:

Hello

The UNI-C and UNI-N entities MUST support the following RSVP objects:

Error Spec [2, 14, 13], Flow Spec [13], Filter Spec [2], Generalized Label [14], Generalized Label Request [14], Component_interface_ID [17], Message_Id [16], Message_Id_Ack [16], Message_Id_Nack [16], Message_ID_List [16], Session [2], RSVP HOP [13], Sender Template [2], Session Attribute [2], Sender_Tsepc [13], Time Value [13], Upstream Label [14], Address_Resolution_Request, Address_Resolution_Response, and Connection_ID.

The UNI-C and UNI-N entities MAY support the following RSVP objects:

Integrity [13], Label Set [14], Policy Data [13], Suggested Label [14], Notify Request [14], Notify [14], Hello [2] and Diversity.

12.5.1 RSVP Messages for UNI Signaling

12.5.1.1 ACK Message

The ACK message is for reliable messaging across the UNI. It has the following format:

```
<ACK message> ::= <Common Header> [ <INTEGRITY> ]
                  <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>
                  [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
[ <Connection_ID> ]
```

12.5.1.2 Hello Message

Hello message is for node failure detection. Hello message is optional. It has the following format:

```
<Hello message> ::= <Common Header> [ <INTEGRITY> ]
                    [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
                    <HELLO>
```

12.5.1.3 Notify Message

The Notify message provides a mechanism to inform “targeted” nodes of connection related events. Notify messages are only generated after a Notify Request object has been received. In general, the Notify message differs from the currently defined error messages (i.e., PathErr and ResvErr messages of RSVP) in that it MAY be “targeted” to a node other than the immediate upstream or downstream neighbor and that it is a generalized notification mechanism. For UNI 1.0, the Notify Messages are only generated from the UNI-N to the UNI-C.

The Notify message MAY be generated at any time to allow expedited notification of change in the status of a connection. Consequently, the UNI-C MUST be prepared to receive a Notify message. The IP destination address is set to the IP address of the UNI-C. The Notify message is sent without the router alert option. The format of the Notify message is as follows ([14]):

```
<Notify message> ::= <Common Header> [<INTEGRITY>] <MESSAGE_ID>
          [ <Connection_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 UNI-N that detected the error or the UNI link that has failed.

12.5.1.4 Query Message

The Query message MAY be generated at any time by an optical node to request and receive requested information from its adjacent node. The Query message has the following format:

```
<Query message> ::= <Common Header> [ <INTEGRITY> ]
                          [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
                          [ <QUERY> ... ]
```

12.5.1.5 Path Message

The Path message is used for connection creation. The format of the UNI Path message is as follows:

```
        <Path Message> ::=
<Common Header>
[ <INTEGRITY> ]
[ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
<MESSAGE_ID>
        <SESSION> <RSVP_HOP>
        <TIME_VALUES>
        [ <EXPLICIT ROUTE> ]
                [ <DIVERSITY> ]
        <GENERALIZED LABEL_REQUEST>
        [ <LABEL SET> ]
        [ <SESSION_ATTRIBUTE> ]
                          [ <NOTIFY_REQUEST> ]
        [ <POLICY_DATA> ... ]
        [ <Connection_ID> ]
        <sender descriptor>
```

The format of the sender descriptor for unidirectional connection is:

```
<sender descriptor> ::=
<SENDER_TEMPLATE> <SENDER TSPEC>
        [ <RECORD_ROUTE> ]
                          [ <DOWNSTREAM_COMPONENT_INTERFACE_ID> ]
                          [ <SUGGESTED LABEL> ]
```

The format of the sender descriptor for bi-directional connection (default in UNI 1.0) is:

<sender descriptor> ::=
<SENDER_TEMPLATE> <SENDER TSPEC>
[<RECORD_ROUTE>]
[<DOWNSTREAM_COMPONENT_INTERFACE_ID>]
[<SUGGESTED LABEL>]
[<UPSTREAM_COMPONENT_INTERFACE_ID>]
<UPSTREAM LABEL>

12.5.1.6 PathErr Message

The PathErr message is used to report errors and connection deletion events by the network.
The format of the UNI PathErr message is shown as follows:

<PathErr message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION>
<ERROR_SPEC>
[<Connection_ID>]
[<sender description>]

12.5.1.7 PathTear Message

The PathTear message is used when a connection is deleted by the source UNI-C. The format of the UNI PathTear message is as follows:

<PathTear Message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION> <RSVP HOP>
[<Connection_ID>]
[<sender descriptor>]

<sender descriptor> ::= (see earlier definition)

12.5.1.8 Resv Message

The Resv message is used for connection creation. The format of the UNI Resv message is as shown below:

<Resv Message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION> <RSVP_HOP>
<TIME_VALUES>
[<RESV_CONFIRM>]
[<NOTIFY_REQUEST>]
[<POLICY_DATA> ...]
[<Connection_ID>]
<STYLE> <flow descriptor list>

<flow descriptor list> ::=
<FF flow descriptor list> | <SE flow descriptor>
<FF flow descriptor list> ::=
<FLOWSPEC> <FF flow descriptor>
| <FF flow descriptor list> <FF flow descriptor>

<FF flow descriptor> ::=
[<FLOWSPEC>] <FILTER_SPEC>
<GENERALIZED LABEL>
[<RECORD_ROUTE>]
<SE flow descriptor> ::= <FLOWSPEC> <SE filter spec list>
<SE filter spec list> ::=
<SE filter spec> | <SE filter spec list> <SE filter spec>
<SE filter spec> ::=
<FILTER SPEC> <GENERALIZED LABEL>
[<RECORD ROUTE>]

12.5.1.9 ResvConf Message

The ResvConf message is sent downstream from the source to acknowledge the receipt of a Resv message. Specifically, under UNI 1.0, the ResvConf messages are sent from the source UNI-C to the corresponding UNI-N, and from the UNI-N to the destination UNI-C. The format of the UNI ResvConf message is shown below:

<ResvConf message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION> <ERROR_SPEC>
<RESV_CONFIRM>
[<Connection_ID>]
<STYLE> <flow descriptor list>

<flow descriptor list> ::= (see earlier definition)

12.5.1.10 ResvErr Message

The ResvErr message is used for reporting errors. The format of the UNI ResvErr message is as follows:

<ResvErr message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION> <RSVP_HOP>
<ERROR_SPEC>
[<POLICY_DATA> ...]
[<Connection_ID>]
<STYLE> <flow error description>

12.5.1.11 ResvTear Message

The ResvTear message is used for connection deletion by the UNI-C that was the destination during the creation of the connection. The format of the UNI ResvTear message is shown below:

<ResvTear Message> ::=
<Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<SESSION> <RSVP HOP>
[<SCOPE>] <STYLE>
[<Connection_ID>]
<flow description list>

<flow description list> ::= (see earlier list)

12.5.1.12 Srefresh Message

The Srefresh message is used for connection status enquiry. The format of the UNI Srefresh Message is shown below:

<Srefresh message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
[<MESSAGE_ID>]
<srefresh list>
<srefresh list> ::= <MESSAGE_ID_LIST>
[<srefrsh list>]
[<Connection_ID>]

12.5.2 UNI RSVP Objects Format

This section describes the RSVP objects used in UNI signaling.

12.5.2.1 Query Object

Two types of Query objects are defined, the Address Resolution Request object and Address Resolution Response object.

12.5.2.1.1 Address Resolution Request Object

The Address Resolution Request object is used in the Query message from UNI-C to UNI-N. It has the following format:

Class = Query Class, C-Type = 1

Resv are reserved bits, should be set to zero on transmission and ignored on reception.
AFI indicates the address family identifier as described in RFC1700. Currently supported values are:
AFI = 0 for 56 bits with unspecified structure.
AFI = 1 for IPv4 address
AFI = 2 for IPv6 address
AFI = 3 for NSAP (ICD and DCC) AESA
AFI = 8 for ITU-T E.164 ATM End System Address (AESA)
Length contains the length of the client address in bytes.
4 bytes for IPv4 address
16 bytes for IPv6 address
20 bytes for ITU-T E.164 ATM End System Address (AESA)
20 bytes for NSAP (ICD, DCC, and general)
7 bytes for unspecified, aligned at 32-bit boundary with zero padding at the end.

Client Address is the client address with zero padding at the end to ensure proper alignment at 32-bit boundary.

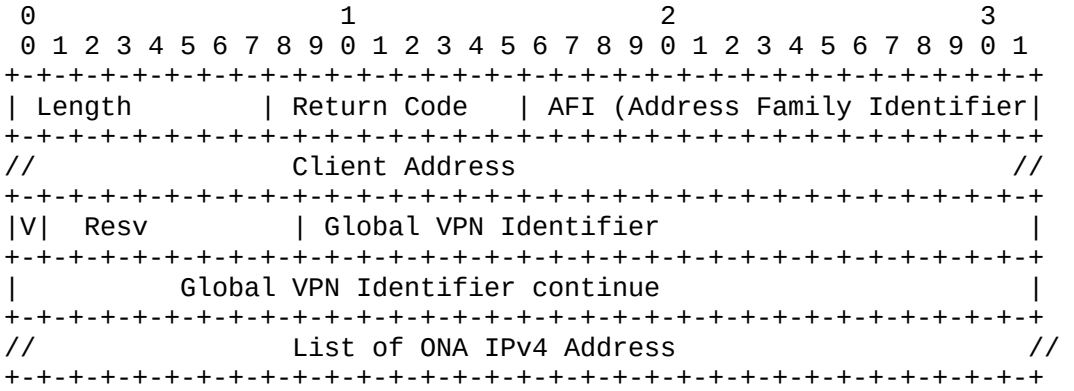
V = 1 indicates that the Global VPN Identifier shall be used to limit the resolution to the identified VPN (user) group. If V = 0, the VPN identifier SHALL NOT be used.

Global VPN Identifier is defined in [RFC2685] and is used to limit the address space from which the client address should be resolved. This field SHALL be set to zero on transmission and ignored on reception if the V bit is set to 0.

12.5.2.1.2 Address Resolution Response Object

The Address Resolution Response Object is used in the Query message from the UNI-N to the UNI-C. It has the following format:

Class = Query Class, C-Type = 2



Return code is used to indicate the result of the resolution:
0 indicates successful resolution and the ONA address portion contains a list of valid ONA addresses.
1 indicates “unrecognized address type”
2 indicates “can not resolve the client address”
Other values are reserved.

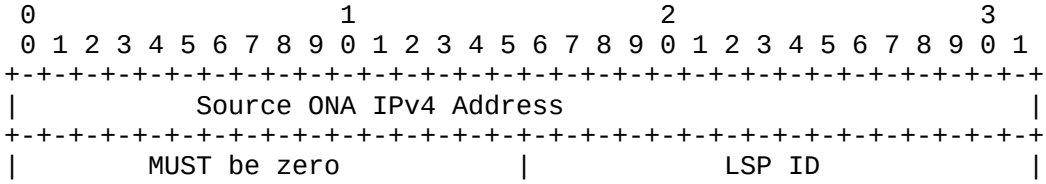
Length, AFI, Client Address, V bit, and Global VPN Identifier are exact copy from the client's Address Resolution Request Object.

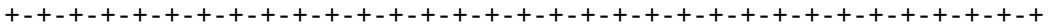
List of ONA IPv4 Address contains a list of the optical network administered (ONA) IPv4 address (32 bits each) corresponding to the Client Address. This is used in the IPv4 address field of a Session object.

12.5.2.2 Sender Template Object

The connection Sender Template Object [2] has the following format:

Class = SENDER_TEMPLATE, LSP_TUNNEL_IPv4 C-Type = 7



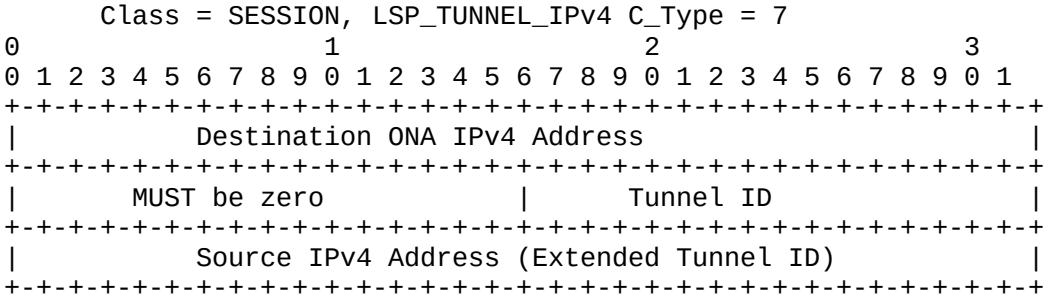


Source IPv4 address is the source client's ONA IP address.

LSP ID is a 16-bit identifier set by RSVP to identify the sender or client (locally significant within the client node) for possibly resource sharing between some new request and old request (from the client itself) within the same session of a connection. It should be set to zero for UNI 1.0.

12.5.2.3 Session Object

LSP_TUNNEL_OIF Session Object [2] has the following format:



Destination ONA IPv4 SHALL be set to one of the ONA address that is resolved by the network through the Query message (or the same address obtained using other mechanism) at the source UNI. At the destination UNI, it SHALL be set to the client's ONA IPv4 address.

Tunnel ID is a 16-bit identifier used in the SESSION that remains constant over the life of the tunnel. It is locally significant within the client, and set up by the RSVP to uniquely identify the connection within the client.

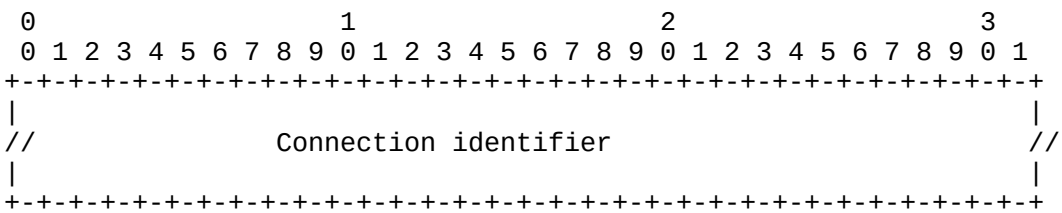
Source IPv4 address is the source client's ONA IP address.

12.5.2.4 Connection_ID Object

The Connection_ID object is used to uniquely identify a connection within the optical network. Upon receiving the first Path message from the initiating UNI-C, the corresponding UNI-N may generate the Connection_ID. This ID is conveyed to the terminating UNI-C by the corresponding UNI-N in the Path message. The same Connection_ID MUST be carried in the Resv message sent by the terminating UNI-C, and delivered to the initiating UNI-C in the Resv message it receives. All subsequent RSVP messages sent in relation to the connection MUST carry the Connection_ID.

Connection_ID object has the following format:

Class = Connection_ID Class, C_Type = 1

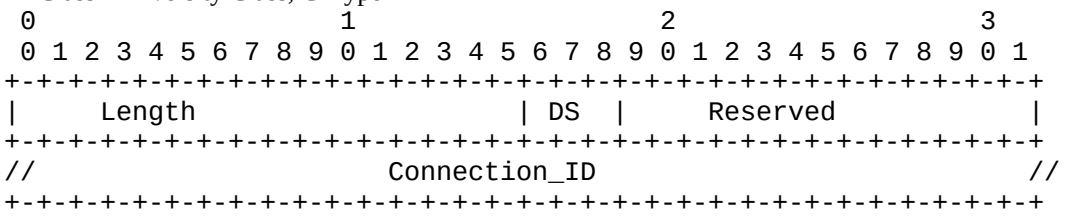


Connection identifier: Connection identifier has variable length in multiple of 32 bits and is at least 64 bits wide.

12.5.2.5 Diversity Object

Diversity object is an OIF-defined object to specify the diversity demand for a connection request. It carries one or more Connection_ID's for the existing connections from which the new connection should be disjoint. The object may be carried by the Path message. It may include multiple subobjects, each of which has the following format:

Class = Diversity Class, C-Type = 1

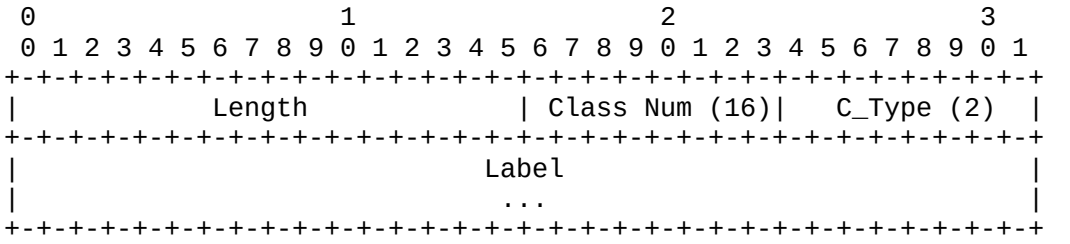


Length indicates the length of Connection ID (in bytes).
DS indicates the diversity requirement as following:
1 - Node diverse
2 - Link Diverse
3 - Shared Risk Link Group diverse
4 - Shared Path
other values are reserved.
The connection(s) identified by the specified Connection_ID(s) MUST terminate or originate at the same UNI. If the condition is not true, the network node SHALL return a PathErr message reporting error code "Connection ID out of scope" in the ErrorSpec object.

12.5.2.6 Generalized Label Object

The Generalized Label [8, 14] 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.

The format of a Generalized Label is [14]:

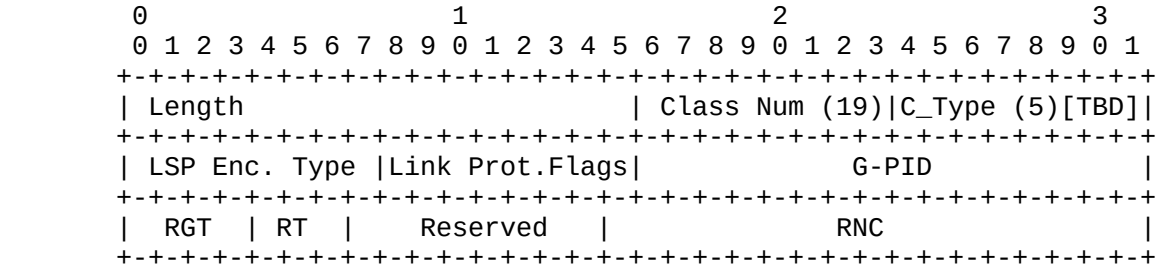


The detailed label format for SONET/SDH and port/wavelength is as defined in [8].

12.5.2.7 Generalized Label Request Object

Generalized Label Request object [14] is used to indicate a connection's bandwidth, protection type, and framing type.

The format of a Generalized Label Request with SONET/SDH Label range is:



LSP Encoding Type indicates the encoding of the LSP being requested. It SHOULD be set as 5 for SDH, 6 for SONET.

Link Protection Flags indicate the desired protection level(s) for each link along the LSP. It SHOULD be set to 0x01 “Unprotected” for UNI 1.0. Support for other Link Protection types are at discretion of network provider and vendor implementations.

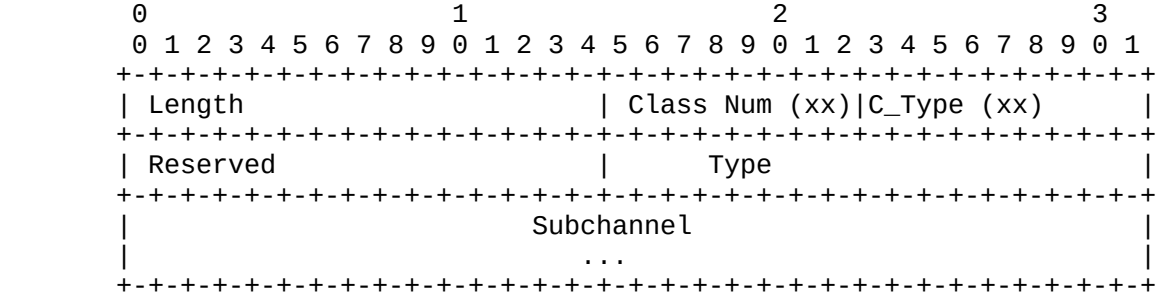
Generalized PID (G-PID) is an identifier of the payload carried by an LSP, i.e. an identifier of the client layer of that LSP. This MUST be interpreted according to the technology encoding type of the LSP and is used by the nodes at the endpoints of the LSP. Requested Grouping Type (RGT) indicates the SDH/SONET type of grouping requested for the LSP, it is used to constraint the type of concatenation. It SHOULD be set to 0 as default to indicate “no concatenation or bundling” since it is not required for UNI 1.0.

Requested Transparency (RT) indicates the type of SDH/SONET transparency (“emulation”) requested for that LSP. Corresponds to the Overhead Termination Type attribute defined in Section 10.

Requested Number of Components (RNC) 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 SHOULD be set to zero to indicate non-concatenation or bundling is requested.

12.5.2.8 Label Set Object

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. LABEL SET object is used to specify the port and channel to be used by a connection at the source. *The support of this object is optional*. The format of a Label_Set is shown below and described in [14]:



12.5.2.9 Suggested Label Object

The Suggested Label [8, 14] is used to provide a downstream node with the upstream node's label preference. This permits the upstream node to start configuring its hardware with the proposed label before the label is communicated by the downstream node. SUGGESTED LABEL object is used by a client

(network) node to suggest a label to a network (client) node. The specified label is a suggestion, and downstream node MAY select a different label in its Resv message.

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.

12.5.2.10 Upstream Label Object

For bi-directional LSPs, two labels MUST be allocated. Bi-directional LSP setup is indicated by the presence of an Upstream Label in the REQUEST/Path message [8, 14]. The directionality of certain framing types is implied, e.g. SONET/SDH. Therefore, when a Framing Type implies a bi-directional connection, this object SHALL be included in the Path Message.

An Upstream Label has the same format as the generalized label. It uses a new class number (TBD of form 0bbbbbbb) and the C-type of the label being suggested.

12.5.2.11 Error Spec Object

The Error_Spec object has the following format [13]:

IPv4 Error_Spec object: Class = 6, C-Type = 1

```
+-------------+-------------+-------------+-------------+
|            IPv4 Error Node Address (4 bytes)          |
+-------------+-------------+-------------+-------------+
|    Flags    |  Error Code |        Error Value        |
+-------------+-------------+-------------+-------------+
```

Error Node Address is the IP address of the node in which the error was detected. This field MAY be set to zero if the address of the failed node is not known or its disclosure is not desirable.

Flags is used only for an Error_Spec object in a ResvErr message. If set, this flag indicates that there still is a reservation in place at the failure point, or indicates that the FLOWSPEC that failed was strictly greater than the FLOWSPEC requested by this receiver.
Error Code is for error description.
Error Value contains additional information about the error.

12.5.2.12 Filter Specification Object

Filter_Spec [2] is used to identify the LSP, per RSVP-TE, required to identify the LSP in Resv.

LSP_TUNNEL_IPv4 Filter Specification Object:

Class = FILTER SPECIFICATION, LSP_TUNNEL_IPv4 C-Type = 7,

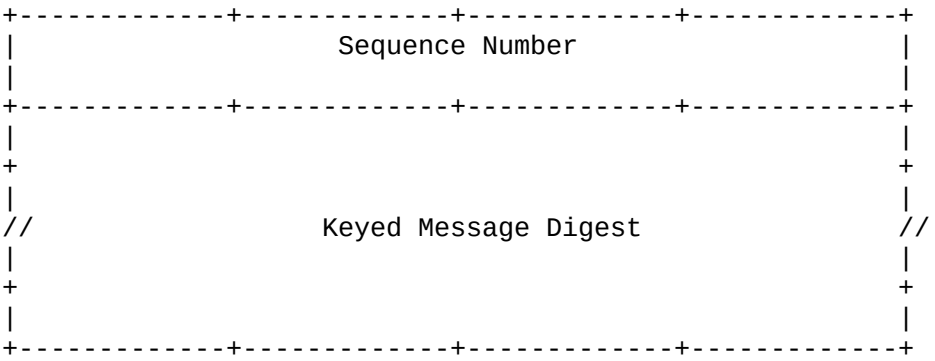
and the format of the object is identical to the LSP_Tunnel_IPv4 Sender_Template object.

12.5.2.13 Integrity Object

The Integrity Object [18] of an RSVP control message contains cryptographic data to authenticate the originating node and to verify the contents of an RSVP message.

INTEGRITY Object: Class = 4, C-Type = 1

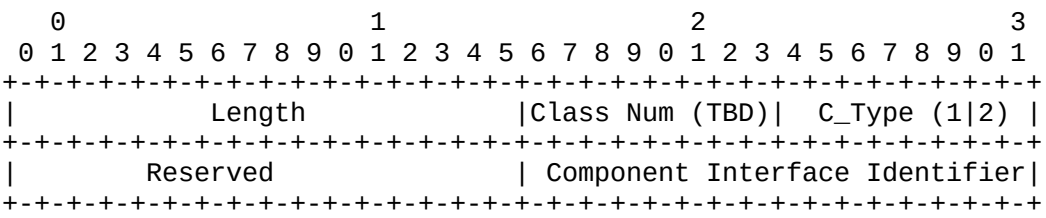
```
+-------------+-------------+-------------+-------------+
|    Flags    | 0 (Reserved)|                           |
+-------------+-------------+                           +
|                   Key Identifier                      |
```



Refer to [18] for the detailed usage and code points.

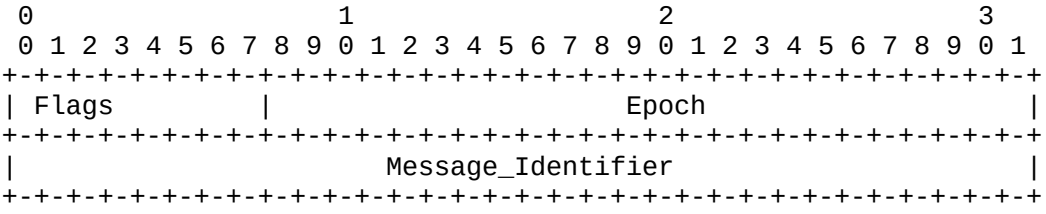
12.5.2.14 Component Interface ID Object

The Component_Interface_ID object's length field is set to 8. The Class Num is TBD of form 0bbbbbbb. The Downstream_Component_Interface_ID object, which has a C_Type of 1, is used to indicate the component interface to be used for traffic flowing in the downstream direction. The Upstream_Component_Interface_ID object, which has a C_Type of 2, is used to indicate the component interface to be used for traffic flowing in the upstream direction. Both objects have the same format and carry a 16-bit Component Interface Identifier. The format of the objects is:



12.5.2.15 Message ID Object

Message_ID object [16] has the following format:



Flags indicate that the sender requests the receiver to send an acknowledgment for the message.

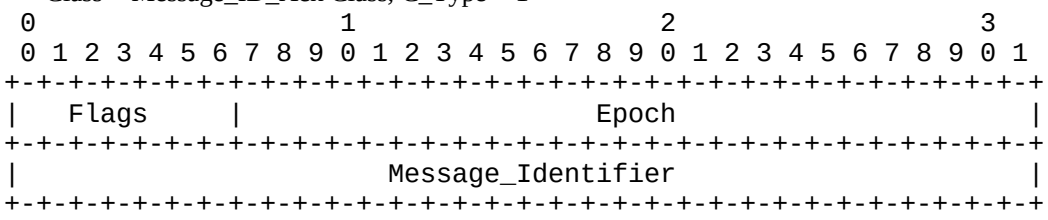
Epoch indicates when the Message_Identifier sequence has reset. SHOULD be randomly generated each time a node reboots or the RSVP agent is restarted. The value SHOULD NOT be the same as was used when the node was last operational. This value MUST NOT be changed during normal operation.

Message_Identifier: When combined with the message generator's IP address, the Message_Identifier field uniquely identifies a message. The values placed in this field change incrementally and only decrease when the Epoch changes or when the value wraps.

12.5.2.16 Message ID ACK Object

Message_ID_Ack object [16] has the following format:

Class = Message_ID_Ack Class, C_Type = 1



Flags: No flags are currently defined. This field MUST be zero on transmission and ignored on receipt.

Epoch: The Epoch field copied from the message being acknowledged.

Message_Identifier: The Message_Identifier field copied from the message being acknowledged.

12.5.2.17 Message ID Nack Object

Message_ID_Nack object [16]:

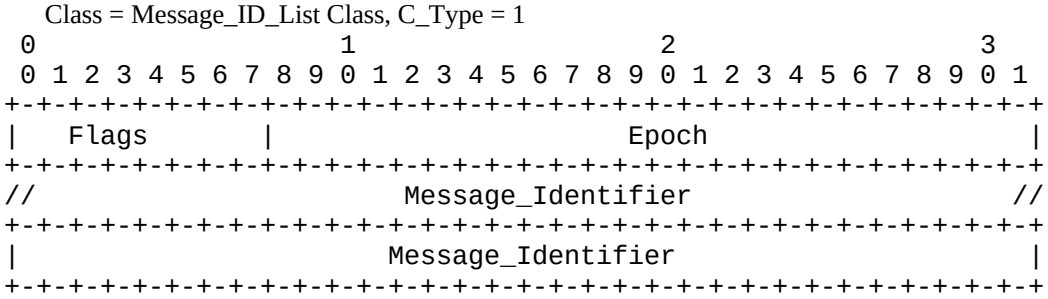
Class = Message_ID_Ack Class, C_Type = 2

It has the same format as the Message_ID_Ack object.

12.5.2.18 Message ID List Object

Message_ID_List object [16] has the following format:

Class = Message_ID_List Class, C_Type = 1



Flags: No flags are currently defined. This field MUST be zero on transmission and ignored on receipt.

Epoch: The Epoch field from the Message_ID object corresponding to the trigger message that advertised the state being refreshed.

Message_Identifier: The Message_Identifier field from the Message_ID object corresponding to the trigger message that advertised the state being refreshed. One or more Message_Identifiers MAY be included.

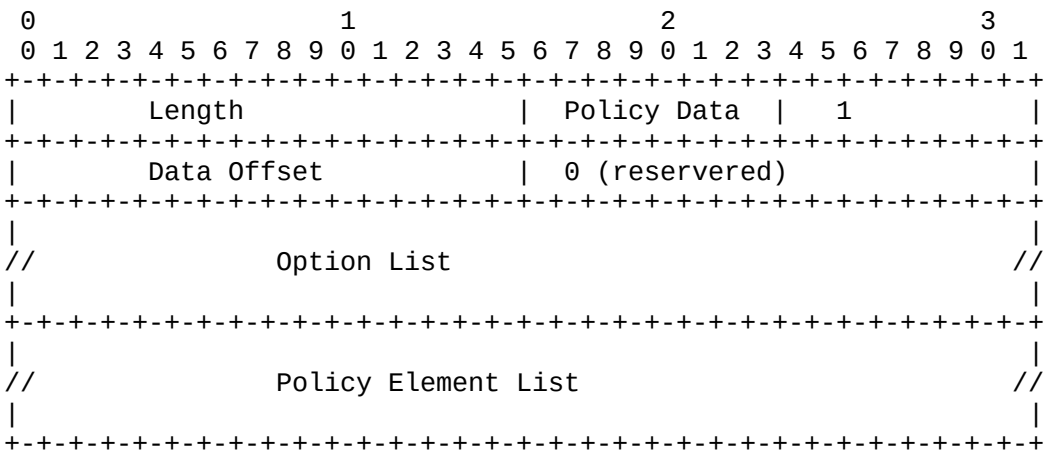
12.5.2.19 Policy Data Class

Policy_Data objects [15] contain policy information and are carried by RSVP messages. Policy_Data class MAY be used to convey Contract ID and client user group ID.

The format of Policy Data Class is specified in [15] and repeated in the following. This specification does not mandate the support of specific Policy Data types. Use of a specific policy type is at the discretion of a network provider and its equipment vendors.

POLICY_DATA class=14

Type 1 POLICY_DATA object: Class=14, C-Type=1



Data Offset: 16 bits

The offset in bytes of the data portion (from the first byte of the object header).

Reserved: 16 bits

Always 0. Ignored on reception.

Option List: Variable length

The list of RSVP objects as defined in Section 3.2 of RFC2750.

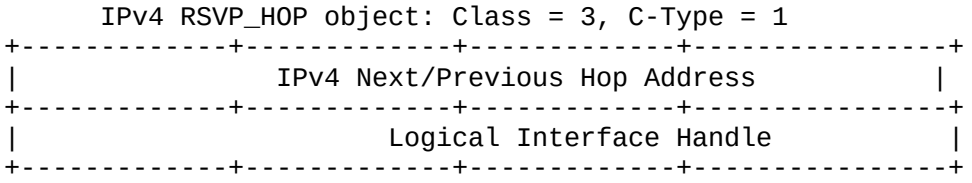
Policy Element List: Variable length

A list of Policy Elements as defined in Section 3.3 of RFC2752. In relation to O-UNI, the POLICY LOCATOR policy element may be used to carry the Contract Identifier and/or the Client Group Identifier.

12.5.2.20 RSVP Hop Object

RSVP_Hop Carries the IP address of the RSVP-capable node that sent the message in reference and a logical outgoing interface handle. The RSVP_Hop object is referred to as the PHOP ("previous hop") object in messages sent downstream or as the NHOP ("next hop") object in messages sent upstream [13]. The RSVP_Hop object of each Path message contains the previous hop address, i.e., the IP address of the interface through which the Path message was most recently sent.

When a node forwards a Path message out a logical outgoing interface, it includes in the message some encoding of the identity of logical outgoing interface, called the "logical interface handle", or LIH. For a UNI interface, the LIH MUST carry a control channel's identity over which the connection is to be established. The LIH value is carried in the RSVP_HOP object. The RSVP_HOP object has the following format [13]:



This object carries the IP address of the interface through which the last RSVP-knowledgeable hop forwarded this message. The LIH is used to distinguish logical outgoing interfaces. A node receiving an LIH in a Path message saves its value and returns it in the HOP objects of subsequent Resv messages sent to the node that originated the LIH. LIH SHALL be set to the ifindex value. It is used to identify a UNI when the signaling transport channel between the UNI-C and UNI-N is not direct (signaling configuration 3 in Section 7).

12.5.2.21 Time Values Object

The Time_Values object [13] in an RSVP control message specifies the time period used for refreshing the state implied by the message. It indicates the refresh rate of a containing Path or a Resv message.

```
    TIME_VALUES Object: Class = 5, C-Type = 1
+-------------+-------------+-------------+-------------+
|                   Refresh Period R                    |
+-------------+-------------+-------------+-------------+
```

Refresh Period is the refresh timeout period R used to generate the containing message, in milliseconds.

12.5.2.22 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.

The Notify Request Object [14] MAY be carried in Path or Resv messages. The NOTIFY_REQUEST class number is TBA (of form 11bbbbbb). The format of a Notify 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             | Class Num(TBD)|  C_Type (1)   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                     IPv4 Notify Node Address                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

IPv4 Notify Node Address: 32 bits, set to the IP address of the node that should be notified when generating an error message.

12.5.2.23 Hello Object

The Hello Class is 22. There are two C-Types defined.

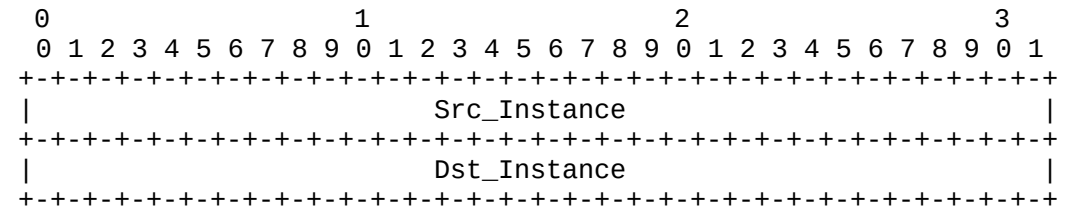
12.5.2.23.1 Hello Request Object

Class = Hello Class, C-Type = 1

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Src_Instance                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          Dst_Instance                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

12.5.2.23.2 HELLO ACK Object

Class = Hello Class, C-Type = 2



- Src_Instance: It represents the sender's instance. The advertiser maintains a per neighbor representation/value. This value MUST change when the sender is reset, when the node reboots, or when communication is lost to the neighboring node and otherwise remains the same. This field MUST NOT be set to zero (0).

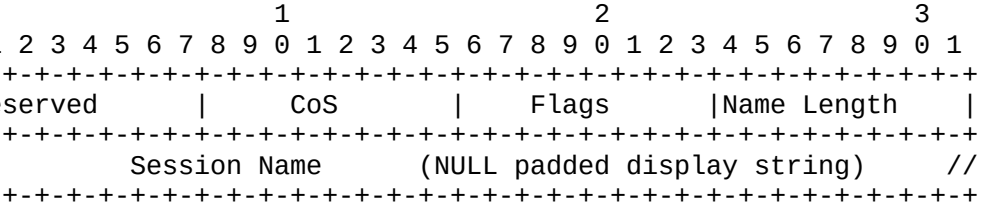
- Dst_Instance: The most recently received Src_Instance value received from the neighbor. This field MUST be set to zero (0) when no value has ever been seen from the neighbor.

12.5.2.24 Session Attribute Object

The Session Attribute class is 207 [2]. The Session Attribute with Class of Service is used to specify service type with predefined characteristics such as connection setup and hold priority. Its format is as following:

12.5.2.24.1 Format With Class Of Service

SESSION_ATTRIBUTE class = 207, LSP_TUNNEL C-Type = TBD



CoS specifies a class of service. Each CoS corresponding to a carrier predefined characteristics, such as type of restoration (e.g. no restoration, 1+1, protection), connection setup and hold priority, reversion strategies for the connection after failures have been repaired, and retention strategies.

Flags indicate additional attributes for the session. Currently no flag is defined.

Name Length is the length of the display string before padding, in bytes.

Session Name is null padded string of characters.

13 UNI Policy and Security Considerations

13.1 UNI Policy Control

The optical network must provide appropriate mechanisms to ensure accurate and authorized usage of network resources and client accountability. Collectively, these mechanisms are often referred to as *policy control*. Policy-based criteria are beyond the regular resource considerations that have to be taken into account in deciding whether a connection request can be accommodated within the optical network. Policy rules may define conditions on parameters such as source and destination addresses, priorities, bilateral agreements among service providers, time-of-day constraints, cost constraints, etc. It is also generally understood that policy control provides the necessary mechanisms to perform accounting.

Upon initial deployment, policy control might rely on simple rules, like for example, *"approve all requests on behalf of a given user group received from a given UNI-C agent, if the identity of the requestor can be verified"*. As more experience is gathered from initial deployments, it is expected that policy rules will become increasingly more sophisticated.

In order to support policy control, two main architectural elements are needed: the Policy Decision Point (PDP) is where policy decisions are made and the Policy Enforcement Point (PEP) is where policy decisions are actually enforced. The PEP resides within an optical network node, such as an OXC. The location of the PDP, however, depends on multiple factors, such as:

- The complexity of policy rules, including the computational load, type of software support and data access required. For example a policy might require complex cryptographic operations not supported within an OXC or access to a credit database which is physically located on a remote server.

- The frequency of events requiring policy decisions. For example, connection set-up requests might be received infrequently thus reducing the computational complexity on the PDP.

- The intelligence and flexibility of the optical networking device. A sophisticated and easily upgrade-able OXC is a better candidate to host the PDP.

The combination of the above factors determines whether the PDP should be co-located with the UNI-N agent or implemented on a remote policy server. If an external policy server is employed, a standardized protocol should be used for the communication between the PEP and PDP. This allows management of multiple PEPs from a single PDP and facilitates the integration of (standard) policy servers with multiple optical network elements. The COPS protocol [19] has been standardized by the IETF for PEP-PDP communication.

Specification of the UNI protocol does not depend on the placement of PEP and PDP modules within the optical network. Whether an external PDP is needed depends on the above factors, namely the frequency and complexity of policy decisions and the intelligence of the optical network element. If an external PDP is required, it is recommended that the COPS protocol be employed. The extension application of COPS to optical networks is described in appendix A.

As shown in Figure 1, the PDP may further access an external server or database, for example to retrieve policy rules, centrally stored accounting information, etc. These additional mechanisms are not being specified by the OIF.

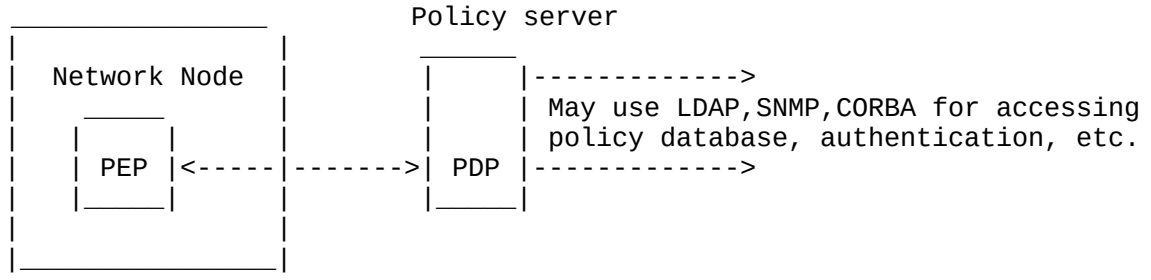


Figure 1. Placement of PDP and PEP for Policy Control

13.2 Sample Policies applicable to Connection Provisioning

This section discusses sample policies that could be employed for controlling connection provisioning across the UNI. It is included for informational purposes only and is not intended to suggest standardization of any policies. In each of the cases presented, the information required to make the policy decision is identified.

13.2.1 Time-Of-Day Based Provisioning

A user's contract with a carrier allows placement of connection requests during a specific time of the day and/or day of the week. As an example, the connection could be used for data backup over a storage area network during night hours. In this scenario the UNI-N agent needs to verify whether connection requests are received during the contracted hours. All information required to make the policy decision (time of request receipt) is implicitly contained in the message.

13.2.2 Identity And Credit Verification For Connection Requestor

A carrier's carrier operates a "*bandwidth exchange*" which allows carriers to dynamically "trade" optical connections. The bandwidth exchange receives requests from a very large number of carriers. Multiple UNI-C agents, representing different carriers, might be contacting the same UNI-N agent in the carrier's carrier network. It is imperative for the carrier's carrier to be able to verify the identity of the originator of the request. Additionally, the ability of the requestor to pay for the service might also need to be verified; this might require information such as an account or credit card number. Policy based admission control at the UNI-N involves positive verification of the identity and creditworthiness of the requestor. In this scenario, the information required to make the policy decision might extend beyond that contained in mandatory attributes. Reference [16] describes identity representation when RSVP is used as the signaling transport.

13.2.3 Usage-Based Accounting

Usage-based accounting can be supported using a *contract identifier*, which refers to the service contract of the connection owner. The contract identifier may be carrier specific; it can be used for accounting, billing and SLA verification. Due to the sensitivity of the information contained, the contract identifier might be encrypted to protect the privacy of the information.

13.3 Policy Control Mechanisms Associated with UNI Signaling Protocols

RSVP already defines a policy data class that can be used to map the UNI policy attribute. The use of the policy data object to carry the contract ID is defined in Section 12. This object may be used to carry other policy-related data in the future.

Under LDP, a new Policy TLV needs to be defined to carry policy-related data in signaling messages. This is described in Section 11.

13.4 UNI Security Considerations

Since optical connections carry high volumes of data and consume significant network resources, security mechanisms are required to safeguard an optical network against attacks on the control plane and/or unauthorized usage of network resources.

Security mechanisms can provide two main properties: *authentication* and *confidentiality*. Authentication mechanisms ensure *origin verification* and *message integrity* of UNI messages so that unauthorized UNI operations can be detected and discarded. For example, UNI message authentication can prevent a malicious UNI-C agent from mounting a denial of service attack against a service provider by placing excessive connection creation requests. Additionally, authentication mechanisms can provide *non-repudiation*, which is desirable for accounting and billing purposes. Authentication and confidentiality can be achieved using either *symmetric* or *public-key cryptographic algorithms*. Symmetric algorithms typically require pair-wise key distribution and do not provide non-repudiation, but are less computationally intensive and easier to deploy. Public-key cryptographic authentication algorithms use digital signatures and can provide non-repudiation, but they rely on deployment of a public-key infrastructure and are typically computationally intensive.

It is expected that from the point of view of UNI 1.0 requirements, the most important feature is message authentication. Confidentiality of UNI messages is also likely to be desirable, especially in the case where UNI message attributes include information private to the communicating parties (client and optical network operator). Examples of such attributes include account numbers, contract identification numbers, etc.

The case of non-co-located equipment presents increased security and policy control requirements. In this scenario, it is assumed that the UNI-C and UNI-N devices are connected via networking devices such as layer 2 switches and IP routers. Since these devices could belong to a public network and might be outside the control of the service provider, communication between the client and ONE is subject to increased security threats, such as IP address spoofing, eavesdropping and unauthorized access. To counter this, appropriate security mechanisms have to be employed to protect the signaling channel.

13.4.1 Security Mechanisms Relevant to UNI 1.0

13.4.1.1 RSVP Security Mechanisms

RSVP cryptographic authentication [18], defines an `INTEGRITY` object that provides hop-by-hop integrity and authentication of RSVP messages. The `INTEGRITY` object contains a message digest, computed using a secret authentication key, shared by the two parties, and a keyed-hash algorithm, such as keyed MD-5. The authentication key is never sent over the network, as in clear text password schemes, thus providing protection against attacks based on capturing information through access to the physical medium or packet data path. This scheme provides protection against forgery or message modification. For each RSVP message, the `INTEGRITY` object is also tagged with a one-time-use sequence number, which allows the message receiver to identify playbacks and hence avoid replay attacks. However, this mechanism *does not afford any confidentiality* since messages stay in the clear. In addition, since it is only based on symmetric cryptography it cannot provide non-repudiation. Standard key management procedures have to be

associated with this scheme. Manual key management can be used; in this case a privileged used manually types in the authentication keys. IKE or Kerberos could also be used for this purpose, as specified in [20].

Keyed MD5 is already being used for cryptographic authentication of IP routing protocols, such as OSPF, IS-IS and BGP. An advantage of using RSVP cryptographic authentication is that the widespread prior use of similar authentication schemes will facilitate deployment of this mechanism.

13.4.1.2 LDP Security Mechanisms

LDP specification [1] defines a mechanism to protect the integrity of LDP messages. This mechanism is based on the use of the TCP MD5 signature option [21], which was defined primarily for BGP authentication, and protects against the introduction of spoofed TCP segments into LDP connection streams. From a cryptographic standpoint the TCP MD5 signature option provides very similar properties as the RSVP authentication mechanism.

13.4.1.3 IP Security Protocols (IPSEC)

IPSEC defines a suite of protocols for providing various security services at the IP layer, for both IPv4 and IPv6. These include two traffic security protocols, the *Authentication Header* (*AH*) [22] and the *Encapsulating Security Payload (ESP)* [23] and one key management protocol, the *Internet Key Exchange (IKE)* [24]. The services offered by IPSEC include access control, connectionless integrity, data origin authentication, protection against replays, confidentiality (encryption). An important characteristic is that these services are provided at the IP layer, *offering protection for IP and/or upper layer protocols*. As such, IPSEC can be applied for either of the above realizations of the UNI.

AH is used to provide connectionless integrity, data origin authentication, and optionally anti-replay service, while ESP provides confidentiality in addition to all the properties offered by AH. The mechanisms are designed to be algorithm independent and can thus accommodate a range of cryptographic algorithms. IKE is an elaborate key management protocol, which includes multiple modes of authentication.

It is recognized in this specification that using the IPSEC suite of protocols to protect UNI traffic has a number of advantages. In particular, IPSEC allows a single solution for both RSVP and LDP implementations, provides confidentiality, which is offered by neither of the signaling transport mechanisms and, additionally includes a key management protocol.

However, it is also recognized that IPSEC is a relatively complex and heavyweight suite of protocols and, since AH and ESP are implemented at the IP layer, it typically requires kernel modifications making implementation harder. Furthermore, it is not clear whether all potential UNI clients will support IPSEC in the immediate future

13.4.2 UNI 1.0 Security Roadmap

UNI 1.0 will use the cryptographic authentication options of the underlying signaling transport mechanisms (RSVP-TE, LDP). This proposal is made in order to accelerate deployment and interoperability for UNI 1.0, given the widespread deployment and experience with similar cryptographic schemes. These mechanisms provide origin authentication and message integrity and hence offer protection against denial of service attacks.

Both RSVP and LDP security mechanisms should use the HMAC-MD5 [25] *algorithm instead of the original MD5 algorithm*. This recommendation is made since HMAC strengthens the cryptographic properties of MD5. HMAC can be used in combination with any iterated cryptographic hash function. Existing hash functions can be used without any modifications and without incurring significant performance degradation.

The standardization of more advanced security options, such as IPSEC, may be considered in future versions of the optical UNI.

14 References

1. L. Andersson, et. al., "LDP Specifications," Internet Draft (Work in Progress), draft-ietf-mpls-ldp-08.txt, work in progress, June 2000.
2. D. Awduche, L. Berger, D-H. Gan, T. Li, G. Swallow and V. Srinivasan, "Extensions to RSVP for LSP Tunnels," Internet Draft (Work in Progress), draft-ietf-mpls-rsvp-lsp-tunnel-07.txt, August 2000.
3. Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", IETF RFC 2119, March 1997.
4. B. Fox and B. Gleeson, "VPN Identifiers," IETF RFC 2685.
5. W. Simpson, Ed., "PPP in HDLC-Like Framing", IETF RFC 1662, July, 1994.
6. A. Malis and W. Simpson, "PPP over SONET/SDH," IETF RFC 2615, July 1999.
7. J. P. Lang, et al., "Link Management Protocol," Internet Draft (Work in Progress), draft-ietf-mpls-lmp-01.txt, November, 2000.
8. P. Ashwood-Smith, et. al, "Generalized MPLS - Signaling Functional Description," Internet Draft (Work in Progress), draft-ietf-mpls-generalized-mpls-signaling-00.txt, November, 2000.
9. B. Mack-Crane, et al., "Enhancements to GMPLS Signaling for Optical Technologies," Internet Draft (Work in Progress), draft-mack-crane-gmpls-signaling-enchancements-00.txt, November, 2000.
10. P. Ashwood-Smith, et. al, "Generalized MPLS - CR-LDP Signaling Functional Description," Internet Draft (Work in Progress), draft-ietf-mpls-generalized-cr-ldp-00.txt, November, 2000.
11. Jamoussi, B. Ed, "Constraint-Based LSP Setup Using LSP," Internet Draft (Work in Progress), draft-ietf-mpls-cr-ldp-04.txt, July 2000.
12. ITU-T, "B-ISDN Application Protocols for Access Signaling"- Q.2931 Specifications, Feb., 1995.
13. R. Braden, Ed., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification," IETF RFC 2205, September, 1997.
14. P. Ashwood-Smith, et. al, "Generalized MPLS - RSVP-TE Signaling Functional Description," Internet Draft (Work in Progress), draft-ietf-mpls-generalized-rsvp-te-00.txt, November, 2000.
15. S. Yadav, "Identity Representation for RSVP," IETF RFC 2752, January 2000.
16. L. Berger, et al., "RSVP Refresh Overhead Reduction Extensions," Internet Draft (Work in Progress), draft-ietf-rsvp-refresh-reduct-05.txt, June 2000.
17. Kompella, K. et al., "Link Bundling in MPLS Traffic Engineering," Internet Draft, (Work in Progress), draft-kompella-mpls-bundle-01.txt, October, 2000.
18. F. Baker et al. "RSVP Cryptographic Authentication," IETF RFC 2747, January 2000.
19. D. Durham, et al., "The COPS (Common Open Policy Service) Protocol," IETF RFC2748.
20. S. Herzog, et al., "RSVP Extensions for Policy Control", IETF RFC 2750.
21. A. Heffernan, "Protection of BGP Sessions via the TCP MD5 Signature Option," IETF RFC 2385.
22. S. Kent and R. Atkinson, "IP Authentication Header," RFC 2402, November 1998.
23. S. Kent and R. Atkinson, "IP Encapsulating Security Payload (ESP)," IETF RFC 2406, November 1998.
24. D. Harkins and D. Carrel, "The Internet Key Exchange (IKE)", IETF RFC 2409, November 1998.
25. H. Krawczyk, M. Bellare and R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, February 1997.

Appendix A: Relation to External Standards

This specification utilizes a number of IETF protocols at various stages of maturity. The following is a list of these protocols:

1. **RSVP**: RSVP is a mature protocol, described in IETF RFC 2205.
2. **RSVP-TE**: The traffic engineering extensions to RSVP are stable in terms of specification, and these are in the process of being approved as an IETF RFC. UNI signaling specified in Section 12 are based on RSVP-TE mechanisms.
3. **LDP**: LDP is a mature protocol in the process of being approved as an IETF RFC. UNI signaling specified in Section 11 is based on LDP.
4. **CR-LDP**: The constraint-based routing extensions to LDP are stable in terms of specification, and they are in the process of being approved as an RFC. UNI signaling specified in Section 11 includes some CR-LDP constructs.
5. **GMPLS**: The Generalized MPLS signaling specification is in the early stages of development. This is presently covered in a set of standards track Internet Drafts. UNI signaling specified in Sections 11 and 12 rely on GMPLS definitions for object formats and semantics.
6. **LMP**: The link management protocol is also in the early stages of development. This is presently covered in a single standards track Internet Draft. The out-of-band neighbor discovery procedure defined in Section 8 and service discovery procedures defined in Section 9 are based on LMP mechanisms.

Appendix B: Multi-Layer Neighbor Discovery

Three different layers of optical interfaces/switching is illustrated in Figure B-1. These are PLR-C, STE, and LTE for SONET/SDH framed signals. Neighbor discovery may be defined for each of these layers, as shown in the figure. For example a client operating at the SONET line termination layer may be connected to a "transparent" switch operating at the PLR-C layer. The discovery of this information rather than the manual provisioning of this information is desired to save on time and reduce errors. The neighbor discovery procedures described in Section 8 covered only LTE-LTE neighbor discovery. In this appendix, a method for discovering neighbors at all levels, LTE, STE and PLR, is outlined. This appendix is included for informational purposes only.

The signal types STE and LTE are inherently bi-directional, i.e., the signals will not correctly function with respect to performance monitoring, fault management, etc., if they are not correctly grouped into corresponding receive/transmit pairs. The PLR-C being essentially a transparent signal type may or may not be grouped into receive/transmit pairs, i.e., it has a unidirectional nature. In this appendix, only the case of bi-directional PLR-C is considered.

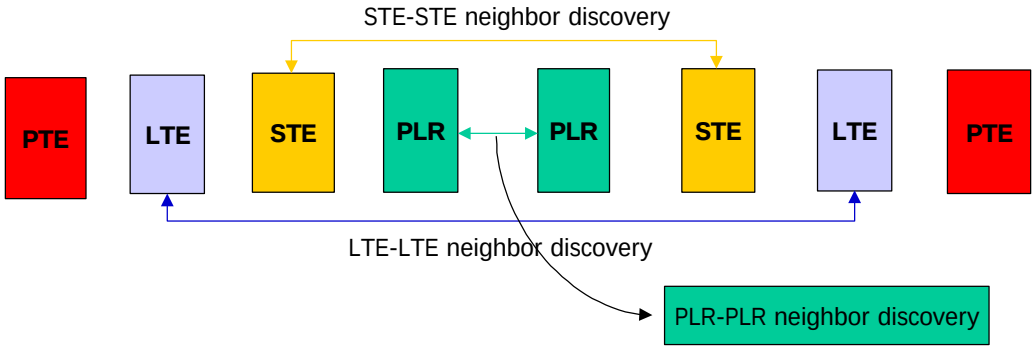


Figure B-1 *Neighbor Discovery at various layers in SONET/SDH.*

B.1 Multi-Layer Neighbor Discovery Protocol: Requirements and Assumptions

The purpose of this protocol is to allow a network element (NE) to find its nearest neighbors at each layer possible. A neighbor at a given transport level may be a number of NEs away if those intermediate NEs operate at lower levels than the NE of interest.

Since linear chains of interconnected NEs are common in the transport environment (different types of regenerator and add/drop multiplexers) any node in the chain should be able to find out its nearest neighbor without disrupting neighbor discovery or other operations further down the chain. The proposal described in this appendix is based on using the J0 bytes for neighbor discovery. In the PLR-C case that this results in a slight loss of transparency while the process is taking place. If changes in J0 bytes causes equipment alarms then the equipment operating around PLR-C circuits would need to be advised ahead of time about the (temporary) use of J0 for neighbor discovery.

No pre-configuration of out of band control channels should be required in the cases covered here. Any needed out-of-band control channel should be bootstrapped via information obtained via the in-band portion of discovery.

B.2 Information Model

Node-id:The node-id is unique within the context of the network and this is currently an IPv4 address. Note that this does not preclude other addresses from being used for signaling and other identification purposes.

Port-id: The port-id is unique within the context of the node-id.

Layer: This information is used to indicate the lowest layer neighbor to be discovered. Current acceptable values are LTE, STE, and PLR.

With some optical services and/or some optical technologies there is a need to be as "transparent" as possible. Hence the overhead or communication mechanism used for UNI neighbor discovery may only be available when services are not occupying the channel. In cases where overhead bytes may be used, it will be desirable to signal that their use should be terminated.

B.3 SONET/SDH Circuit Types and Options

B.3.1 Physical Layer Regenerator Circuits (PLR-C)

Considering neighbor discovery at the PLR level between SONET/SDH section level (capable equipment) and PLRC equipment, it is required that the PLRC circuit must snoop SONET/SDH overhead bytes with neighbor discovery information. This information may be in section or path trace byte strings.

B.3.2 Section Layer Terminating Circuits (STE-C)

Considering neighbor discovery at the STE level, there is a choice of utilizing section trace (J0) bytes or section DCC bytes for neighbor discovery. In the latter case, the neighbor discovery information may be in an HDLC-encapsulated packet format. Neighbor discovery information would be sent repetitively like a "beacon" signal.

B.3.3 Line (Multiplex Section) Layer Terminating Circuits (LTE-C)

Considering neighbor discovery at the LTE level, if the intermediate STE equipment does not change J0 bytes, the same can be used for neighbor discovery. Otherwise, a packet based mechanism over line DCC bytes could be used as described in Section 8.

In this appendix, a unified neighbor discovery procedure for all layers based on J0 bytes is described.

B.4 J0 String based Mechanism

B.4.1 Message Coding

SDH places the most stringent constraints on the contents of the J0 section trace string. Per G.707 (section 9.2.2.2) 16 bytes are used for the regenerator section trace. The first byte is a combination frame start marker and CRC-7 code. This leaves 15 bytes available. Due to the frame start procedure bit 0 of all bytes but the frame start/CRC-7 byte are require to be set to 0. Hence this results in a reduced 7 bit content for each byte. Per G.707 a T.50 character is to be used in these bytes. The advantage of abiding by these restrictions is that existing equipment can interact with a minimum of software only modifications. In the current T1X1.5 letter ballot on SONET rates and formats (T1X1.5/2000-193) SONET section traces are aligned with the ITU's used of the section trace with an option for a fixed 64-byte fixed length ASCII string left for further study.

Possible byte assignments (using a format similar to that of G.831 March 2000, appendix I):

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
T	I	X	X	X	X	X	X	X	X	P1	P2	P3	P4	P5
Type	Dis Id	Node Identifier								Port Identifier				
F1	F2	F3								F4				

Notes: all entries will be readable characters.

F1 Type/Layer Indicator

This field is used to indicate either the type of format for the (node identifier, port identifier) pair or an action to be taken. The currently suggested values are:

T = 0 “Do nothing”. Indicates to successive equipment downstream not to respond to this discovery request.

T = 1 PLR-C equipment discovery using an Ipv4 address and port number in the format detailed in F2 and F3.

T = 2 STE-C equipment discovery using an Ipv4 address and port number in the format detailed in F2 and F3.

T = 3 LTE-C equipment discovery using an Ipv4 address and port number in the format detailed in F2 and F3.

Note that other address and port types maybe considered in the future, however IPv4 addresses are the most important for the rest of the UNI functionality.

F2 Distinguishing Identifier

This character is to be determined, but its purpose is to discriminate this format of SONET/SDH J0 string from other optional formats, e. g., that specified in G.831 Appendix I.

F3 Node Identifier (IPv4 address in hex characters)

IPv4 addresses are four byte quantities. These are encoded in each byte with hex characters. This requires 8 hex characters thus saving 4 characters over decimal notation. For example the IP address of 192.168.2.23 is encoded “C0 A8 02 17”.

F4 Port Identifier (Port number in hex characters)

This permits port numbers from 0 to FFFFF (1,048,575).

B.4.2 Basic Discovery Procedure

Note that this includes LTE-C, STE-C and bi-directional PLR-C case.

1. A device initiates discovery of its neighbor at level L by setting the type T field in the J0 string to the appropriate value and placing its IPv4 address and port number in the J0 string.

2. A device receiving an appropriately formatted J0 string checks to see if it operates at the level L (or above) as indicated in the string.

3. If the level indicated in the string is higher than that at which the device operates the J0 string is passed on without modification, i.e., the device will remain “transparent”.

4. If the level indicated in the string is equal or lower than that level the device operates then the device will modify the J0 string sent downstream by replacing the type field with T=0 (the “do nothing” indication).

5. If after a specified time (e.g., 2 seconds) the J0 string request remains, i.e., it has not changed to a message with T=0 then on the return fiber the receiver will transmit a J0 signal with the type field set to the highest level at which the port will operate. It will also send its IPv4 address and port number information.

6. The originating device can now record the received connectivity information contained in the J0 string.

7. If no J0 string is received at the originating device at a level equal to or higher than the original level request than either there is no neighbor at that layer or higher or the neighbor does not support this procedure. Or a lower layer neighbor is not transparent to this procedure.

The neighbor discovery procedure is complete if a device reaches Step 6 and no more needs to be done. If the originating device wishes to discover its neighbor at a different layer (if one exists), it can start at Step 1 with a new type value. In the SONET/SDH LTE case no action needs to be taken on the J0 string since it is terminated at this layer.

B.4.3 Detecting Mis-Wired fibers (bi-directional case)

Since the 64 byte long J0 string is for further study in the SONET and not currently a part of the T1.105 series of specifications, only the SDH compliant format is considered for the present. In the SDH case the strings used are too short to contain the return fiber information and therefore this must be sent out of band. This procedure is similar to the out-of-band procedures described in Section 8.

APPENDIX C: COPS Usage for the UNI

In this appendix we describe the use of the Common Open Policy Service (COPS) protocol for outsourcing policy provisioning within the optical domain. A very brief outline of the COPS protocol framework is first given. This summary is only intended to serve as a background reference, and interested readers are referred to [19] for full details.

C.1 Functional Overview of the COPS Protocol

COPS is a query/response protocol based upon a client/server model and has been designed for policy and admission control for a generic set of network resources. The framework defines client entities, termed Policy Enforcement Points (PEPs), and server entities, termed Policy Decision Points (PDPs), which exchange policy information through various message types. At least one PDP entity has to be defined for an administrative domain. The messages include request, update, and delete messages from the PEP and decision and update messages from the PDP server. COPS designed to support multiple policy clients, and examples include quality of service (QOS), security, customer-specific, etc. Therefore, to distinguish between different client types, a client type must be specified in each message. Each client type can have its own specific data and policy decision rules. Note that a single PEP or PDP can support policy provisioning for multiple client types.

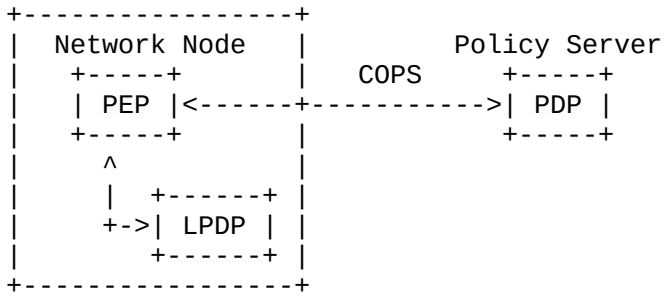


Figure C-1: A COPS Illustration

The overall interaction between the respective COPS protocol entities is shown in Figure C-1. Specifically, the COPS protocol message set details the information exchange between the client PEP and remote PDP server. Additionally, an optional Local Policy Decision Point (LPDP) entity can also be associated with a network device to execute “local” policy decisions. However, all local decision information must be relayed to the PDP also (i.e., by the PEP in the form of an LPDP decision object), and the PDP entity remains as the binding decision point for all PEP requests.

COPS can function in two basic models, outsourcing and provisioning and defines various objects for its functioning. The outsourcing model represents a client-driven approach, where the PDP server actively responds to incoming policy requests from PEP clients. This model works well for signaled protocols, e.g., as in RSVP-COPS [20]. Client requests are carried via COPS-PR defined data objects, which communicate Client Specific Information objects. The format of this data is independent of the actual messaging protocol, and hence this decouples the information model from the actual signaling semantics.

Meanwhile, the provisioning model represents a server-driven approach, where the PDP downloads (pre-provisions) policy information to client PEPs beforehand. Such policy information is based upon redicted configurations/demands and works well for non-signaled protocols/scenarios. Specifically, after the PEP-PDP connection is setup, the PEP sends a configuration request message to the PDP, which in turn replies with a message containing all provisionable policies for the PEP device. Alternatively, the PDP can also asynchronously “push” policy state onto the PEP. Meanwhile, the actual COPS policy data is represented via a named “self-identifying” data structure, termed the Policy Information Base (PIB). This structure specifies the type and purpose of the policy information arriving from the PDP, and hence must also be

client specific. This information is installed by the PEP. Conceptually, PIB can be described using a tree structure, consisting of Policy Rule Classes (PRCs) and their associated Policy Rule instances (PRIs), see [19].

C.2 COPS Applied to O-UNI

Since the PEP deals with optical network policy administration, it resides within the optical network and communicates with the UNI-N agent. This is shown in Figure C-2.

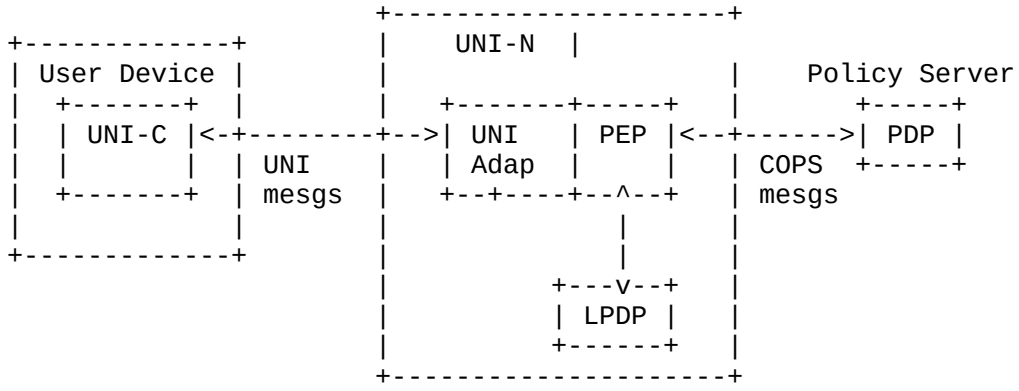


Figure C-2: COPS Applied to O-UNI

The UNI-N agent residing in the edge optical device passes messages between the (optical domain) PEP and the (electronic domain) user device, e.g., such as an IP router or SONET digital cross-connect. In particular, this entity translates UNI actions into appropriate PEP client actions. Subsequently, the optical PEP entity sends back appropriate reply messages to the UNI-C. The outsourcing policy management model applied for COPS-OUNI interworking is akin to RSVP-COPS interworking [15], and O-UNI actions are transmitted to the PDP via the local PEP entities. In this case, the PDP (LPDP) returns policy responses to the PEP, which are in turn translated into appropriate O-UNI actions and/or messages. Therefore, when PEPs open communications with PDP servers, their initial Client-Open (OPN) messages have the client-type set to indicate a signaled client. In particular, *a new signaled client type O-UNI is defined*. If the PDP supports this client-type, it responds with a Client-Accept (CAT) message or otherwise with a Client-Close (CC) message. Alternatively, the PDP can also redirect the PEP to other PDPs in the network domain (via a Redirect address in the CC message). After receiving a CAT message, the PEP can issue the first (O-UNI-driven) request message to the PDP server.

Meanwhile, the information transfer model assumes that all objects received in the UNI messages are encapsulated in a Client Specific Information Object (Signaled ClientSI) and sent from the PEP to the PDP. Specifically, UNI connection create request messages will contain topological information (source, destination IP addresses and port numbers), bandwidth requirements, protection levels, preemption priorities, and setup priority. It assumed that the PDP (O-UNI policy server) is fully OUNI-aware and can handle these message contents and generate appropriate policy control decisions.

C.3 Message Contents

The COPS protocol provides the capability for different COPS clients to define their own “named”, i.e. client-specific, information for various messages. This section describes the messages exchanged between a COPS server (PDP) and COPS O-UNI clients (PEP) that carry client-specific data objects. Since OUNI is a signaled client, a new signaled client-type value is defined (TBD) for the common header field.

C.3.1 Request (REQ): PEP → PDP

The REQ message is sent by COPS-OUNI clients to issue a connection establishment request to the PDP. The REQ message is used to request a new connection establishment, to query and to release a connection. The Client Handle associated with the REQ message originated by a client must be unique for that client. The Client Specific info object is used to specify the requested resources. Each REQ message contains a single request. The PDP responds to the resource allocation request with a DEC message containing the answer to the query. The REQ message has the following format:

<Request> ::= <Common Header>
<Client Handle>
<Context >
[<ClientSI: all objects in OUNI connection Create Request>]
[<LPDPDecision(s)>]
[<Integrity>]

The only OUNI message types supported by COPS are light create request, connection delete request, connection modify request and connection status inquiry. The other OUNI response messages and notification are not supported by COPS. The four types of actions are specified in the context object in the M-Type field.

All objects that are received in O-UNI request messages are encapsulated inside the Client Specific information (ClientSI) Object sent from PEP to the remote PDP.

<ClientSI:> all objects in connection Create Request>

If the request is a connection modify request, the previous value for starting time should be appended at the end of ClientSI object (for accounting purpose):

<ClientSI:>::
OUNI connection modify request message
<starting time> (new value)
<starting time> (old value)

The LPDPDecision can be specified if applicable.

C.3.2 Decision (DEC): PDP → PEP

The DEC message is sent from the PDP to a COPS-OUNI client in response to the REQ message received from the PEP. Unsolicited DEC messages cannot be sent for this client type (therefore the solicited decision flag is always set). The Client Handle must be the same Handle that was received in the REQ message.

PDP performs admission control for connection request messages based on the User Group Id, and some other criteria. The Decision object will contain in the Client Specific info. Each DEC message contains a single decision. The DEC message for the COPS-OUNI client type has the following format:

<Decision Message>::= <Common Header>
<Client Handle>
<Decision> | <Error>
[<Integrity>]

The decision object in turn has the following format:

<Decision> ::= <Context>
<Decision: Command code>
<Decision: Client SI data>

The context object will be the same as contained in the REQ message.
The Decision: command code object will contain the answer in the Command-code field according to the COPS specifications. In particular the Command-code will be “Install” to mean a positive answer and “Remove” to mean a negative answer. The following text clarifies how Install and Remove Decisions map into the different request types.

REQ (M-Type= Add)
DEC (Install) -> The requested resources are successfully allocated
DEC (Remove) -> The requested resources are not allocated

REQ (M-Type = Release)
DEC (Install) -> The resources are released
DEC (Remove) -> The resources are still allocated.

REQ (M-Type= Modify)
DEC (Install) -> The modification is accepted. The newly requested resources are allocated, while the previous ones have been released
DEC (Remove) -> The modification is not accepted. Previous allocation is still active.

REQ (M-Type=Query)
DEC (Install) -> The request for query is accepted.
DEC (Remove) -> The request for query is not accepted.

The Error object is used to communicate COPS protocol error to the PEP, according to the definition in [13]. No client specific error sub-codes are used by COPS-O-UNI.

The Decision ClientSI data object carries the information needed to correlate the decision with the answer and some optional information to explain negative Decisions. It has the following format:

<Decision: Client SI data> ::= <Reject Reason>

Possible reasons are:
Resource unavailable
Unsupported resource type
Unacceptable source address
Unacceptable destination address

No report is sent by the PEP to confirm the reception of a Decision message. Only in case of specific errors, the PEP will send back a Report State message to the PDP.

C.3.3 Report State (RPT): PEP → PDP

For COPS-O-UNI client type, the Report State message is sent by the PEP to the PDP in case of problems with a received Decision message. More specifically it is used to communicate that the Decision contains a handle which cannot be correlated to a previous request. This event is the manifestation of abnormal behavior. On reception of a Report State message the PDP could start a Synchronization procedure. The RPT message for the COPS-O-UNI client type has the following format:

<Report State Message>::= <Common Header>
<Client Handle>
<Report Type>
<ClientSI: >
[<Integrity>]

C.3.4 Synchronize State Request (SSQ): PDP → PEP

The Synchronize State Request message is sent by the PDP to the PEP to "reset" the state information. It requests the PEP to send the set of resource allocation REQ messages needed to rebuild the state. The SSQ can apply to the whole set of PEP active reservations PEP, or to a specific resource type and source-destination couple, depending on the information contained in the Client SI object.

< Synchronize State> ::= <Common Header>
<Client Handle>
<ClientSI: SSQ scope>]
[<Integrity>]

C.3.5 Synchronize State Complete (SSC): PEP -> PDP

The Synchronize State Complete message is sent by the PEP to the PDP to inform that all the REQ messages needed to rebuild the state have been sent. The Client SI object is the same received in the SSQ message and specifies the scope of the synchronization procedure which has been completed.

< Synchronize State Complete> ::= <Common Header>
<Client Handle>
<ClientSI: SSQ scope>]
[<Integrity>]

C.4 Example

In this section, an illustrative example is presented

A PEP requests an OC48 connection between OXC A (192.234.124.1) and OXC B (192.234.4.1).

PEP -- > PDP REQ: = <Handle C>
<Context: Connection request, Add>
<ClientSI:

** start of OUNI connection Create Request message **
User Group: X
Source Address: 192.234.124.1
Destination address: 192.234.4.1
Bandwidth type: OC48
** end of OUNI Create message ***
>

The PDP accepts the request.

PDP--- > PEP DEC: = <Handle C>
<Context: Connection Request, Add>
<Decision: Command = Install>
<Decision: Client SI
>

Appendix D: Companies Belonging to the OIF

This section is to be completed.