User Network Interface (UNI) 2.0 Signaling Specification Release 2

OIF-UNI-02.0-R2-RSVP - RSVP Extensions for User Network Interface (UNI) 2.0 Signaling Release 2

January 29, 2013

Implementation Agreement created and approved
by the Optical Internetworking Forum
www.oiforum.com

Working Group: **Networking and Operations**

TITLE: **User Network Interface (UNI) 2.0 Signaling Specification Release 2: RSVP**

SOURCE:

TECHNICAL EDITOR

Evelyne Roch
Huawei Technologies
303 Terry Fox Dr.
Ottawa, ON K2K 3J1
Canada
Phone: . +1.613.595.1900 Ext. 1612
Email: evelyne.roch@huawei.com

WORKING GROUP CHAIR

Remi Theillaud
Marben Products
176 rue Jean Jaures
92800 Puteaux
France
Phone: +33 (1) 79 62 10 22
Email: remi.theillaud@marben-products.com

ABSTRACT: This Implementation Agreement (IA) specifies the content and operation of the OIF UNI 2.0 signaling protocol in a protocol specific manner. It allows a client device to dynamically request the establishment of a service across an operator's network. UNI signaling functions, along with the OIF E-NNI 2.0 and I-NNI signaling protocols (the latter not specified by OIF), are used to establish end-to-end connection services. This document contains the RSVP-TE extensions for UNI 2.0 signaling. It obsoletes the OIF IA OIF-UNI2.0-RSVP by updating the Ethernet signaling related objects and sub-objects.

technologies. Optical internetworks are data networks composed of routers and data switches interconnected by optical networking elements.

With the goal of promoting worldwide compatibility of optical internetworking products, the OIF actively supports and extends the work of national and international standards bodies. Formal liaisons have been established with The ATM Forum, IEEE 802.3, IETF, ITU-T Study Group 13, ITU-T Study Group 15, MEF, NPF, ATIS-TMOC, ATIS-OPTXS, TMF, UXPi and the XFP MSA Group.

For additional information contact:
The Optical Internetworking Forum, 48377 Fremont Blvd.,
Suite 117, Fremont, CA 94538
510-492-4040 Φ info@oiforum.com

www.oiforum.com

List of Contributors

The UNI 2.0 was based on the UNI 1.0r2 and we acknowledge the work of the contributors to the UNI 1.0r2:

Osama Aboul-Magd	William Goodson	Zhi-Wei Lin	Arnold Sodder
Stefan Ansorge	Gert Grammel	Ling-Zhong Liu	John Strand
K. Arvind	Richard Graveman	Ben Mack-Crane	George Swallow
Krishna Bala	Eric Gray	Larry McAdams	Ewart Tempest
Sandra Ballarte	Riad Hartini	Wilson Nheu	Eve Varma
Ayan Banerjee	Eric Mannie	Lyndon Ong	Cary Wright
Rick Barry	Raj Jain	Dimitiri Papadimitriou	Yangguang Xu
Debashis Basak	LiangYu Jia	Dimitrios Pendarakis	Yong Xue
Greg Bernstein	Jim Jones	Kavi Prabhu	Tao Yang
Richard Bradford	Suresh Katukam	Bala Rajagopalan	Jennifer Yates
Curtis Brownmiller	Nooshin Komaee	Anil Rao	John Z. Yu
Yang Cao	Jonathan P. Lang	Robert Rennison	Alex Zinin
John Drake	Monica Lazer	Jonathan Sadler	Zhensheng Zhang
Hans-Martin Foisel	Fong Liaw	Stephen Shew	

We acknowledge the work of the contributors to the UNI 2.0 extensions:

Evelyne Roch (Co-editor)	Lyndon Ong
Alessandro D'Alessandro	Vijay Pandian
Hans-Martin Foisel	Rajender Razdan
Richard Graveman	Jonathan Sadler
Fred Gruman	Stephen Shew (Co-editor)
Jim Jones	Eve Varma
Monica Lazer	Lucy Yong
Thierry Marcot	

1 Table of Contents

2 List of Figures

3 List of Tables

4 Overview

RSVP (Resource reSerVation Protocol) is a protocol for establishing network resources for IP sessions (or "flows") [RFC2205]. The RSVP definition consists of basic procedures, messages and object formats for signaling in an IP network. RSVP with Traffic Engineering extensions (RSVP-TE) has been defined for establishing connections subject to routing constraints in an MPLS network [RFC3209]. The RSVP-TE definition includes additional procedures, messages and object formats as extensions to the base RSVP definition. Generalized MPLS (GMPLS) extensions for RSVP-TE signaling [RFC3471, RFC3473] extend RSVP-TE signaling procedures and objects to cover different types of switching applications such as circuit switching, wavelength switching, etc.

In this document, UNI signaling based on [G.7713.2] is defined. [G.7713.2] adapts GMPLS RSVP-TE [RFC3473] to ASON requirements by using objects defined in [RFC3476] and [RFC3474] to carry additional information. Some extensions have been made to G.7713.2 to support services not in its original scope. This specification defines how GMPLS RSVP-TE messages and objects are used to provide UNI signaling between UNI-C/UNI-N and UNI-N/UNI-C. The OIF UNI 2.0 is agnostic to the protocol used within the network, however it assumes both source and destination UNI-C and UNI-N support RSVP-TE signaling. This definition leverages the above specifications to the maximum extent possible and uses messages and objects defined in RFCs with the exception of an Ethernet-related sub-object that is defined in this document. This document also specifies the applicable values for certain objects and the execution of specific procedures where the above RSVP-related specifications allow variants. The changes introduced by this revision obsolete the OIF IA OIF-UNI2.0-RSVP by updating the Ethernet signaling related objects. The changes are in blue colored font and the changes are summarized in Section 11.

This specification is not fully compatible with GMPLS [RFC3473] in the sense that an interworking function is required between domains using the different specifications. The OIF has published a guideline document on this topic [OIF-G-Sig-IW-01.0]. Major areas of differences between GMPLS [RFC3473] and this specification include:

- Call/Connection concepts
- Single end-to-end RSVP session vs. multiple RSVP sessions
- Support for [RFC3474] and [RFC3476]
- ResvTear/ResvErr usage
- Support of SPC services
- Terminology
- Addressing

This specification supports all UNI 1.0 signaling [OIF-UNI-01.0-R2-RSVP] and additional objects added for UNI 2.0 features. Backwards compatibility is discussed in Section 6.

In this document, the "source UNI" refers to the A-end of the connection as defined in [G.7713], the "destination UNI" refers to the Z-end of the connection as defined in [G.7713], "upstream" is towards the "source UNI" and "downstream" is towards the "destination UNI".

4.1 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 [RFC2119].

5 Terminology and Abbreviations

The key terminology and abbreviations used in the rest of the document are summarized below.

5.1 Terminology

CE-VLAN ID	Customer Edge Virtual Local Area Network Identifier: The identifier derivable from the content of a service frame that allows the service frame to be associated with an EVC at the UNI.
Connection	A series of contiguous link connections and/or subnetwork connections between termination connection points (G.805).
In-Fiber Signaling	The transport of signaling traffic over a communication channel embedded in the data-bearing physical link.
Logical Port ID	A control plane identifier for a port. For SONET/SDH and OTN links, there is a one-to-one correspondence between a logical port ID and a port. For Ethernet, it is possible to have a single logical port ID representing multiple ports in the case where link aggregation is used as this is modeled by a single logical port by the control plane.
Node ID	Control plane identifier for the network element on the client or network sides of the UNI reference point.
Out-of-Fiber Signaling	The transport of signaling traffic over a communication link, separate from the data-bearing link, between the signaling entities.
Port	The hardware interface in an optical or user network element that terminates a bi-directional link between network elements. Examples include OC-48 or OC-192 ports in a TNE.
SC PC ID	Signaling Controller Protocol Controller Identifier. The SC PC provides the protocol specific processing of signaling messages, including mapping to and from abstract interfaces of the control plane components.
Signal Type	A SDH/SONET or OTN signal type, such as STS-1 or ODU1.
Signaling Communications Network (SCN)	A network that transports signaling messages between the signaling controllers.
Transport Network	The functional resources of the network that convey user information from one to another location bi-directionally or uni-directionally. A transport network can also transfer various kinds of network control information (e.g., operations and maintenance information).
Transport Network Assigned (TNA) Name	A name assigned to data bearing links connecting a UNI-N and a UNI-C. The TNA name is assigned by the transport service provider, either via a protocol or by configuration.
Transport Network Element (TNE)	A network element (within the transport network) having optical interfaces, such as an optical cross-connect (OXC) or an optical add/drop multiplexer.
UNI	The user-network interface is the service control interface between a client device and the transport network.

UNI-C	The logical entity that performs UNI signaling on the user device side.
UNI-N	The logical entity that performs UNI signaling on the network device side.
User or Client	Network equipment that is connected to the transport network for utilizing optical transport services. Examples of clients include IP routers, ATM switches, Ethernet Switches, SDH/SONET Cross-connects, etc.

5.2 Abbreviations

ASON	Automatically Switched Optical Network
CBS	Committed Burst Size
CE-VLAN ID	Customer Edge Virtual Local Area Network Identifier
CIR	Committed Information Rate
CoS	Class of Service
DCSC	Data Channel Switching Capable
DIP	Deletion In Progress
EBS	Excess Burst Size
EIR	Excess Information Rate
EPL	Ethernet Private Line
EVC	Ethernet Virtual Connection
EVPL	Ethernet Virtual Private Line
FF	Fixed Filter
FSM	Finite State Machine
GPID	Generalized Payload IDentifier
GMPLS	Generalized Multi-Protocol Label Switching
GRE	Generic Routing Encapsulation
GSMP	Generic Switch Management Protocol
IANA	Internet Assigned Numbers Authority
IP	Internet Protocol
L2SC	Layer-2 Switch Capable
LSC	Lambda Switch Capable
LSP	Label Switched Path
MEF	Metro Ethernet Forum
MPLS	Multi-Protocol Label Switching
MT	Multiplier
MTU	Maximum Transmission Unit
NCC	Number of Contiguous Components
NMC	Number of Multiplexed Components
NSAP	Network Service Access Protocol
NVC	Number of Virtual Components
OTN	Optical Transport Network as defined in [G709]
PDU	Protocol Data Unit
RFC	Request For Comments
RCC	Requested Contiguous Concatenation
RSVP	Resource ReSerVation Protocol
RSVP-TE	RSVP with Traffic Engineering extensions
SC	Switched Connection
SC PC ID	Signaling Controller Protocol Controller Identifier
SC PC SCN	Signaling Controller Protocol Controller Signaling Communications Network
SDH	Synchronous Digital Hierarchy
SE	Shared Explicit
SONET	Synchronous Optical NETwork
SPC	Soft Permanent Connection
SPE	SONET Payload Envelope
SRLG	Shared Risk Link Group
STS-N	Synchronous Transport Signal level N
TDM	Time Division Multiplexing
TNA	Transport-Network Assigned

TNE	Transport network Element
UNI	User Network Interface
UNI-N	UNI Signaling Agent – Network
UNI-C	UNI Signaling Agent – Client
UDP	User Datagram Protocol
UNI	User Network Interface
RSVP	Resource ReSerVation Protocol
SCN	Signaling Communications Network
VLAN	Virtual Local Area Network

6 UNI 2.0r2 Compatibility with UNI 1.0r2 and UNI 2.0

The UNI 2.0r2 obsoletes UNI 2.0 [OIF-UNI-02.0-RSVP]. The UNI 2.0r2 supports all RSVP messages and procedures described in UNI 1.0r2. These messages and procedures are included in this specification for completeness. UNI 2.0 also supports the following features described in [OIF-UNI-02.0-Common]:

1. Call Control
2. SONET/SDH Low-order Signals
3. Transport of Ethernet Services
4. Transport of OTN Interfaces
5. Enhanced Security
6. Call Modification

The UNI 2.0 specification also modifies certain procedures for existing features:

7. Network Initiated Graceful Deletion
8. Address Separation of Node Id and SC PC ID
9. Hello Procedures Clarification

Backwards compatibility considerations for each feature are covered in this section.

A UNI implementation MAY allow for manual configuration of a neighbors' version but this is not required.

6.1 Call Control

The Call Control feature introduces the CALL_ID RSVP object in Path message (Section 9.1.3), PathErr message (Section 9.1.4), PathTear message (Section 9.1.5) and Resv Message (Section 9.1.6). The CALL_ID format is defined in [RFC3474] with further clarifications provided in Section 9.2.12.

The CALL_ID object is mandatory in UNI 2.0. The presence of the CALL_ID object in messages sent to UNI 1.0 implementation should not cause any problems because the Class-Number for the CALL_ID is of the form 11bbbbbb and, according to [RFC2205], an implementation that receives an unknown object of this type should ignore and forward the object unmodified.

In order to use features associated with Call and Connection separation, all nodes involved in the call/connection MUST support UNI 2.0.

6.2 SONET/SDH Low-order Signals

SONET/SDH low-order signals are requested using the SONET/SDH SENDER_TSPEC [RFC4606] format that is identical to the format used in UNI 1.0r2. No new object is required, but the restriction that low-order signals cannot be requested has been lifted. All UNI nodes must support UNI 2.0 in order for this service to be available. Furthermore, this feature is optional under UNI 2.0, and nodes must be able to switch at the requested low-order signal rate level to provide this service.

6.3 Transport of Ethernet Services

Ethernet Services require a new type of SENDER_TSPEC/FLOWSPEC described in [ETH_PARAM] and two new label formats for Ethernet Private Line and Ethernet Virtual Private Line as described in Section 9.2.1. To be able to request this service, all UNI nodes MUST support UNI 2.0 and Ethernet Services.

6.4 Transport of OTN Interfaces

OTN Interfaces require a new type of SENDER_TSPEC/FLOWSPEC and a new label format described in [RFC4328]. In order to be able to request this service, all UNI nodes MUST support UNI 2.0 and OTN (G.709) Interfaces.

6.5 Enhanced Security

Enhanced Security does not have a direct impact on this specification and does not introduce any backwards compatibility issues in the RSVP signaling implementation.

6.6 Call Modification

To be able to modify a call, all UNI nodes MUST support UNI 2.0 and a common mechanism for the optional call modification extensions.

The first mechanism is call modification by adding or removing connections to an existing call. This mechanism only requires support for the CALL_ID object. See Section 6.1 for backwards compatibility discussion of the CALL_ID.

The second mechanism is call modification by modifying the service parameters of an existing connection. This mechanism relies on the use of the Shared-Explicit (SE) reservation style requested in the SESSION_ATTRIBUTE object. The use of this new reservation style and object are described in Section 8.10.2.

6.7 Network Initiated Graceful Deletion

The UNI 2.0 uses the Notify message as opposed to the ADMIN_STATUS object A+R bits in the Path/Resv message to initiate graceful deletion from the network. Network deletion includes deletion initiated by the source and destination UNI-Ns.

The procedures for graceful deletion initiated by the source or destination UNI-C remain the same as in UNI 1.0.

6.7.1 UNI-N 2.0 Interworking with UNI-C 1.0

A UNI-C 2.0 implementation MUST include a NOTIFY_REQUEST object in every Path and Resv message it generates and fill it with its SC PC ID as described in Section 9.2.15. A UNI-N 2.0 implementation determines which network deletion procedures to apply based on the absence or presence of a properly filled NOTIFY_REQUEST object in the Path/Resv message received from the UNI-C.

The deletion procedures described in Section 8.11 apply unless the source UNI-C is UNI 1.0 compliant and does not send the NOTIFY_REQUEST to the source UNI-N in Path messages.

If the source UNI-N 2.0 implementation receives a Path message without the NOTIFY_REQUEST object, it MUST follow the network initiated graceful deletion procedures described in [OIF-UNI-01.0-R2-RSVP] when deletion is triggered locally, in the network or the destination UNI-N. This is illustrated in Figure 1. If a source UNI-C does not respond to a network initiated graceful deletion Resv message, the network MUST continue the message flow illustrated below the dashed line in Figure 1.

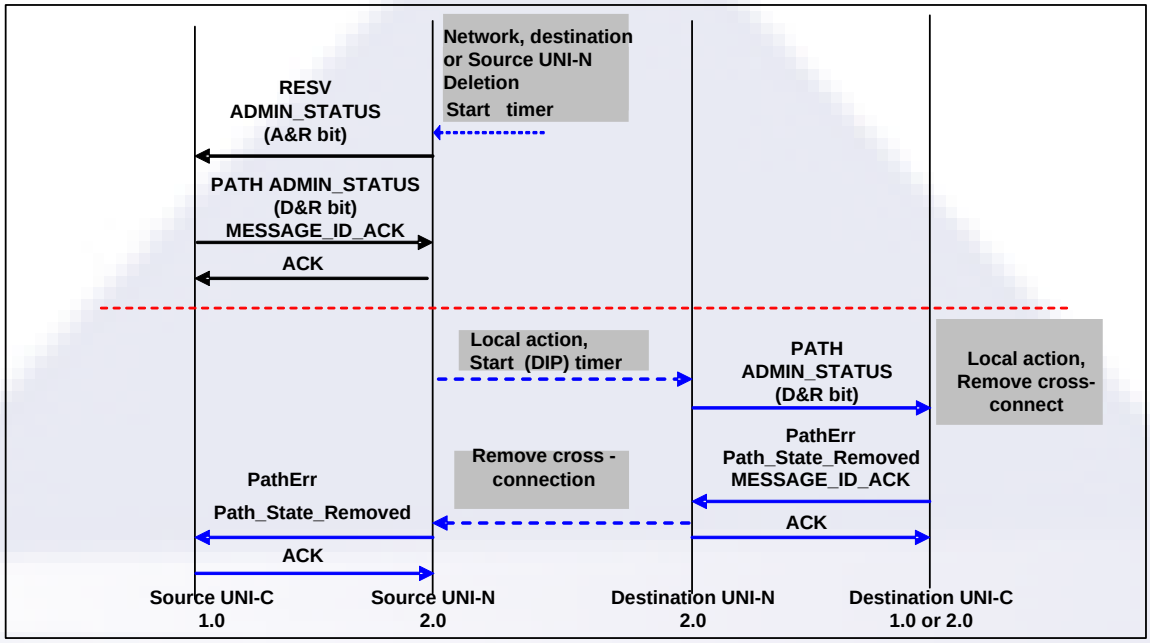


Figure 1 - Network Initiated Graceful Deletion with a Source UNI-C UNI1.0

6.7.2 UNI-N 1.0 Interworking with UNI-C 2.0

There are no compatibility issues between a UNI-C 2.0 and UNI-N 1.0 as long as the UNI-N implementation follows [RFC2205] for unknown class numbers.

The UNI-C 2.0 implementation MUST include the NOTIFY_REQUEST object, even if the UNI-N implementation is 1.0. If a UNI-C 2.0 receives an ADMIN_STATUS object with the A+R bits set in the Path or Resv message, it SHOULD respond with a Resv or Path message with the D+R bits set in the ADMIN_STATUS object to trigger graceful deletion as per UNI 1.0 procedures. The NOTIFY_REQUEST object Class Number is 195. That Class Number is of the type that should be forwarded but ignored by implementations that do not recognize the Class Number [RFC2205]. The reception of a NOTIFY_REQUEST object by a UNI-N 1.0 implementation should not cause any problems.

6.8 Address Separation of Node Id, SC PC ID and SC PC SCN Address

The UNI 2.0 specification introduces separation of the following identifiers: Node Id, SC PC ID and SC PC SCN Address. In UNI 1.0, all these identifiers were combined together to form the Node Id. If a UNI 2.0 implementation has to support UNI 1.0 interfaces, it has to use the same value for Node Id, SC PC ID and SC PC SCN address.

Identifier separation is described in Section 6.1 of [OIF-UNI-02.0-Common].

The address separation did not result in new protocol-specific objects beyond those already defined in UNI 1.0. For backwards compatibility, it is sufficient for a UNI 2.0 implementation to use the same value for Node Id, SC PC ID and SC PC SCN address.

6.9 Hello Procedures Clarification

The UNI 2.0 specification states that Hello messages MUST be exchanged successfully prior to accepting any other RSVP message. Interworking with a UNI 1.0 implementation requires one of the following:

- The UNI 1.0 implementation supports Hello Messages prior to other RSVP message exchanges or;
- The UNI 2.0 implementation allows for configuration of the Hello Message behavior to support UNI 1.0 implementations.

The UNI 2.0 specification mandates the use of the 0xFFFFFFFF value for the RESTART_TIME field within the RESTART_CAP object of the Hello Message and a non-zero value for the RECOVERY_TIME. When interworking with UNI 1.0, a UNI 2.0 MAY ignore the RESTART_TIME field within the RESTART_CAP object value if it differs from the 0xFFFFFFFF value. If the RECOVERY_TIME advertised by the UNI 1.0 is 0 when the Hello adjacency is recovered, then the UNI 2.0 MAY delete the connections to/from UNI 1.0.

7 UNI 2.0 Signaling Messages and RSVP Objects

The UNI 2.0 abstract messages and procedures are described in [OIF-UNI-02.0-Common]. Most of these are directly supported by re-using existing procedures, messages, and objects defined under RSVP-TE [RFC3209] and GMPLS extensions for RSVP-TE [RFC3471, RFC3473]. This is summarized in Table 1.

[OIF-UNI-02.0-Common] also defines the set of attributes to be signaled. Table 2 summarizes those attributes and the corresponding RSVP objects. Specific UNI-related object formats and usage are described in Section 9.

Message No.	Abstract Message and Procedure Description	RSVP Message
1	Connection Setup Request	Path
2	Connection Setup Indication	Resv, PathErr
3	Connection Setup Confirm	ResvConf
4	Connection Release Request	Path or Resv with ADMIN_STATUS "Deletion in Progress" bit
5	Connection Release Indication	PathErr with Path_State_Removed flag, PathTear
6	Connection Query Request	implicit
7	Connection Query Indication	implicit
8	Connection Notify	PathErr, Notify[1]
9	Connection Modify Request	Path
10	Connection Modify Indication	Resv, PathErr
11	Connection Modify Confirm	ResvConf
12	Signaling Adjacency Maintenance	Hello

Table 1 – Mapping between UNI Abstract Messages and RSVP Messages

[1] Notify Message is only used for notification of network initiated graceful deletion

UNI Attributes	RSVP object	Reference
Source TNA Name	GENERALIZED_UNI/Source TNA	Section 9.2.5.1
Source Logical Port Identifier	IPv4_IF_ID_RSVP_HOP	[RFC3473]
Source Generalized Label	GENERALIZED_LABEL	[RFC3473]
Destination TNA Name	GENERALIZED_UNI/Destination TNA	Section 9.2.5.5
Destination Logical Port Identifier	GENERALIZED_UNI/Egress Label, GENERALIZED_UNI/SPC_Label	Section 9.2.5.10
Destination Generalized Label	GENERALIZED_UNI/Egress Label GENERALIZED_UNI/SPC_Label	Section 9.2.5.10
Local Connection ID	(UNI_IPv4_SESSION, LSP_TUNNEL_IPv4_SENDER_TEMPLATE) or (UNI_IPv4_SESSION, LSP_TUNNEL_IPv4_FILTER_SPEC)	Section 9.2.4/9.2.1
Contract ID	POLICY_DATA	[RFC2205], [OIF-UNI-02.0-Common]
SONET/SDH Traffic Parameters	SONET/SDH_SENDER_TSPEC, SONET/SDH_FLOWSPEC	Section 9.2.13
G.709 Traffic Parameters	G.709 SENDER_TSPEC, G.709 FLOWSPEC	[RFC4328]
Ethernet Traffic Parameters	ETHERNET SENDER_TSPEC, ETHERNET FLOWSPEC	[RFC6003]
Directionality	UPSTREAM_LABEL	[RFC3473]
Generalized Payload Identifier	GENERALIZED_LABEL_REQUEST/G-PID	[RFC3473]
Service Level	GENERALIZED_UNI, Service Level	Section 9.2.5.11
Diversity	GENERALIZED_UNI, Diversity	Section 9.2.5.9
Error Code	IPv4_ERROR_SPEC / IF_ID ERROR_SPEC	[RFC3473]
Connection Status	ADMIN_STATUS	[RFC3473]
Encoding Type	GENERALIZED_LABEL_REQUEST/ LSP_ENC_TYPE	[RFC3473]
Switching Type	GENERALIZED_LABEL_REQUEST/ SWITCHING_TYPE	[RFC3473]
Call Name	CALL_ID	[RFC3474] and Section 9.2.12

Table 2 – Mapping between UNI Attributes and RSVP Objects

8 UNI RSVP Signaling Procedures

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 network. In fact, the UNI RSVP messages contain values as if they are used to set up two separate single-hop connections, one between the source UNI-C and source UNI-N, and the other between the destination UNI-N and the destination UNI-C. The network is assumed to provide coordination of signaling information between the source and destination UNI-Ns. UNI 2.0 does not make any assumption about the signaling protocol supported within the network; it does require that the network support the (transparent) transport of the information from the GENERALIZED_UNI object from the source UNI-N towards the destination UNI-N.

8.1 UNI Interfaces, Signaling Adjacency, SCN , Logical Port Identifier and Addressing

RSVP messages are exchanged over the SCN as described in [OIF-UNI-02.0-Common]. The identification of connection endpoints is described in [OIF-UNI-02.0-Common].

8.2 Sending UNI RSVP Messages

When a UNI-C (UNI-N) is sending an RSVP message, it MUST address the message directly to its UNI-N (UNI-C) peer. The peer's SC PC SCN address is used for this purpose (see [OIF-UNI-02.0-Common]). A node SHOULD use simple IP encapsulation and the router-alert option MUST NOT be included in any RSVP/IPv4 messages. This is shown in Table 3.

Either GRE or IP-in-IP encapsulation MAY be used (by configuration) if simple IP encapsulation poses operational problems.

IPv4 Header Values for UNI RSVP messages	
Version	4
Header Length	5
TOS	As defined in [RFC 2205]
Total Length	Message length
Identification	As defined in [RFC 791]
Flags	As defined in [RFC 791]
Fragment Offset	As defined in [RFC 791]
TTL	≥ 1
Protocol	46
Header Checksum	As defined in [RFC 791]
Source Address	UNI-C/UNI-N SC PC SCN IP address
Destination Address	UNI-C/UNI-N SC PC SCN IP address

Table 3 – IP header for UNI RSVP messages

8.3 Receiving UNI RSVP Messages

A UNI-C or a UNI-N node SHOULD process a received RSVP message as specified in [RFC2205, RFC3209] only if all security validation procedures (if implemented as described in [OIF-UNI-02.0-Common]) have been successfully performed. Specifically, a received RSVP PDU that fails security validation MUST be dropped and an ACK message MUST NOT be generated, even if an ACK was requested.

8.4 Reliable Messaging

To support reliable messaging across the UNI, UNI-C and UNI-N implementations MUST support the RSVP Refresh Reduction Extensions [RFC2961]. In particular, the MESSAGE_ID object MUST be

included in Notify, Path, PathTear, PathErr, Resv, ResvTear, ResvErr, ResvConf, and Srefresh messages. The Ack_Desired flag in a MESSAGE_ID object MUST be set in Path and Resv trigger[2] messages, Notify, PathTear, PathErr, ResvTear, ResvErr, ResvConf, and Srefresh message, and MAY be set in Path and Resv refresh messages.

Message identification and acknowledgment are done on a per RSVP hop basis. Each MESSAGE_ID object contains a message identifier. This identifier MUST uniquely identify a message with respect to a node's Signaling Controller Protocol Controller Identifier (SC PC ID).

Failure to receive an acknowledgment for a full Path/Resv refresh or Srefresh message MUST NOT result in the deletion of the corresponding connection.

Note that ACK messages or MESSAGE_ID_ACK object may not appear in the exact manner as shown in the timing diagrams in this section.

8.5 Connection State Maintenance

RSVP takes a "soft state" approach to manage the connection state. RSVP soft state is created and periodically refreshed by Path and Resv messages. The state is deleted if no matching refresh messages arrive before the expiration of a "cleanup timeout" interval. State MAY also be deleted by an explicit PathTear, PathErr with Path_State_Removed flag, or ResvTear message.

UNI signaling maintains the soft state approach but requires explicit tear-down messages from the user. That is, connection deletion should normally be in response to an explicit tear-down request rather than soft-state timeout. Therefore, a state timeout occurring at a UNI-C or a UNI-N indicates a problem. In response to a state timeout an alarm MAY be reported.

A control plane failure MUST not result in the release of established connections. Setup requests in the process of being completed MAY be removed (either during the failure or after recovery from failure). Established connections associated with a pending release request MUST be released (either during the failure or after recovery from failure).

The use of reliable messaging, via the MESSAGE_ID and MESSAGE_ID_ACK objects, does not remove the need to refresh connection states. To reduce the number of refresh messages, a node SHOULD use the summary refresh (Srefresh) message [RFC2961] to refresh the connection states.

8.6 Reservation Style

RSVP reservation "styles" are defined in [RFC2205]. UNI signaling makes use of the Fixed Filter (FF) style for connections that do not support connection modification. UNI signaling uses Shared Explicit (SE) style for connections that support connection modification.

8.7 Local Connection Identification

A Local Connection Identifier is used to uniquely identify a connection at a UNI. Under RSVP signaling, this is realized using the UNI_IPv4_SESSION object in Path, PathTear, PathErr, Resv, ResvTear, ResvErr, ResvConf and Notify messages. The local connection identifier is unique and remains the same for the lifetime of the connection, even when connection modification is performed. In the case of connection modification, multiple Path state FSMs are maintained and are distinguished by the LSP_TUNNEL_IPv4_SENDER_TEMPLATE in Path, PathTear, PathErr and Notify messages and by the

[2] Trigger and refresh messages are defined in [RFC2961].

LSP_TUNNEL_IPv4_FILTER_SPEC in the Resv, ResvTear, ResvErr and ResvConf messages. This is described in Section 8.10.2.

8.8 Connection Traffic Parameters

The traffic parameters of a connection are technology dependent and are encoded in the SENDER_TSPEC and the FLOWSPEC objects. For SONET/SDH service, a UNI-N and a UNI-C node MUST support standard SONET/SDH traffic parameters defined in GMPLS SONET-SDH specification [RFC4606] and covered in [OIF-UNI-02.0-Common]. For G.709 service, a UNI-N and a UNI-C node MUST support standard G.709 traffic parameters defined in the GMPLS G.709 specification [RFC4328] and covered in [OIF-UNI-02.0-Common]. For Ethernet service, a UNI-N and a UNI-C node MUST support the Ethernet traffic parameters and Ethernet Bandwidth Profile defined in [RFC6003]. When the requested connection traffic parameters are not supported by the network, a PathErr message with the following error code MUST be generated by the network: "Traffic Control Error/ Service unsupported" [RFC2205].

8.9 Connection Creation

To create a connection, a UNI-C node sends a Path message to its adjacent UNI-N node. The Path message SHALL include a GENERALIZED_LABEL_REQUEST object, which indicates that a label binding is requested for this connection. The traffic parameters (characteristics) of the connection are encoded in an appropriate object in the Path message and an appropriate FLOWSPEC object in the corresponding Resv Message. Figure 2 shows the timing diagram and message flow during successful connection creation.

The IPv4_IF_ID_RSVP_HOP object in a Path message specifies the interface over which the connection is to be established. The GENERALIZED_LABEL object in the Resv message specifies the allocated label(s) (on the interface) to be used by the connection.

To request a bi-directional connection, a UNI-C MUST insert an UPSTREAM_LABEL Object in the Path message to select the upstream label(s) for the connection.

For EVPL, SONET/SDH and G.709 connections, if a node requires the GENERALIZED_LABEL and UPSTREAM_LABEL to be of the same value, it SHOULD include a LABEL_SET object such that the GENERALIZED_LABEL is limited to the same value as the UPSTREAM_LABEL. This is the typical configuration for SONET/SDH, G.709 and Ethernet connections.

For EPL connections, the LABEL_SET MUST NOT be used as the GENERALIZED_LABEL represents the logical port identifier and can be inferred from the RSVP_HOP sub-objects.

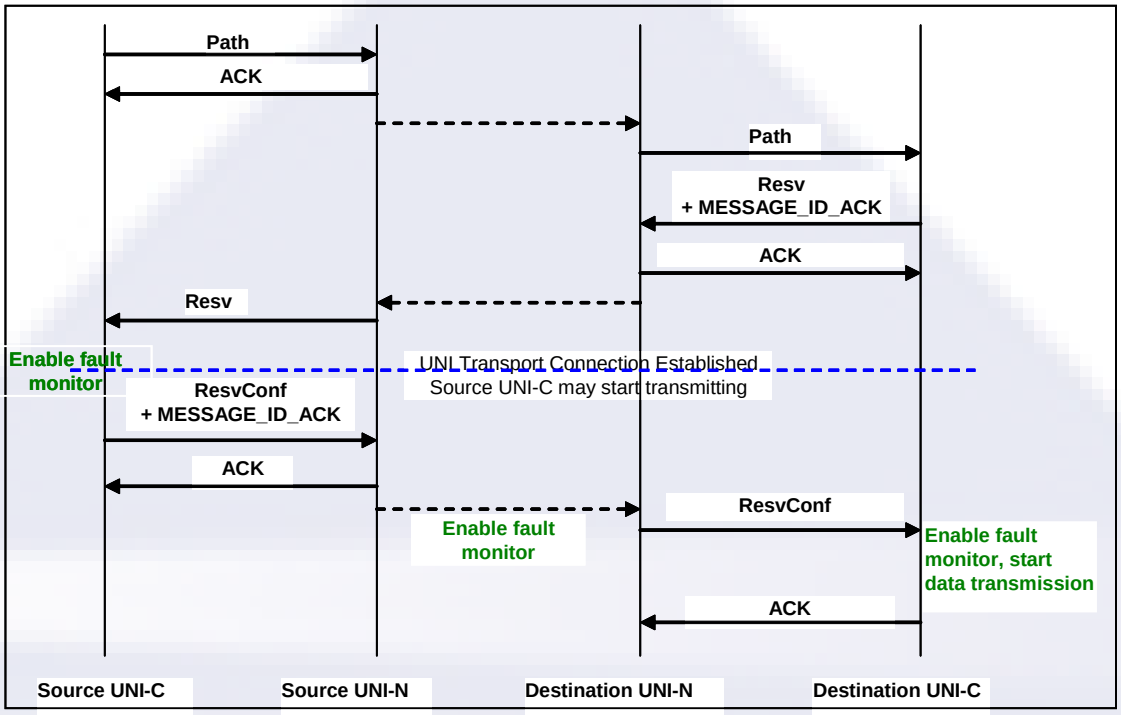


Figure 2 – Successful Connection Establishment

It can be assumed that the connection segment within the transport network has been established only when the UNI-N sends the Resv message to the source UNI-C. Therefore, to avoid loss of data, the source client SHOULD NOT start data transmission before the Resv message is received by the corresponding UNI-C. The destination client MAY choose to insert the RESV_CONFIRM object in the Resv message. In this case, the destination client SHOULD NOT start data transmission before the ResvConf message is received by the corresponding UNI-C. If the destination UNI-C inserted a RESV_CONFIRM object in the Resv message but did not receive a RESV_CONFIRM in the timeframe determined by local policy or configuration, it MAY delete or alarm the connection. The decision to delete or alarm the connection is local to the destination client and may be based on local policy or configuration. The destination client SHOULD be ready to receive data before the corresponding UNI-C sends the Resv message, and source client SHOULD be ready to receive data before the corresponding UNI-C initiates the ResvConf message. If a node monitors a connection, it SHOULD NOT raise a service affecting alarm until it has verified that the connection has been established end-to-end. This is shown by the dashed line in Figure 2.

Contention for labels may occur between two connection creation requests. The contention resolution procedure in this case is described in [RFC3473].

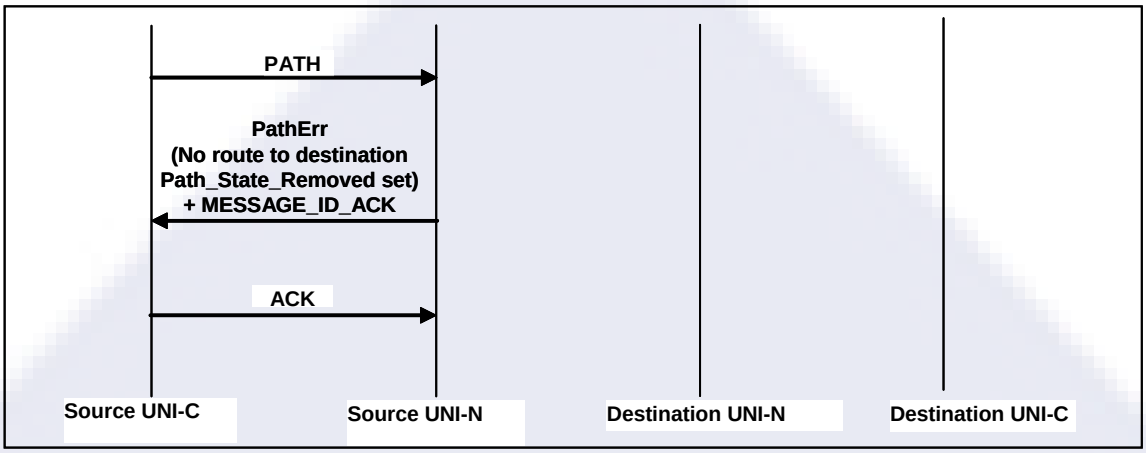


Figure 3 – Connection Rejection by the Network using the Path_State_Removed Flag

A node rejecting a connection request MUST delete its own path state and set the Path_State_Removed flag in the PathErr message. This is illustrated in Figure 3.

A connection may not be successfully created due to resource unavailability, policy or reachability constraints. Figure 4 and Figure 5 illustrate timing diagrams and message flows for certain unsuccessful connection creation scenarios.

If the Path_State_Removed flag is not set in a PathErr message, the source UNI-C MUST delete the connection explicitly. To delete the connection explicitly, the UNI-C MUST send a PathTear message. A node that receives a PathTear, that does not match any path state MUST acknowledge the message if the PathTear carries a MESSAGE_ID with the Ack_Desired flag set, and then discard the PathTear message. This is shown in Figure 4.

A node receiving a PathErr with Path_State_Removed flags set, SHOULD NOT send a PathTear downstream. This is shown in Figure 5.

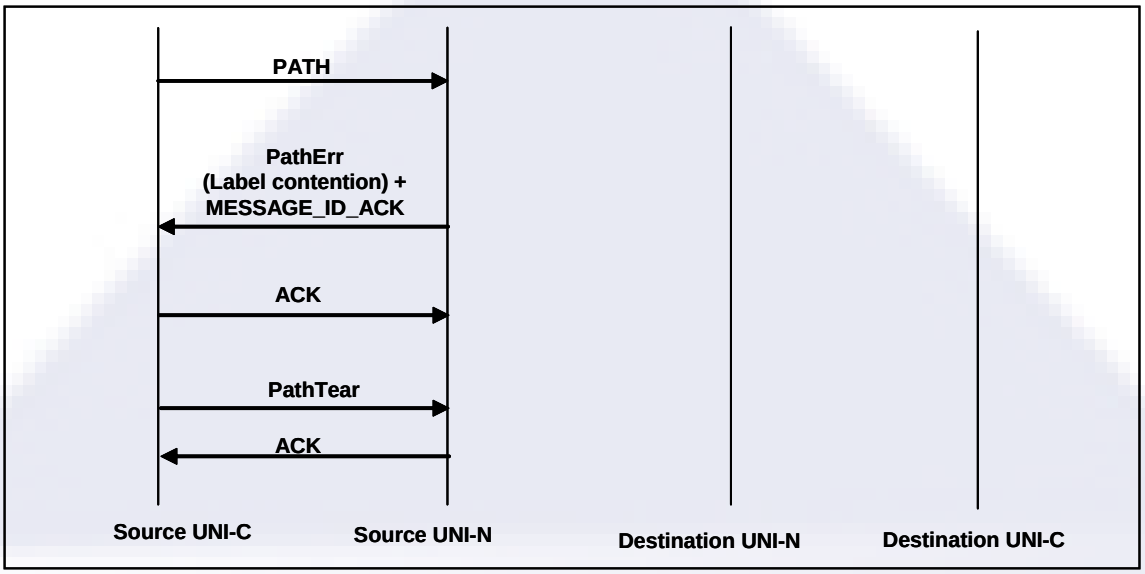


Figure 4 – Connection Rejection by the Network, without the Use of Path_State_Removed Flag

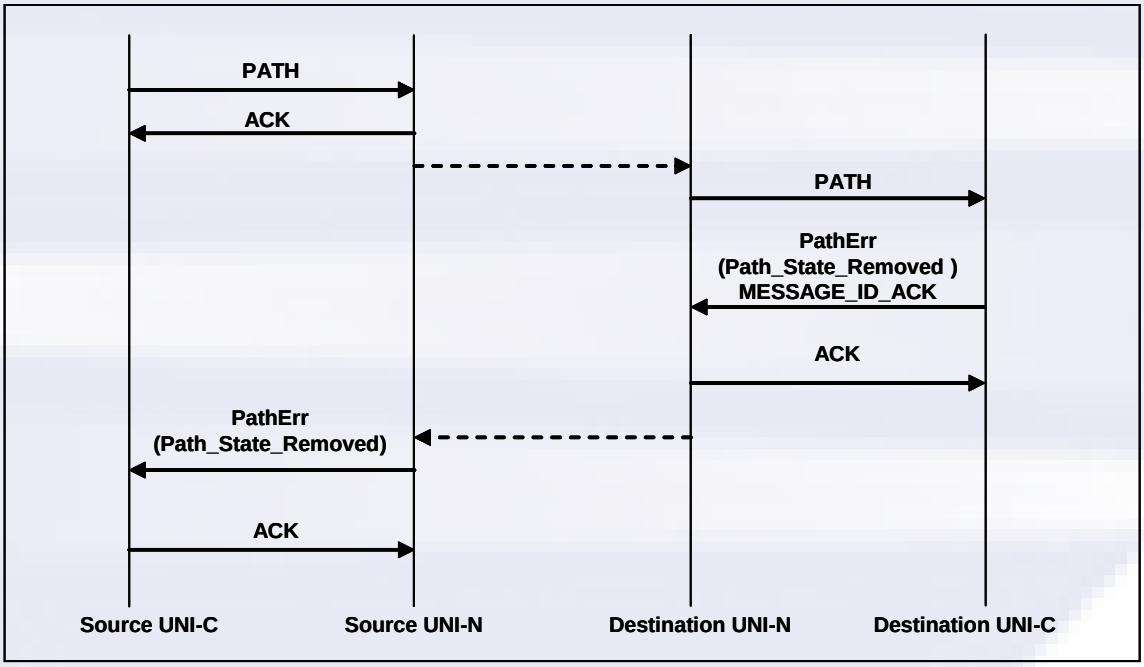


Figure 5 – Connection Set-Up Rejection by the Destination UNI-C

8.10 Call Modification

Call Modification support is optional. This feature allows for CE-VLAN ID modification for Ethernet services and non-disruptive bandwidth modification for all services. Two mechanisms can be used to non-disruptively modify the bandwidth of a call. The first mechanism described in Section 8.10.1 modifies a call by adding or removing connections associated with the call. The second mechanism described in Section

8.10.2 modifies a call by non-disruptively modifying the service parameters of one or more of its connections. The second mechanism MAY be used to modify the CE-VLAN ID associated with an EVC.

The two mechanisms described above MAY be applied to both Ethernet and TDM (SONET/SDH, G.709). However, the value of each mechanism differs for each technology. In particular, the first mechanism of simply adding multiple connections for Ethernet seems to have limited value in that its use would require managing multiple QoS mechanisms within the same Ethernet call. Implementations that support bandwidth modification for TDM clients (SONET/SDH, G.709) MUST support call modification by adding and removing connections to an existing call as described in Section 8.10.1 and MAY support call modification by modifying the bandwidth of an existing connection as described in Section 8.10.2. Implementations that support bandwidth modification for Ethernet clients MUST support the procedures to modify a call by modifying an existing connection as described in Section 8.10.2 and MAY support bandwidth modification by adding and removing connections to an existing call as described in Section 8.10.1.

8.10.1 Call Modification by Adding and Removing Connections

Adding and removing connections to an existing call can be used to modify the bandwidth of a call. A failure to add or remove a connection does not impact other connections in the call. That is, the connections remain independent of each other within the call.

Figure 6 shows a successful addition of a connection to an existing call. The call is established when the first connection is established. The UNI-C sends a Path message to the UNI-N to request a new call and connection creation. The UNI-C MUST send the null CALL_ID when sending a Path message for the first connection within the call. The CALL_ID is assigned by the source UNI-N and used in all subsequent messages at the source and destination UNI-C. If the source UNI-C subsequently wants to modify the call by adding another connection, it will generate a new Path message with the same CALL_ID it received in the Resv for the first connection. The presence of the CALL_ID in a Path message for a new connection is used to infer that a connection is being added to the specified call. CALL_ID is used to correlate the various connections at the UNI-C and UNI-N. This new Path message will have a different connection identifier (LSP_ID and TUNNEL_ID) and a new MESSAGE_ID. Refer to Table 4 and Table 5 for a list of rules explaining which objects are allowed to differ for connections that are part of the same call. When a Path message is received at the source UNI-N with a non-null CALL_ID, a PathErr with Invalid/unknown CALL_ID error as listed in Section 9.2.16 MUST be sent to the UNI-C if no corresponding call state exists on the UNI-N.

Failure to add a connection to an existing call does not impact other connections as each connection has its own RSVP state.

The connection deletion message sequence is described in Section 8.11. Individual connections within a call can be deleted from the source UNI-C, the destination UNI-C, the source UNI-N, the destination UNI-N or the network. Each connection deletion is performed independently. A call without connections is not supported. Removal of the last connection results in the removal of the call state.

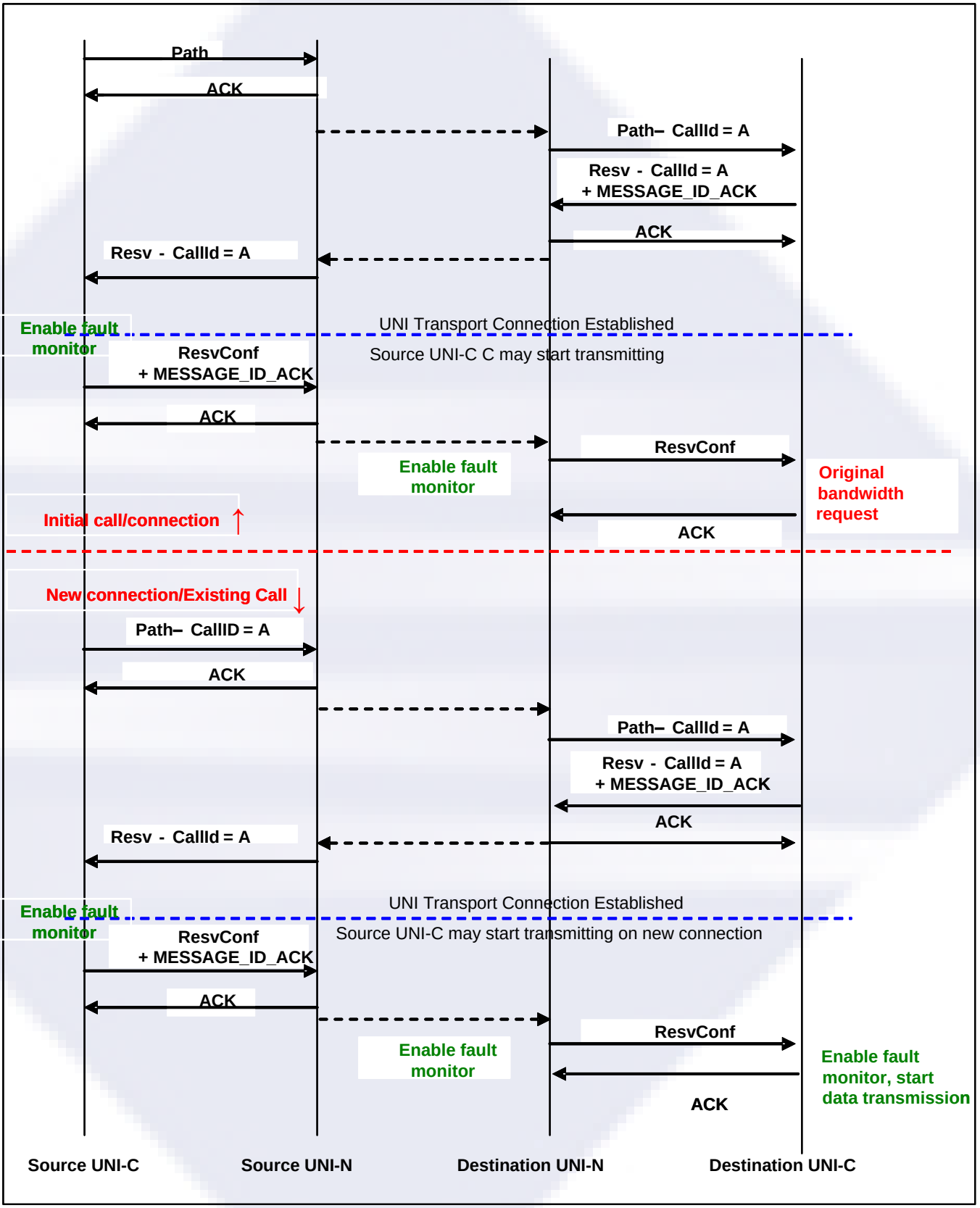


Figure 6 – Successful Call Modification – Adding a Connection

8.10.2 Connection Modification

Some parameters can be modified in an existing connection. Connection modification can be used to modify the bandwidth of a call by modifying the bandwidth of underlying connections or to modify the CE-VLAN ID mapping to an EVC. Note that this mechanism MAY be combined with the connection addition/removal mechanism described in the previous section. It is possible to have a call with multiple connections and to modify the bandwidth of one of the connections to increase/decrease the total bandwidth of the call. For SONET/SDH and G.709 calls, this mechanism modifies the connection multiplier. It differs from the previously described connection addition/removal mechanism by introducing shared protocol state.

This mechanism does not permit the Characteristic Information of a call to change. For example, modifying an STS-3c connection to an STS-12c connection requires a change in the Characteristic Information and is not a valid connection modification.

Modifying the service parameters of an existing connection can be achieved in RSVP by using the Shared Explicit (SE) reservation style. This allows two or more RSVP Path states to share the same bandwidth profile and CE-VLAN ID information. A UNI-C that supports non-disruptive service parameter modification SHOULD request the SE reservation style in the SESSION_ATTRIBUTE. If the SE reservation style is not used in the Resv message, the UNI-C MUST NOT request non-disruptive connection modification. If the UNI-N receives a request to non-disruptively modify a connection for which the FF (Fixed Filter) reservation style is used, a PathErr with "Traffic Control Error: Service Unsupported" error as listed in Section 9.2.16 MUST be sent to the UNI-C.

A service parameters connection modification is illustrated in Figure 7. The same message flow could be used to modify the bandwidth of an existing connection or CE-VLAN ID mapping to an EVC.

The call is established when the first connection is established. The UNI-C sends a Path message to the UNI-N to request call and connection creation. The Path request contains a SESSION_ATTRIBUTE object with SE Style requested flag set. The UNI-C MUST send the null CALL_ID. The CALL_ID is assigned by the source UNI-N and used in subsequent messages at the source and destination UNI-C. If the source UNI-C subsequently wants to modify the call by modifying the connection service parameters, it generates a new Path message with the same CALL_ID it received in the Resv for the first connection. CALL_ID is used to correlate the various connections at the UNI-C and UNI-N. This new Path message will have a different LSP_ID and MESSAGE_ID but the same TUNNEL_ID. When a Path message is received at the source UNI-N with a non-null CALL_ID, a PathErr with Invalid/unknown CALL_ID error as listed in Section 9.2.16 MUST be sent to the UNI-C if no corresponding call state exists on the UNI-N. Refer to Table 4 and Table 5 for a list of rules explaining which objects are allowed to differ for Path/Resv messages that share the same bandwidth. A new Resv message is generated in the reverse direction. This can be achieved using two different mechanisms. The recipient of the RESV message(s) MUST be able to handle both procedures as it is up to the sender of the RESV message to determine which mechanism is implemented. The mechanisms are as follows:

- As illustrated in Figure 7, a single Resv state MAY correspond to all Path states that share the service parameters, i.e. have the same TUNNEL_ID. In this case, a single Resv message is sent, but its contents will change when LSPs are added or removed. The Resv message includes the bandwidth of the largest bandwidth request it received in the Path messages and the SE flow descriptor includes the FILTER_SPECs and labels of all corresponding Path messages, including all CE-VLAN IDs for EVPL. Even if the labels for all states are identical, multiple labels are included, one after each FILTER_SPEC object.
- A new Resv message MAY be sent for only the new Path state. It contains the FILTER_SPEC and labels of the corresponding Path message only. The RECOMMENDED procedure is to keep refreshing the previous Resv message corresponding to the state that is awaiting teardown until the PathErr with PATH_STATE_REMOVED flag set has been received.

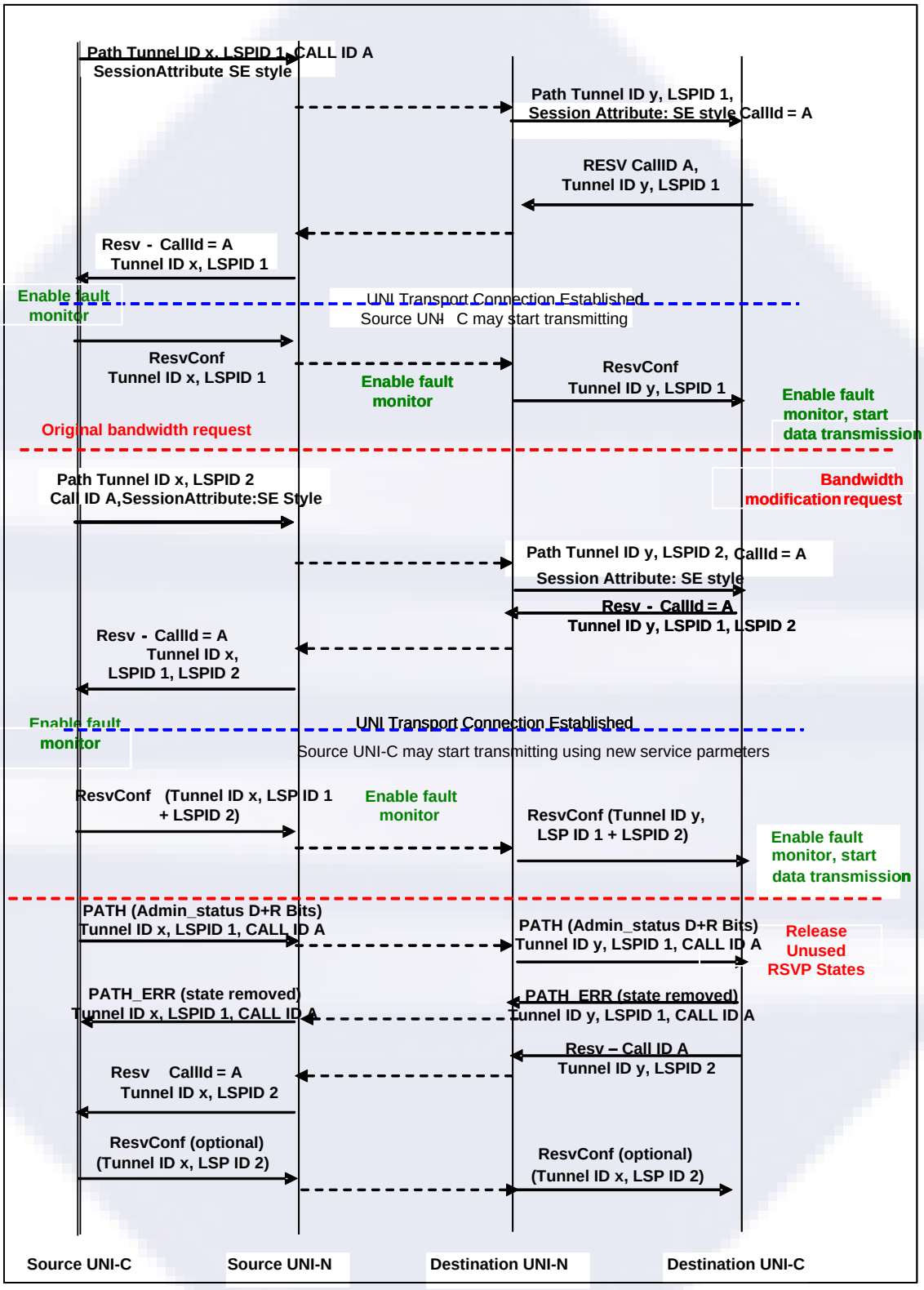


Figure 7 – Successful Connection Modification[3]

[3] Reliable message has been omitted from the figure but occurs for each trigger message.

For both mechanisms, if the Resv message includes a RESV_CONFIRM object, the source UNI-C MUST send a new ResvConf message which includes at least the new LSP_ID value. At this point, the data can be transmitted end to end with the new service parameters and the source UNI-C SHOULD trigger the original Path state removal by generating a Path message with the Delete and Reflect bits set in the ADMIN_STATUS object. This results in the graceful removal of the RSVP Path state at each node when the destination UNI-C responds with a PathErr message with the PATH_STATE_REMOVED flag set. If the source UNI-C does not initiate the graceful removal of the Path states, unnecessary RSVP Path states will be maintained across all nodes along the signaling path. The deletion of the Path state will result in the removal of the corresponding Resv state after the PathErr message with the PATH_STATE_REMOVED flag set has been received. The Resv corresponding to the deleted state will stop being refreshed. If a single Resv message was used for all Path states, the Resv message SE flow descriptor contents will contain one fewer FILTER_SPEC. A final ResvConf message MAY be sent to confirm the removal of the Resv state.

A failure to increase a connection bandwidth SHOULD result in a PathErr being sent for the Path message requesting more bandwidth. This MUST NOT impact the existing connections, other Path messages and RSVP states. Figure 8 illustrates a failure to increase the bandwidth in the network. The failure could also happen at the destination UNI-C. In this case, the destination UNI-C would generate the PathErr message.

A bandwidth decrease can be achieved with an identical message flow although the ResvConf message may not be necessary in that case. The new bandwidth would become effective at the PathErr stage as opposed to the ResvConf message stage.

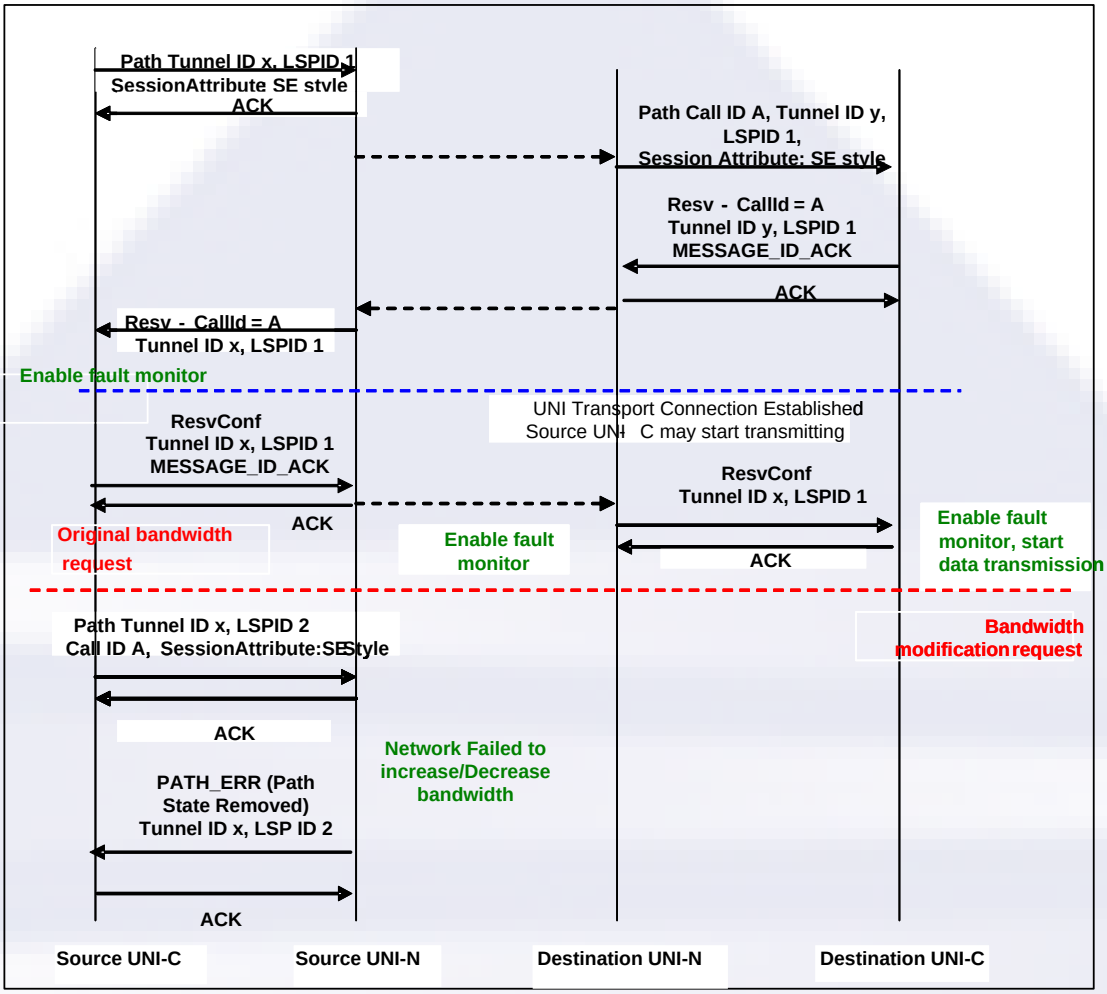


Figure 8 –Connection Modification – Failure to Increase the bandwidth in the Network

A failure to decrease a connection bandwidth SHOULD result in a PathErr being sent for the Path message requesting less bandwidth. This MUST NOT impact the existing connections, other Path messages and RSVP states.

Figure 9[4] illustrates a failure to decrease the bandwidth in the network. If the network fails to decrease the bandwidth, it returns a successful code to the UNI-C. It is the responsibility of the network to clear its resources at a later time. The failure could also happen at the destination UNI-C. In this case, the destination UNI-C would generate the PathErr message. In case of failure, the UNI-C may retry at a configured interval.

[4] While Figure 9 shows bandwidth modification being performed with Forced Teardown, use of Graceful Deletion is not precluded.

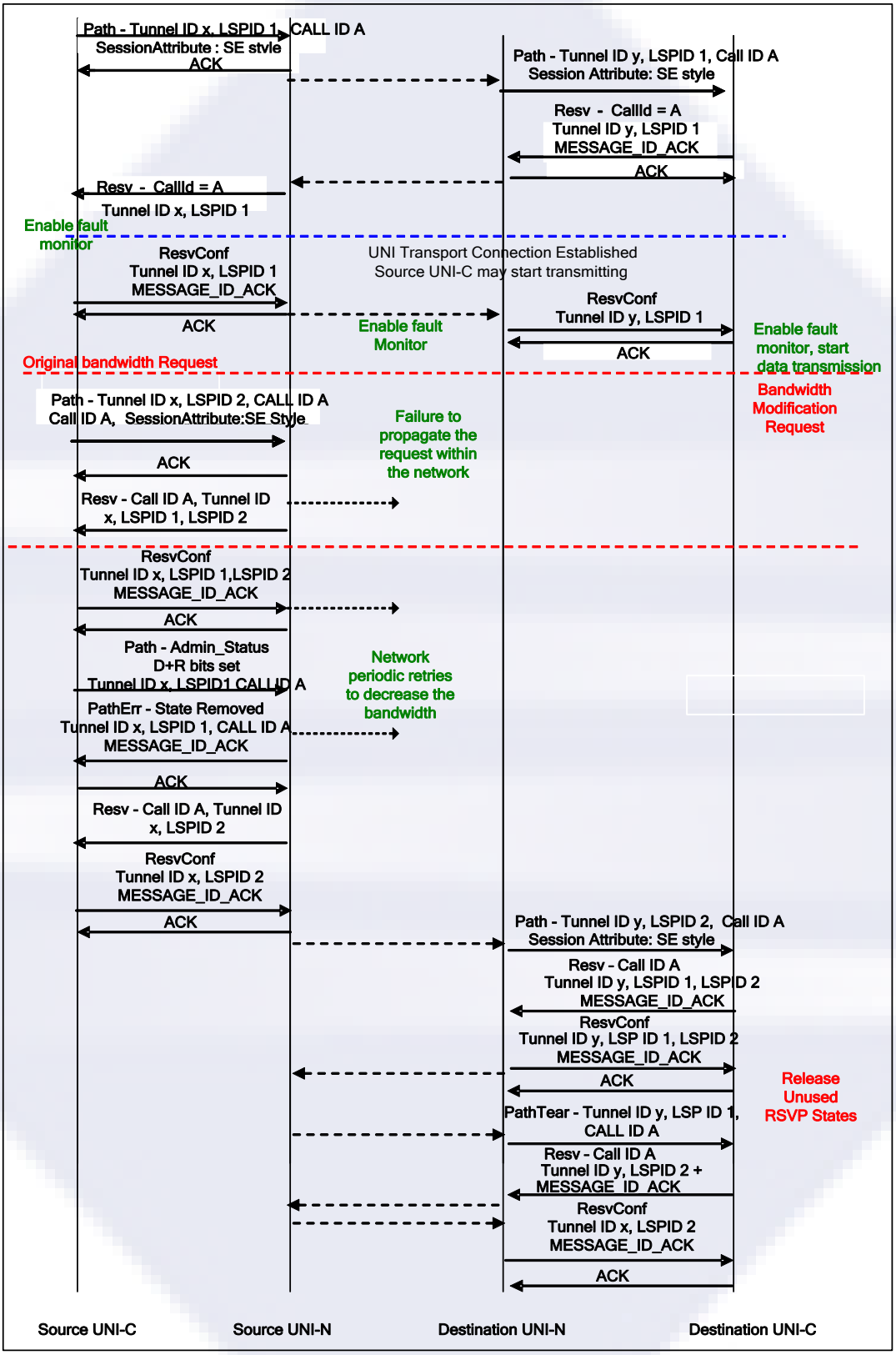


Figure 9 - Connection bandwidth - Failure to decrease the bandwidth within the network

8.10.3 Rules Associated with Call and Connection Modification

The Path message is used for call and connection creation and modification. The content of the Path message includes the following objects. Table 4 lists the rules associated with connection addition/removal and connection modification.

RSVP Path Msg object	**Abstract Attribute**	**Applicability**	**Rules Add/Remove Connections**	**Rules Modify Existing Connection**
INTEGRITY	None – RSVP specific (Optional)	N/A	Minimally recomputed each time the message contents changes.	Minimally recomputed each time the message contents changes.
MESSAGE_ID_ACK/ MESSAGE_ID_NACK	None – RSVP specific (Optional)	N/A	Sent as required.	Sent as required.
MESSAGE_ID	None – RSVP specific (Mandatory)	N/A	Must be different for each connection.	Must be different for each related Path message .
UNI_IPv4_SESSION (UNI-C SC PC ID , tunnel id, UNI-N SC PC ID)	Local connection ID (Mandatory)	Connection	Different for each connection.	Tunnel ID MUST be the same for all related Path messages that share resources.
LSP_TUNNEL_IPv4_ SENDER_TEMPLATE	Local connection ID (Mandatory)	Connection	LSP_ID MUST be different for all connections having the same TunnelID.	LSP_ID MUST be different for all related Path messages sharing the same Tunnel ID.
IPv4_IF_ID_RSVP_HOP	Source Logical Port Identifier (Mandatory)	Connection	Can differ or be the same for the connections.	Same for all related Path message sharing resources
TIME_VALUES	None – RSVP specific (Mandatory)	N/A	Can differ for each connection but would typically be the same.	Can differ for each Path message but would typically be the same.
GENERALIZED_LABEL_ REQUEST (LspEncodingType, SwitchingType, GPID)	Encoding Type (Mandatory)	Call/Con	Same for all connections	Same for all related Path messages.
	SwitchingType (Mandatory)	Call/Con		
	Generalized Payload Identifier (Mandatory)	Call/Con		
LABEL_SET	RSVP Specific to help in Source Generalized Label selection (Optional)	Connection	Likely differs for each connection	Likely differs for each related Path message.
ADMIN_STATUS	None – RSVP specific	Connection	Used for teardown of connections	Used for teardown of connections

RSVP Path Msg object	Abstract Attribute	Applicability	Rules Add/Remove Connections	Rules Modify Existing Connection
	(Optional for creation)			
POLICY_DATA	Contract ID (Optional)	Call	Same for all connections	Same for all related Path messages
DESTINATION_TNA (G-UNI)	Destination TNA Name (Mandatory)	Call	Same for all connections	Same for all related Path messages.
SOURCE_TNA (G-UNI)	Source TNA Name (Mandatory)	Call	Same for all connections	Same for all related Path messages.
DIVERSITY (G-UNI)	Diversity (Optional)[5]	Call/Connection	Same for all connections	Same for all related Path messages.
SERVICE_LEVEL (G-UNI)	Service Level (Optional)	Call	Same for all connections	Same for all related Path messages.
EGRESS_LABEL/ SPC_LABEL (G-UNI)	Destination Logical Port Identifier + Destination Generalized Label (Optional)	Connection	Differs for all connections.	Differs for all related Path messages. Logical port identifier is the same, label differs.
SONET/SDH_ SENDER_TSPEC - Signal Type - RCC - NCC - NVC - Multiplier - Transparency - Profile	SONET/SDH Traffic Parameters (Mandatory for SONET/SDH requests)	Call/Connection	Same for all connections or multiplier field differs.	Only the multiplier field is allowed to differ in related Path messages.
G.709_SENDER_TSPEC [RFC4328] - Signal Type - NMC - NVC - MT (multiplier)	G.709 Traffic Parameters (Mandatory for G.709 requests)	Call/Connection	Same for all connections or multiplier field differs..	Only the multiplier field is allowed to differ in related Path messages.
ETHERNET_ SENDER_TSPEC [RFC6003] - Profile - CIR - CBS - EIR - EBS	Ethernet Traffic Parameters (Mandatory for Ethernet requests)	Call/Connection	Only the bandwidth fields are allowed to differ (CIR, CBS, EIR, EBS).	Only the bandwidth fields are allowed to differ (CIR, CBS, EIR, EBS) in related Path messages.
RECOVERY_LABEL	RSVP Specific (Not used in initial request)	N/A	Not used in initial path request.	Not used in initial path request.
UPSTREAM_LABEL	Directionality and Source Generalized Label	Connection	Differs for all connections	Differs for all related Path messages.

[5] It is mandatory to specify the type of diversity required as per [OIF-UNI-02.0-Common]. If the DIVERSITY object is not included in the RSVP message, the type of diversity is "No diversity".

RSVP Path Msg object	Abstract Attribute	Applicability	Rules Add/Remove Connections	Rules Modify Existing Connection
	(Mandatory for bi-directional connections)			
CALL_ID	CALL_ID (Mandatory except for 1st path msg from UNI-C to UNI-N)	Call	Same for all connections	Same for all related Path messages.
SESSION_ATTRIBUTE – - To be used if the bandwidth of a connection can be changed without tearing it down.	RSVP Specific – Indicates that this call uses a reservation style that allows modification of the bandwidth of a connection without tearing down the connection. (Optional)	Call	Not required. Each connection can have different contents if required.	Same for all related Path messages: - Flags = SE Style desired (0x04) if connection's bandwidth can be modified.

Table 4 – RSVP Path Message objects

The Resv message is used for call and connection creation and modification. The content of the Resv message includes the following objects. Table 5 lists the rules associated with connection addition/removal and connection modification.

RSVP Resv Msg object	Abstract Attribute	Applicability	Rules Add/Remove Connections	Rules Modify Existing Connection
INTEGRITY	None – RSVP specific (Optional)	N/A	Minimally recomputed each time the message contents changes.	Minimally recomputed each time the message contents changes.
MESSAGE_ID_ACK/ MESSAGE_ID_NACK	None – RSVP specific (Optional)	N/A	Sent as required.	Sent as required.
MESSAGE_ID	None – RSVP specific (Mandatory)	N/A	Must be different for each connection.	New message id MAY be allocated when Resv message contents has changed after FILTER_SPEC has been added/removed.
UNI_IPv4_SESSION	Local connection	Connection	LSP_ID MUST	The same

RSVP Resv Msg object	Abstract Attribute	Applicability	Rules Add/Remove Connections	Rules Modify Existing Connection
(UNI-C SC PC ID, tunnel id, UNI-N SC PC ID)	ID (Mandatory)		be different for all connections having the same TunnelID.	SESSION object value MUST be used for all related states, whether signaled in the same RESV message or distinct RESV messages.
LSP_TUNNEL_IPv4_ FILTER_SPEC	Local connection ID (Mandatory)	Connection	LSP_ID MUST be different for all connections.	Multiple FILTER_SPEC objects are included, one for each connection.
IPv4_IF_ID_RSVP_HOP	Source Logical Port Identifier (Mandatory)	Connection	Can differ or be the same for connections.	The same RSVP_HOP object value MUST be used for all related states, whether signaled in the same RESV message or distinct RESV messages.
TIME_VALUES	None – RSVP specific (Mandatory)	N/A	Can differ for each connection but would typically be the same.	The same value for TIME_VALUES object value MUST be used for all related states, whether signaled in the same RESV message or distinct RESV messages.
RESV_CONFIRM	None – RSVP specific (Optional)	Connection	Can differ for each connection but would typically be the same. ResvConf messages SHOULD be sent separately if requested on a per connection basis.	ResvConf presence in the Resv message after a bandwidth modification MUST trigger the generation of a new ResvConf message.
ADMIN_STATUS	None – RSVP specific (Optional)	Connection	Used for teardown of connections	Used for teardown of connections
POLICY_DATA	Contract ID (Optional)	Call	Same for all connections	Same for all connections
STYLE	None – RSVP specific	Call	Same for all connections. If	Same for all connections. If SE

RSVP Resv Msg object	Abstract Attribute	Applicability	Rules Add/Remove Connections	Rules Modify Existing Connection
	(Mandatory)		SE style was requested in Session Attribute. FF style or SE style can be used.	style was requested in Session Attribute. SE style SHOULD be used to allow connection modification.
SONET/SDH_ FLOWSPEC [RFC4606] - Signal Type - RCC - NCC - NVC - MT (multiplier) - Transparency - Profile	SONET/SDH Traffic Parameters (Mandatory for SONET/SDH requests)	Call/Connection	Same for all connections or multiplier field differs.	Only the multiplier field is allowed to change.
G.709_FLOWSPEC [RFC4328] - Signal Type - NMC - NVC - MT (multiplier)	G.709 Traffic Parameters (Mandatory for G.709 requests)	Call/Connection	Same for all connections or multiplier field differs.	Only the multiplier field is allowed to change.
ETHERNET_ FLOWSPEC [RFC6003] - Profile - CIR - CBS - EIR - EBS	Ethernet Traffic Parameters (Mandatory for Ethernet requests)	Call/Connection	The bandwidth fields are allowed to differ.	The bandwidth fields are allowed to change.
CALL_ID – UNI 2.0 Attribute	CALL_ID (Mandatory)	Call	Same for all connections	Same for all connections

Table 5 – RSVP Resv Message objects

8.11 Connection Deletion

RSVP allows for deletion of connections using either a single pass PathTear message, or a ResvTear and PathTear message combination. Upon receipt of the PathTear message, a node deletes the connection state and forwards the message. In optical networks, however, it is possible that the deletion of a connection (e.g., removal of the cross-connect) in a node may cause the connection to be perceived as failed in downstream nodes (e.g., loss of frame, loss of light, etc.). This may in turn lead to management alarms and perhaps the triggering of restoration/protection for the connection.

To address this issue, the graceful connection deletion procedure MUST be followed. Under this procedure, an ADMIN_STATUS object with the D-bit set MUST be sent in a Path or Resv message along the connection's path to inform all nodes enroute of the intended deletion, prior to the actual deletion of the connection. A PathErr with an error code /value of 0/0 SHOULD be returned. The procedure is described in [RFC3473] and shown in Figure 10 and Figure 11.

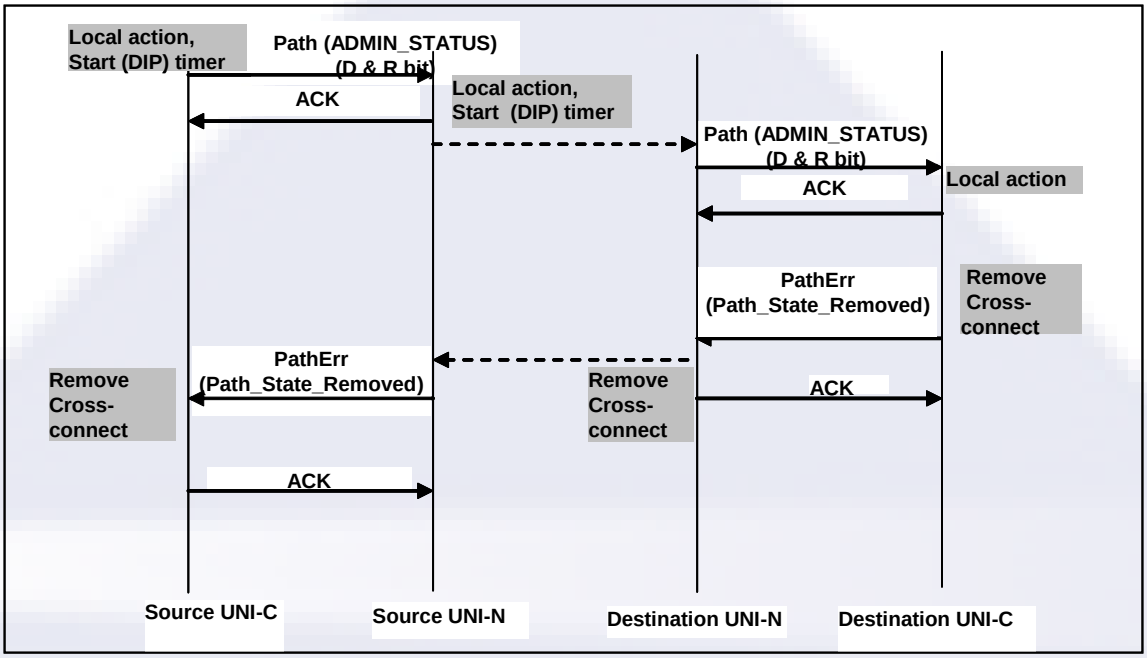


Figure 10 – Connection Teardown Initiated by the Source UNI-C

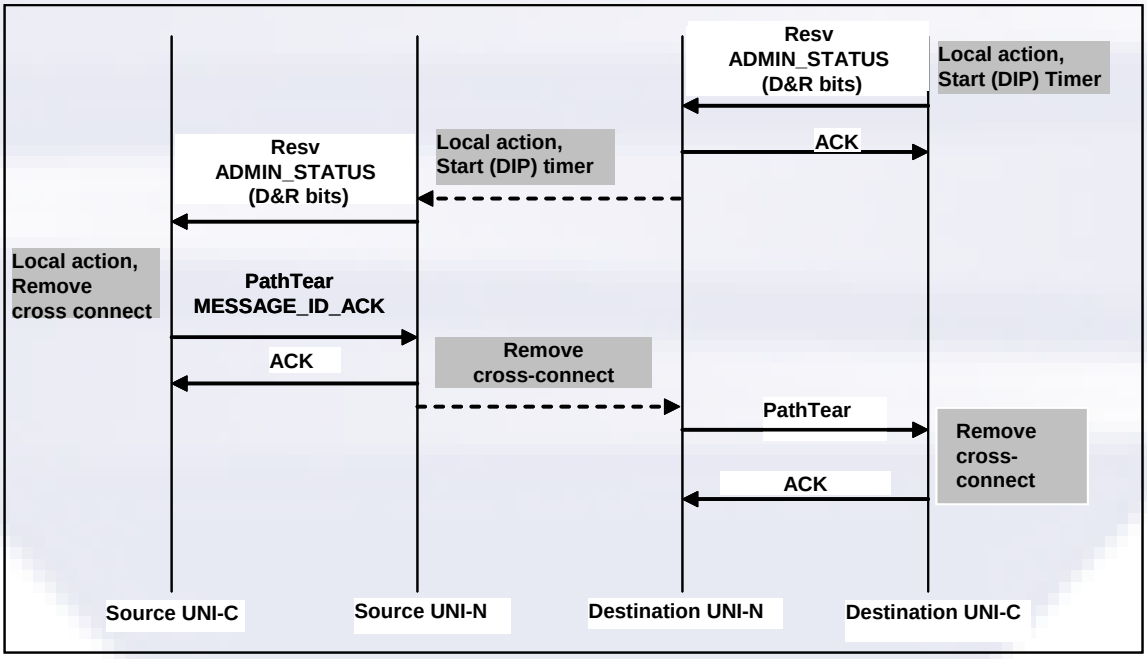


Figure 11 – Connection Teardown Initiated by the Destination UNI-C

The UNI-N MAY trigger connection deletion by sending a Notify message to the UNI-C that includes the ADMIN_STATUS object with the D bit set. This is shown in Figure 12 and Figure 13. The UNI-C receiving the message SHOULD initiate the graceful deletion procedure illustrated in Figure 12 and Figure 13. If a source UNI-C does not respond to a network initiated graceful deletion Notify message, the network MUST continue the message flow illustrated below the dashed line in Figure 12 and Figure 13.

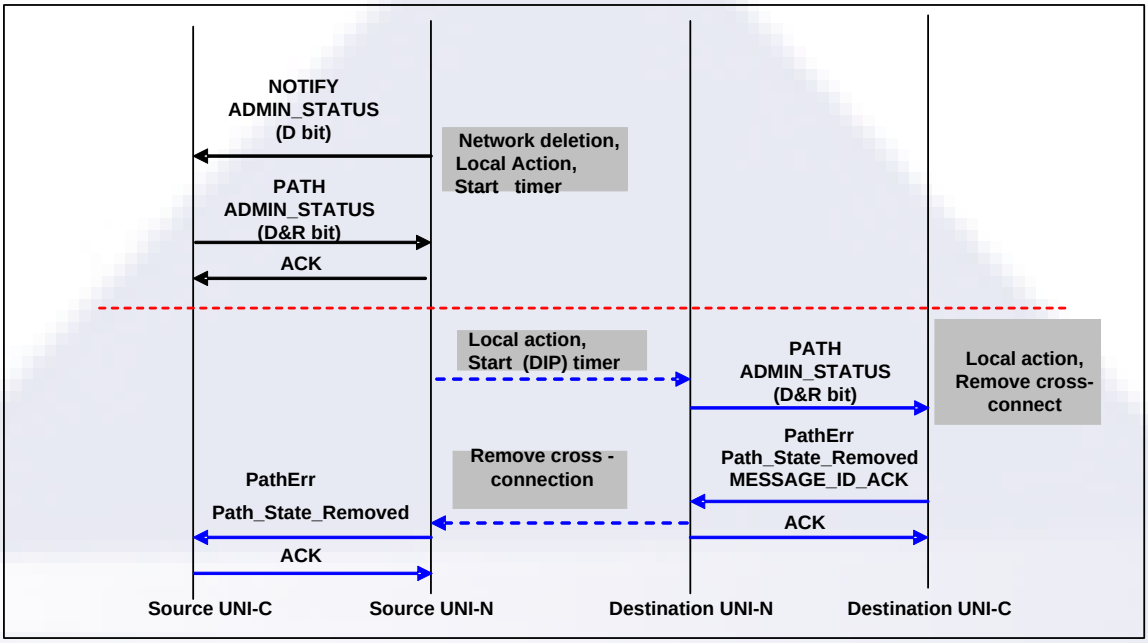


Figure 12 - Connection Teardown Initiated by the Source UNI-N

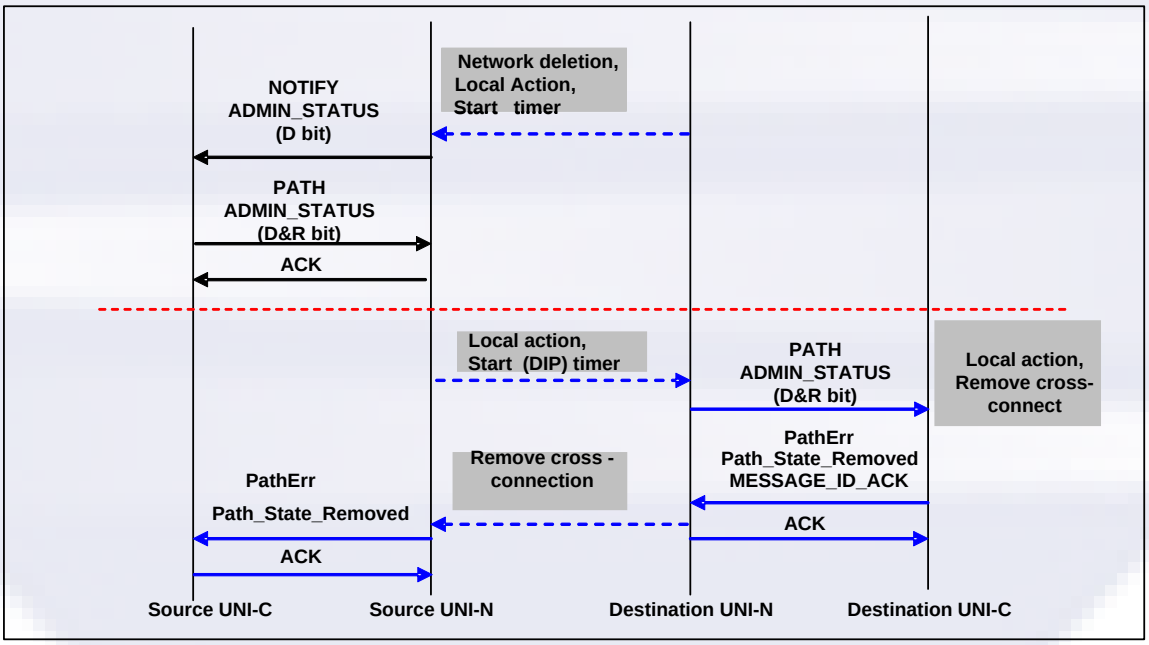


Figure 13 - Connection Teardown Initiated by Destination UNI-N

8.12 Forced Deletion

The forced deletion procedure SHOULD be used to handle scenarios that require unilateral deletion. In general this occurs due to a network-generated event that requires the connection to be deleted. For example:

- Internal network failures, which force the network to terminate connections.

- When the "Deletion In Progress" timer of an ADMIN_STATUS object expires.

A UNI-N initiates a forced deletion by sending a PathErr toward the source UNI-C and, simultaneously, a PathTear toward the destination UNI-C. The PathErr message sent to the source UNI-C has the "Path_State_Removed" flag to indicate that the path state is deleted. In this case, a PathTear from the source UNI-C is not required to terminate the connection; in fact, such a PathTear would be discarded (but acknowledged) since Path state will have already been removed. The message flow for the forced deletion procedure is illustrated in Figure 14. This flow reflects the fact that connection tear-down can be initiated by a UNI-N unilaterally, without the consent of the other party.

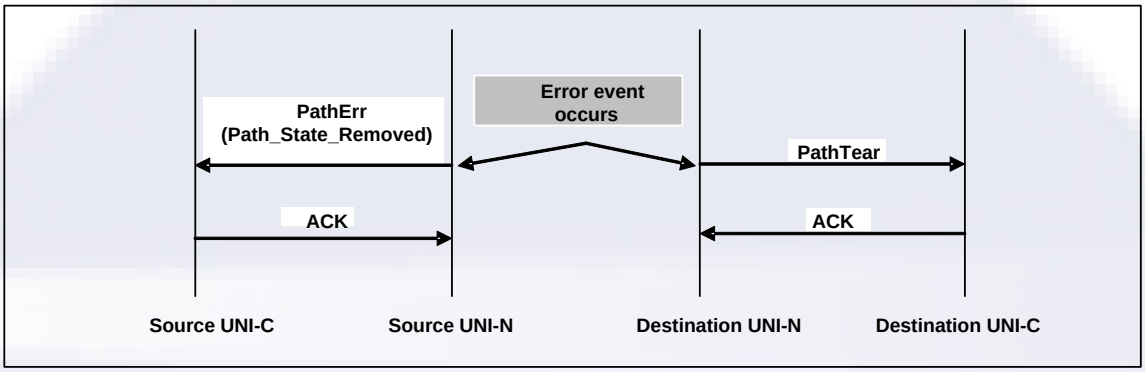


Figure 14 - Forced Connection Tear-Down by the Network

To initiate a forced deletion, a source UNI-C sends a PathTear message to the source UNI-N and a destination UNI-C sends a PathErr with Path_State_Removed flag to the destination UNI-N. This is illustrated in Figure 15 and Figure 16. A UNI-C node SHOULD initiate a forced deletion only under one of the following error conditions:

- When the "Deletion In Progress" timer expires.
- No response to connection setup or connection modify request is received.
- Destination UNI-C does not receive a confirmation for a connection setup or modify indication

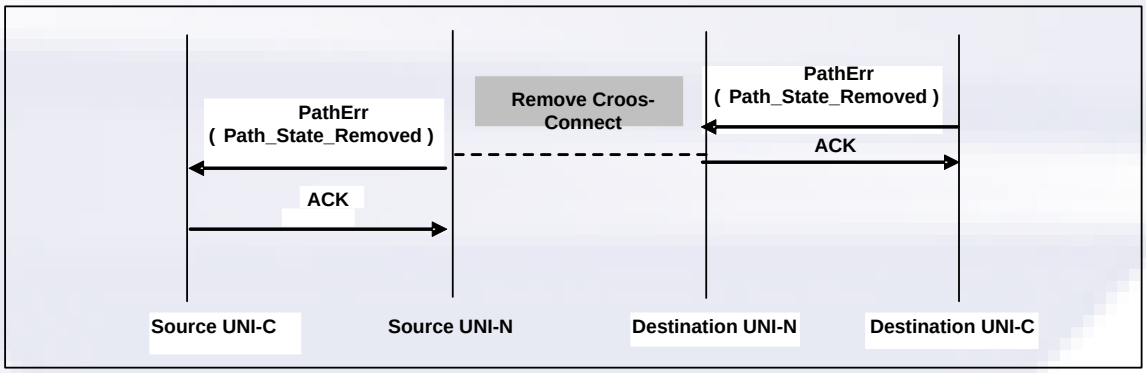


Figure 15 - Forced Deletion Initiated by Destination UNI-C

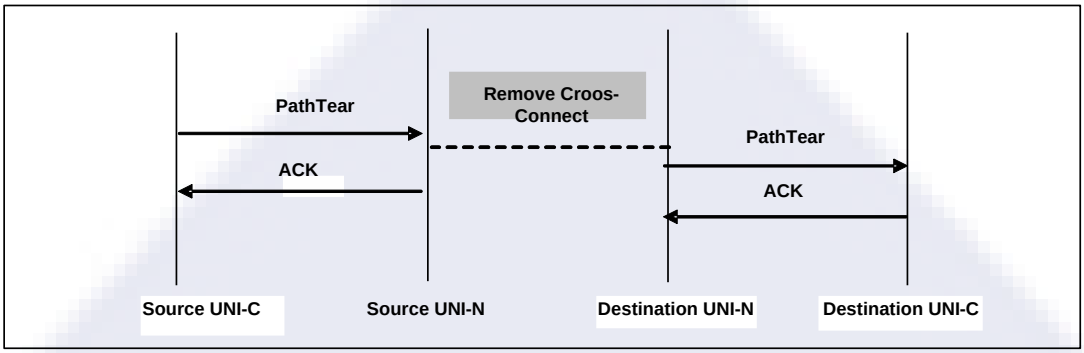


Figure 16 - Forced Deletion Initiated by Source UNI-C

8.13 Connection Status Enquiry And Response

The RSVP protocol periodically exchanges refresh messages to synchronize connection states between adjacent UNI-N and UNI-C nodes. This can be viewed as a continual query and response process that keeps the states on both nodes synchronized. There is, therefore, no need for explicit query and response messages to find out a neighbor's connection state.

8.14 Signaling Channel Failure Detection and Recovery

Under the RSVP protocol, signaling messages relating to a data link are sent on the same link and the failure of a link or the control plane results in the eventual deletion of reservations made on the link. This, however, SHOULD NOT be the case under UNI signaling where the control plane is separate from the data plane and where the failure of the control plane does not imply data plane failure. The handling of control state failure (without loss of the forwarding state) and UNI-C to UNI-N signaling adjacency failure is described in [RFC3473] through the support of the RESTART_CAP object, which requires the use of Hello messages. Here, in particular, the failure of a signaling channel or control protocol entities MUST NOT result in the deletion of previously established connections. Connections in the process of being deleted prior to or during the failure MAY be deleted after the recovery.

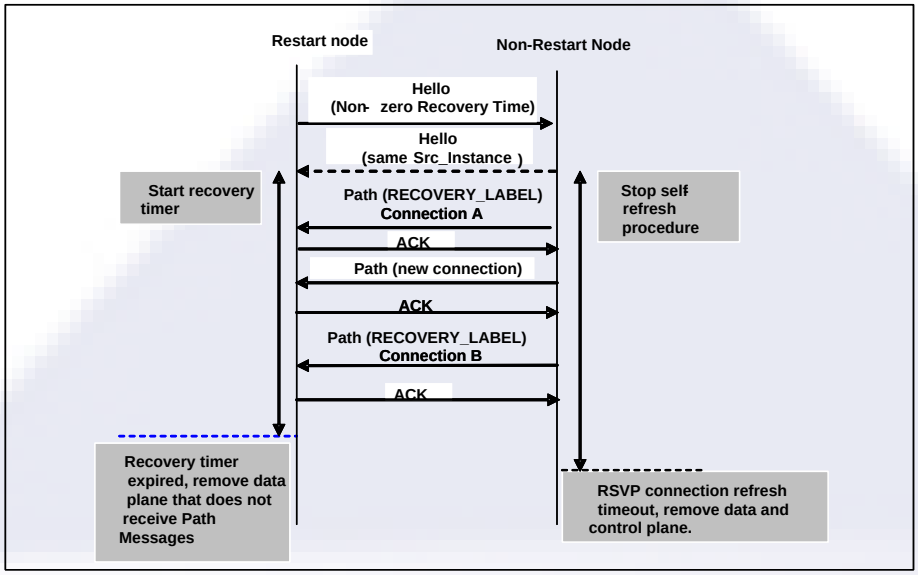


Figure 17 - Recovery from Node Failure

To address this requirement, a node MUST support the fault handling procedure described in Section 9 of [RFC3473], illustrated in Figure 17 and Figure 18. A node MUST NOT accept any RSVP request unless a successful signaling adjacency is established in order to ensure RSVP Hellos are supported. A received connection related message from a node that has not sent out any Hello message yet MAY be used as a trigger to initiate the Hello procedure. A node SHOULD respond to unexpected or erroneous Hello messages by setting the Dst_Instance to 0 in the Hello Request or HelloAck object, indicating that the received message is not accepted. Figure 17 illustrates the message flow in the case of recovery from a node failure (i.e., complete failure of the control plane). Figure 18 illustrates the message flow, based on Srefresh messages, between a pair of adjacent nodes in the case of recovery from signaling channel failures (i.e., where the signaling protocol entities did not fail).

The UNI-C and UNI-N MUST set the Restart Time to 0xFFFFFFFF in the RESTART_CAP object of the Hello Messages. This indicates that the restart of the sender's control plane may occur over an indeterminate interval and that the operation of its data plane is unaffected by control plane failures. The UNI-C and UNI-N MUST set the Recovery Time to a value greater than 0, i.e. the UNI-N and UNI-C MUST preserve data plane state during restarts and control plane failures. The "self-refresh" procedure in Figure 17 and Figure 18 refers to the behavior of a UNI-C or a UNI-N node that acts as if it is receiving periodic RSVP refresh messages from the neighbor during the failure situation. A UNI-C or a UNI-N stops the self-refresh behavior when the RSVP adjacency is re-established and resynchronizes the state during the Recovery period that is determined by the Recovery Time exchanged in the RESTART_CAP object of the Hello Messages. When the recovery period is completed, any connection not refreshed by the neighbor is deleted, and corresponding control and data planes are removed. On the UNI-N, this includes the release of any remaining network resources. This allows any deletion triggered by the neighbor during the control plane failure to complete end-to-end.

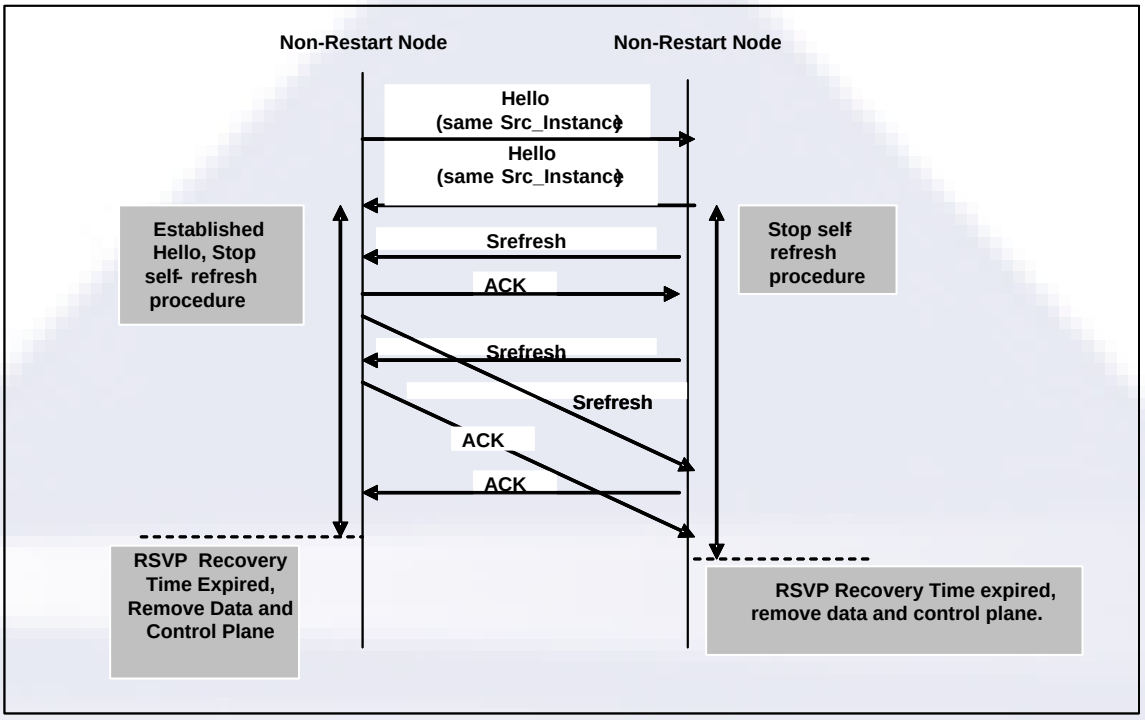


Figure 18 - Recovery from Signaling Channel Failure

8.15 Data Plane Failure and Recovery

The data plane provides rich features for failure detection. For example, the SONET/SDH overhead bytes allows a SONET/SDH node to detect transport layer failure. Recovery MAY be attempted at the node that detects the failure.

To allow different restoration and protection schemes within the network, a node SHOULD NOT delete a connection upon detecting a fault, but SHOULD continue to accept and send RSVP refresh messages until it receives explicit tear-down messages or the connection state times out.

9 RSVP Messages And Objects For UNI Signaling

This section describes the specific usage and procedures of RSVP/GMPLS RSVP-TE objects and messages that apply to UNI 2.0. Only objects, messages and behavior that are not already captured in standard IETF specifications, or for which additional details are necessary, are described in this document. The standard behavior is captured in relevant IETF documents as referenced.

Table 6 and Table 7 show the complete list of messages and objects supported by UNI 2.0. RSVP trigger messages and ResvConf message have end-to-end significance and the network MUST relay these messages from the source UNI to the destination UNI and vice versa. Also, some objects MUST maintain the same value at the source and destination UNI-C. These are marked as end-to-end significant in Table 7. The particular approach used to trigger Srefresh message based refreshes at both source and destination UNI is implementation specific (see [RFC2961]).

RSVP Messages	Direction	Message Type	Reference
Path	(Source) UNI-C→ UNI-N & (Dest) UNI-N → UNI-C	1	Section 9.1.3
PathTear	(Source) UNI-C→ UNI-N & (Dest) UNI-N → UNI-C	5	Section 9.1.5
PathErr	(Source) UNI-N→ UNI-C & (Dest) UNI-C → UNI-N	3	Section 9.1.4
Resv	(Source) UNI-N→ UNI-C & (Dest) UNI-C → UNI-N	2	Section 9.1.6
ResvErr	(Source) UNI-C→ UNI-N & (Dest) UNI-N → UNI-C	4	Section 9.1.8
ResvTear	(Source) UNI-N → UNI-C & (Dest) UNI-C→ UNI-N &	6	Section 9.1.9
ResvConf	(Source) UNI-C→ UNI-N & (Dest) UNI-N → UNI-C	7	Section 9.1.7
Hello	UNI-N ↔UNI-C	20	Section 9.1.2
Ack	UNI-N ↔ UNI-C	13	[RFC2961]
Srefresh	UNI-N ↔ UNI-C	15	Section 9.1.10
Bundle	UNI-N ↔ UNI-C	12	[RFC2961]
Notify	(Source) UNI-N → UNI-C	21	Section 9.1.11

Table 6 – RSVP Messages Supported Under UNI 2.0

RSVP Object or Sub-Objects	End-to-End Significant?	C-Num/ C-Type [/ Type [/ Sub-type]]	Reference
ACCEPTABLE _LABEL_SET	NO	130/1	[RFC3473], [RFC4328], Section 9.2.1
ADMIN_STATUS	YES	196/1	[RFC3473]
CALL_ID	YES	230/<0,1,2>	[RFC3474]
SONET/SDH_FLOWSPEC	YES	9/4	Section 9.2.13
G.709 FLOWSPEC	YES	9/5	[RFC4328]
ETHERNET FLOWSPEC	YES	9/6	[RFC6003]
LSP_TUNNEL_IPv4_FILTER_SPEC	NO	10/7	Section 9.2.9
GENERALIZED_LABEL	NO	16/<2,4,5>	[RFC3473], [RFC4328], Section 9.2.1
GENERALIZED_LABEL_REQUEST	YES	19/<4,5>	Section 9.2.6, [RFC3473] and [RFC6004]

RSVP Object or Sub-Objects		End-to-End Significant?	C-Num/ C-Type [/ Type [/ Sub-type]]	Reference
GENERALIZED_ UNI_ATTRIBUTES	SOURCE _TNA	YES	229/1/1/<1,2,3>	Section 9.2.5.1 and [RFC3476]
	DESTINATION_TNA	YES	229/1/2/<1,2,3>	Section 9.2.5.5 and RFC3476]
	DIVERSITY	NO	229/1/3/1	Section 9.2.5.9 and [RFC3476]
	EGRESS_LABEL/ SPC_LABEL	NO	229/1/4/<1,2>	Section 9.2.5.10 and [RFC3476] [RFC3473], [RFC4328] and Section 9.2.1
	SERVICE_LEVEL	YES[6]	229/5	Section 9.2.5.11 and [RFC3476],
HELLO_REQUEST		NO	22/1	[RFC3473]
HELLO ACK		NO	22/2	[RFC3473]
INTEGRITY		NO	4/1	Section 9 of [OIF-UNI-02.0-Common]
IPv4_ERROR_SPEC, error code and value IF_ID ERROR_SPEC		YES	6/<1,3>	Section 9.2.8 [RFC3473]
IPv4_IF_ID_RSVP_HOP		NO	3/3	Section 9.2.11
IPv4_RESV_CONFIRM		NO	15/1	Section 9.2.7
LABEL_SET		NO	36/1	[RFC3473] [RFC4328] Section 9.2.1
MESSAGE ID		NO	23/1	[RFC2961]
MESSAGE ID_ACK		NO	24/1	[RFC2961]
MESSAGE ID_NACK		NO	24/2	[RFC2961]
MESSAGE ID_LIST		NO	25/1	[RFC2961]
POLICY_DATA		YES[7]	14/1	Section 9 of [OIF-UNI-02.0-Common]
RECOVERY_LABEL		NO	34/2	[RFC3473]/ [RFC4328]/ Section 9.2.1
RESTART_CAP		NO	131/1	[RFC3473]
LSP_TUNNEL_IPv4_SENDER_TEMPLATE		NO	11/7	Section 9.2.1
SONET/SDH_SENDER_TSPEC		YES	12/4	Section 9.2.13
G.709 SENDER_TSPEC		YES	12/5	[RFC4328]
ETHERNET_SENDER_TSPEC		YES	12/6	[RFC6003]
STYLE		YES	8/1	[RFC2205]
TIME_VALUE		NO	5/1	[RFC2205]
UNI_IPv4_SESSION		NO	1/11	Section 9.2.4
UPSTREAM_LABEL		NO	35/2	[RFC3473], [RFC4328], Section 9.2.1
SESSION_ATTRIBUTE		YES	207/7	[RFC3209]
NOTIFY_REQUEST		NO	195/1	Section 9.2.15

Table 7 – Summary of UNI RSVP Objects

[6] This parameter is processed by the network but not transmitted to the destination UNI-C.
[7] This parameter is processed by the network but not transmitted to the destination UNI-C.

9.1 RSVP Messages for UNI Signaling

9.1.1 RSVP Common Message Header

The flag field of the RSVP common header MUST be set to 1, to indicate support of the Bundle and Srefresh messages. As a result, an implementation MUST be able to process Bundle and Srefresh messages received from a neighbor.

9.1.2 Hello Message (Msg Type = 20 [RFC3209])

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

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

Hello messages are retransmitted periodically to an adjacent UNI signaling peer. The retransmission interval SHALL be administratively configurable. The default value is 5 seconds .

9.1.3 Path Message (Msg Type = 1 [RFC2205])

The Path message is used for connection creation, call and connection modification, graceful connection teardown and failure recovery. The format of the Path message is as follows:

<Path Message> ::=
<Common Header>
[<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<UNI_IPv4_SESSION> <IPv4_IF_ID_RSVP_HOP>
<TIME_VALUES>
<GENERALIZED_LABEL_REQUEST>
<CALL_ID>
[<LABEL_SET> ...]
[<SESSION_ATTRIBUTE>]
<NOTIFY_REQUEST>[8]
[<ADMIN_STATUS>]
<Generalized UNI>
[<POLICY_DATA> ...]
<sender descriptor>

<Generalized UNI> ::=
<Common Object Header>
<DESTINATION_TNA>
<SOURCE_TNA>
[<DIVERSITY> …][9]
[<SERVICE_LEVEL>]

[8] The NOTIFY_REQUEST is mandatory in Path messages generated by the UNI-C but not required in Path messages generated by the UNI-N.
[9] It is mandatory to specify the type of diversity required as per [OIF-UNI-02.0-Common]. If the DIVERSITY object is not included in the RSVP message, the type of diversity is “No diversity”.

[<EGRESS_LABEL> [<EGRESS_LABEL>]] | [<SPC_LABEL> [<SPC_LABEL>]]

The format of the sender descriptor for a unidirectional connection is:

For SONET/SDH calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <SONET/SDH_SENDER_TSPEC>
[< RECOVERY_LABEL >]

For G.709 calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <G.709 SENDER_TSPEC>
[< RECOVERY_LABEL >]

For Ethernet calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE><ETHERNET_SENDER_TSPEC>
[< RECOVERY_LABEL >]

The format of the sender descriptor for a bi-directional connection (default under UNI 2.0) is:

For SONET/SDH calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <SONET/SDH_SENDER_TSPEC>
<UPSTREAM_LABEL> [< RECOVERY_LABEL >]

For G.709 calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <G.709 SENDER_TSPEC>
<UPSTREAM_LABEL> [< RECOVERY_LABEL >]

For Ethernet calls
<sender descriptor> ::=
<LSP_TUNNEL_IPv4_SENDER_TEMPLATE><ETHERNET_SENDER_TSPEC>
<UPSTREAM_LABEL> [< RECOVERY_LABEL >]

9.1.4 PathErr Message (Msg Type = 3 [RFC2205])

The PathErr message is used to report errors and for connection deletion.

The format of the UNI PathErr message is shown as follows:

<PathErr message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<UNI_IPv4_SESSION>
<CALL_ID>
<IPv4_ERROR_SPEC>
[<ACCEPTABLE_LABEL_SET>]
[<POLICY_DATA> …]
<sender descriptor>

9.1.5 PathTear Message (Msg Type = 5 [RFC2205])

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

```
<PathTear Message> ::= <Common Header> [ <INTEGRITY> ]
        [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
        <MESSAGE_ID>
        <UNI_IPv4_SESSION>
        <CALL_ID>
        <IPv4_IF_ID_RSVP_HOP>
        <sender descriptor>

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

9.1.6 Resv Message (Msg Type = 2 [RFC2205])

The Resv message is used for connection creation and call/connection modification. If the RESV_CONFIRM object is inserted, it MUST be included in all future full Resv refreshes.

The format of the UNI Resv message is as shown below:

```
<Resv Message> ::= <Common Header> [ <INTEGRITY> ]
        [  [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
        <MESSAGE_ID>
        <UNI_IPv4_SESSION>  < IPv4_IF_ID_RSVP_HOP >
        <TIME_VALUES>
        <CALL_ID>
        [ <IPv4_RESV_CONFIRM> ]
        <NOTIFY_REQUEST>[10]
        [ <ADMIN_STATUS>]
        [ <POLICY_DATA> ... ]
        <STYLE>
        <FF flow descriptor> | <SE flow descriptor>

<FF flow descriptor> ::=
        <SONET/SDH_FLOWSPEC> | <G.709 FLOWSPEC> | <ETH FLOWSPEC>
        <LSP_TUNNEL_IPv4_FILTER_SPEC>
        <GENERALIZED_LABEL>

<SE flow descriptor> ::=
        <SONET/SDH_FLOWSPEC> | <G.709 FLOWSPEC> | <ETH FLOWSPEC>
        <filter spec list>

<filter spec list> ::= [<filter spec list>]
        <LSP_TUNNEL_IPv4_FILTER_SPEC>
        <GENERALIZED_LABEL>
```

[10] The NOTIFY_REQUEST is mandatory in Resv messages generated by the UNI-C but not required in Resv messages generated by the UNI-N.

9.1.7 ResvConf Message (Msg Type = 7 [RFC2205])

The ResvConf message MUST be sent downstream from the source UNI-C to acknowledge the receipt of a trigger[11] Resv message that includes a RESV_CONFIRM Object. Specifically, under UNI 2.0, ResvConf messages are sent from the source UNI-C to the corresponding UNI-N, and from the destination UNI-N to the destination UNI-C. The use of the RESV_CONFIRM object in the Resv is not required. However, if such an object is received, the receiver MUST generate a ResvConf message in response. The network MUST relay the ResvConf message from source UNI-N to destination UNI-N. The CALL_ID is not included in the ResvConf message, in line with [RFC3474]. As a result, the attributes are not inline with the abstract Connection Setup Confirm message. It is possible to correlate the ResvConf message with the proper Resv state based on the UNI_IPv4_SESSION and flow descriptor.

The format of the UNI ResvConf message is shown below:

```
<ResvConf message> ::= <Common Header> [ <INTEGRITY> ]
        [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
        <MESSAGE_ID>
        <UNI_IPv4_SESSION> <IPv4_ERROR_SPEC>
        <IPv4_RESV_CONFIRM>
        <STYLE>
        <FF flow descriptor > | <SE flow descriptor>
```

9.1.8 ResvErr Message (Msg Type = 4 [RFC2205])

The ResvErr message is used for reporting reservation errors. For example, it MAY be generated if a Resv message is received but no corresponding Path state can be found. The format of the UNI ResvErr message is as follows:

```
<ResvErr message> ::= <Common Header> [ <INTEGRITY> ]
        [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
        <MESSAGE_ID>
        <UNI_IPv4_SESSION> <IPv4IF_ID_RSVP_HOP>
        <IPv4_ERROR_SPEC>
        [ <ACCEPTABLE_LABEL_SET> ]
        [ <POLICY_DATA> ... ]
        <STYLE>
        <FF flow description> | <SE flow descriptor>
```

9.1.9 ResvTear Message (Msg Type = 6 [RFC2205])

The ResvTear message is supported by UNI 2.0 to be compatible with RSVP. A UNI 2.0 implementation MUST NOT generate ResvTear messages but it MUST be able to process this message if it is received as it may be used to trigger a forced deletion by a UNI 1.0 implementation. The format of the UNI ResvTear message is shown below:

```
<ResvTear Message> ::=
        <Common Header> [ <INTEGRITY> ]
        [ [ <MESSAGE_ID_ACK> | <MESSAGE_ID_NACK> ] ... ]
        <MESSAGE_ID>
        <UNI_IPv4_SESSION> <IPv4_IF_ID_RSVP_HOP>
```

[11] A trigger Resv message is a message that modifies the reservation state. Examples include the original Resv message sent during connection establishment in response to the first Path message and the Resv message that includes a change in the FILTER_SPEC object sent during connection modification.

<STYLE>
<FF flow descriptor > | <SE flow descriptor>

9.1.10 Srefresh Message (Msg Type = 15 [RFC2961])

If Srefresh is used, periodically sending full Path and Resv refresh messages is RECOMMENDED. The Srefresh interval SHOULD be smaller than or equal to the smallest refresh interval for all Path and Resv states being refreshed. 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>
[<srefresh list>]

9.1.11 Notify Message (Msg Type = 21 [RFC3473])

The format of the UNI Notify Message is shown below. In UNI 2.0, the Notify Message is only supported from the source UNI-N to the source UNI-C to indicate that graceful deletion should be initiated by the source UNI-C.

<Notify message> ::= <Common Header> [<INTEGRITY>]
[[<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ...]
<MESSAGE_ID>
<ERROR_SPEC>
<notify session lists>

<notify session lists> ::= [<notify session lists>]
<upstream notify session>

<upstream notify session> ::= <SESSION> <CALL_ID> <ADMIN_STATUS>
<sender descriptor>

9.2 UNI RSVP Objects Format

This section describes the RSVP objects that require specific behavior.

9.2.1 Label Related Objects

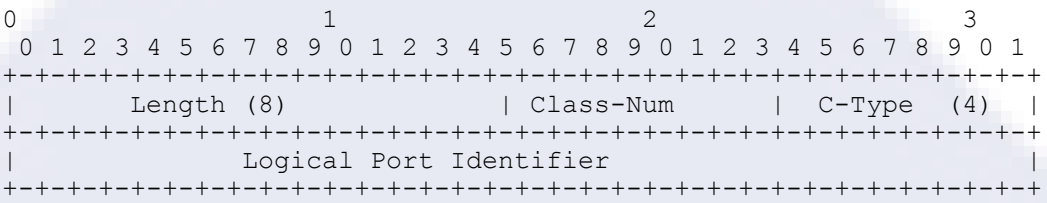
The following OBJECTs carry label information:

- ACCEPTABLE_LABEL_SET
- GENERALIZED_LABEL
- LABEL_SET
- RECOVERY_LABEL
- UPSREAM_LABEL
- EGRESS_LABEL sub-object of GENERALIZED_UNI_ATTRIBUTES
- SPC_LABEL sub-object of GENERALIZED_UNI_ATTRIBUTES

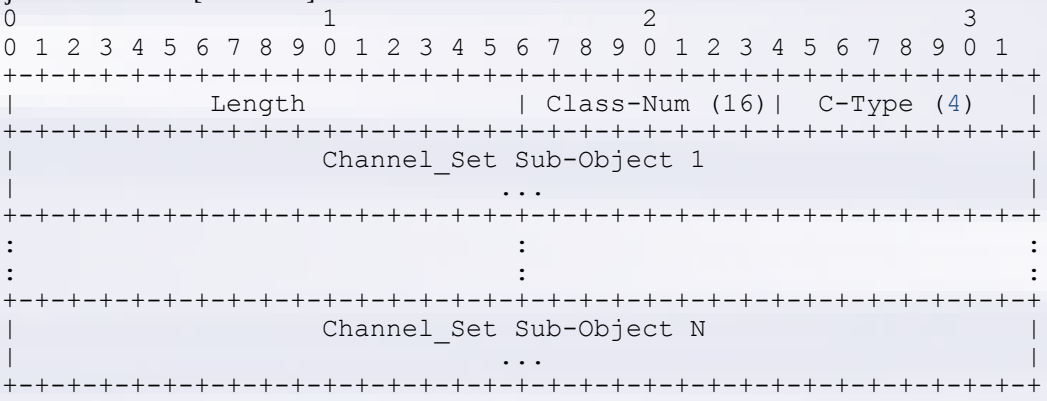
The format of the label differs based on the type of connection requested. For SONET/SDH connections, the label format is described in [RFC4606]. For G.709 connections, the label format is described in [RFC4328].

In case of signal multiplication, i.e. the multiplier field of the SONET/SDH SENDER_TSPEC or G.709 SENDER_TSPEC is greater than one, an ordered list of labels MUST appear in the label object.

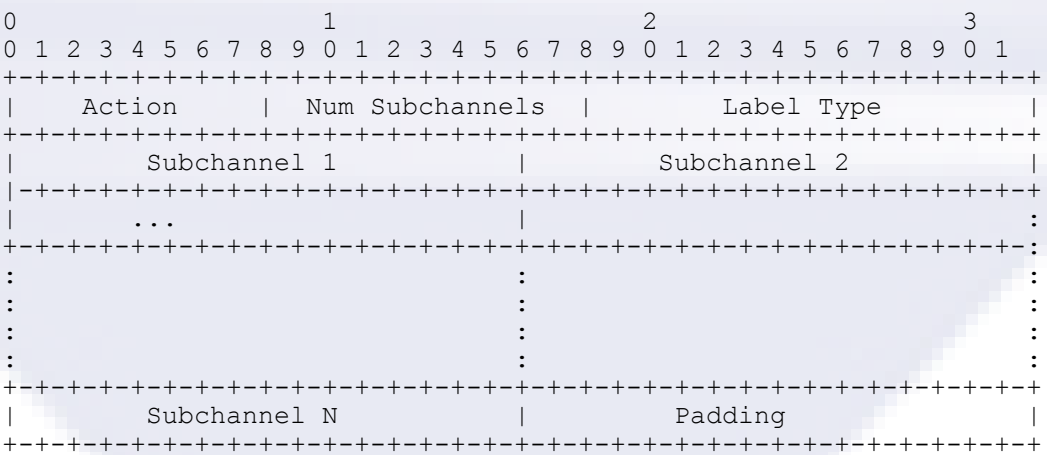
For Ethernet Private Line (EPL) connections, the label format is as follows where the Class Number varies for the type of label, i.e. the RECOVERY_LABEL Class Number is 34 and the ACCEPTABLE_LABEL_SET Class Number is 130.



For Ethernet Virtual Private Line (EVPL) connections, the label format is the Generalized Channel Set Label Object as defined in [RFC6002] and included below:



The Channel_Set Sub-Object size is measured in bytes and MUST always be a multiple of 4, and at least 4, and has the following format:



Action: 8 bits

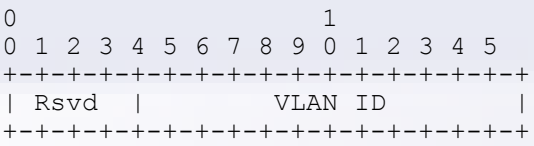
- See [RFC3471] for definition of actions. Range actions SHOULD be used when possible to minimize the size of the Channel_Set LABEL Object.

Number of Subchannels: 10 bits

- Indicates the number of subchannels carried in the sub-object. When the number of subchannels required exceeds the limit of the field, i.e., 1023, multiple Sub-Objects MUST be used. A value of zero (0) has special meaning and MUST NOT be used except in the UPSTREAM_LABEL object.
- A value of zero (0) is used in an UPSTREAM_LABEL object to indicate that the subchannel(s) used in the upstream direction MUST match the subchannel(s) carried in the LABEL object. When value of zero (0) is used, no Subchannels are included in the Channel_Set Sub-Object and only one Channel_Set Sub-Object may be present.

Label Type: 14 bits

See [RFC3473] for a description of this field. It MUST be set to Generalized Label, i.e. 2. The format for the Generalized Label used with EVPL services is defined in [RFC6004]:



Reserved: 4 bits

- This field is reserved. It MUST be set to zero on transmission and MUST be ignored on receipt. These bits SHOULD be passed through unmodified by transit nodes.

VLAN ID: 12 bits

- CE-VLAN identifier.

Subchannel: 16 bits

- See [RFC3471] for a description of this field. For UNI 2.0, this field contains the Generalized Label for EVPL services described above.

Padding: Variable

- Padding is used to ensure that the length of a Channel_Set Sub-Object meets the multiple of 4 byte size requirement. The field is only required when the Subchannel field is not 32 bit aligned and the number of included Subchannel fields result in the Sub-Object not being 32 bit aligned.
- The Padding field MUST be included when the number of bits represented in all the Subchannel fields included in a Generalized Channel_Set Sub-Object result in the Sub-Object not being 32 bit aligned. When present, the Padding field MUST have a length that results in the Sub-Object being 32 bit aligned. When present, the Padding field MUST be set to a zero (0) value on transmission and MUST be ignored on receipt. These bits SHOULD be passed through unmodified by transit nodes.

For simplicity in management, a single LSP SHOULD be used for each EVPL connection whose Path and Resv messages fit within a single unfragmented IP packet. This allows the reuse of all standard LSP modification procedures. Of particular note is the modification of the CE-VLAN IDs associated with the Ethernet connection. Specifically, when a single LSP is used to support an EVPL connection, make-before-break procedures, see [RFC3209], SHOULD be used to modify the Channel_Set LABEL object.

Multiple LSPs MAY be used to support an EVPL service connection. All such LSPs MUST be established within the same call. The primary purpose of multiple LSPs is to support the case where the related objects result in a Path message being larger than a single unfragmented IP packet.

When using multiple LSPs, all LSPs associated with the same call / EVPL connection MUST be signaled with the same LSP objects with the exception of the SENDER_TEMPLATE, SESSION and label related objects. Each LSP has its own Tunnel ID as different SESSION objects are used. The LSP_ID in the SENDER_TEMPLATE is reserved for make-before-break procedures. The CALL_ID is used to identify all LSPs belonging to the same EVPL call/connection. The CALL_ID is null when the source UNI-C sends an initial Path message. The UNI-C has to wait for the Resv message that includes a valid CALL_ID before sending future Path messages for the remaining CE-VLAN IDs. All such LSPs SHOULD share resources. When using multiple LSPs, CE-VLAN IDs MAY be added to the EVPL connection using either a new LSP or the make-before-break procedures mentioned in the previous section. Make-before-break procedures on individual LSPs SHOULD be used to remove CE-VLAN IDs.

To modify the bandwidth, it is necessary to resignal all LSPs associated with the call make-before-break procedures.

9.2.2 Label Set Object

The label set object MUST NOT be used for EPL services as the label can be inferred from the RSVP_HOP object. The label set MAY be used for EVPL, SONET/SDH and OTN to enforce the use of a particular label in the downstream direction.

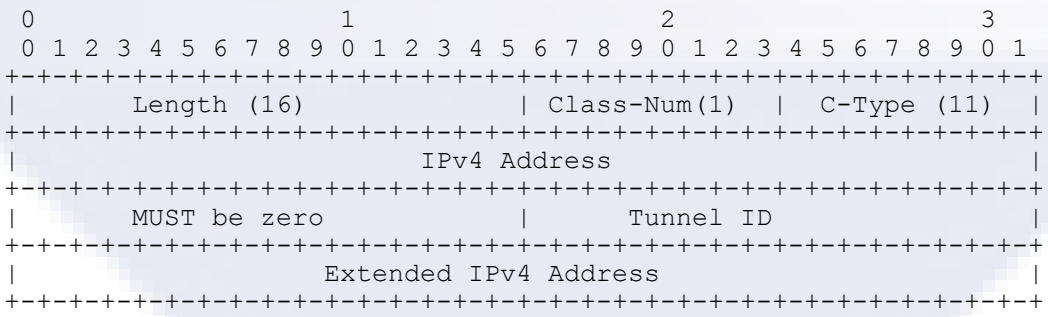
9.2.3 LSP_TUNNEL_IPv4_SENDER_TEMPLATE Object (Class-Num = 11 [RFC3209])

The format of the LSP_TUNNEL_IPv4_SENDER_TEMPLATE object is defined in [RFC3209]. For UNI 2.0, the IPv4 Tunnel sender address MUST be set to the source UNI-C's SC PC ID at the source UNI and MUST be set to the destination UNI-N's SC PC ID at the destination UNI. The LSP ID is assigned by the sender of the Path message. The LSP ID changes when the connection is modified as described in Section 8.10.2 but otherwise it MUST remain constant during the life of a connection.

9.2.4 UNI_IPv4_SESSION Object (Class-Num = 1 [RFC3476])

UNI_ IPv4_SESSION Object [RFC3476] has the following format:

- UNI_ IPv4_SESSION object: Class = 1, C-Type = 11



IPv4 Address: This MUST be set to the source UNI-N's SC PC ID at the source UNI and it MUST be set to the destination UNI-C's SC PC ID at the destination UNI.

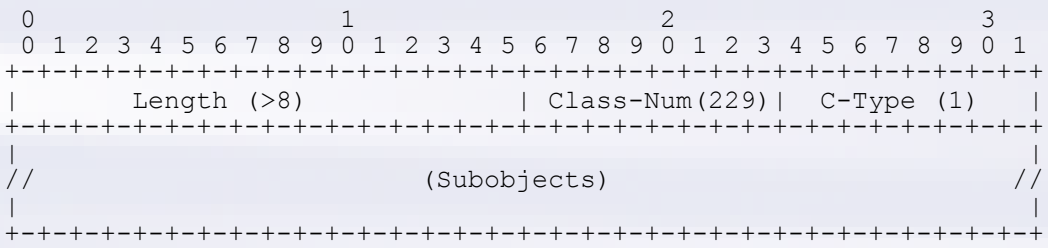
Tunnel ID: A 16-bit identifier, assigned by the sender of the Path message. This ID remains constant during the life of a connection. The Tunnel ID is the only way to distinguish between single hop LSPs established between the same UNI-C/UNI-N pair if the Source/Destination TNA name is not part of their description.

Extended IPv4 address: This MUST be set to the SC PC ID of source UNI-C at the source UNI, and it MUST be set to the SC PC ID of the destination UNI-N at the destination UNI.

The combination of the LSP_TUNNEL_IPv4_SENDER_TEMPLATE object and UNI_IPv4_SESSION object MUST uniquely identify a connection at a local UNI and remain unmodified for the duration of the connection. An unrecognized connection ID SHOULD result in an error message with error code “Routing Problem: Invalid/Unknown Connection ID”.

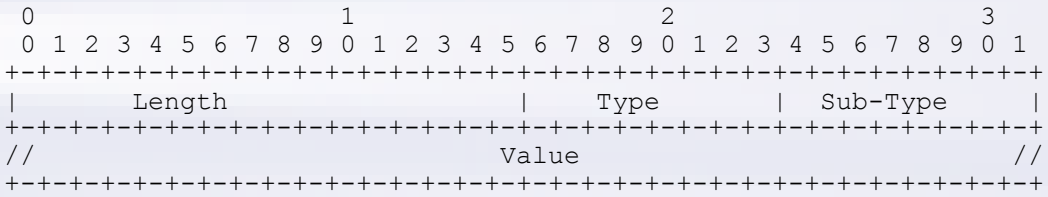
9.2.5 GENERALIZED_UNI Object (Class-Num=229) [RFC3476]

UNI specific attributes are specified via the GENERALIZED_UNI object. The GENERALIZED_UNI object has the following format:



Subobjects:

The contents of a GENERALIZED_UNI object are a series of variable-length data items. The common format of the sub-objects is shown below:



For future compatibility, the Type and Sub-Type are assigned according to the rules pertaining to RSVP objects, but from their own number space. The treatment of future Type and Sub-Type is the same as specified for RSVP Class-Num and C-Type, respectively. If an error message is to be sent due to unrecognized Type or SubType, a node SHOULD use the error code “unknown Class-Number” or “unknown C-Type with known Class-Number” and the error value set to Class-Num and C-Type of GENERALIZED_UNI object.

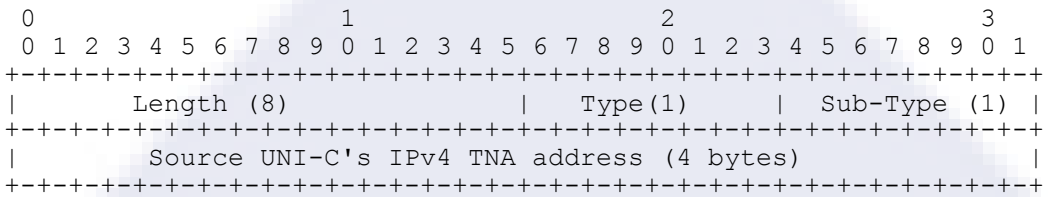
When the source or destination address specified cannot be serviced due to policy reasons, an error message with the error code “Policy control failure: Unauthorized sender,” or “Policy control failure: Unauthorized receiver”, respectively.

9.2.5.1 Source TNA Address Sub-Object (Type =1)

The source TNA address sub-object contains the Source UNI-C's TNA name. It may be in one of the three formats, i.e. IPv4, IPv6, or NSAP. If a GENERALIZED_UNI object contains more than one source TNA address sub-object, only the first is meaningful and all others MUST be ignored.

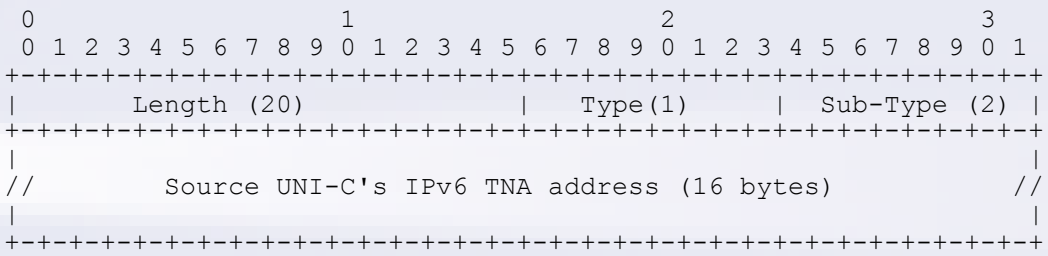
9.2.5.2 Source IPv4 TNA Address (Type=1, Sub-Type = 1)

The source IPv4 TNA address sub-object has the following format:



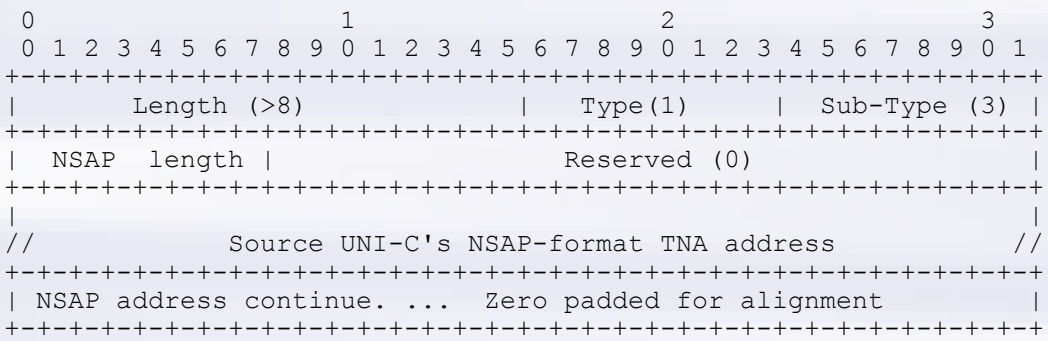
9.2.5.3 Source IPv6 TNA Address Sub-Object (Type=1, Sub-Type=2)

The source IPv6 TNA address sub-object has the following format:



9.2.5.4 Source NSAP TNA Address Sub-Object (Type=1, SubType = 3)

The source NSAP TNA address sub-object has the following format.



NSAP Length (8 bits): The length of source NSAP in bytes.

The Source NSAP-format TNA address is a variable length field structured according to [X.213].

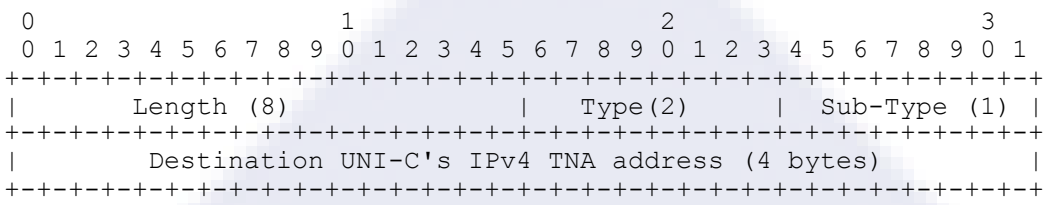
9.2.5.5 Destination TNA Address Sub-Object (Type =2)

The destination TNA address sub-object contains the destination UNI-C's TNA name. It may be in one of the three formats, i.e. IPv4, IPv6, or NSAP. If a GENERALIZED_UNI object contains more than one destination TNA address sub-object, only the first is meaningful and all others MUST be ignored.

If a destination TNA name is unknown or not reachable, a signaling node SHOULD generate a PathErr message with error code “Routing problem: no route available toward destination”.

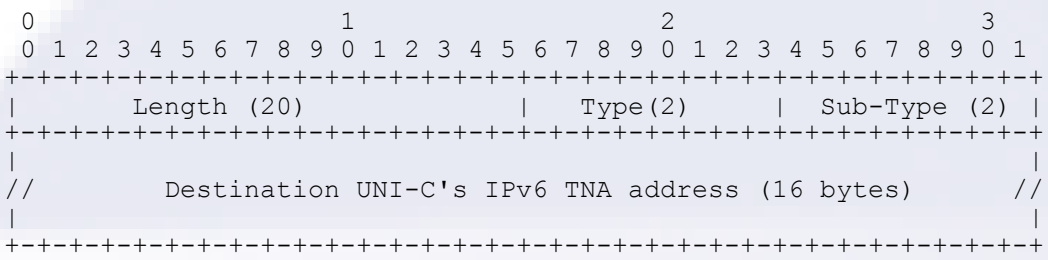
9.2.5.6 Destination IPv4 TNA Address Sub-Object (Type =2, Sub-Type = 1)

The destination IPv4 TNA address sub-object has the following format:



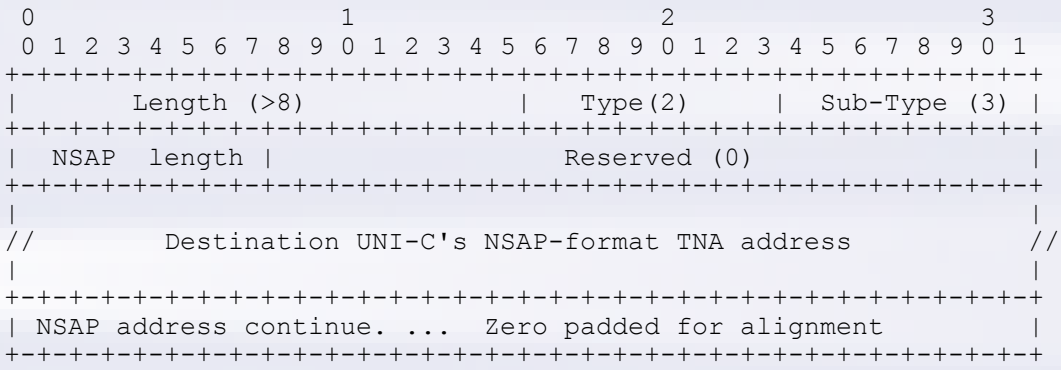
9.2.5.7 Destination IPv6 TNA Sub-object (Type =2, Sub-Type =2)

The destination IPv6 TNA address sub-object has the following format:



9.2.5.8 Destination NSAP-format TNA Address Sub-Object (Type =2, Sub-Type=3)

The destination NSAP TNA address sub-object has the following format:

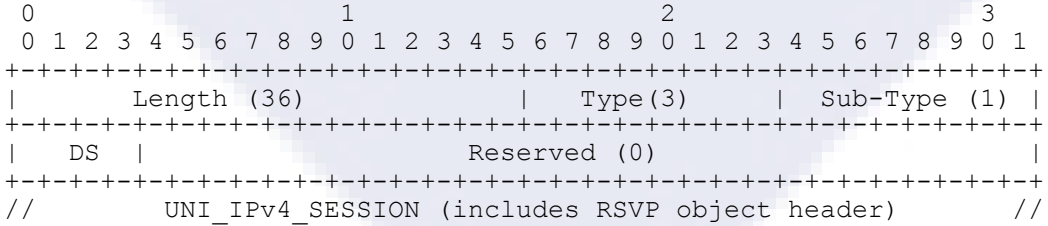


NSAP length (8 bits): The NSAP in bytes.

The destination NSAP-format TNA address is a variable length field structured according to ISO/IEC 8348, 1993 and it is identical to ITU X.213, 1992.

9.2.5.9 Diversity Sub-Object (Type= 3, Sub-Type = 1)

The Diversity sub-object is an UNI defined object to specify the physical diversity required for a new connection. It carries the local connection identifier of an existing connection, and it MAY be carried in the Path message. Multiple instances of the diversity sub-object MAY be used. The sub-object has the following format:



```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
//                LSP_TUNNEL_IPv4_SENDER_TEMPLATE                //
//                (includes RSVP object header)                  //
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

DS indicates the diversity requirement as following:

1 - Node diverse. The new connection SHALL NOT use any nodes that are on the path of the listed connection.

2 - Link Diverse. The new connection SHALL NOT use any links that are on the path of the listed connection.

3 - Shared Risk Link Group diverse. The new connection SHALL NOT use any links that have the same SRLG of the listed connection.

4 - Shared Path. The new connection SHALL use the same links as the listed connection.

0, 5-15 Reserved.

The diversity sub-object is processed by the source UNI-N and it is of no significance to the destination UNI-N or UNI-C. Hence, this sub-object may not be delivered to the destination UNI-C.

The source UNI-N MUST report error “Routing Problem: Diversity not available” if it cannot provide the requested diversity or if it receives a diversity object specifying an unsupported value.

9.2.5.10 Egress Label Sub-Object / SPC Label Sub-Object(Type=4, Sub-Type=1/2)

The Egress label sub-object is a UNI defined object and is used to specify the port, and label(s), to be used at the destination UNI. Up to two Egress label sub-objects MAY be included. The SPC label sub-object is an object defined in E-NNI [E-NNI] to specify the port and label(s) to be used at the destination. Up to two SPC label sub-objects MAY be included. The EGRESS_LABEL/SPC_LABEL MUST be carried from the source to the destination UNI-N but MUST not be signaled at the destination UNI.

The SPC label sub-object SHOULD only be used by the source UNI-C if the UNI-C can determine the destination is an SPC endpoint. If the source UNI-C does not know whether the destination is UNI enabled or not, it SHOULD use the Egress Label sub-object.

The Egress label and SPC Label sub-objects have the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|        Length (>=16)          |   Type(4)     |Sub-Type(1/2)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|U|        Reserved (0)                         |   Label Type  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                    Logical Port Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             Label                             |
|                              ...                              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Label Type:

This field indicates the C-Type of the label carried in the object. For egress label, sub-type is 1 [RFC3476]; for SPC label, sub-type is 2 [RFC3474].

U-bit:

This bit indicates the direction of the label and the logical port ID. It is 0 for the downstream label and port ID and 1 for the upstream label and port ID. It is only used on bi-directional connections.

Logical Port Identifier:

This field indicates a logical port identifier assigned at the destination UNI-C. It is used to select a link at the destination UNI. The identified link MUST be used to allocate resources for the connection.

Label:

This field identifies the label to be used. The format of this field is as described in Section 9.2.1.

The following SHOULD result in a "Bad EXPLICIT_ROUTE object" error:

- If the Logical Port Identifier does not identify a valid link at the destination.
- If the U-bit is set when requesting a unidirectional connection set-up.
- If there are two label sub-objects with the same U-bit values.

The destination UNI-N examines one sub-object for unidirectional connections and two sub-objects for bi-directional connections. If the U-bit of the sub-object being examined is clear (0) then value of the label is copied into a new LABEL_SET object. For EPL, the value of the EGRESS_LABEL / SPC_LABEL MUST be sent from the destination UNI-N to destination UNI-C in the RSVP_HOP object and the LABEL_SET object MUST NOT be sent. For EVPL, SONET/SDH and OTN, this LABEL_SET object MUST be included by the destination UNI-N in the corresponding outgoing Path message. If the destination UNI-C, is unable to pick the label from the Label Set or if there is a problem parsing the LABEL_SET object, then the request is terminated and a PathErr message with a "Routing problem/Label Set" indication MUST be generated.

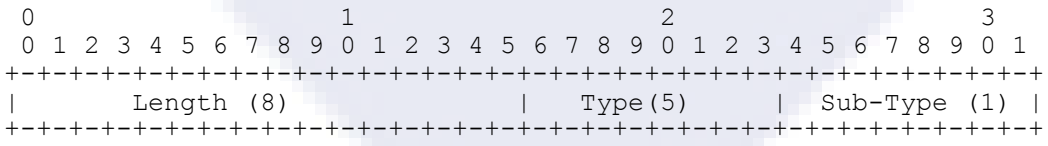
If the U-bit of the sub-object being examined is set (1) then the value of the label is to be used for upstream traffic associated with the bi-directional LSP. If the label is not acceptable for the destination UNI-N, then it MUST issue a PathErr message with a "Routing Problem/ label allocation failure" indication. If the label is acceptable by the destination UNI-N, then the label is copied into a new Upstream Label object. This Upstream Label object MUST be included on the corresponding outgoing Path message. If the label is not acceptable by the destination UNI-C, then it MUST issue a PathErr message with a "Routing problem/ Unacceptable label value" indication.

The generated PathErr message MAY include an Acceptable LABEL_SET object, see [RFC3473]. Otherwise, the Path message is processed as usual.

After processing is complete, the label sub-objects are removed from the GENERALIZED_UNI object. The logical_port id is mapped (on local basis) into the "interface_id" fields of the IF_ID_RSVP_HOP object (see [RFC3471, RFC3473]).

9.2.5.11 Service Level Sub-Object (Type=5, Sub-Type=1)

The Service Level sub-object specifies an integer within the range 0-255 (called the "service level" (see Section 7.12.2.6 of [OIF-UNI-02.0-Common])), and it MAY be included in the GENERALIZED_UNI object. Each service level corresponds to carrier predefined characteristics, such as type of restoration (e.g. unprotected, 1+1 protection), reversion strategies for the connection after failures have been repaired, and retention strategies. The sub-object has the following format:



```
| service Level |                Reserved (0)                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

A node MUST report error “Routing Problem: Service level not available” if it receives a service level object specifying an unsupported value.

9.2.6 GENERALIZED_LABEL_REQUEST Object (Class-Num = 19 [RFC3473])

The format of the GENERALIZED_LABEL_REQUEST object is defined in [RFC3473]. For UNI 2.0, a node MUST support SONET/SDH LSP encoding type and TDM Switching Type [RFC3471]. For G.709, a node MUST support one or more of the encoding types defined in [RFC4328] and TDM and/or Lambda Switch Capable (LSC) encoding type [RFC3471]. For Ethernet, a node MUST support Ethernet encoding type. For EPL services, a node MUST support Data Channel Switching Capable (DCSC) switching type as defined in [RFC6002] with a value of 125. For EVPL services, a node MUST support Layer-2 Switch Capable (L2SC) switching type [RFC3471].

9.2.7 IPv4_RESV_CONFIRM Object (Class-Num = 15, [RFC2205])

The format of the IPv4_RESV_CONFIRM object is defined in [RFC2205]. For UNI 2.0, the IPv4 receiver address is set to the destination UNI-C's SC PC ID at the destination UNI, and it is set to the source UNI-N's SC PC ID at the source UNI.

9.2.8 IPv4_ERROR_SPEC Object (Class-Num = 6 [RFC2205])

The format of the IPv4_ERROR_SPEC object is defined in [RFC2205] and [RFC3473]. For UNI 2.0, the IPv4 Error Node Address is set to the SC PC ID of the UNI-C or the UNI-N which reported the error. UNI-N and UNI-C entities MUST support InPlace, NotGuilty, and Path_State_Removed flags.

There are cases where it is useful to indicate a specific interface associated with an error. To support these cases the IF_ID ERROR_SPEC Objects are defined. The format of the IPv4/IPv6 IF_ID ERROR_SPEC Object is defined in [RFC3473].

Nodes wishing to indicate that an error is related to a specific interface SHOULD use the appropriate IF_ID ERROR_SPEC Object in the corresponding PathErr or ResvErr message. IF_ID ERROR_SPEC Objects SHOULD be generated and processed as any other ERROR_SPEC Object.

9.2.9 LSP_TUNNEL_IPv4_FILTER_SPEC Object (Class-Num = 10 [RFC3209])

The LSP_TUNNEL_IPv4_FILTER_SPEC, combined with the UNI_IPv4_SESSION object, is used to uniquely identify a connection. The format of this object is identical to the LSP_TUNNEL_IPv4_SENDER_TEMPLATE object.

9.2.10 MESSAGE_ID Object (Class-Num = 23 [RFC2961])

The format of MESSAGE_ID object is defined in [RFC2961]. For UNI 2.0, the Ack_Desired flag in a MESSAGE_ID object MUST be set in Path and Resv trigger messages, Notify, PathErr, PathTear, ResvErr, ResvTear and ResvConf messages, and in Srefresh message in order to support reliable messaging. The Ack_Desired flag MAY be set in a refresh message. Lack of acknowledgement of a refresh message at a node MUST NOT result in deletion of a connection.

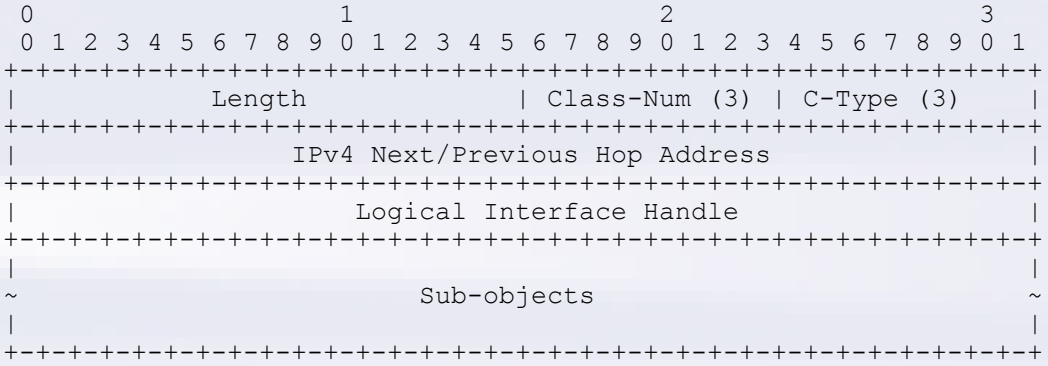
9.2.11 IPv4_IF_ID_RSVP_HOP Object (Class-Num = 3, [RFC3473/RFC3471])

The format of the IPv4_IF_ID_RSVP_HOP object is defined in [RFC3473]. For UNI 2.0, the IPv4_IF_ID_RSVP_HOP object MUST be used to select the data link where a connection’s resource should be allocated. Data links are specified from the viewpoint of the sender of the Path message. A node receiving one or more objects in a Path message saves their values and returns them in the HOP objects of

subsequent Resv messages sent to the node that originated the objects. In this object, sub-objects are used to identify the data channel(s) associated with an LSP:

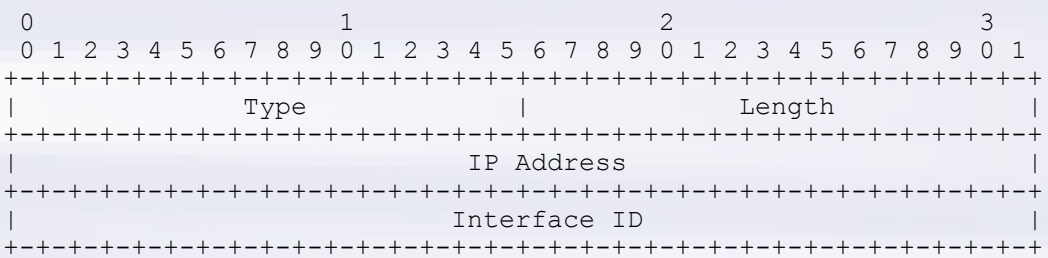
- For a unidirectional LSP, a downstream data link MUST be indicated.
- For bi-directional LSPs, a common downstream and upstream data link is normally indicated. In the special case where a bi-directional LSP traverses a bundled link, it is possible to specify a downstream data link that differs from the upstream data link. When two RSVP_HOP sub-objects are required, Type 3 sub-object MUST be used as follows:
  - The first sub-object MUST represent the downstream data link
  - The second sub-object MUST represent the upstream data link

The IPv4_IF_ID_RSVP_HOP Object has the following format. The C-Type 3 MUST be used:



A node's SC PC ID is used to fill the IP address field in the IPv4 IF_ID_RSVP_HOP object.

Sub-objects have the following format:



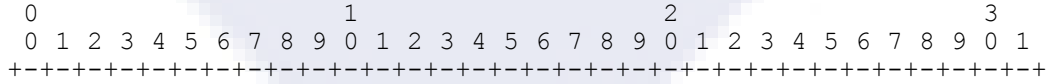
The node ID is used to fill the IP Address field of the sub-object fields.

9.2.12 Call_ID Object (Class-Num = 230, [RFC3474])

The CALL_ID is defined in [RFC3474] with further clarifications below.

To guarantee uniqueness of the CALL_ID across multiple domains, the LSR address MUST be set to the SC PC ID of the source node generating the CALL_ID.

In the initial request for a call/connection, [RFC3474] indicates that "the source user MUST set the CALL_ID's C-Type and call identifier value to all zeros". [RFC3474] does not specify the C-TYPE or length of the null CALL_ID. The following format MUST be used for the null CALL_ID with C-Type = 0. The last field is the C-Type and no further fields are present. The length is 4, representing the length of the header.



```
|            Length = 4          |Class-Num (230)|    C-Type (0) |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Once a call identifier has been assigned, the format SHOULD be one of the formats described in [RFC3474].

9.2.13 SONET/SDH Sender Tspec / Flowspec (Class-Num = 12/9, C-Type = 4 [RFC4606])

For SONET/SDH requests, the encoding of the SONET/SDH Sender Tspec / Flowspec follows [RFC4606] except for the STS-3c SPE/VC-4 signal. To simplify SONET/SDH interoperability, the encoding for the STS-3c SPE/VC-4 signal MUST be sent as follows: Signal Type=6, RCC=0, NCC=0. For backwards compatibility, an implementation MUST be able to accept the following value: Signal Type=6, RCC=1, NCC=1.

The NVC field MUST be set to 0 by the sender. The receiver MUST reject the request if the NVC value is not 0 with a "Traffic Control Error: Service Not Supported" error.

9.2.14 MEF Ethernet Sender Tspec /Flowspec(Class-Num = 12/9, C-Type = 6 [RFC6003])

For Ethernet requests, the encoding of the Ethernet Sender Tspec / Flowspec follows [RFC6003] with the following guidelines. The format of the MEF Ethernet Sender Tspec is as follows:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length              | Class-Num (12)| C-Type(6)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Switching Granularity       |              MTU              |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                                 |
~                        Sub-objects                              ~
|                                                                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Switching Granularity:

EPL MUST use value 1 (Ethernet Port)
EVPL MUST use value 2 (Ethernet Frame)

MTU – This is initialized by the source UNI-C.

Bandwidth Profile sub-object

Two types of sub-objects are supported. The Ethernet Bandwidth Profile sub-object MUST be used for bandwidth profile per UNI and bandwidth profile per EVC. The OIF Vendor Private Ethernet Bandwidth Profile sub-object MUST be used for bandwidth profile per CoS. There MUST be exactly one bandwidth profile.

Ethernet Bandwidth Profile (Type 2, Length 24)

For signaling a bandwidth profile per UNI or a bandwidth profile per EVC, the Bandwidth Profile sub-object MUST be:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Profile    |     Index     |            Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                              CIR                                |
```

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               CBS                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               EIR                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               EBS                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Profile (8 bits):

Bit 0 - Coupling Flag (CF)
Bit 1 - Color Mode (CM)
Bits 2-7 - Reserved (set to 0, ignored on receive)

Index:

MUST be set to 0.

CIR, CBS, EIR, EBS:

Each field is encoded as a 32-bit IEEE single-precision floating-point number. The bandwidth profile algorithm applied to the frames based on <CIR, CBS, EIR, EBS> is as described in [MEF10.1].

OIF Vendor Private Ethernet Bandwidth Profile

For signaling a bandwidth profile per CoS, the OIF Vendor Private Ethernet Bandwidth Profile MUST be:

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Type = 240                 |            Length               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  SMI Enterprise Code (26041)                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Profile    |     Index     |            Reserved             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               CIR                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               CBS                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               EIR                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                               EBS                             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

For EVPL services with a bandwidth profile per CoS, the Index field must indicate to which CoS the bandwidth profile applies.

Profile (8 bits):

Bit 0 - Coupling Flag (CF)
Bit 1 - Color Mode (CM)
Bits 2-7 - Reserved (set to 0, ignored on receive)

Index:

Contains one or more user priorities as follows:

User Priority	111	110	101	100	011	010	001	000

Index Bit	7	6	5	4	3	2	1	0

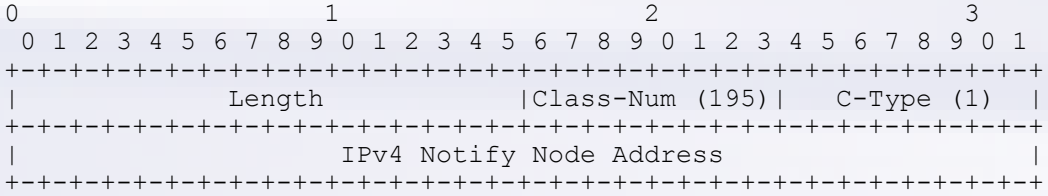
For example, the Index Field for User Priority 101 contains 00100000 and the Index Field for User Priority 001 and 000 contains 00000011.If a bandwidth profile sub-object Index contains multiple user priorities, the bandwidth profile applies to the combined traffic received for all the specified user priorities. For example, an index 00000011 with a CIR of 100Mbps and EIR of 0Mbps would allow a total of 100Mps for all frames with user priorities 001 or 000. To request 100Mbps for each user priority, the user needs to request two profiles, one with an index 00000001 with CIR of 100Mbps and one with an index 00000100 with CIR of 100Mbps. Frames are dropped if they are received with a priority for which no bandwidth profile is defined.

CIR, CBS, EIR, EBS:

Each field is encoded as a 32-bit IEEE single-precision floating-point number. The bandwidth profile algorithm applied to the frames based on <CIR, CBS, EIR, EBS> is as described in [MEF10.1].

9.2.15 Notify Request Object (Class-Num = 195, C-Type = 1 [RFC3473])

The Node Address field contains the SC PC ID of the UNI node that generates the object.



9.2.16 Code Points

UNI signaling using RSVP defined in this specification utilizes several RSVP objects. Most objects are defined in IETF RFCs and have been assigned a class number and a C-Type by the Internet Assigned Numbers Authority (IANA) based on policies that can be found in [RFC2434]. The only exceptions are the Ethernet Sender Tspec sub-object described for which an OIF Vendor private codepoint is defined in Section 9.2.14.

Furthermore, notification messages defined in this section have error codes for which unique values have been assigned. Table 8 summarizes the error codes specific to UNI signaling, as defined in this document. A number of other error codes generally applicable to RSVP and RSVP-TE signaling have been defined in [RFC2205] and [RFC3209].

Error Description	Error code/Value	Reference
"Routing problem: no route available toward destination"	24/5	[RFC3209]
"Routing Problem: Diversity not available"	24/100	[RFC3476]
"Bad EXPLICIT_ROUTE object"	24/1	[RFC3209]
"Routing Problem/Label Set"	24/11	[RFC3473]
"Routing Problem: Service level not available"	24/101	[RFC3476]
"Routing Problem: Invalid/Unknown connection ID"	24/102	[RFC3476]
"Traffic Control Error: Service Unsupported"	21/02	[RFC2205]
"Policy Control Failure: Unauthorized sender"	2/100	[RFC3476]
"Policy Control Failure: Unauthorized receiver"	2/101	[RFC3476]
No route available toward source	24/103	[RFC3474]
Unacceptable interface ID	24/104	[RFC3474]
Invalid/unknown CALL_ID	24/105	[RFC3474]
Invalid SPC interface ID/label	24/106	[RFC3474]

Table 8 – RSVP Error Codes

10 References

[G7713] ITU-T Recommendation G.7713/Y.1704 (2001) /Am1 (2004), "Distributed Call and Connection Management (DCM)"

[G7713.2] ITU-T Rec. G.7713.2, DCM Signalling Mechanism Using GMPLS RSVP-TE

[MEF10.1] Metro Ethernet Forum, MEF.10.1 - Ethernet Services Attributes Phase 2, Nov. 2006

[OIF-UNI-02.0-Common] "User Network Interface (UNI) 2.0 Signaling Specification: Common Part," February 2008[OIF-UNI-02.0-RSVP] "User Network Interface (UNI) 2.0 Signaling Specification Release 2 – RSVP", February 2008

[OIF-G-Sig-IW-01.0] "OIF Guideline Document: Signaling Protocol Interworking of ASON/GMPLS Network Domains", June 2008

[RFC791] "Internet Protocol", IETF RFC 791, September 1981

[RFC2205] R. Braden, Ed., "Resource Reservation Protocol (RSVP) - Version 1 Functional Specification," IETF RFC 2205, September, 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," IETF RFC 2434.

[RFC2961] L. Berger, et al., "RSVP Refresh Overhead Reduction Extensions," IETF RFC 2961.

[RFC3209] D. Awduche, et. al, "Extensions to RSVP for LSP Tunnels," IETF RFC 3209.

[RFC3471] L. Berger, et. al, "Generalized MPLS - Signaling Functional Description, IETF RFC 3471.

[RFC3473] L. Berger, et. al, "Generalized MPLS - RSVP-TE Signaling Functional Description," IETF RFC 3473.

[RFC3474] Z. Lin , et Al, "Documentation of IANA assignments for Generalized Multiprotocol Label Switching (GMPLS) Resource Reservation Protocol – Traffic Engineering (RSVP-TE) Usage and Extensions for Automatically Switched Optical Network (ASON)", IETF RFC 3474

[RFC3476] B. Rajagopalan, "Documentation of IANA Assignments for Label Distribution Protocol (LDP), Resource ReSerVation Protocol (RSVP), and Resource ReSerVation Protocol-Traffic Engineering (RSVP-TE) Extensions for Optical UNI Signaling", IETF RFC3476

[RFC4328] D. Papadimitriou, "GMPLS Signaling Extensions for G.709 Optical Transport Networks Control", January 2006

[RFC4606] E. Mannie, et. al, "Generalized Multi-Protocol Label Switching (GMPLS) Extensions for Synchronous Optical Network (SONET) and Synchronous Digital Hierarchy (SDH) Control" IETF RFC 4606.

[RFC6002] L. Berger, et al., "Generalized MPLS (GMPLS) Data Channel Switching Capable (DCSC) and Channel Set Label Extensions, IETF RFC 6002.

[RFC6003] D. Papadimitriou, "Ethernet Traffic Parameters", IETF RFC 6003.

[RFC6004] L. Berger, et al., "Generalized MPLS (GMPLS) Support For Metro Ethernet Forum and G.8011 Ethernet Service Switching", IETF RFC 6004.

[X.213] ITU-T Recommendation X.213, "Information Technology – Open Systems Interconnection – Network Service Definition"

11 Appendix A: Summary of UNI 2.0r2 deltas from UNI 2.0

The following changes were made as part of the UNI 2.0r2 revision:

- Updated references Ethernet-related RFCs in section 0, Table 2, section 8.3, Table 4, Table 5, Table 7, sections 9.2.1 and 10.
- Updated the Ethernet SENDER_TSPEC /FLOWSPEC in section 9.2.14.
- Obsoletes UNI 2.0 in section 6.

12 Appendix B: List of companies belonging to OIF when document is approved

Acacia Communications
ADVA Optical Networking
Agilent Technologies
Alcatel-Lucent
Altera
AMCC
Amphenol Corp.
Anritsu
Applied Communication Sciences
AT&T
Avago Technologies Inc.
Broadcom
Brocade
Centellax, Inc.
China Telecom
Ciena Corporation
Cisco Systems
ClariPhy Communications
Cogo Optronics
Comcast
Cortina Systems
CyOptics
Dell, Inc.
Department of Defense
Deutsche Telekom
EigenLight.com
Emcore
Ericsson
ETRI
EXFO
FCI USA LLC
Fiberhome Technologies Group

Finisar Corporation
France Telecom Group/Orange
Fujitsu
Fundacao.CPqD
Furukawa Electric Japan
GigOptix Inc.
Hewlett Packard
Hitachi
Hittite Microwave Corp
Huawei Technologies
IBM Corporation
Infinera
Inphi
Intel
IPtronics
JDSU
Juniper Networks
Kandou
KDDI R&D Laboratories
Kotura, Inc.
LeCroy
LSI Corporation
Luxtera
M/A-COM Technology Solutions, Inc.
Marben Products
Metaswitch
Mindspeed
Mitsubishi Electric Corporation
Molex
MoSys, Inc.
MultiPhy Ltd
NEC
NeoPhotonics
Nokia Siemens Networks
NTT Corporation
Oclaro
Optoplex
PETRA
Picometrix

PMC Sierra
QLogic Corporation
Reflex Photonics
Semtech
SHF Communication Technologies
Skorpios Technologies
Sumitomo Electric Industries
Sumitomo Osaka Cement
TE Connectivity
Tektronix
Tellabs
TeraXion
Texas Instruments
Time Warner Cable
TriQuint Semiconductor
u2t Photonics AG
Verizon
Vitesse Semiconductor
Xilinx
Xtera Communications
Yamaichi Electronics Ltd.