



OIF OPTICAL
INTERNETWORKING
FORUM

**User Network Interface (UNI) 1.0
Signaling Specification, Release 2**

**OIF-UNI-01.0-R2-RSVP -
RSVP Extensions for User Network Interface (UNI) 1.0
Signaling, Release 2**

February 27, 2004

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



The OIF is an international non profit organization with over 170 member companies, including the world's leading carriers and vendors. Being an industry group uniting representatives of the data and optical worlds, OIF's purpose is to accelerate the deployment of interoperable, cost-effective and robust optical internetworks and their associated 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, T1M1, T1X1, TMF, UXPi and the XFP MSA Group.

For additional information contact:
The Optical Internetworking Forum, 39355 California Street,
Suite 307, Fremont, CA 94538
510-608-5928 ☎ info@oiforum.com

www.oiforum.com

Notice: This Technical Document has been created by the Optical Internetworking Forum (OIF). This document is offered to the OIF Membership solely as a basis for agreement and is not a binding proposal on the companies listed as resources above. The OIF reserves the rights to at any time to add, amend, or withdraw statements contained herein. Nothing in this document is in any way binding on the OIF or any of its members.

The user's attention is called to the possibility that implementation of the OIF implementation agreement contained herein may require the use of inventions covered by the patent rights held by third parties. By publication of this OIF implementation agreement, the OIF makes no representation or warranty whatsoever, whether expressed or implied, that implementation of the specification will not infringe any third party rights, nor does the OIF make any representation or warranty whatsoever, whether expressed or implied, with respect to any claim that has been or may be asserted by any third party, the validity of any patent rights related to any such claim, or the extent to which a license to use any such rights may or may not be available or the terms hereof.

© 2001 Optical Internetworking Forum

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction other than the following, (1) the above copyright notice and this paragraph must be included on all such copies and derivative works, and (2) this document itself may not be modified in any way, such as by removing the copyright notice or references to the OIF, except as needed for the purpose of developing OIF Implementation Agreements.

By downloading, copying, or using this document in any manner, the user consents to the terms and conditions of this notice. Unless the terms and conditions of this notice are breached by the user, the limited permissions granted above are perpetual and will not be revoked by the OIF or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE OIF DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY, TITLE OR FITNESS FOR A PARTICULAR PURPOSE.

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

SOURCE:	Editor(s)'s Name	Working Group Chair
	Lyndon Ong Ciena Corporation 5965 Silver Creek Valley Rd San Jose, CA 95138 Phone: 408.834.7894 Email: lyong@ciena.com	Jim D. Jones Alcatel 601 Data Dr Plano, TX 75075 Phone: 972.519.2744 Email: jim.d.jones@alcatel.com

DATE: February 27, 2004

List of Contributors

Osama Aboul-Magd

Stefan Ansoerge

K. Arvind

Krishna Bala

Sandra Ballarte

Ayan Banerjee

Rick Barry

Debashis Basak

Greg Bernstein

Richard Bradford

Curtis Brownmiller

Yang Cao

John Drake

Hans-Martin Foisel

William Goodson

Gert Grammel

Richard Graveman

Eric Gray

Riad Hartini

Eric Mannie

Raj Jain

LiangYu Jia

Jim Jones (Editor)

Suresh Katukam

Nooshin Komae

Jonathan P. Lang

Monica Lazer

Fong Liaw

Zhi-Wei Lin

Ling-Zhong Liu

Ben Mack-Crane

Larry McAdams

Wilson Nheu

Lyndon Ong

Dimitiri Papadimitriou

Dimitrios Pendarakis

Kavi Prabhu

Bala Rajagopalan

Anil Rao

Robert Rennison

Jonathan Sadler

Stephen Shew

Arnold Sodder

John Strand

George Swallow

Ewart Tempest

Eve Varma

Cary Wright

Yangguang Xu

Yong Xue

Tao Yang

Jennifer Yates

John Z. Yu

Alex Zinin

Zhensheng Zhang

Table of Contents

1	OVERVIEW	8
2	UNI 1.0 SIGNALING MESSAGES AND RSVP OBJECTS	8
3	UNI RSVP SIGNALING PROCEDURES	9
3.1	UNI INTERFACES, SIGNALING CHANNEL, CONTROL CHANNELS, LOGICAL PORT IDENTIFIER AND ADDRESSING	9
3.2	SENDING UNI RSVP MESSAGES	10
3.3	RECEIVING UNI RSVP MESSAGES.....	10
3.4	RELIABLE MESSAGING	10
3.5	CONNECTION STATE MAINTENANCE.....	11
3.6	RESERVATION STYLE	11
3.7	LOCAL CONNECTION IDENTIFICATION	11
3.8	CONNECTION TRAFFIC PARAMETERS	11
3.9	CONNECTION CREATION	11
3.10	CONNECTION MODIFICATION	13
3.11	CONNECTION DELETION.....	13
	FORCED DELETION	15
3.13	CONNECTION STATUS ENQUIRY AND RESPONSE.....	19
3.14	SIGNALING CHANNEL FAILURE DETECTION AND RECOVERY	19
3.15	DATA PLANE FAILURE AND RECOVERY	19
4	RSVP MESSAGES AND OBJECTS FOR UNI SIGNALING	22
4.1	RSVP MESSAGES FOR UNI SIGNALING.....	23
4.1.1	<i>RSVP Common Message Hearer</i>	23
4.1.2	<i>Hello Message (Msg Type = 20 [RFC3209])</i>	23
4.1.3	<i>Path Message (Msg Type = 1 [RFC2205])</i>	24
4.1.4	<i>PathErr Message (Msg Type = 3 [RFC2205])</i>	24
4.1.5	<i>PathTear Message (Msg Type = 5 [RFC2205])</i>	25
4.1.6	<i>Resv Message (Msg Type = 2 [RFC2205])</i>	25
4.1.7	<i>ResvConf Message (Msg Type = 7 [RFC2205])</i>	26
4.1.8	<i>ResvErr Message (Msg Type = 4 [RFC2205])</i>	26
4.1.9	<i>ResvTear Message (Msg Type = 6 [RFC2205])</i>	26
4.1.10	<i>Srefresh Message (Msg Type = 15 [RFC2961])</i>	26
4.2	UNI RSVP OBJECTS FORMAT	27
4.2.1	<i>LSP_TUNNEL_IPv4_SENDER_TEMPLATE Object (Class-Num = 11 [RFC3209])</i>	27
4.2.2	<i>UNI_IPv4_SESSION Object (Class-Num = 1 [RFC3209])</i>	27
4.2.3	<i>GENERALIZED_UNI Object (Class-Num=229)</i>	28
4.2.4	<i>GENERALIZED_LABEL_REQUEST Object (Class-Num = 19 [RFC 3473])</i>	33
4.2.5	<i>IPv4_RESV_CONFIRM Object (Class-Num = 15, [RFC2205])</i>	33
4.2.6	<i>IPv4_ERROR_SPEC Object (Class-Num = 6 [RFC2205])</i>	33
4.2.7	<i>LSP_TUNNEL_IPv4_FILTER_SPEC Object (Class-Num = 10 [RFC3209])</i>	33
4.2.8	<i>MESSAGE_ID Object (Class-Num = 23 [RFC2961])</i>	33
4.2.9	<i>IPv4_IF_ID_RSVP_HOP Object (Class-Num = 3, [RFC-3473])</i>	34
5	RSVP CODE POINTS	34

6	REFERENCES	35
----------	-------------------------	-----------

1 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 [RFC-3209]. 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 [RFC-3471, RFC-3473] extends 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 adapting GMPLS RSVP-TE [RFC-3473] specifications is defined. UNI signaling re-uses the existing GMPLS RSVP-TE between UNI-C/UNI-N and UNI-N/UNI-C. The OIF UNI 1.0 is agnostic to protocol used within the network, however it assumes both source and destination UNI-C and UNI-N support GMPLS RSVP-TE.. This definition leverages the above specifications to the maximum extent possible. A few new objects are defined for supporting connection attributes that are unique to UNI 1.0. In addition to defining these new objects, 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 GMPLS RSVP-TE signaling specifications in OIF UNI 1.0 Release 1 have been used, unchanged, within ITU-T Rec. G.7713.2.

In this document, the directional terms “source” vs. “destination”, “originating” vs. “terminating”, “upstream” vs. “downstream”, “previous hop” vs. “next hop”, and “incoming interface” vs. “outgoing interface” are defined with respect to the direction of control message flow as in [RFC2205, RFC 3209, RFC 3473].

2 UNI 1.0 Signaling Messages and RSVP Objects

The UNI 1.0 abstract messages are described in [OIF2003.248]. 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.

[OIF2003.248] 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 4.

Message No.	Abstract Message Description	RSVP Message
1	Connection Create Request	Path
2	Connection Create Response	Resv, PathErr
3	Connection Create Confirmation	ResvConf
4	Connection Delete Request	Path or Resv with ADMIN_STATUS “Deletion in Progress” bit
5	Connection Delete Response	PathErr with Path_State_Removed flag, PathTear
6	Connection Status Enquiry	implicit
7	Connection Status Response	implicit
8	Notification	PathErr, ResvErr

Table 1: Mapping between UNI Abstract Messages and RSVP Messages

UNI Attributes	RSVP object	Reference
Source TNA Address	GENERALIZED_UNI/Source TNA Address	Section 4.2.3.1
Source Port ID	IPv4_IF_ID_RSVP_HOP	[RFC3473]
Source Generalized Label	GENERALIZED_LABEL	[RFC3473]
Destination TNA Address	GENERALIZED_UNI/Destination TNA	Section 4.2.3.5
Destination Port ID	GENERALIZED_UNI/Egress Label	Section 4.2.3.10
Destination Generalized Label	GENERALIZED_UNI/Egress Label	Section 4.2.3.10
Local Connection ID	(UNI_IPv4_SESSION, LSP_TUNNEL_IPv4_SENDER_TEMPLATE) or (UNI_IPv4_SESSION, LSP_TUNNEL_IPv4_FILTER_SPEC)	Section 4.2.2/4.2.1
Contract ID	POLICY_DATA	[RFC2205],
SONET/SDH Traffic Parameters	SONET/SDH_SENDER_TSPEC, SONET/SDH_FLOWSPEC	[GMPLS SONET]
Directionality	UPSTREAM_LABEL	[RFC3473]
Payload	GENERALIZED_LABEL_REQUEST/G-PID	[RFC3473]
Service Level	GENERALIZED_UNI, Service Level	Section 4.2.3.11
Diversity	GENERALIZED_UNI, Diversity	Section 4.2.3.9
Error Code	IPv4_ERROR_SPEC / IF_ID ERROR_SPEC	[RFC3473]
Connection Status		

Table 2: Mapping between UNI Attributes and RSVP Objects

3 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 setup two separate single-hop connections, one between the initiating UNI-C and UNI-N, and the other between the UNI-N and the terminating UNI-C. The network is assumed to provide coordination of signaling information between the initiating and the terminating side of the connection. UNI 1.0 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.

3.1 UNI Interfaces, Signaling Channel, Control Channels, Logical Port Identifier and Addressing

RSVP messages are exchanged over the UNI signaling channel. The signaling channel may be realized over one or more underlying IP control channels, as described in Section 6 of [OIF2003.248]. The

determination of the UNI control interfaces and the maintenance of the IP control channel are described in Section 8 of [OIF2003.248]. The identification of connection endpoints is described in Section 7 of [OIF2003.248].

3.2 Sending UNI RSVP Messages

When a UNI-C (UNI-N) is sending a RSVP message, it **MUST** address the message directly to its UNI-N (UNI-C) peer. The peer's Node ID is used for this purpose (see Section 7 of [OIF2003.248]). A node **SHOULD** use the simple IP encapsulation and the router-alert option **MUST NOT** be included in any RSVP 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.

IP Header Values for UNI RSVP messages	
Version	4
Header Length	5
TOS	As defined in RFC 2205
Total Length	Message length
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 Node ID
Destination Address	UNI-C/UNI-N Node ID

Table 3: IP header for UNI RSVP messages

3.3 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 Section 11 of OIF2003.248) 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.

3.4 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 Path, PathTear, PathErr, Resv, ResvTear, ResvErr, ResvConf, and Srefresh messages. The Ack_Desired flag in a MESSAGE_ID object **MUST** be set in all RSVP trigger messages, PathErr, ResvErr, ResvConf, and Srefresh message, and **MAY** be set in RSVP refresh messages.

Message identification and acknowledgment are done on a per-hop basis. Each MESSAGE_ID object contains a message identifier. This identifier **MUST** uniquely identify a message with respect to a node's Node Identifier.

Failure to receive a MESSAGE_ID_ACK for a refresh 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.

3.5 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.

3.6 Reservation Style

RSVP reservation “styles” are defined in [RFC2205]. UNI signaling makes use of the Fixed Filter (FF) style only.

3.7 Local Connection Identification

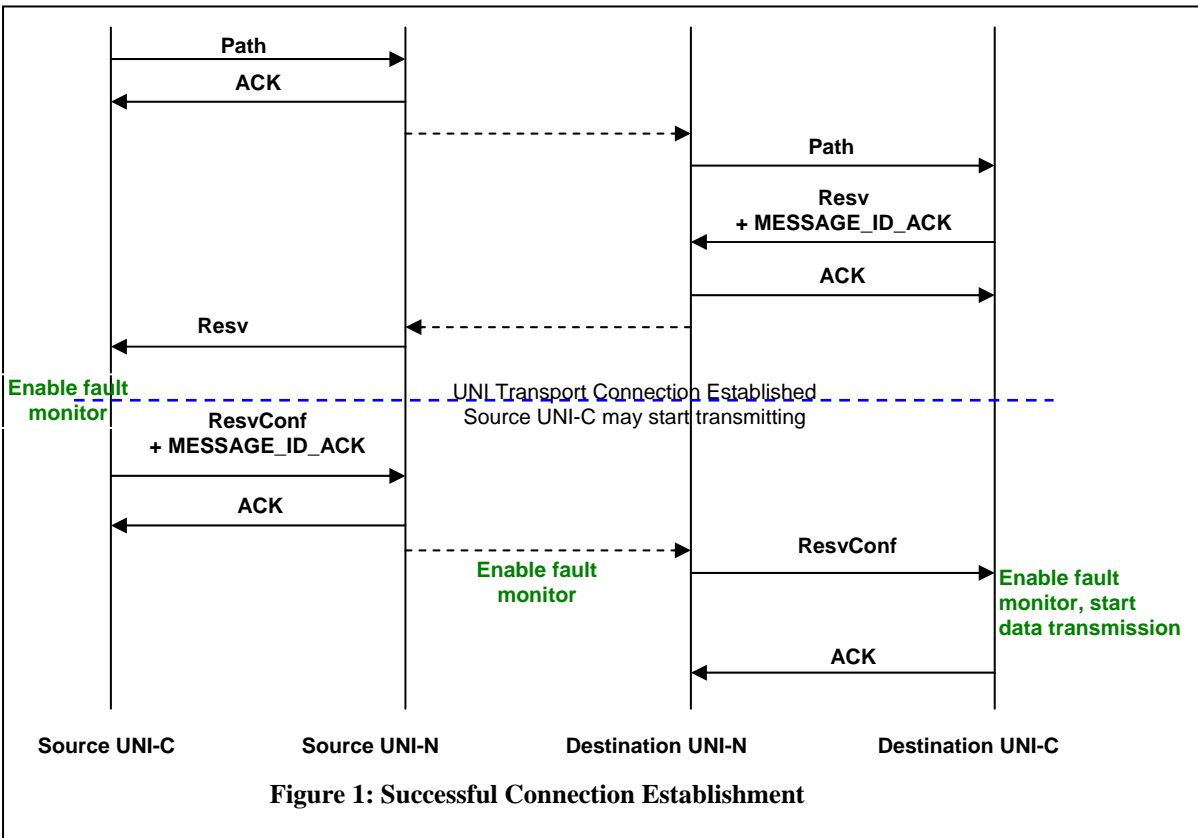
A Local Connection Identifier is used to uniquely identify a connection at a UNI. Under RSVP signaling, this is realized using the combination of the UNI_IPv4_SESSION object and the LSP_TUNNEL_IPv4_SENDER_TEMPLATE object in Path, PathTear, and PathErr messages, and the combination of the UNI_IPv4_SESSION object and the LSP_TUNNEL_IPv4_FILTER_SPEC object in Resv, ResvTear, ResvErr, and ResvConf messages. The local connection identifier remains unmodified during the lifetime of a connection.

3.8 Connection Traffic Parameters

The traffic parameters of a connection are technology dependent and are encoded in the SENDER_TSPEC and the FLOWSPEC object classes. A UNI-N and a UNI-C node MUST support standard SONET/SDH traffic parameters defined in GMPLS SONET-SDH specification [GMPS SONET] and covered in Section 10 of [OIF2003.248]. 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].

3.9 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 a SONET/SDH SENDER_TSPEC object in the Path message and a SONET/SDH FLOWSPEC object in the corresponding Resv Message. Figure 1 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.

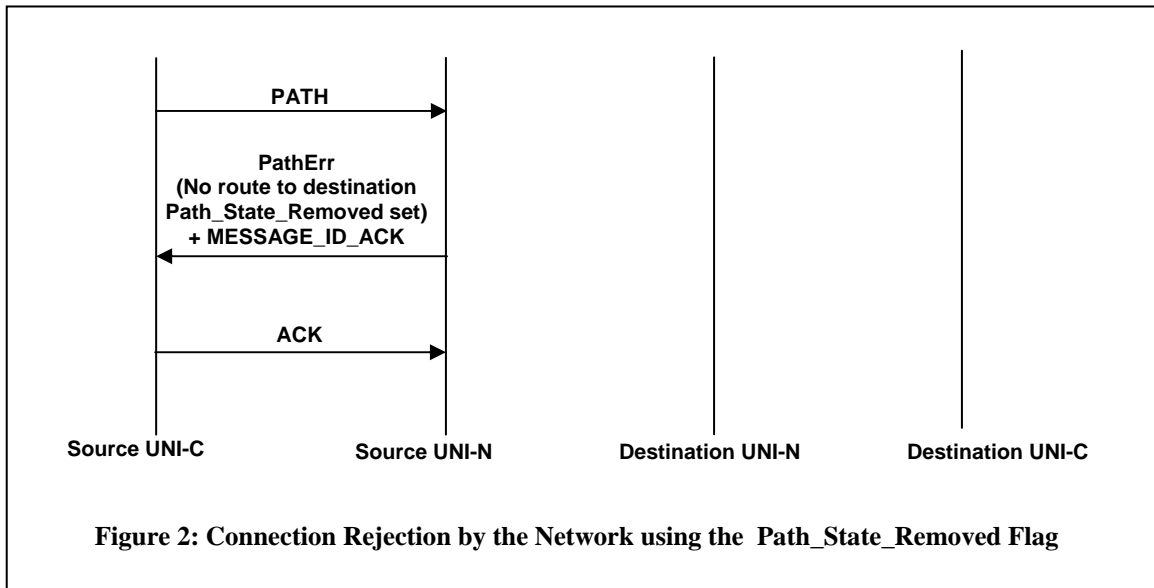
To request a bi-directional connection, a UNI-C MUST insert an UPSTREAM_LABEL Object in the Path message to select the upstream label(s) for the connection.

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 connections.

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. 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 Path 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 1.

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

A connection may not be successfully created due to resource unavailability, policy or reachability constraints. Figures 2, 3 and 4 illustrate timing diagrams and message flows for certain unsuccessful connection creation scenarios.



A node rejecting a connection request SHOULD delete its own path state and set the Path_State_Removed flag in the PathErr message. A node receiving a PathErr with Path_State_Removed flags set, SHOULD NOT send a PathTear downstream. This is shown in Figures 2 and 4.

If the Path_State_Removed flag is not set in a PathErr message, the source UNI-C MUST decide whether to delete the connection explicitly or allow it to time out. To delete the connection explicitly, the UNI-C MUST send a PathTear message. A node that receives a PathTear, which 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 3.

3.10 Connection Modification

UNI 1.0 does not support destructive connection modification. Modification of RSVP objects that do not change traffic parameters, e.g. ADMIN_STATUS object, MESSAGE_ID object, etc., is supported.

3.11 Connection Deletion

RSVP currently deletes 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 MUST be sent in 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. The procedure is described in [RFC3473] and shown in Figures 5 and 6.

The UNI-N may trigger connection deletion by sending a Resv or Path message to the UNI-C that includes the ADMIN_STATUS object with A&R bits set. This is shown in Figures 7 and 8. The UNI-C receiving the message SHOULD initiate the deletion procedure illustrated in Figures 7 and 8. If a UNI-C does not respond to a network initiated graceful deletion, the network SHOULD continue the message flow illustrated below the (red) dashed line in Figures 7 and 8.

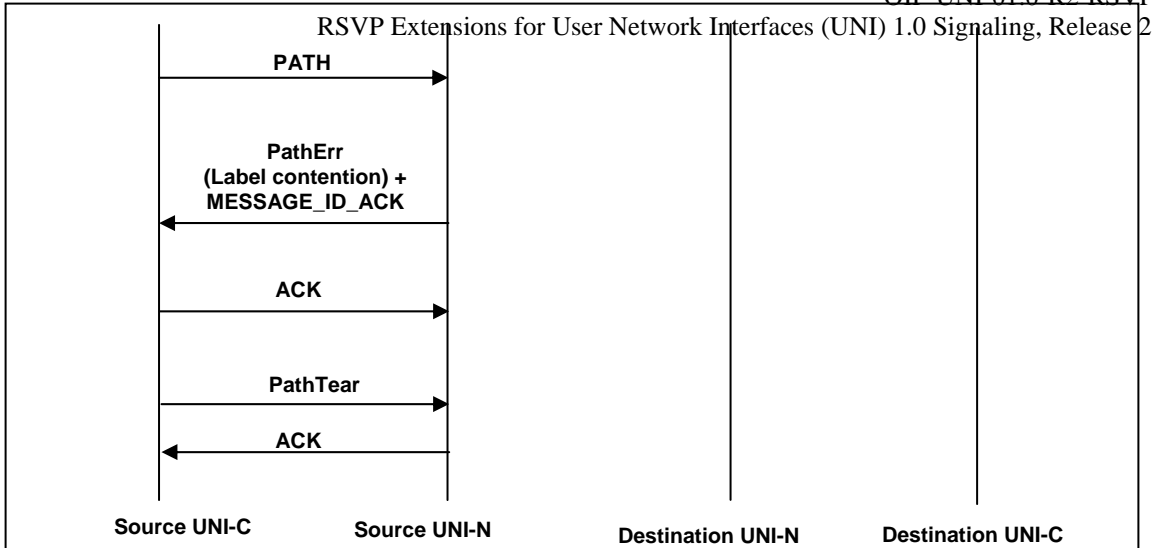
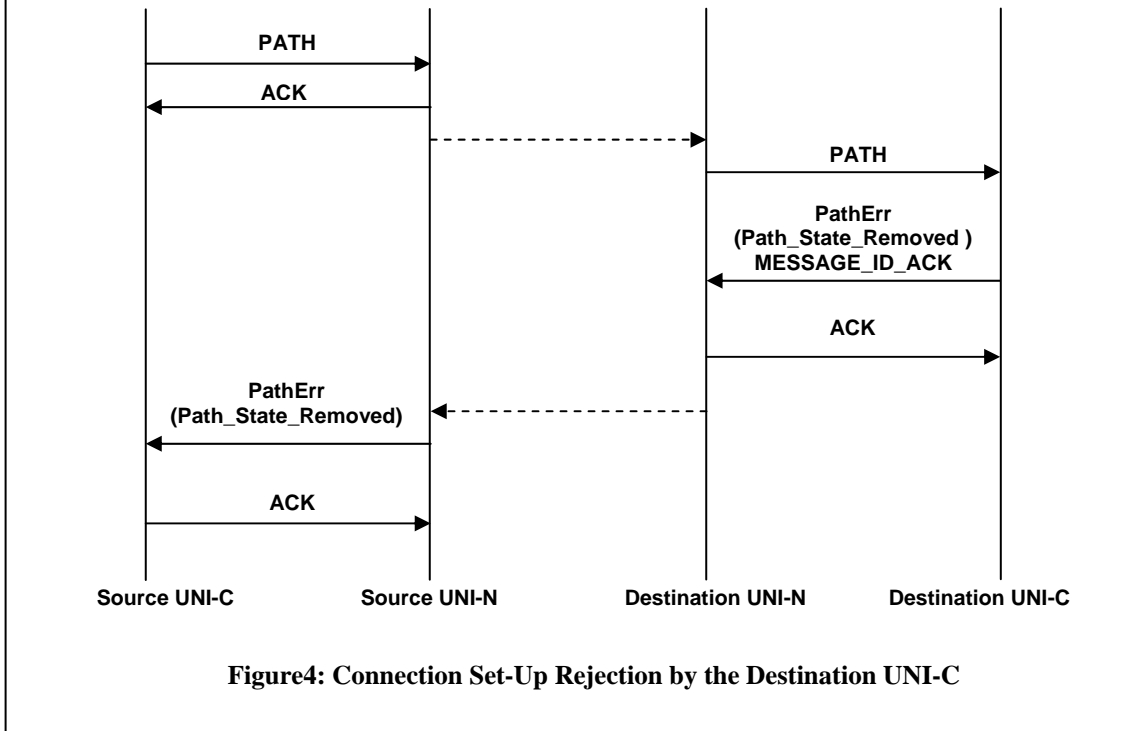


Figure3: Connection Rejection by the Network, without the Use of Path_State_Removed Flag



3.12 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.

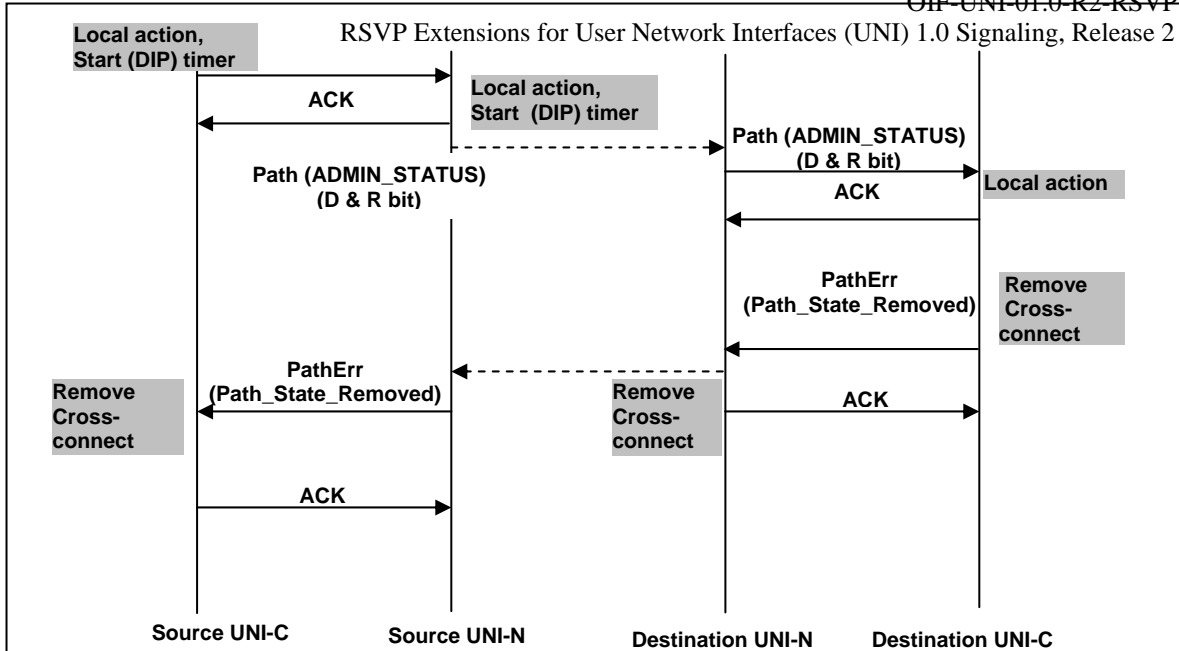


Figure 5: Connection Teardown Initiated by the Source UNI-C

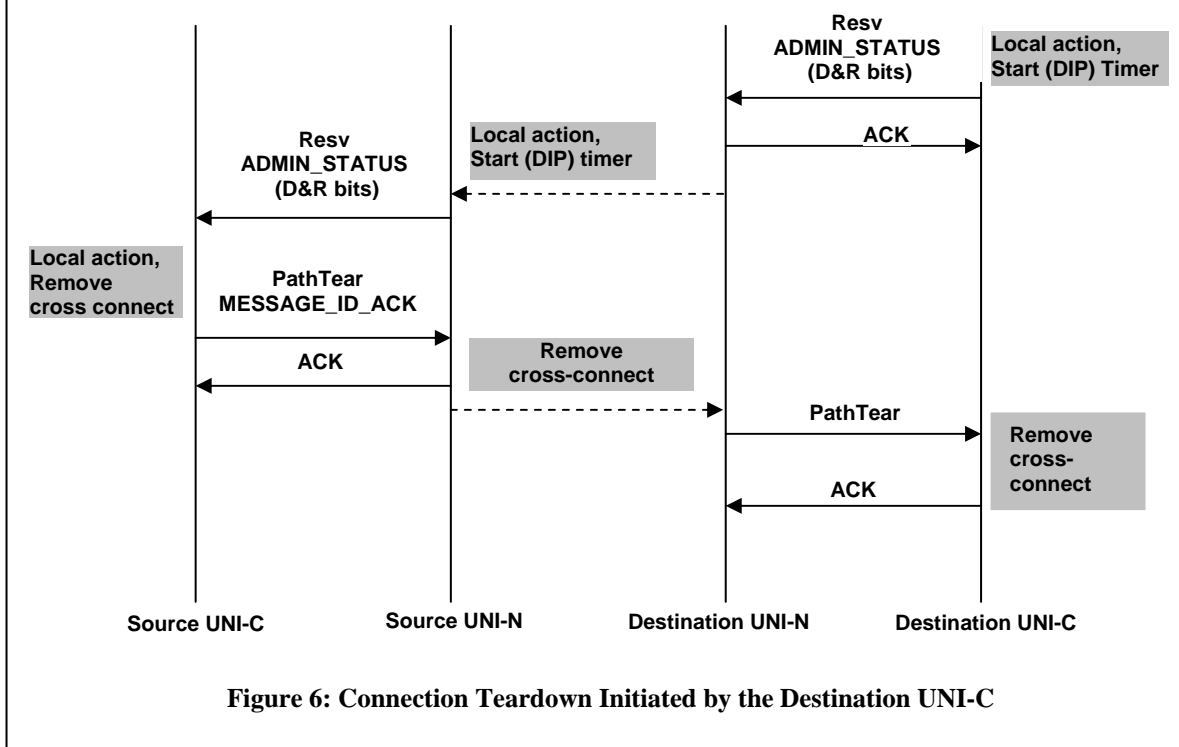
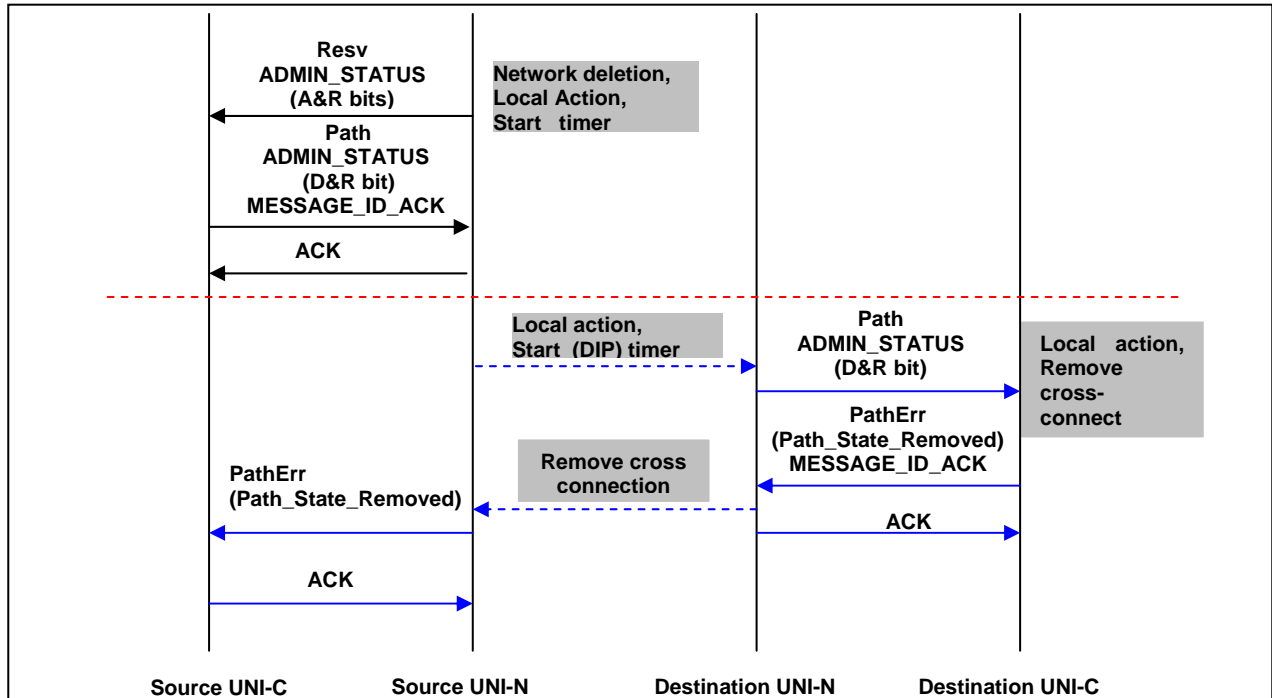
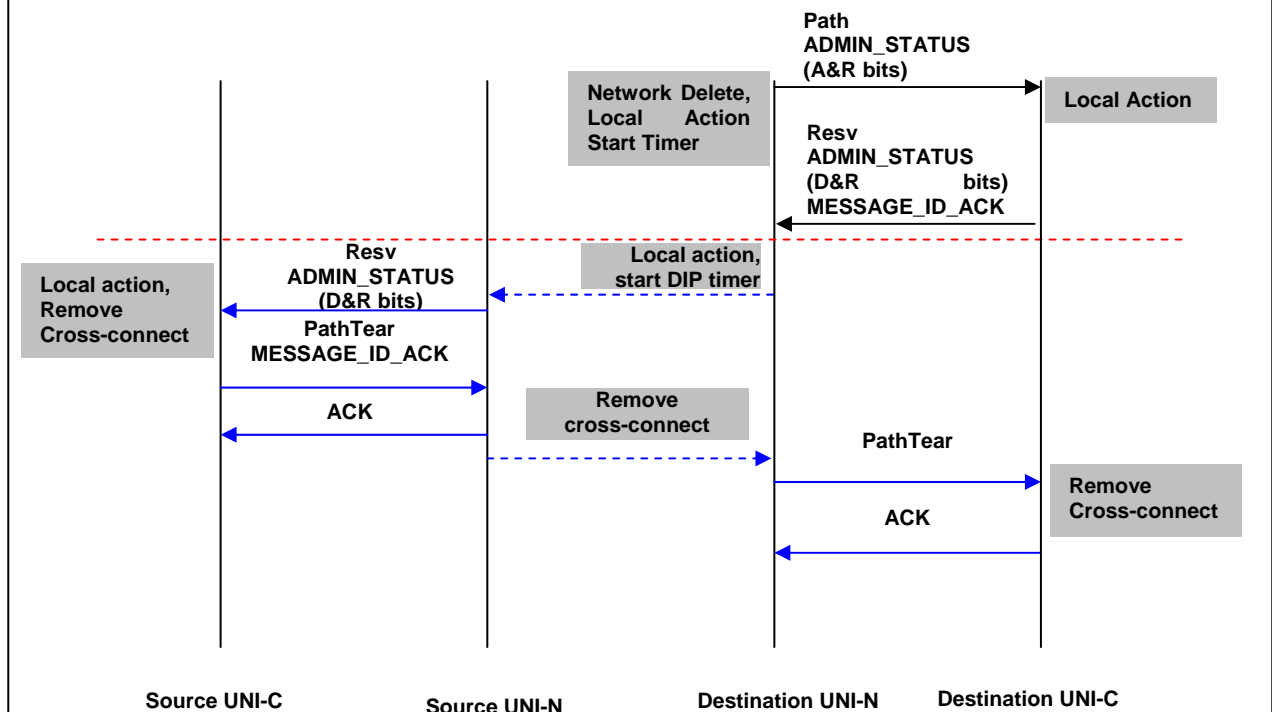
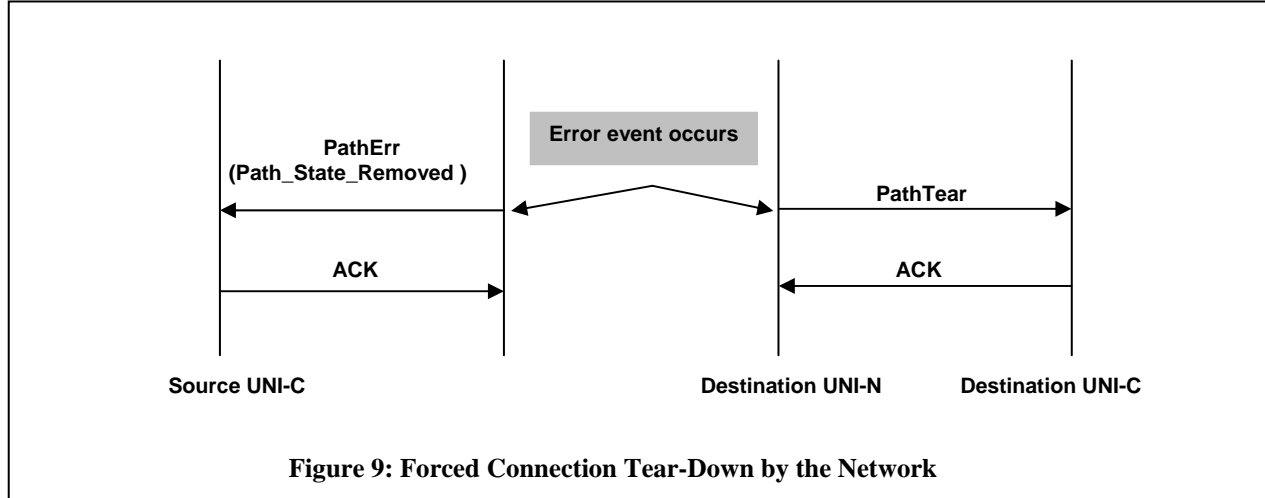


Figure 6: Connection Teardown Initiated by the Destination UNI-C

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 9. 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 7: Connection Teardown Initiated by the Source UNI-N

Figure 8: Connection Teardown Initiated by Destination UNI-N



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. A forced deletion initiated by a PathErr with Path_State_Removed flag may be replaced by using the combination of a ResvTear and a PathTear message. A UNI-C node should initiate a forced deletion only when the “Deletion In Progress” timer expires or when the soft state times out.

3.13 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 continuous query and response process that keeps the states on both nodes synchronized. There is, therefore, no need for explicit query and response messages in order to find out a neighbor’s connection state.

3.14 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 control channel failure is described in [RFC 3473] 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.

To address this requirement, a node MUST support the fault handling procedure described in Section 9 of [RFC-3473], illustrated in Figures 10 and 11. Figure 10 illustrates the message flow in the case of recovery from a node failure (i.e., complete failure of the control plane). Figure 11 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 “self-refresh” procedure in Figures 10 and 11 refers to the behavior of a UNI-C or a UNI-N node which 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 RSVP adjacency is re-established or when the restart timer expires.

3.15 Data Plane Failure and Recovery

SONET/SDH provides rich features for failure detection, in particular 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.

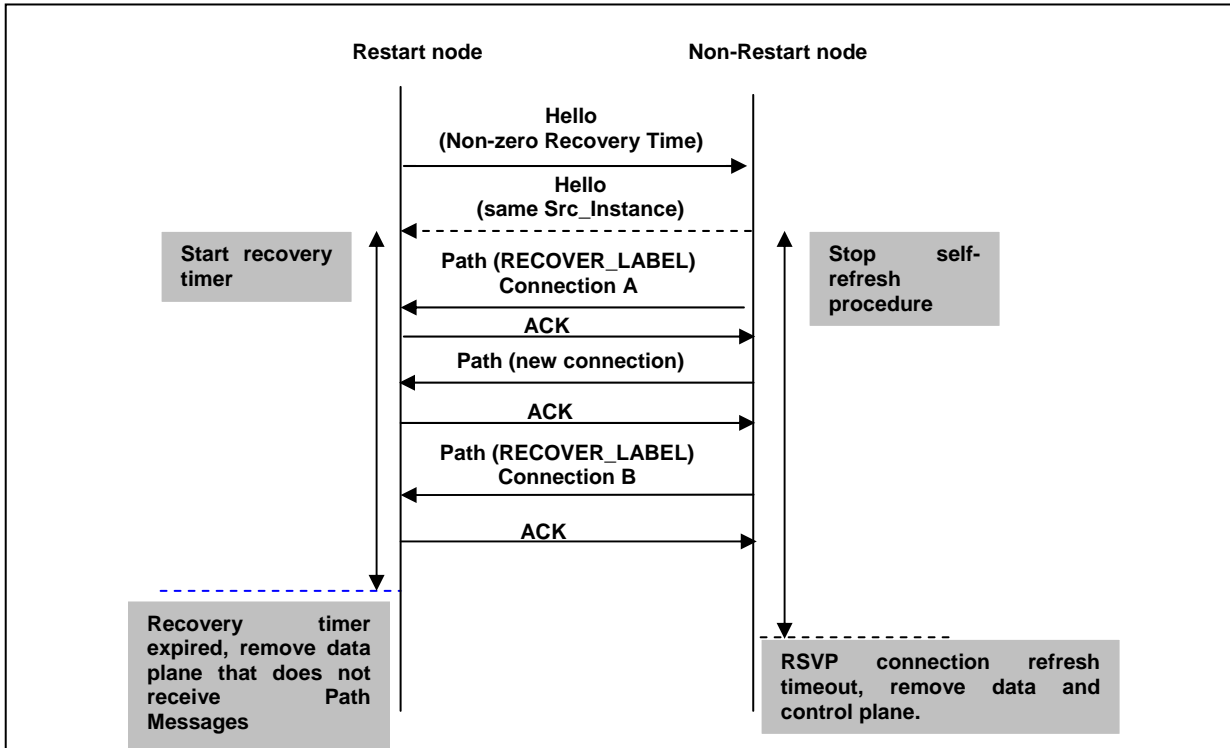


Figure 10: Recovery from Complete Failure

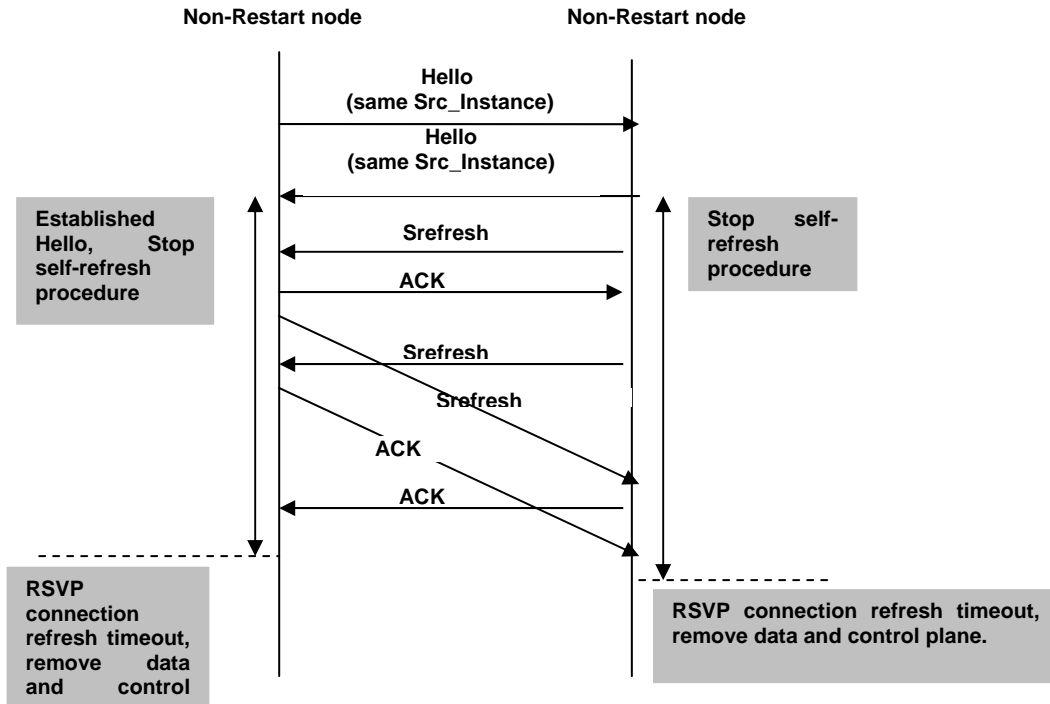


Figure 11: Recovery from Signaling Channel Failure

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.

4 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 1.0. Only objects, messages and behavior that are not already captured in standard IETF specifications, or for which specific details are necessary are described in this document. The standard behavior is captured in relevant IETF documents as referenced.

Table 4 and 5 show the complete list of messages and objects supported by UNI 1.0. 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 5. The particular approach used to trigger Srefresh message based refreshes at both source and destination UNI is implementation specific (see [RFC2961]). The EGRESS_LABEL and the pair Source/Destination TNA are the only Generalized UNI attributes that MUST be carried from the source to the destination UNI-N.

RSVP Messages	Direction	Message Type	Reference
Path	(Source) UNI-C → UNI-N & (Dest) UNI-N → UNI-C	1	Section 4.1.3
PathTear	(Source) UNI-C → UNI-N & (Dest) UNI-N → UNI-C	5	Section 4.1.5
PathErr	(Source) UNI-N → UNI-C & (Dest) UNI-C → UNI-N	3	Section 4.1.4
Resv	(Source) UNI-N → UNI-C & (Dest) UNI-C → UNI-N	2	Section 4.1.6
ResvErr	(Source) UNI-C → UNI-N & (Dest) UNI-N → UNI-C	4	Section 4.1.8
ResvTear	(Source) UNI-C → UNI-N & (Dest) UNI-N → UNI-C	6	Section 4.1.9
ResvConf	(Source) UNI-C → UNI-N & (Dest) UNI-N → UNI-C	7	Section 4.1.7
Hello	UNI-N ↔ UNI-C	20	[RFC3473]
Ack	UNI-N ↔ UNI-C	13	[RFC2961]
Srefresh	UNI-N ↔ UNI-C	15	[RFC2961]
Bundle	UNI-N ↔ UNI-C	12	[RFC2961]

Table 4: RSVP Messages Supported Under UNI 1.0

RSVP Object or Sub-Objects	End-to-End Significant?	C-Num/ C-Type [/ Type [/ Sub-type]]	Reference	
ACCEPTABLE_LABEL_SET	NO	130/1	[RFC3473]	
ADMIN_STATUS	YES	196/1	[RFC3473]	
SONET/SDH_FLOW_SPEC	YES	9/4	[GMPLS SONET]	
LSP_TUNNEL_IPv4_FILTER_SPEC	NO	10/7	Section 4.2.7	
GENERALIZED_LABEL	NO	16/2	[RFC3473]	
GENERALIZED_LABEL_REQUEST	YES	19/4	Section 4.2.4	
GENERALIZED_UNI_ATTRIBUTES	SOURCE_TNA	YES	229/1/1/<1,2,3>	Section 4.2.3.1
	DESTINATION_TNA	YES	229/1/2/<1,2,3>	Section 4.2.3.5
	DIVERSITY	NO	229/1/3/1	Section 4.2.3.9
	EGRESS_LABEL	NO	229/1/4/1	Section 4.2.3.10
	SERVICE_LEVEL	YES	229/5	Section 4.2.3.11
HELLO_REQUEST	NO	22/1	[RFC3473]	
HELLO_ACK	NO	22/2	[RFC3473]	
INTEGRITY	NO	4/1	Section 13 of [OIF2003.248]	
IPv4_ERROR_SPEC, error code and value	YES	6/1	Section 4.2.6	
IF_ID_ERROR_SPEC			[RFC3473]	
IPv4_IF_ID_RSVP_HOP	NO	3/3	Section 4.2.9	
IPv4_RESV_CONFIRM	NO	15/1	Section 4.2.5	
LABEL_SET	NO	36/1	[RFC3473]	
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	14/1	Section 13 of [OIF2003.248]	
RECOVERY_LABEL	NO	34/2	[RFC3473]	
RESTART_CAP	NO	131/1	[RFC3473]	
LSP_TUNNEL_IPv4_SENDER_TEMPLATE	NO	11/7	Section 4.2.1	
SONET/SDH_SENDER_TSPEC	YES	12/4	[GMPLS SONET]	
STYLE	YES	8/1	[RFC2205]	
TIME_VALUE	NO	5/1	[RFC2205]	
UNI_IPv4_SESSION	NO	1/11	Section 4.2.2	
UPSTREAM_LABEL	NO	35/2	[RFC3473]	

Table 5: Summary of UNI RSVP Objects

4.1 RSVP Messages for UNI Signaling

4.1.1 RSVP Common Message Hearer

The flag field of RSVP common header MUST be set to 1, to indicate support of the Bundle and Srefresh messages.

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

The Hello message is 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 .

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

The Path message is used for connection creation. 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> [ <LABEL_SET> ... ]
    [ < ADMIN_STATUS> ]
    <Generalized UNI>
    [ <POLICY_DATA> ... ]
    <sender descriptor>

<Generalized UNI> ::=
    <Common Object Header>
    <DESTINATION_TNA>
    <SOURCE_TNA>
    [<DIVERSITY> ...]
    [<SERVICE_LEVEL>]
    [<EGRESS_LABEL>]

```

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

```

<sender descriptor> ::=
    <LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <SONET/SDH_SENDER_TSPEC>
    [ < RECOVER_LABEL > ]

```

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

```

<sender descriptor> ::=
    <LSP_TUNNEL_IPv4_SENDER_TEMPLATE> <SONET/SDH_SENDER_TSPEC>
    <UPSTREAM_LABEL> [ < RECOVER_LABEL > ]

```

4.1.4 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>
    <IPv4_ERROR_SPEC>
    [ <ACCEPTABLE_LABEL_SET> ]
    [ <POLICY_DATA> ... ]
    <sender descriptor>

```

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

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

```

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

```

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

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

The Resv message is used for connection creation. A destination UNI-C SHOULD stop inserting the ResvConfirm Object in the Resv message if it receives a matching ResvConf message.

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>
    [ <IPv4_RESV_CONFIRM> ]
    [ <ADMIN_STATUS> ]
    [ <POLICY_DATA> ... ]
    <STYLE>
    <FF flow descriptor>

```

```

<FF flow descriptor> ::=
    <SONET/SDH_FLOWSPEC> <LSP_TUNNEL_IPv4_FILTER_SPEC>
    <GENERALIZED_LABEL>

```

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

The ResvConf message SHOULD be sent downstream from the source UNI-C to acknowledge the receipt of a Resv message that includes a RESV_CONFIRM Object. Specifically, under UNI 1.0, 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 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 >
```

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

The ResvErr message is used for reporting reservation errors. 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>
```

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

The ResvTear message is supported by UNI 1.0 to be compatible with RSVP. 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>
    <STYLE>
    <FF flow description >
    <FF flow description > ::= (see earlier list)
```

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

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> ]
```

4.2 UNI RSVP Objects Format

This section describes the RSVP objects that require specific behavior.

4.2.1 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 1.0, the IPv4 Tunnel sender address MUST be set to the source UNI-C's Node ID at the source UNI and MUST be set to the destination UNI-N's Node ID at the destination UNI. The LSP ID is assigned by the sender of the Path message and it remains constant during the life of a connection.

The combination of the LSP_TUNNEL_IPv4_SENDER_TEMPLATE object and UNI_IPv4_SESSION object MUST uniquely identify a connection at a local UNI.

4.2.2 UNI_IPv4_SESSION Object (Class-Num = 1 [RFC3209])

UNI_IPv4_SESSION Object [RFC3209] has the following format:

- UNI_IPv4_SESSION object: Class = 1, C-Type = 11

```

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)          | Class-Num(1) | C-Type (11) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               IPv4 Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+
|          MUST be zero          |          Tunnel ID          |
+-----+-----+-----+-----+-----+-----+-----+-----+
|                               Extended IPv4 Address                               |
+-----+-----+-----+-----+-----+-----+-----+-----+

```

IPv4 Address: This MUST be set to the source UNI-N's Node ID at the source UNI and it MUST be set to the destination UNI-C's Node ID at 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 establish between the same UNI-C/UNI-N pair if the Source/Destination TNA is not part of their description.

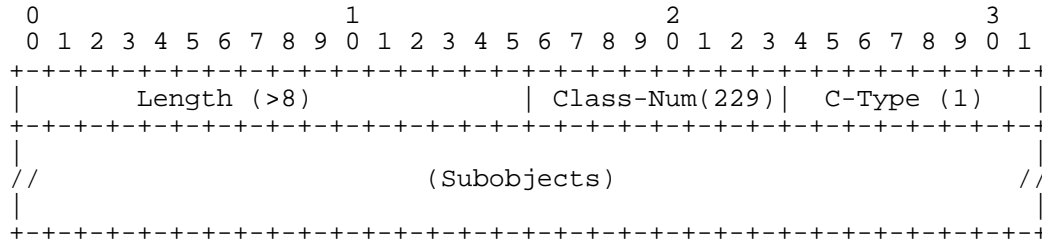
Extended IPv4 address: This MUST be set to the Node ID of source UNI-C at the source UNI and it MUST be set to the Node ID of the destination UNI-N at the destination UNI.

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”.

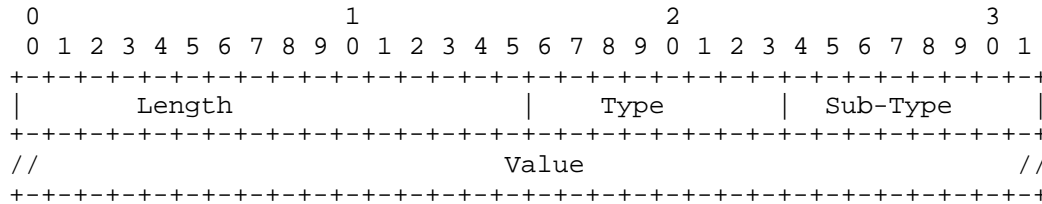
4.2.3 GENERALIZED_UNI Object (Class-Num=229)

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.

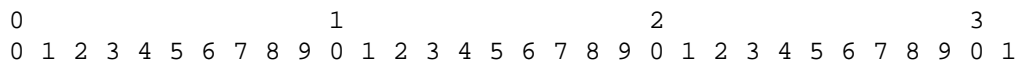
When the source or destination address specified cannot be serviced due to policy reasons, an error message with the error code “Policy control failure: Unauthorized sender,” or “Policy control failure: Unauthorized receiver”, respectively.

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

The source TNA address sub-object contains the Source UNI-C's TNA address. 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.

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

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



```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          Length (8)          |      Type(1)      |  Sub-Type (1)  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          Source UNI-C's IPv4 TNA address (4 bytes)          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

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

```

          0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          Length (20)          |      Type(1)      |  Sub-Type (2)  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                                                    |
//          Source UNI-C's IPv6 TNA address (16 bytes)          //
|                                                                    |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

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

```

          0          1          2          3
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          Length (>8)          |      Type(1)      |  Sub-Type (3)  |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  NSAP length |          Reserved (0)          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|                                                                    |
//          Source UNI-C's NSAP-format TNA address          //
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|  NSAP address continue. ... Zero padded for alignment          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

Source NSAP-format TNA address: Variable length field. Structured according to ISO/IEC 8348, 1993. Identical to ITU X.213, 1992.

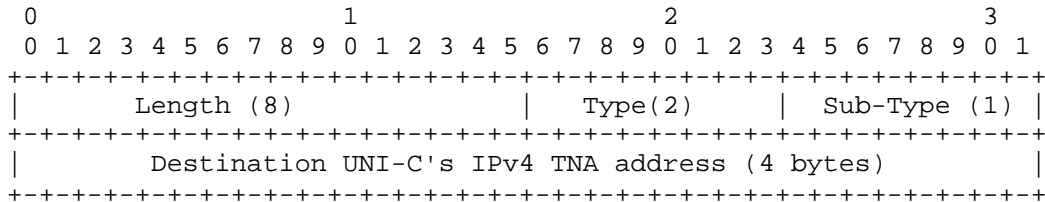
4.2.3.5 Destination TNA Address Sub-Object (Type =2)

The destination TNA address sub-object contains the destination UNI-C's TNA. 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 address is unknown or not reachable, a signaling node SHOULD generate a PathErr message with error code "Routing problem: no route available toward destination".

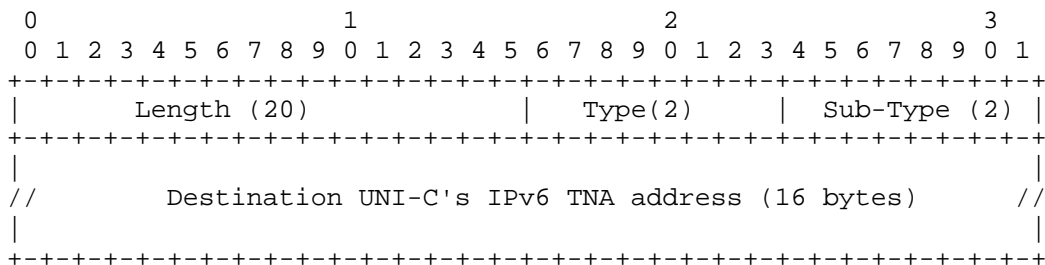
4.2.3.6 Destination IPv4 TNA Address Sub-Object (Type =2, Sub-Type = 1)

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



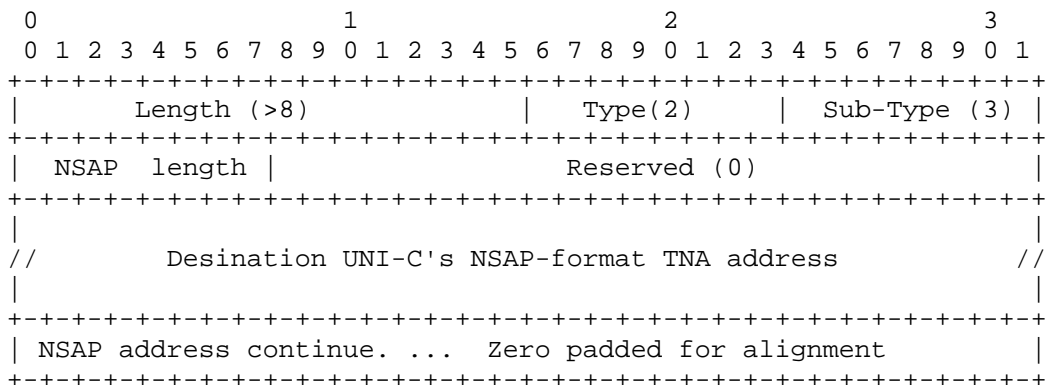
4.2.3.7 Destination IPv6 TNA Sub-object (Type =2, Sub-Type =2)

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



4.2.3.8 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.

Destination NSAP-format TNA: Variable length field. Structured according to ISO/IEC 8348, 1993. Identical to ITU X.213, 1992.

4.2.3.9 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:

```

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 (36)          |          Type(3)          |          Sub-Type (1) |
+-----+-----+-----+-----+-----+-----+-----+-----+
|   DS   |          Reserved (0)          |
+-----+-----+-----+-----+-----+-----+-----+
//          UNI_IPv4_SESSION (includes RSVP object header)          //
+-----+-----+-----+-----+-----+-----+-----+-----+
//          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 link 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.

4.2.3.10 Egress Label Sub-Object (Type=4, Sub-Type=1)

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. It has the following format:

```

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

```

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.

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.

Label:

This field identifies the label to be used. The format of this field is identical to the one used by the Label field in GMPLS SONET/SDH Label [GMPLS SONET].

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. 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 to the TLV of type 3, 4, 5 of the IF_ID_RSVP_HOP object (see [RFC3471, RFC3473]).

4.2.3.11 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 10 of [OIF2003.248])), 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:

0	1	2	3
0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9	0 1 2 3 4 5 6 7 8 9 0 1

```

+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
|          Length (8)          |          Type(5)          |          Sub-Type (1) |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| service Level |          Reserved (0)          |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

```

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

4.2.4 GENERALIZED_LABEL_REQUEST Object (Class-Num = 19 [RFC 3473])

The format of the GENERALIZED_LABEL_REQUEST object is defined in GMPLS RSVP-TE. For UNI 1.0, a node MUST support SONET/SDH LSP encoding type and TDM Switching Type.

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

The format of the IPv4_RESV_CONFIRM object is defined in [RFC2205]. For UNI 1.0, the IPv4 receiver address is set to the destination UNI-C’s Node ID at the destination UNI, and it is set to the source UNI-N’s Node ID at the source UNI.

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

The format of the IPv4_ERROR_SPEC object is defined in [RFC2205]. For UNI 1.0, the IPv4 Error Node Address is set to the Node 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.

Note: IF_ID ERROR_SPEC Object (Class-Num = 6 [RFC-3471])

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.

4.2.7 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.

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

The format of MESSAGE_ID object is defined in [RFC296]. For UNI 1.0, the Ack_Desired flag in a MESSAGE_ID object MUST be set in trigger messages, PathErr, ResvErr 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.

4.2.9 IPv4_IF_ID_RSVP_HOP Object (Class-Num = 3, [RFC-3473])

The format of the IPv4_IF_ID_RSVP_HOP object is defined in GMPLS RSVP-TE. For UNI 1.0, 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 TLVs 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 TLVs. In this object, TLVs 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 that traverses a bundled link, it is possible to specify a downstream data link that differs from the upstream data link.

A node's Node identifier is used to fill an IP address field in the IPv4 IF_ID_RSVP_HOP object.

Note: The IF_ID RSVP_HOP object **SHOULD NOT** be used when no TLVs are needed.

5 RSVP Code Points

UNI signaling using RSVP defined in this section utilizes a number of new objects. Many of these objects have been defined as part of the GMPLS RSVP-TE signaling specification [RFC3473] and some new objects have been defined specifically for UNI signaling. All these objects must have unique RSVP class number and class type values assigned. Furthermore, notification messages defined in this section have error codes for which unique values must be assigned. These assignments are performed by the Internet Assigned Numbers Authority (IANA) based on policies that can be found in [RFC2434].

The following table 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	
"Bad EXPLICIT_ROUTE object"	24/1	[RFC3209]
"Routing Problem/Label Set"	24/11	[RFC3473]
"Routing Problem: Service level not available"	24/101	
"Routing Problem: Invalid/Unknown connection ID"	24/102	
"Routing Problem: Connection parameters not supported"	21/02	[RFC2205]
"Policy Control Failure: Unauthorized sender"	2/100	
"Policy Control Failure: Unauthorized receiver"	2/101	

6 References

- [OIF2003.248] “User Network Interface (UNI) 1.0 Signaling Specification, Release 2: Common Part,” OIF Contribution [OIF2003.248], July, 2003.
- [RFC3212] Jamoussi, B. Ed, “Constraint-Based LSP Setup Using LDP,” IETF RFC 3212.
- [RFC3471] L. Berger, et. al, “Generalized MPLS - Signaling Functional Description, IETF RFC 3471.
- [RFC3472] P. Ashwood-Smith, et. al, “Generalized MPLS - CR-LDP Signaling Functional Description,” IETF RFC 3472.
- [RFC3473] L. Berger, et. al, “Generalized MPLS - RSVP-TE Signaling Functional Description,” IETF RFC 3473.
- [GMPLS SONET] E. Mannie, et. al, “GMPLS Extensions for SONET and SDH Control,” IETF RFC to be: draft-ietf-ccamp-gmpls-sonet-sdh-08.txt.
- [LMP] J. P. Lang, et al., “Link Management Protocol,” IETF RFC to be: draft-ietf-ccamp-lmp-09.txt.
- [OIF2000.155] J. Strand, et. al., “Carrier Optical Services Framework and Associated Requirements for UNI,” OIF2000.155, September, 2000.
- [RFC1662] W. Simpson, Ed., “PPP in HDLC-Like Framing”, IETF RFC 1662.
- [RFC1701] S. Henks, et. al, “Generic Routing Encapsulation (GRE)”, IETF RFC 1701.
- [RFC1853] W. Simpson, et. al, “IP in IP tunnel”, IETF RFC 1853.
- [RFC2104] H. Krawczyk, M. Bellare and R. Canetti, “HMAC: Keyed-Hashing for Message Authentication,” IETF RFC 2104.
- [RFC2119] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels“, IETF RFC 2119.
- [RFC2205] R. Braden, Ed., “Resource Reservation Protocol (RSVP) - Version 1 Functional Specification,” IETF RFC 2205, September, 1997.
- [RFC2209] RSVP - Message Processing Rules”, IETF RFC 2209.
- [RFC2385] A. Heffernan, “Protection of BGP Sessions via the TCP MD5 Signature Option,” IETF RFC 2385.
- [RFC2402] S. Kent and R. Atkinson, “IP Authentication Header,” IETF RFC 2402.
- [RFC2403] C. Madson and R. Glenn, “The Use of HMAC-MD5-96 within ESP and AH,” IETF RFC 2403.
- [RFC2406] S. Kent and R. Atkinson, “IP Encapsulating Security Payload (ESP),” IETF RFC 2406.
- [RFC2407] D. Piper, “The Internet IP Security Domain of Interpretation for ISAKMP,” IETF RFC 2407.
- [RFC2409] D. Harkins and D. Carrel, “The Internet Key Exchange (IKE)”, IETF RFC 2409.
- [RFC2434] Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” IETF RFC 2434.
- [RFC2615] A. Malis and W. Simpson, “PPP over SONET/SDH,” IETF RFC 2615.
- [RFC2747] F. Baker et al. “RSVP Cryptographic Authentication,” IETF RFC 2747.
- [RFC2748] D. Durham, et al., “The COPS (Common Open Policy Service) Protocol,” IETF RFC2748.
- [RFC2750] S. Herzog, et al., “RSVP Extensions for Policy Control”, IETF RFC 2750.
- [RFC2752] S. Yadav, “Identity Representation for RSVP,” IETF RFC 2752, January 2000.
- [RFC2961] L. Berger, et al., “RSVP Refresh Overhead Reduction Extensions,” IETF RFC 2961.
- [RFC3036] L. Andersson, et. al., “LDP Specifications,” IETF RFC 3036.
- [RFC3209] D. Awduche, et. al, “Extensions to RSVP for LSP Tunnels,” IETF RFC 3209.

- [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”
- [G707] ITU-T Rec. G.707, Network Node Interface for the Synchronous Digital Hierarchy (SDH)
- [G7713.2] ITU-T Rec. G.7713.2, DCM Signalling Mechanism Using GMPLS RSVP-TE
- [G7713.3] ITU-T Rec. G.7713.3, DCM Signalling Mechanism Using GMPLS CR-LDP
- [G807] ITU-T Rec. G.807/Y.1301 (2001), Requirements for the Automatic Switched Transport Network (ASTN)
- [G8080] ITU-T Rec. G.8080/Y.1304 (2001), Architecture of the Automatically Switched Optical Network (ASON)

Appendix A: List of companies belonging to OIF when document was approved

Accelerant Networks
Acuid
Aeluros
Agere Systems
Agilent Technologies
Agility Communications
Alcatel
Altera
AMCC
America Online
Analog Devices
Analogix Semiconductor
Anritsu
Artisan Components
ASTRI
AT&T
Atrica Inc.
Avici Systems
Azna
Big Bear Networks
Bit Blitz Communications
Bookham Technology
Booz-Allen & Hamilton
Broadcom
Cadence Design Systems
Calient Networks
Caspian Networks
China Telecom
Chunghwa Telecom Labs
Ciena Corporation

Circadiant Systems
Cisco Systems
CIVCOM
CoreOptics
Corrigent Systems
Corvis Corporation
Cypress Semiconductor
Data Connection
Department of Defense
Diablo Technologies
ELEMATICS
Elisa Communications
Emcore
Ericsson
ETRI
FCI
Finisar Corporation
Flextronics
Force 10 Networks
Foxconn
France Telecom
Fujitsu
Furukawa America
Galazar Networks
Gennum Corporation
Harris Corporation
Helix AG
Hi/fn
Hitachi
Ibiden
IBM Corporation
IDT
Industrial Technology Research Institute
Infineon Technologies
Infinera
Intel
Intelligent Photonics Control
Interoute
Intune Technologies, Ltd.
Iolon
Japan Telecom
JDS Uniphase
Juniper Networks
KDDI R&D Laboratories
Kodeos Communications

KT Corporation
Lambda Optical Systems
Lattice Semiconductor
LSI Logic
Lucent
Lumentis
Marconi Communications
MCI
MergeOptics GmbH
Mindspeed
Mintera
Mitretek Systems
Mitsubishi Electric Corporation
Molex
Multiplex
Mysticom
Navtel Communications
NEC
NIST
Nortel Networks
NTT Corporation
OpNext
PhotonEx
Photuris, Inc.
Phyworks
PMC Sierra
Pontusys
Princeton Optronics
Procket Networks
Quake Technologies
Quellan
Qwest Communications
Sandia National Laboratories
Santur
SBC
Scintera Networks
Siemens
Silicon Access Networks
Silicon Laboratories
Silicon Logic Engineering
ST Microelectronics
StrataLight Communications
Sun Microsystems
Sycamore Networks
Tektronix

Telcordia Technologies
Telecom Italia Lab
Tellabs
Tellium
Teradyne
Texas Instruments
Toshiba Corporation
TriQuint Semiconductor
T-Systems/ Deutsche Telekom
Turin Networks
Tyco Electronics
Velio Communications
Verizon
Vitesse Semiconductor
W.L. Gore & Associates
Winchester Electronics
Xanoptix
Signal Technologies
Xilinx