INTERNATIONAL TELECOMMUNICATION UNION

ITU-T G.7713.2/Y.1704.2

TELECOMMUNICATION
STANDARDIZATION SECTOR
OF ITU

(03/2003)

SERIES G: TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

Digital terminal equipments – Operations, administration and maintenance features of transmission equipment

SERIES Y: GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

Internet protocol aspects – Operation, administration and maintenance

Distributed Call and Connection Management: Signalling mechanism using GMPLS RSVP-TE

ITU-T Recommendation G.7713.2/Y.1704.2

ITU-T G-SERIES RECOMMENDATIONS

TRANSMISSION SYSTEMS AND MEDIA, DIGITAL SYSTEMS AND NETWORKS

INTERNATIONAL TELEPHONE CONNECTIONS AND CIRCUITS	G.100–G.199
GENERAL CHARACTERISTICS COMMON TO ALL ANALOGUE CARRIER-TRANSMISSION SYSTEMS	G.200–G.299
INDIVIDUAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON METALLIC LINES	G.300–G.399
GENERAL CHARACTERISTICS OF INTERNATIONAL CARRIER TELEPHONE SYSTEMS ON RADIO-RELAY OR SATELLITE LINKS AND INTERCONNECTION WITH METALLIC LINES	G.400–G.449
COORDINATION OF RADIOTELEPHONY AND LINE TELEPHONY	G.450–G.499
TESTING EQUIPMENTS	G.500–G.599
TRANSMISSION MEDIA CHARACTERISTICS	G.600–G.699
DIGITAL TERMINAL EQUIPMENTS	G.700–G.799
DIGITAL NETWORKS	G.800–G.899
DIGITAL SECTIONS AND DIGITAL LINE SYSTEM	G.900–G.999
QUALITY OF SERVICE AND PERFORMANCE	G.1000–G.1999
TRANSMISSION MEDIA CHARACTERISTICS	G.6000–G.6999
DIGITAL TERMINAL EQUIPMENTS	G.7000–G.7999
General	G.7000–G.7099
Coding of analogue signals by pulse code modulation	G.7100–G.7199
Coding of analogue signals by methods other than PCM	G.7200–G.7299
Principal characteristics of primary multiplex equipment	G.7300–G.7399
Principal characteristics of second order multiplex equipment	G.7400–G.7499
Principal characteristics of higher order multiplex equipment	G.7500–G.7599
Principal characteristics of transcoder and digital multiplication equipment	G.7600–G.7699
Operations, administration and maintenance features of transmission equipment	**G.7700–G.7799**
Principal characteristics of multiplexing equipment for the synchronous digital hierarchy	G.7800–G.7899
Other terminal equipment	G.7900–G.7999
DIGITAL NETWORKS	G.8000–G.8999

For further details, please refer to the list of ITU-T Recommendations.

ITU-T Recommendation G.7713.2/Y.1704.2

Distributed Call and Connection Management: Signalling mechanism using GMPLS RSVP-TE

Summary

This Recommendation meets the requirements of ITU-T Rec. G.7713/Y.1704 and is functionally similar to ITU-T Recs G.7713.1/Y.1704.1 and G.7713.3/Y.1704.3. This Recommendation covers the areas associated with the signalling aspects of automatic switched transport network (ASTN). Specifically it provides the signalling protocol based on the GMPLS RSVP-TE. This Recommendation focuses on the UNI and E-NNI interface specification. While these protocol specifications are generally applicable to the I-NNI as well, the I-NNI interface specification is for further study. This Recommendation encompasses support for Soft Permanent Connection (SPC) services. This version also includes support for Switched Connection (SC) services for intra-carrier application. As such, name translation/directory services and call capability sets are not included. This signalling protocol is used for the communications of call controller, connection controller and link resource manager. Areas covered include:

– message specifications;

– attribute specifications;

– signal flows.

This Recommendation does not cover any aspects related to routing, or automatic discovery.

Source

ITU-T Recommendation G.7713.2/Y.1704.2 was prepared by ITU-T Study Group 15 (2001-2004) and approved under the WTSA Resolution 1 procedure on 16 March 2003.

History

This Recommendation forms part of a suite of Recommendations covering the full functionality of the automatic switched transport network (ASTN).

Document history	
Issue	**Notes**
0.1	Version 0.1 of G.7713.2/Y.1704.2
0.2	Modifications based on WD40 of 2/02 Q.14/15 meeting
0.3	Editorial modifications to provide clarifications in some sections of the Recommendation
0.4	Included new text on SPC, Call, Recovery
0.5	Modifications to text for call processing
0.6	Modification based on comments received at 10/02 Q.14/15 meeting
0.7	Accept the revision mark of version 0.6
0.8	Editorial changes to align with IANA codepoint assignments

Keywords

Automatic switched optical network, Automatic switched transport network, Distributed call and connection management, Exterior network-node interface, GMPLS RSVP-TE, User network interface.

FOREWORD

The International Telecommunication Union (ITU) is the United Nations specialized agency in the field of telecommunications. The ITU Telecommunication Standardization Sector (ITU-T) is a permanent organ of ITU. ITU-T is responsible for studying technical, operating and tariff questions and issuing Recommendations on them with a view to standardizing telecommunications on a worldwide basis.

The World Telecommunication Standardization Assembly (WTSA), which meets every four years, establishes the topics for study by the ITU-T study groups which, in turn, produce Recommendations on these topics.

The approval of ITU-T Recommendations is covered by the procedure laid down in WTSA Resolution 1.

In some areas of information technology which fall within ITU-T's purview, the necessary standards are prepared on a collaborative basis with ISO and IEC.

NOTE

In this Recommendation, the expression "Administration" is used for conciseness to indicate both a telecommunication administration and a recognized operating agency.

INTELLECTUAL PROPERTY RIGHTS

ITU draws attention to the possibility that the practice or implementation of this Recommendation may involve the use of a claimed Intellectual Property Right. ITU takes no position concerning the evidence, validity or applicability of claimed Intellectual Property Rights, whether asserted by ITU members or others outside of the Recommendation development process.

As of the date of approval of this Recommendation, ITU had not received notice of intellectual property, protected by patents, which may be required to implement this Recommendation. However, implementors are cautioned that this may not represent the latest information and are therefore strongly urged to consult the TSB patent database.

CONTENTS

ITU-T Recommendation G.7713.2/Y.1704.2

Distributed Call and Connection Management: Signalling mechanism using GMPLS RSVP-TE

1 Scope

This Recommendation covers the areas associated with the signalling aspects of automatic switched transport network (ASTN). Specifically it provides the signalling protocol based on the GMPLS RSVP-TE. This Recommendation focuses on the UNI and E-NNI interface specification. While these protocol specifications are generally applicable to the I-NNI as well, the I-NNI interface specification is for further study. This Recommendation encompasses support for Soft Permanent Connection (SPC) services. This version also includes support for Switched Connection (SC) services for intra-carrier application. As such, name translation/directory services and call capability sets are not included. This signalling protocol is used for the communications of call controller, connection controller and link resource manager. Areas covered include:

- message specifications;
- attribute specifications;
- signal flows.

This Recommendation provides the attribute and message specification, and signalling exchange that allow support for hierarchical, source and step-by-step routing. Other areas of ASTN such as routing mechanism, parameters associated with routing mechanisms, discovery, and naming and addressing are not in the scope of this Recommendation.

This Recommendation uses the control plane architectural requirements in ITU-T Rec. G.8080/Y.1304 and the protocol neutral requirements as outlined in ITU-T Rec. G.7713/Y.1704 as the basis for the specification.

2 References

The following ITU-T Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below. A list of the currently valid ITU-T Recommendations is regularly published. The reference to a document within this Recommendation does not give it, as a stand-alone document, the status of a Recommendation.

- ITU-T Recommendation E.360.1 (2002), *Framework for QoS routing and related traffic engineering methods for IP-, ATM- and TDM-based multiservice networks*.
- ITU-T Recommendation G.703 (2001), *Physical/electrical characteristics of hierarchical digital interfaces*.
- ITU-T Recommendation G.707/Y.1322 (2000), *Network node interface for the Synchronous Digital Hierarchy (SDH)*.
- ITU-T Recommendation G.709/Y.1331 (2003), *Interfaces for the Optical Transport Network (OTN)*.
- ITU-T Recommendation G.7713/Y.1704 (2001), *Distributed Call and Connection Management (DCM)*.
- ITU-T Recommendation G.803 (2000), *Architecture of transport networks based on the Synchronous Digital Hierarchy (SDH)*.

- ITU-T Recommendation G.805 (2000), *Generic functional architecture of transport networks*.
- ITU-T Recommendation G.807/Y.1302 (2001), *Requirements for Automatic Switched Transport Network (ASTN)*.
- ITU-T Recommendation G.872 (2001), *Architecture of optical transport networks*.
- ITU-T Recommendation G.8080/Y.1304 (2001), *Architecture for Automatic Switched Optical Network (ASON)*.
- IETF RFC 2205 (1997), *Resource ReSerVation Protocol (RSVP) – Version 1 Functional Specification*.
- IETF RFC 2747 (2000), *RSVP Cryptographic Authentication*.
- IETF RFC 2750 (2000), *RSVP Extensions for Policy Control*.
- IETF RFC 2961 (2001), *RSVP Refresh Overhead Reduction Extensions*.
- IETF RFC 3097 (2001), *RSVP Cryptographic Authentication – Updated Message Type Value*.
- IETF RFC 3209 (2001), *RSVP-TE: Extensions to RSVP for LSP Tunnels*.
- IETF RFC 3471 (2003), *Generalized Multi-Protocol Label Switching (GMPLS) – Signalling Functional Description*.
- IETF RFC 3473 (2003), *Generalized Multi-Protocol Label Switching (GMPLS) Signalling – Resource Reservation Protocol – Traffic Engineering (RSVP-TE) Extensions*.
- OIF UNI-01.0 (2001), *User Network Interface (UNI) 1.0 signalling specification*.

3 Terms and definitions

The following terms are defined in ITU-T Rec. G.805:

- administrative domain;
- layer network;
- link connection;
- management domain;
- subnetwork;
- subnetwork connection.

The following terms are defined in ITU-T Rec. G.8080/Y.1304:

- agent;
- component;
- call controller;
- connection controller;
- connection admission control;
- route controller;
- neighbour discovery;
- link resource manager;
- policy;
- protocol controller;

- subnetwork point;
- subnetwork point pool.

The following terms are defined in ITU-T Rec. G.807/Y.1302:

- soft permanent connection;
- switched connection.

4 Abbreviations

This Recommendation uses the following abbreviations:

ASON	Automatic Switched Optical Network
ASTN	Automatic Switched Transport Network
CallC	Call Controller
CC	Connection Controller
CCC	Calling/Called Party Call Controller
DCM	Distributed Call and Connection Management
E-NNI	Exterior NNI
GMPLS	Generalized Multi-Protocol Label Switching
I-NNI	Interior NNI
LRM	Link Resource Manager
NCC	Network Call Controller
NNI	Network Node Interface
RSVP-TE	Resource reSerVation Protocol – Traffic Engineering
SC	Switched Connection
SNP	Subnetwork Point
SNPP	Subnetwork Point Pool
SPC	Soft Permanent Connection
UNI	User Network Interface

5 Conventions

In this Recommendation, the acronym GMPLS is used to denote the signalling protocol portion of GMPLS based on the GMPLS RSVP-TE and should be read as synonymous to GMPLS RSVP-TE.

6 Assumptions

This Recommendation assumes the messages and objects defined by RFC 2205, RFC 2961, RFC 3209, [RFC 3471 GMPLS-SIG], [RFC 3473 GMPLS-RSVP-TE], and OIF UNI-01.0 as the basis for the protocol specification for the ASON network.

ITU-T Rec. G.8080/Y.1304 defines a UNI Transport Resource Addresses for the bearer links at the UNI reference point. For this Recommendation, an instantiation of those addresses will follow the OIF Transport Network Address (TNA) from OIF UNI-01.0 which complies with the G.8080/Y.1304 architecture. Allowable address formats in the OIF TNA are IPv4, IPv6, and NSAP addresses.

Call routing services are assumed to be available that associate a UNI Transport Resource Addresses with internal routable addresses. This is not within the scope of this Recommendation.

Addressing of transport resources in the protocol is done by SNPP identifiers. A pair of these would identify an SNPP link. SNPP names are defined from transport name spaces (see clause 10/G.8080/Y.1304) and it is important to note that control plane names/addresses are not used for these. For example, neither routing controller nor connection manager identifiers are used for bearer link names.

The terms Quality of Service (QoS), Class of Service (CoS), and Grade of Service (GoS) with respect to the transport plane are used in this Recommendation in the sense of ITU-T Rec. E.360.1. It is expected that ASON specific characteristics and parameters will be associated with these terms in later versions of this Recommendation.

7 Overview and application of GMPLS RSVP-TE to distributed connection management

Figure 1 shows an overall view of control plane partition.

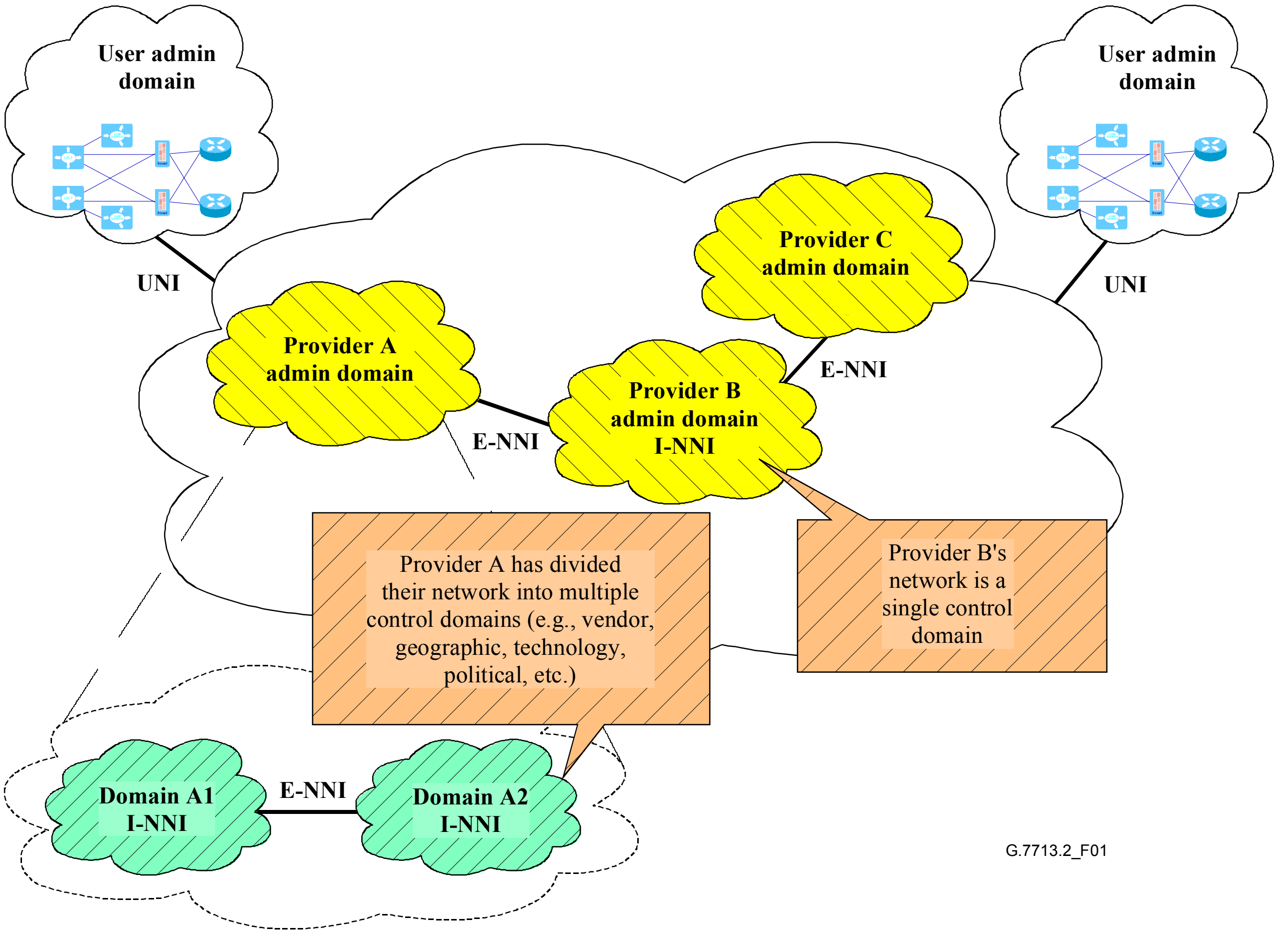

User admin domain
User admin domain
UNI
UNI
Provider C admin domain
Provider A admin domain
E-NNI
E-NNI
Provider B admin domain I-NNI
Provider A has divided their network into multiple control domains (e.g., vendor, geographic, technology, political, etc.)
Provider B's network is a single control domain
Domain A1 I-NNI
E-NNI
Domain A2 I-NNI
G.7713.2_F01

Figure 1/G.7713.2/Y.1704.2 – Overall view of control plane partition

7.1 Overview of GMPLS RSVP-TE

The Resource reSerVation Protocol (RSVP) is an IETF-defined protocol [RFC 2205] for establishing network resources for IP datagram sessions (or "flows"). The definition of RSVP consists of basic procedures, message and object formats for signalling in an IP network. RSVP with Traffic Engineering extensions (RSVP-TE) [RFC 3209] has been defined for establishing connections subject to routing constraints in an MPLS network. The RSVP-TE definition includes additional procedures, message and object formats over the base RSVP definition. Generalized MPLS (GMPLS) signalling extends basic MPLS signalling procedures and abstract messages to cover different types of switching applications such as time-division-multiplexing (TDM) switching, port switching, wavelength switching, etc. Figure 2 shows the message flow for the relevant messages defined for GMPLS RSVP-TE. See below for a detailed description of these messages.

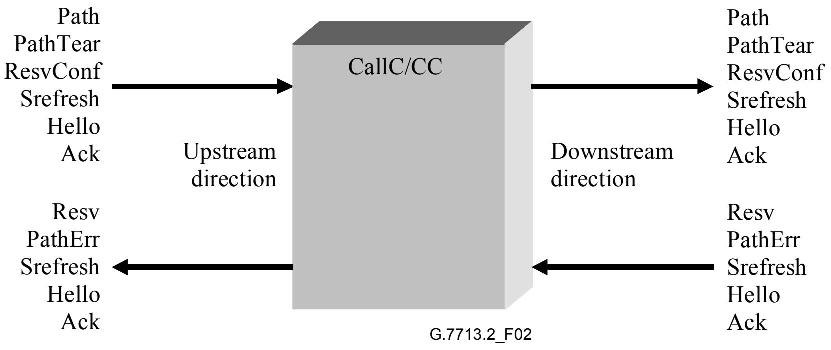


Figure 2/G.7713.2/Y.1704.2 – GMPLS RSVP-TE message flow directions

The GMPLS RSVP-TE is extended to support the requirements as specified in ITU-T Rec. G.7713/Y.1704. The GENERALIZED_UNI object is defined to encapsulate the A-end and Z-end names, as well as the CoS and GoS specifications to support the service requests at the UNI interface. See OIF UNI-01.0 for a definition of the GENERALIZED_UNI object. This information is also summarized as part of Annex A. In addition to this object, extensions are also made to support the basic call concept and to support the soft permanent connection service.

7.1.1 Support for basic call identifier

GMPLS RSVP-TE may be extended to support the basic call model as specified in ITU-T Rec. G.7713/Y.1704. This call model assumes that a request message handles both a call and its associated connections within the same message between the calling party call controller and the network call controller, as well as between the network call controller and the called party call controller. Addition or release of connections from an existing call is considered a call modify procedure, i.e., modification of specific connection attributes. As such, a call session remains constant across call modification operations. An established call may be identified via a call identifier object, CALL_ID. The format and structure of the CALL_ID information is:

– CALL_ID Class = 230, C-Type = 1

0 1 2 7 8 15	16 23	24 31
Length	Class-Num	C-type
… Call identifier …		

Where the following C-types are defined:

– C-Type = 1 (operator specific): The call identifier contains operator specific identifier.

– C-Type = 2 (globally unique): The call identifier contains globally unique part plus an operator specific identifier.

The following structures are defined for the call identifier:

– Call identifier: generic [Length*8-32] bit identifier. The number of bits for call identifier must be multiples of 32 bits, with minimum size of 32 bits.

The structure for the globally unique call identifier (to guarantee global uniqueness) is to concatenate a globally unique fixed ID (composed of country code, carrier code, unique access point code) with an operator specific ID (where the operator specific ID is composed of a source transport network element address – and a local identifier).

Therefore, a generic CALL_ID with global uniqueness includes <global ID> (composed of <country code> plus <carrier code> plus <unique access point code>) and <operator specific ID> (composed of <source transport network element address> plus <local identifier>). For a CALL_ID that only requires operator specific uniqueness only the <operator specific ID> is needed, while for a CALL_ID that requires to be globally unique both <global ID> and <operator specific ID> are needed.

The <global ID> shall consist of a three-character International Segment (the <country code>) and a twelve-character National Segment (the <carrier code> plus <unique access point code>). These characters shall be coded according to ITU-T Rec. T.50. The International Segment (IS) field provides a 3-character ISO 3166 Geographic/Political Country Code. The country code shall be based on the three-character uppercase alphabetic ISO 3166 Country Code (e.g., USA, FRA).

The National Segment (NS) field consists of two sub-fields: the ITU Carrier Code followed by a Unique Access Point Code. The ITU Carrier Code is a code assigned to a network operator/service provider, maintained by the ITU-T Telecommunication Standardization Bureau in association with ITU-T Rec. M.1400. This code shall consist of 1-6 left-justified characters, alphabetic, or leading alphabetic with trailing numeric. The unique access point code shall be a matter for the organization to which the country code and ITU carrier code have been assigned, provided that uniqueness is guaranteed. This code shall consist of 6-11 characters, with trailing NULL, completing the 12-character National Segment.

The format of the Call identifier field for C-Type = 1:

<table>
<tr><th>0 1 2 7</th><th>8 15</th><th>16 23</th><th>24 31</th></tr>
<tr><td colspan="2">Length</td><td>Class-Num</td><td>C-type</td></tr>
<tr><td>Type</td><td colspan="3">Reserved</td></tr>
<tr><td colspan="4">Source transport network element address
…</td></tr>
<tr><td colspan="4">Local Identifier</td></tr>
</table>

The format of the Call identifier field for C-Type = 2:

<table>
<tr><th>0 1 2 7</th><th>8 15</th><th>16 23</th><th>24 31</th></tr>
<tr><td colspan="2">Length</td><td>Class-Num</td><td>C-type</td></tr>
<tr><td>Type</td><td colspan="3">IS (3 bytes)</td></tr>
<tr><td colspan="4">NS (12 bytes)</td></tr>
<tr><td colspan="4">Source transport network element address
…</td></tr>
<tr><td colspan="4">Local Identifier</td></tr>
</table>

In both cases, a "Type" field is defined to indicate the type of format used for the source transport network element address. The Type field has the following meaning:

- For Type = 0x01, the source transport network element address is 4 bytes;
- For Type = 0x02, the source transport network element address is 16 bytes;
- For Type = 0x03, the source transport network element address is 20 bytes;
- For type = 0x04, the source transport network element address is 6 bytes;
- For type = 0x7f, the source transport network element address has the length defined by the vendor.

Source transport network element address:

- An address of the transport network element (SSN) controlled by the source network.

Local identifier:

- A 64-bit identifier that remains constant over the life of the call.

Note that if the source transport network element address is assigned from an address space that is globally unique, then the operator-specific CALL_ID may also be used to represent a globally unique CALL_ID. However, this is not guaranteed since this address may be assigned from an operator-specific address space.

The following processing rules are applicable to the CALL_ID object:

- For initial calls, the calling/originating party call controller must set the CALL_ID's C-Type and call identifier value to all-zeros.
- For a new call request, the source network call controller (SNCC) sets the appropriate C-type and value for the CALL_ID.
- For an existing call (in case CALL_ID is non-zero) the SNCC verifies existence of the call.
- The CALL_ID object on all messages MUST be sent from ingress call controller to egress call controller by all other (intermediate) controllers without altering.
- The destination user/client receiving the request uses the CALL_ID value as reference to the requested call between the source user and itself. Subsequent actions related to the call uses the CALL_ID as the reference identifier.

7.1.2 Support for Soft permanent connection

GMPLS RSVP-TE may be extended to support SPC services. An SPC service assumes that both source and destination user-to-network connection segments are provisioned while the network connection segment is set up via the control plane. For example, when an initial request is received from an external source (e.g., from management system), there is an implicit assumption that the control plane has adequate information to determine the specific destination (network-to-user) link connection to use. Support for SPC is provided via the SPC_LABEL object.

The SPC_LABEL is a sub-object of the GENERALIZED_UNI object, and has the same format and structure as the EGRESS_LABEL sub-object of the GENERALIZED_UNI object. The SPC_LABEL information is:

- SPC_LABEL (Type = 4, Sub-type = 2).

Note that to support the case of SPC, the GENERALIZED_UNI object is used. This object is used to support the SPC label information as well as service level and diversity specifications that are relevant to the SPC connection request. For a SPC request, the source and destination TNA addresses contain the addresses of the transport network elements controlled by the source the destination network call controllers, respectively. Thus the source TNA contains the address of the transport network element controlled by the source network call controller and the destination

TNA contains the address of the transport network element controlled by the destination network call controller.

7.2 Defect handling of GMPLS RSVP-TE

There are different types of defects that may affect the control plane. These defects may range from a simple signalling channel failure to multiple control plane node failures. The control plane needs to support appropriate behaviours to recover from these defects, initially attempting to recover from failures based on local control plane mechanisms, local interaction with the transport plane, and subsequently attempting to recover based on control plane interactions with external components. General guidelines for defect handling include:

- Control plane failures are notified to the management plane. The management plane may direct the control plane to take certain actions due to the failure. These actions may include entering into a self-refresh state, cleaning up of partial connections, release of certain connections, or other protocol-specific actions for state maintenance and recovery.
- A control plane node may provide a persistent storage of relevant information, such as call and connection state information, configuration information, and control plane neighbour information.
- After repair if connection/call states cannot be recovered, the control plane node may communicate with an external component to attempt state information recovery. External components may include neighbour control plane nodes or a persistent storage provided by a centralised (e.g., management plane) component. Note that although the restart mechanism allows neighbour control plane nodes to automatically recover (and thus infer) the states of calls/connections, this mechanism can also be used for verification of neighbour states while the persistent storage provides the local recovery of lost state. In this case, if during the Hello synchronization the restarting node determines that a neighbour does not support state recovery (i.e., local state recovery only), and the restarting node maintains its state on a per neighbour basis, the restarting node should immediately consider the Recovery as completed.
- A control plane node notifies the management plane of the inability to recover (subset of) relevant information (e.g., inability to synchronize state of connections). The management plane may respond with the following actions (the default control plane action should be to retain the connections):
  - Release the impacted connections.
  - Retain the impacted connections. In this case, a connection may remain non-synchronized from the control plane perspective; however, the connection may remain valid.
- A control plane node (after recovering from node failure) may not be able to recover neighbour connection state from its local persistent storage and thus may lose information on connections. In this case the control plane node should request an external controller (e.g., the management system) for information to recover the connections. Similarly call state may be un-recovered and require management intervention to resolve. Specifics of the interactions between the control plane and management plane are beyond the scope of this Recommendation.

Thus, as a general rule:

- A control plane failure must not result in the release of established connections. Setup requests in the process of been 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).

– Additional actions by the control plane may be dependent on provisioned default behaviour for a particular type of connection.

However, a transport plane node failure may result in the release of established connections. This depends on the type of connection and the service level associated with each connection. For example, a "best-effort unprotected" connection may be released during a transport plane node failure while a "protected" connection must be restored (or maintained) based on the service level specification associated with that connection. Note that even in the case of a protected connection, the original connection may be released while a new connection is set up (this also depends on the type of protection used for the particular connection).

In the case of an initial failure to set up the call, an error message is sent upstream towards the source call controller. The source call controller, upon receiving indication of a failure (e.g., information in the ERROR_SPEC object) to set up a call, may initiate a retry of the call request. Crankback is a mechanism that supports the ability of the control plane to automatically retry the establishment of a connection, using an alternative path, when a connection setup request fails. The route controller may use the information returned in the error message to determine an alternate path. Specification for the crankback mechanism is for further study.

7.2.1 Signalling channel failure

In the case of signalling channel failure between control plane nodes A and B, connection #1, #4 and #6 will be affected. As the RSVP-TE state refresh are point-to-point, there will be three Path refresh messages (or a single Srefresh message) between nodes A and B that are disrupted. According to the behaviours described above, both nodes A and B will notify the management plane of the communications failure between nodes A and B. Management plane determines that the failure is a communications channel failure (since it is still receiving notifications from both control plane nodes) and thus instructs both nodes to continue with self-refresh. Figure 3 illustrates the failure scenario.

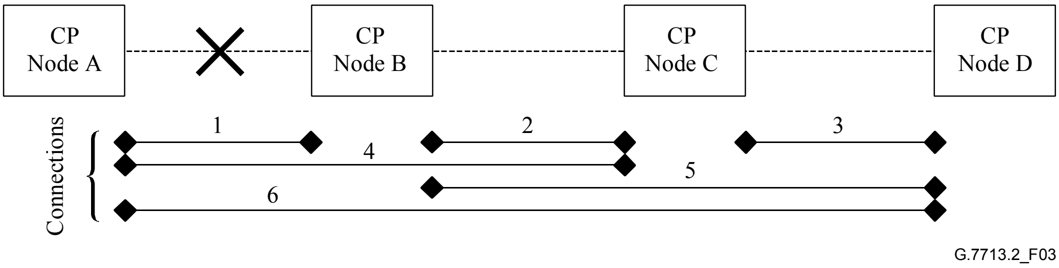


Figure 3/G.7713.2/Y.1704.2 – Signalling channel failure between control plane nodes A and B

Upon repair of the signalling channel, nodes A and B initiate mechanism (as per GMPLS RSVP-TE restart mechanism – send Srefresh for state verification) for synchronizing the states of the affected connections and calls (e.g., states of connections #1, #4 and #6). If during the synchronization procedure connection states are found out of synchronization, a notification is sent to the management plane according to the behaviour described above.

7.2.2 Single control plane node failure

In the case of a control plane node failure, for example failure of node B in Figure 4, both neighbouring nodes A and C will notify the management plane of loss of communication with node B. The management plane then determines whether any connections (and calls) are affected by the control plane node failure. For connections (and calls) that are not affected, it instructs nodes A and C to enter self-refresh procedures; for connections affected, it may instruct nodes A and C to initiate connection release, e.g., if there is also an associated transport plane failure.

Note that in addition to management plane notification, connections that are disrupted due to the control plane node failure will be detected by nodes A and C (e.g., LOS) and as such the control plane may also initiate release of the connection based on this status for certain types of connections, e.g., "best effort" connections. Figure 4 illustrates this scenario.

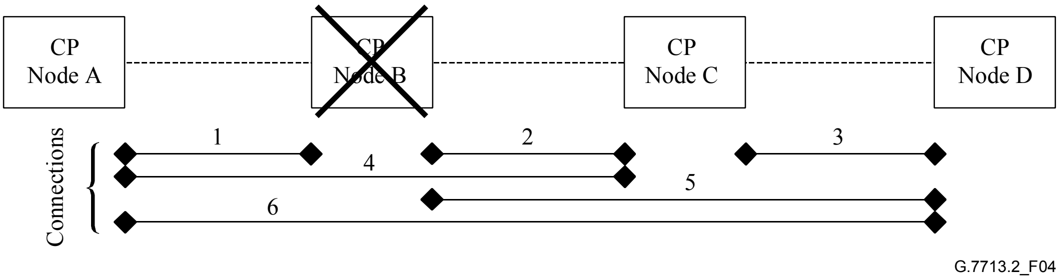


Figure 4/G.7713.2/Y.1704.2 – Control plane node B failure

In the case of self-refresh, upon node B's recovery, node B restores the connection states based on its "last-known" connection state status via a local persistent storage. Two possible scenarios exist:

– Node B's connections are lost due also to transport plane failure (note that for this case the connections (#1, #2, #4, #5, #6) have already been released by the non-failed control plane node since there would be indications of service interruption from the transport plane to the control plane – via transport plane's OAM mechanism); in this case node B may be instructed by the management system (if it did not already perform this via examining its fabric status) to remove the states of the affected connections.

– Node B's connections are retained (i.e., connections #1, #2, #4, #5, #6 remain active); in this case, node B initiates recovery procedure with node A and node C.

  NOTE – This may be done simultaneously or per-neighbour.

  Thus, node B's neighbour may not distinguish the node failure from a signalling channel failure. If status of any active connections is out-of-sync, then the management plane needs to provide information to correct the synchronization as per above behaviours.

7.3 Example signalling flows

This clause illustrates basic signal flows for the GMPLS RSVP-TE for Soft Permanent Connections and Switched Connections. The basic signal flows considered within the document are operations associated with supporting Soft Permanent Connections and Switched Connections. A general description of the signalling flow for the setup procedure:

– A Path message is sent from the source to the destination to request a connection.

– Upon reception of the Path message by the destination node, an RSVP session is set up between the source and destination.

– The destination node responds to the Path message via one of two messages sent in the upstream direction:

  - Resv (for normal setup response); or
  - PathErr (for error in the setup procedure); in this case the connection is not set up. If the Path state is not removed, then an explicit PathTear is needed to remove any extraneous states.

– Upon reception of the Resv message by the source node, an optional ResvConf message may be sent. This is dependent on the RESV_CONFIRM object within the Resv message.

7.3.1 Example SPC signal flow

Figures 5 and 6 show an example signal flow for SPC request. For a SPC connection, it is assumed that the user-to-network link connection is provisioned and information is provided to the control plane regarding the identity of the link connection. Setting up the switched portion of the SPC connection remains the same as that for setting up a switched connection. This is true of the SPC request signal flow as well.

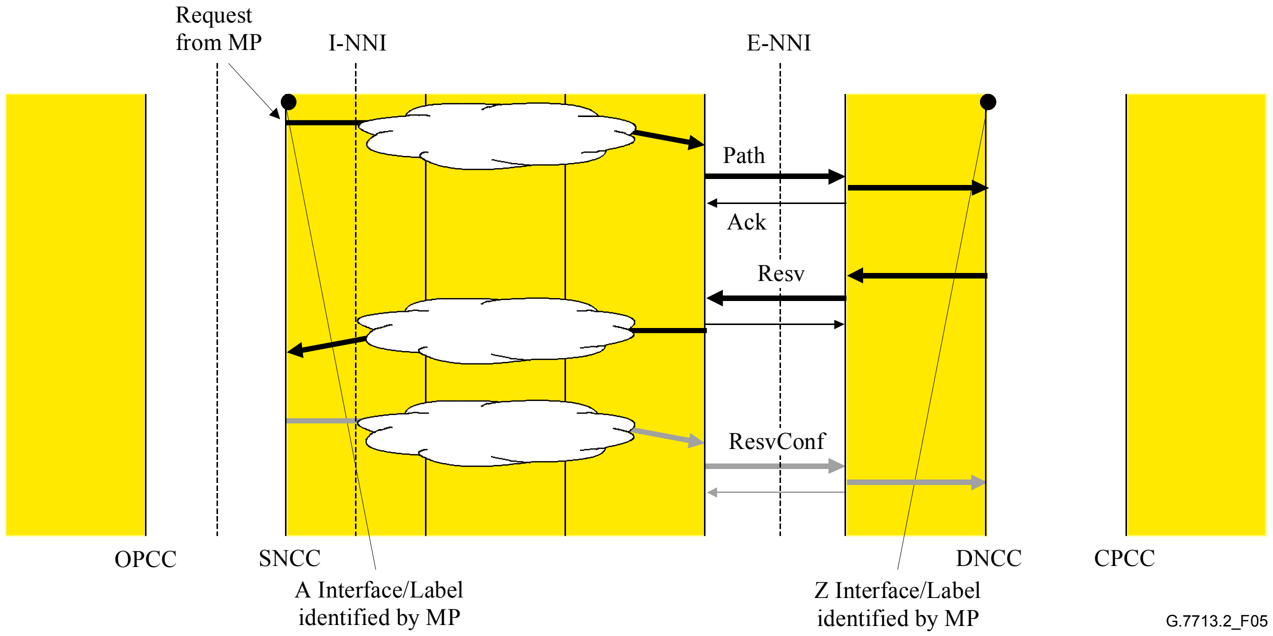
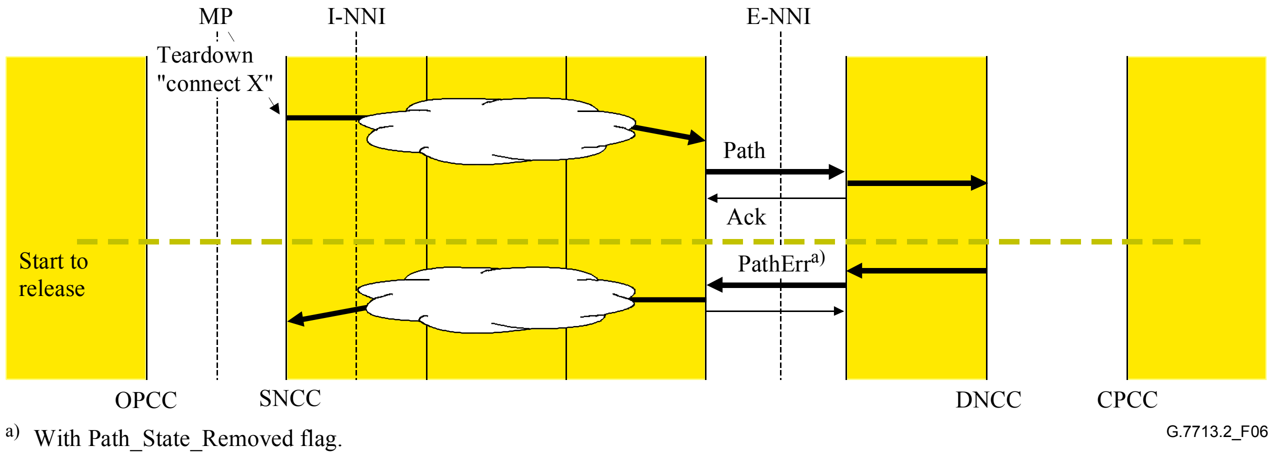


Figure 5/G.7713.2/Y.1704.2 – Basic soft permanent connection setup



[a] With Path_State_Removed flag.

Figure 6/G.7713.2/Y.1704.2 – Basic SPC release

7.3.2 Basic SC signal flow

Figure 7 illustrates the setup of the SC. To set up a SC call, a user initiates the request across the UNI interface. The request is propagated across the network to the destination user. Upon verification/acceptance of the request, a positive indication is sent to the source user. Optionally, the source user may also transmit a final response. This third phase message is introduced to support explicit destination notification of completed connection setup.

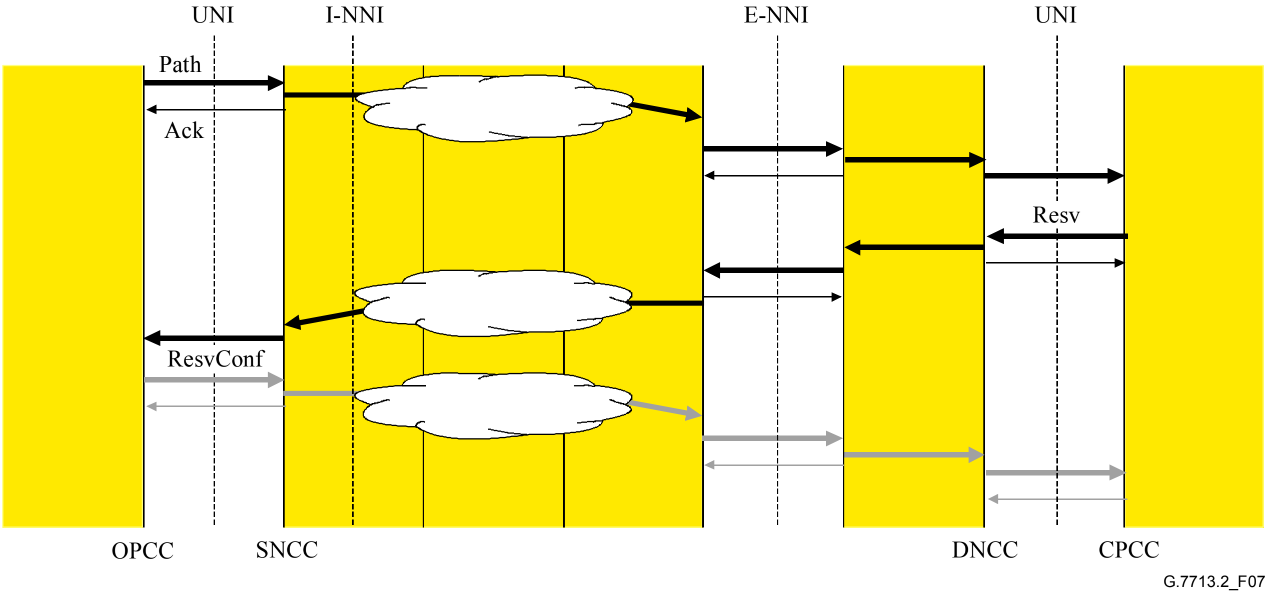
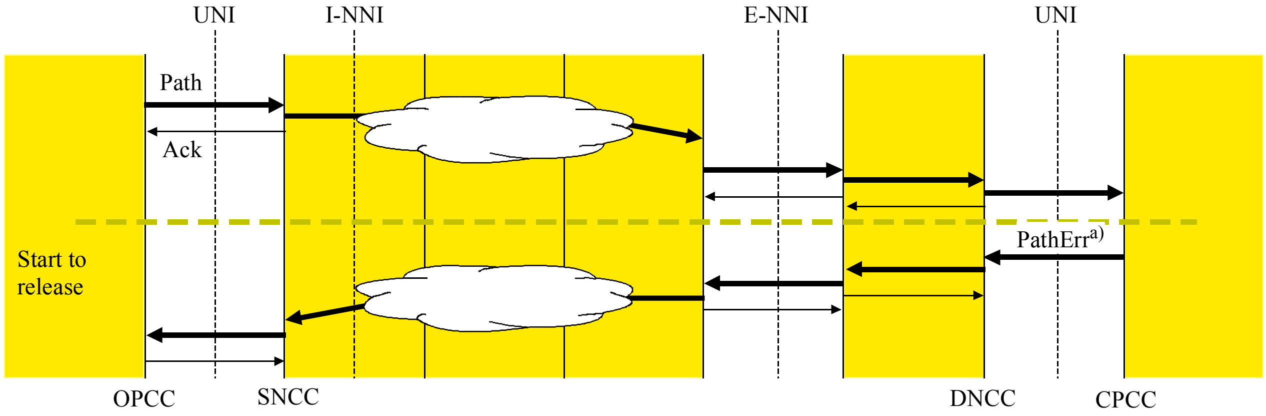


Figure 7/G.7713.2/Y.1704.2 – Basic SC setup

In the case for SC release, a release request may be initiated by different controllers, e.g., either originating party call controller, called party call controller, or any one of network call controllers may initiate the release. Figure 8a illustrates the originating party initiated release request, Figure 8b illustrates the called party initiated release request, and Figures 8c to 8f illustrate a network call controller initiating a release request.



[a)] With Path_State_Removed flag.

Figure 8a/G.7713.2/Y.1704.2 – Basic SC release (OPCC initiated)

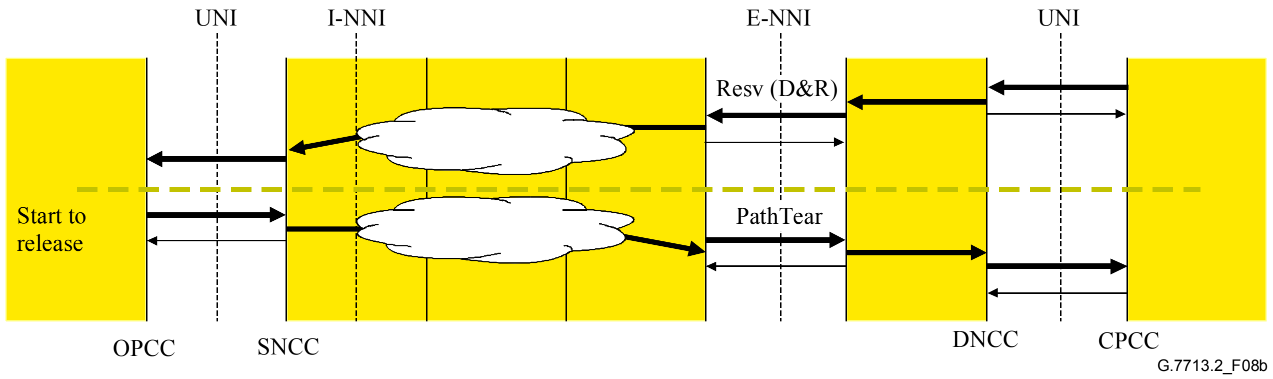


Figure 8b/G.7713.2/Y.1704.2 – Basic SC release (CPCC initiated)

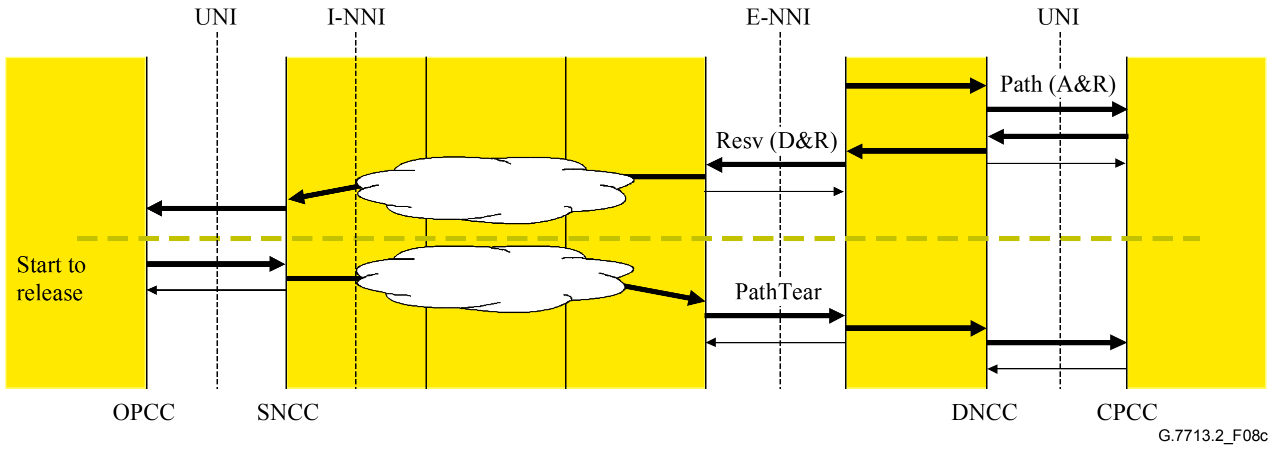
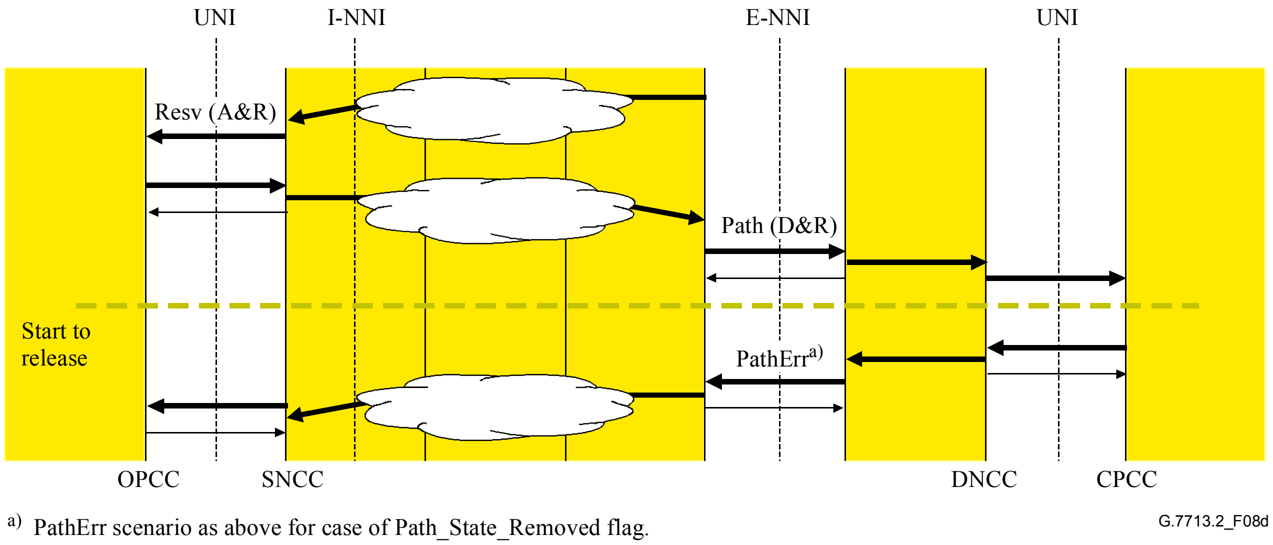


Figure 8c/G.7713.2/Y.1704.2 – SC release: Intermediate controller initiated (towards UNI downstream)



[a] PathErr scenario as above for case of Path_State_Removed flag.

Figure 8d/G.7713.2/Y.1704.2 – SC release: Intermediate controller initiated (towards UNI upstream)

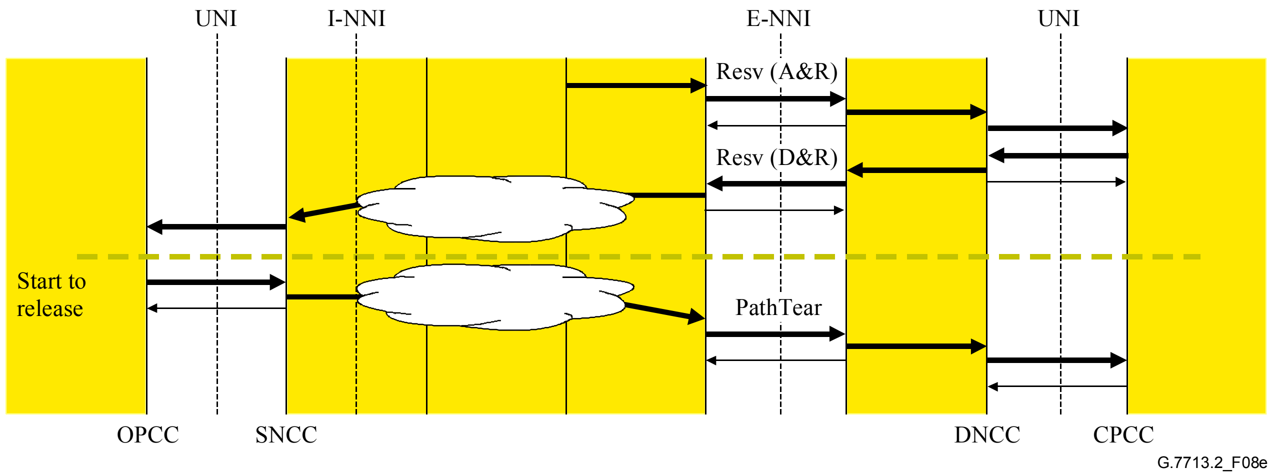
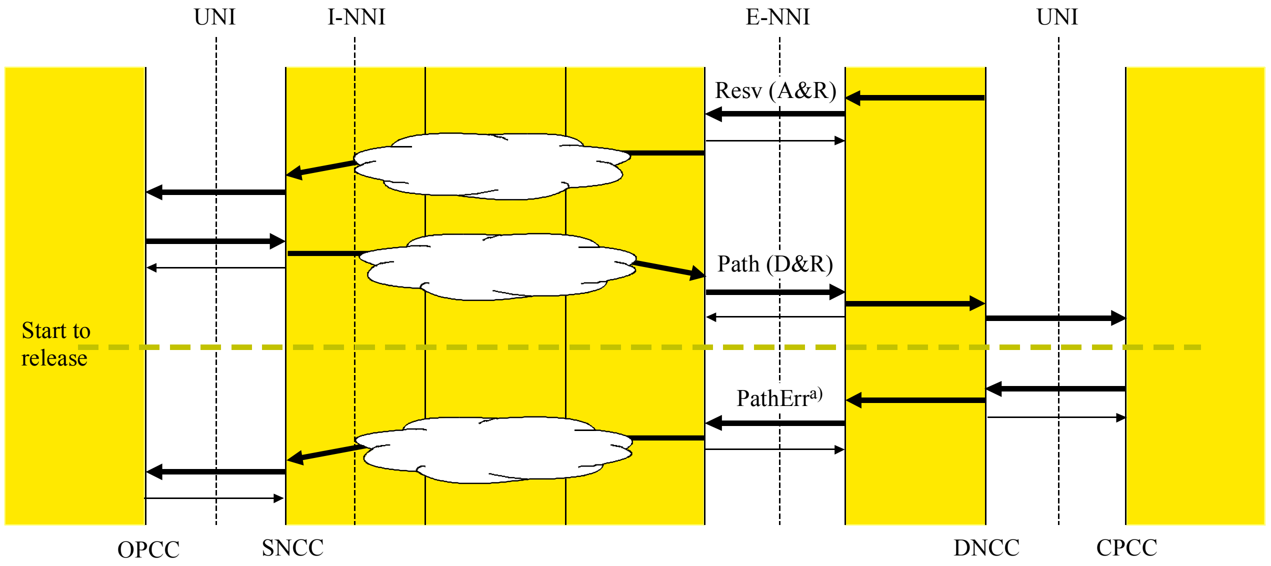


Figure 8e/G.7713.2/Y.1704.2 – SC release: Intermediate controller initiated (towards E-NNI downstream)

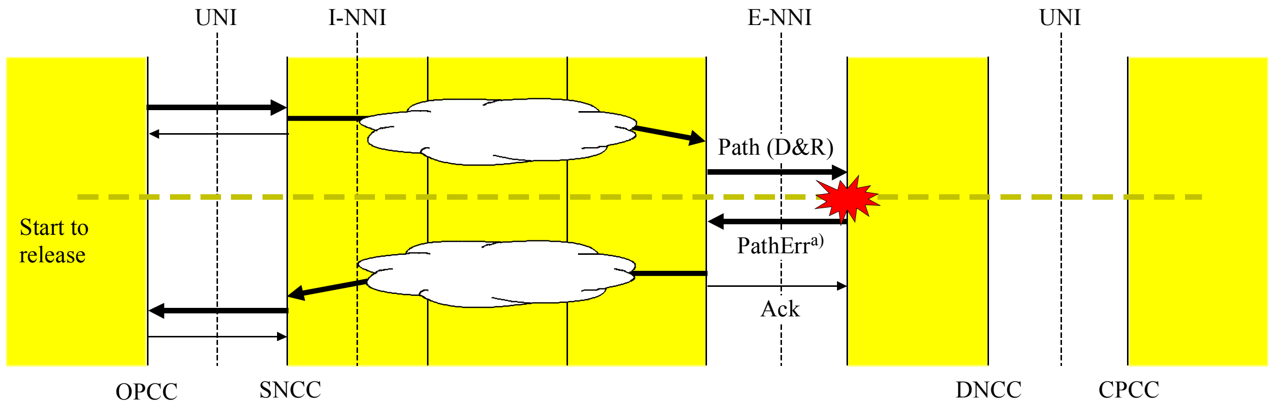


[a] PathErr scenario as above for case of Path_State_Removed flag.

Figure 8f/G.7713.2/Y.1704.2 – SC release: Intermediate controller initiated (towards E-NNI upstream)

7.3.3 Setup rejection signal flow

Figure 9 illustrates the case where a request to set up a connection is immediately rejected by an intermediate node. This may occur due to various reasons as described in ITU-T Rec. G.7713/Y.1704, e.g., during the initial request, no resources were available.

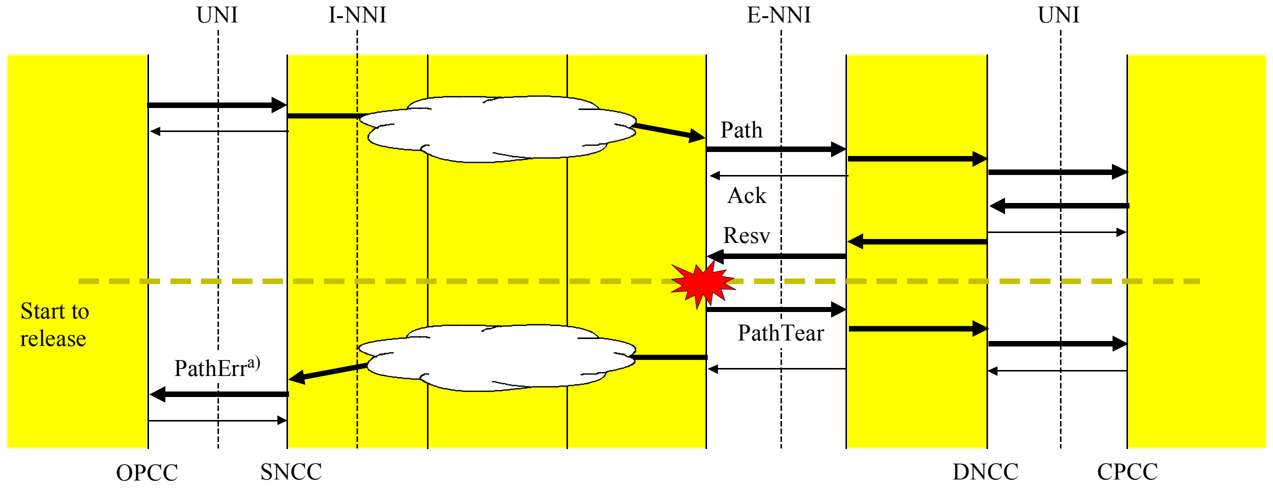


[a] PathErr with Path_State_Removed flag set.

G.7713.2_F09

Figure 9/G.7713.2/Y.1704.2 – Setup: Rejection by intermediate node (with Path_State_Removed set)

Figure 10 illustrates the case where a request to set up a connection is rejected by an intermediate node after receiving an indication from the destination. For example, this may occur due to inability to complete assignment of a resource to the requested connection due to transport plane error.

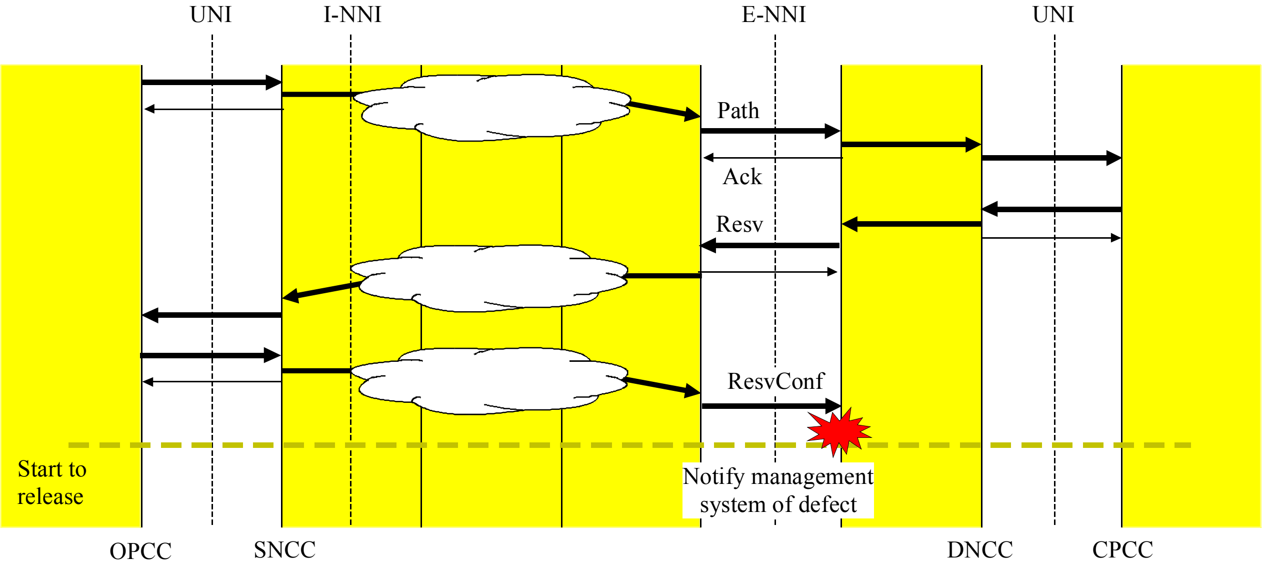


[a] PathErr scenario as above for case of Path_State_Removed flag set.

G.7713.2_F010

Figure 10/G.7713.2/Y.1704.2 – Setup: Rejection by intermediate node after receiving indication

Figure 11 illustrates the case where a request to set up a connection is rejected by an intermediate node after receiving a confirmation from the source. For example, this may occur due to loss of message (either loss of the ResvConf message, or the Ack message associated with the ResvConf message). In this case the connection has in fact been established, and connection monitoring (if any) may be in place. As such this defect constitutes a control plane defect, and thus should not be service impacting. A possible action should be to notify the management system of the control plane defect.



a) PathErr scenario as above for case of Path_State_Removed flag.

G.7713.2_F11

Figure 11/G.7713.2/Y.1704.2 – Setup: Rejection by intermediate node after receiving confirmation

8 GMPLS RSVP-TE messages

A GMPLS RSVP-TE message format is based on the basic structure as defined by RFC 2205. An RSVP message is composed of a common header plus a number of objects specific to each message type. The structure of the common header is shown in Table 1:

Table 1/G.7713.2/Y.1704.2 – Common header

0 1 2 7 8 15 16 23 24 31

Vers	Flags	Msg type	RSVP checksum
Send_TTL		(Reserved)	RSVP length

Definitions of the fields may be found in RFC 2205, with specific message type extensions provided by RFC 2961 and RFC 3209. For clarity, the message type field is re-produced below:

Msg Type:

1: Path

2: Resv

3: PathErr

5: PathTear

7: ResvConf

13: Ack

15: Srefresh

20: Hello

21: Notify

8.1 Path

This message is modified from definitions in RFC 2205, RFC 2961 and RFC 3209 with further extensions to support distributed connection management.

This message is used to:

- Initiate a connection setup request.
- Initiate a source-initiated release request (using ADMIN_STATUS with D and R bit set).
- Initiate an intermediate-initiated downstream release request (using ADMIN_STATUS with A and R bit set).
- Respond to a received Resv (with A and R bit set) connection release request (using ADMIN_STATUS with D and R bit set).

```
<Path Message> ::=
          <Common Header>
          [ <INTEGRITY> ]
          [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
          [ <MESSAGE_ID> ]
          <SESSION>
          <RSVP_HOP>
          <TIME_VALUES>
          [ <EXPLICIT_ROUTE> ]
          <LABEL_REQUEST>
          <CALL_ID>
          [ <PROTECTION> ]
          [ <LABEL_SET> ... ]
            [ <SESSION_ATTRIBUTE> ]
            [ <NOTIFY_REQUEST> ]
            [ <ADMIN_STATUS> ]
            <GENERALIZED_UNI>
            [ <POLICY_DATA> ... ]
          <sender descriptor>
```

The format of the sender description for unidirectional LSPs is:

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

The format of the sender description for bidirectional LSPs is:

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

The <common header> must come first before any objects. When present, the <INTEGRITY> object must precede all other objects.

8.2 Resv

This message is modified from definitions in RFC 2205, RFC 2961 and RFC 3209 with further extensions to support distributed connection management.

This message is used to:

- Respond to a connection setup request indicated by a Path message.
- Initiate a destination-initiated release request (using ADMIN_STATUS with D and R bit set).
- Initiate an intermediate-initiated upstream release request (using ADMIN_STATUS with A and R bit set).
- Respond to a received Path (with A and R bit set) connection release request (using ADMIN_STATUS with D and R bit set).

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

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

<FF flow descriptor list> ::=
     <FLOWSPEC>
     <FILTER_SPEC>
     <LABEL>
     [ <RECORD_ROUTE> ] | <FF flow descriptor list>
     <FF flow descriptor>

<FF flow descriptor> ::=
     [ <FLOWSPEC> ]
     <FILTER_SPEC>
     <LABEL>
     [ <RECORD_ROUTE> ]

<SE flow descriptor> ::=
     <FLOWSPEC>
     <SE filter spec list>

<SE filter spec list> ::=
     <SE filter spec> | <SE filter spec list>
     <SE filter spec>

<SE filter spec> ::=
     <FILTER_SPEC>
     <LABEL>
     [ <RECORD_ROUTE> ]
```

The <common header> must come first before any objects. When present, the <INTEGRITY> object must precede all other objects. The <STYLE> and <flow descriptor list> must come last after all other objects.

8.3 ResvConf

This message is modified from definitions in RFC 2205 by RFC 2961. No additional modifications are necessary to support distributed connection management. For clarity, the format of this message is re-produced below:

This message is used to:

– Respond to a Resv connection setup request.

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

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

The <RESV_CONFIRM> object is copied from the same object in the Resv message. For <ERROR_SPEC>, the error code and error value are "0/0" to indicate confirmation.

8.4 PathTear

This message is modified from definitions in RFC 2205 by RFC 2961. No additional modifications are necessary to support distributed connection management. For clarity, the format of this message is re-produced below:

This message is used to:

– Respond to a Resv (with D & R bit set) connection release request.

– Respond to a PathErr (without Path_State_Removed flag set) during setup and release operations.

– Be sent as a result of unsuccessful setup operation (when no response is received to sending Path message).

– Be sent as a result of unsuccessful release request (when no response is received to sending Path or Resv message with (A or D) & R bits set).

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

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

The <SENDER_TSPEC> and <ADSPEC> must be ignored.

8.5 PathErr

This message is modified from definitions in RFC 2205 and RFC 2961, with further extensions to support distributed connection management.

This message is used to:

- Respond to a Path connection setup request when the connection cannot be set up successfully (using ERROR_SPEC with Path_State_Removed flag set).
- Respond to a Path (D & R bit set) connection release request (using ERROR_SPEC with Path_State_Removed flag set).
- Be sent as a result of unsuccessful setup operation (when no response is received to sending Resv message).
- Be sent as a result of unsuccessful release request (when no response is received to sending Path or Resv message with (A or D) & R bits set).

```
<PathErr Message> ::=
     <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <SESSION>
     <CALL_ID>
     <ERROR_SPEC>
     [ <ACCEPTABLE_LABEL_SET> ... ]
     [ <POLICY_DATA> ... ]
     <sender descriptor>
```

The <sender descriptor> is copied from the message in error.

8.6 Notify

This message is defined to support distributed connection management.

This message is used to:

- Asynchronously notify the connection controller (specified in NOTIFY_REQUEST object) of errors associated with a connection.

For connections setup that are monitored, the transport plane will provide associated monitoring based on existing transport plane OAM mechanisms. For example, if an ODU1 link connection is set up, a tandem connection monitoring may be set up to support exchange of connection status instead of the Notify message.

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

<notify session list> ::=
     [ <notify session list> ]
     <upstream notify session> | <downstream notify session>

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

```
<downstream notify session> ::=
          <SESSION>
          <CALL_ID>
          [ <POLICY_DATA>...]
          <flow descriptor list descriptor>
```

8.7 Hello

This message is modified from definitions in RFC 3209, with further extensions to support distributed connection management.

This message is used to:

- Ensure RSVP session is up (using request and acknowledge objects).
- Initiate restart procedures by exchanging recovery and restart timers.

```
<Hello Message> ::=
        <Common Header>
        [ <INTEGRITY> ]
        <HELLO>
        [ <RESTART_CAP> ]
```

8.8 Ack

This message is modified from definitions in RFC 2961, with further extensions to support distributed connection management.

This message is used to:

- Provide acknowledgement of sent messages. The acknowledgement function can be provided either directly, using the Ack message, or indirectly, when the sent message has a corresponding reply message on a specific link (e.g., Resv is Path's corresponding reply message). In the latter case, the ACK function is provided by including a MESSAGE_ID_ACK object within the reply message.

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

8.9 Srefresh

This message is defined in RFC 2961. No modifications are necessary to support distributed connection management. For clarity, the format of this message is re-produced below:

This message is used to:

- Refresh RSVP-TE state without the transmission of Path or Resv messages. This results in a reduction of the amount of information that must be transmitted and processed in order to maintain call and connection state synchronization. An Srefresh message carries a list of Message_Identifier fields corresponding to Path and Resv trigger messages that established the state.

```
<Srefresh Message> ::=
     <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <srefresh list> | <source srefresh list>
```

```
<srefresh list> ::=
     <MESSAGE_ID LIST> | <MESSAGE_ID MCAST_LIST>
     [ <srefresh list> ]

<source srefresh list> ::=
     <MESSAGE_ID SRC_LIST>
     [ <source srefresh list> ]
```

9 GMPLS RSVP-TE attributes

9.1 GMPLS RSVP-TE objects

GMPLS RSVP-TE re-uses attributes defined by RFC 2205, RFC 2961 and RFC 3209. RFC 2961 and RFC 3209 modifies certain attributes originally defined by RFC 2205, and takes precedence over attributes defined in RFC 2205.

In addition to modifications made by RFC 3209, the following attributes are further modified to support distributed connection management. Table 2 provides the list of attributes modified to support distributed connection management (the values are in decimal notation):

Table 2/G.7713.2/Y.1704.2 – List of attributes defined by RFC 2205, RFC 2961 and RFC 3209, modified for distributed connection management

Class Num	Object	Object format (C-Type)
1	SESSION	7 LSP Tunnel IPv4 8 LSP Tunnel IPv6 11 UNI_IPv4 12 UNI_IPv6 15 ENNI_IPv4 16 ENNI_IPv6
3	RSVP_HOP (Note 1)	1 IPv4 2 IPv6 3 IPv4 IF_ID 4 IPv6 IF_ID For C-type 3, 4, the following sub-TLVs are defined: 1 IPv4 2 IPv6 3 IF_INDEX 4 COMPONENT_IF_DOWNSTREAM 5 COMPONENT_IF_UPSTREAM
4	INTEGRITY	1 Type 1 Integrity Value
5	TIME_VALUES	1 Type 1 Time Value
6	ERROR_SPEC (Notes 1 and 2)	1 IPv4 2 IPv6 3 IPv4 IF_ID 4 IPv6 IF_ID same sub-TLV as RSVP_HOP
7	SCOPE	1 IPv4 2 IPv6
8	STYLE	1 Type 1 Style
9	FLOWSPEC	2 Int-serv Flowspec

Table 2/G.7713.2/Y.1704.2 – List of attributes defined by RFC 2205, RFC 2961 and RFC 3209, modified for distributed connection management

Class Num	Object	Object format (C-Type)
10	FILTER_SPEC	7 LSP Tunnel IPv4 8 LSP Tunnel IPv6
11	SENDER_TEMPLATE	7 LSP Tunnel IPv4 8 LSP Tunnel IPv6
12	SENDER_TSPEC	2 Int-serv
14	POLICY_DATA	1 Type 1 policy data
15	RESV_CONFIRM	1 IPv4 2 IPv6
16	RSVP_LABEL	1 Type 1 label 2 GENERALIZED_LABEL 3 Waveband_Switching_Label
19	LABEL_REQUEST	1 Without Label Range 2 With ATM Label Range 3 With Frame Relay Label Range 4 Generalized_Label_Request
20	EXPLICIT_ROUTE (Note 1)	1 Type 1 explicit route also sub-type: 1 IPv4 prefix 2 IPv6 prefix 3 Label 4 Unnumbered Interface ID 32 Autonomous System
21	RECORD_ROUTE (Note 1)	1 Type 1 record route also sub-type: 1 IPv4 address 2 IPv6 address 3 Label 4 Unnumbered Interface ID
22	HELLO	1 Request 2 Acknowledgment
23	MESSAGE_ID	1 Type 1 message id
24	MESSAGE_ID_ACK	1 MESSAGE_ID_ACK 2 MESSAGE_ID_NACK
25	MESSAGE_ID_LIST	1 Message ID list 2 IPv4 Message ID Source list 3 IPv6 Message ID Source list 4 IPv4 Message ID Multicast list 5 IPv6 Message ID Multicast list
34	RECOVERY_LABEL	same as RSVP_LABEL
35	UPSTREAM_LABEL	same as RSVP_LABEL
36	LABEL_SET	1 Type 1
37	PROTECTION	1 Type 1
129	SUGGESTED_LABEL	same as RSVP_LABEL

Table 2/G.7713.2/Y.1704.2 – List of attributes defined by RFC 2205, RFC 2961 and RFC 3209, modified for distributed connection management

Class Num	Object	Object format (C-Type)
130	ACCEPTABLE_LABEL_SET	same as LABEL_SET
131	RESTART_CAP	1 Type 1
195	NOTIFY_REQUEST	1 IPv4 2 IPv6
196	ADMIN_STATUS	1 Type 1
207	SESSION_ATTRIBUTE	1 LSP_TUNNEL_RA 7 LSP Tunnel
229	GENERALIZED_UNI (Note 1)	Under C-Type = 1, the following Type, Sub-type are defined: Type Description (Sub-Type, if multiple)) 1 Source TNA address IPv4 (sub-type = 1) IPv6 (sub-type = 2) NSAP (sub-type = 3) 2 Destination TNA address IPv4 (sub-type = 1) IPv6 (sub-type = 2) NSAP (sub-type = 3) 3 Diversity 4 Egress label (sub-type = 1) SPC_LABEL (sub-type = 2) 5 Service level
230	CALL_ID	1 Operator specific 2 Globally unique Type 0x01 4-byte Source Transport NE address 0x02 16-byte Source Transport NE address 0x03 20-byte Source Transport NE address 0x04 6-byte Source Transport NE address 0x7F Vendor defined length
NOTE 1 – The address formats used in these objects are drawn from transport name spaces. NOTE 2 – Connection controller wishing to indicate that an error is related to a specific SNP or SNPP should use the appropriate IF_ID ERROR_SPEC object in the corresponding PathErr or ResvErr message, as specified in [RFC 3473].		

Note that an object's Class-number determines how a control plane node reacts to these objects when the object is not recognized:

- Class-Num = 0bbbbbbb

  The entire message should be rejected and an "Unknown Object Class" error returned.

- Class-Num = 10bbbbbb

  The node should ignore the object, neither forwarding it nor sending an error message.

- Class-Num = 11bbbbbb

  The node should ignore the object but forward it, unexamined and unmodified, in all messages resulting from this message.

9.2 GMPLS RSVP-TE error/status codes

Table 3/G.7713.2/Y.1704.2 – Error codes and values for status/error reporting

Connection setup – success	Resv (or ResvConf) message
Connection setup – failed: message error	ERROR_SPEC (general)
Connection setup – failed: called party busy	ERROR_SPEC 24/5
Connection setup – failed: calling party busy	ERROR_SPEC 24/103
Connection setup – failed: timeout	ERROR_SPEC 24/5 or 24/103 (Note 1)
Connection setup – failed: identity error: invalid A-end user name	ERROR_SPEC 2/100
Connection setup – failed: identity error: invalid Z-end user name	ERROR_SPEC 2/101
Connection setup – failed: identity error: invalid connection name	ERROR_SPEC 24/102
Connection setup – failed: identity error: invalid call name	ERROR_SPEC 24/105
Connection setup – failed: service error: invalid SNP ID	ERROR_SPEC 24/6 or 24/11 or 24/12 or 24/14
Connection setup – failed: service error: unavailable SNP ID	ERROR_SPEC 24/6 or 24/11 or 24/12 or 24/14
Connection setup – failed: service error: invalid SNPP ID	ERROR_SPEC 24/104
Connection setup – failed: service error: unavailable SNPP ID	ERROR_SPEC 24/104
Connection setup – failed: identity error: invalid SPC Label	ERROR_SPEC 24/106
Connection setup – failed: policy error: invalid CoS	ERROR_SPEC 24/101 also additional values from 2/any
Connection setup – failed: policy error: unavailable CoS	ERROR_SPEC 24/101 also additional values from 2/any
Connection setup – failed: policy error: invalid GoS	ERROR_SPEC 24/101 also additional values from 2/any
Connection setup – failed: policy error: unavailable GoS	ERROR_SPEC 24/101 also additional values from 2/any
Connection setup – failed: policy error: failed security check	ERROR_SPEC 2/100 or 2/101 (Note 2)
Connection setup – failed: policy error: invalid explicit resource list	ERROR_SPEC 24/1, 24/2, 24/3, or 24/7
Connection setup – failed: policy error: invalid recovery	ERROR_SPEC 24/15 also ERROR_SPEC 24/100
Connection setup – failed: connection error: failed to create SNC	ERROR_SPEC 1/2
Connection setup – failed: connection error: failed to establish LC	ERROR_SPEC 24/9
Connection release – success	PathTear or PathErr (with Path_State_Removed flag)
Connection release – failed: message error	ERROR_SPEC
Connection release – failed: timeout	ERROR_SPEC 24/5, 24/103
Connection release – failed: identity error: invalid call name	ERROR_SPEC 24/102

Table 3/G.7713.2/Y.1704.2 – Error codes and values for status/error reporting

Connection release – failed: policy error: failed security check	(if security failed, GMPLS drops the request)
Connection release – failed: connection error: failed to release SNC	Error value in error code = 21 (general)
Connection release – failed: connection error: failed to free LC	Error value in error code = 21 (general)
Connection error – non-service affecting	ERROR_SPEC (general)
Connection error –service affecting	ERROR_SPEC (general)
Connection error – unexpected call release	Error value in error code = 21 (general)
NOTE 1 – Timeout is an internal event. As such, the error reported is one of: (1) no route available towards source, or (2) no route available towards destination. NOTE 2 – Security check failure is reported as: (1) unauthorized source, or (2) unauthorized destination.	

In addition to the above error codes and values used for distributed connection management, Table 4 below provides the set of error codes and values that are used for identifying other protocol specific errors.

Table 4/G.7713.2/Y.1704.2 – List of error codes and values defined by RFC 2205 and RFC 3209, modified for distributed connection management

Error code	Error value
00: Confirmation	
01: Admission Control Failure	bits in this format: ssur cccc cccc cccc ss = 00: Low order 12 bits contain a globally defined sub-code (values listed below). ss = 10: Low order 12 bits contain are organization-specific sub-code. RSVP is not expected to be able to interpret this except as a numeric value. ss = 11: Low order 12 bits contain a service-specific sub-code. RSVP is not expected to be able to interpret this except as a numeric value. Since the traffic control mechanism might substitute a different service, this encoding may include some representation of the service in use. u = 0: RSVP rejects the message without updating local state. u = 1: RSVP may use the message to update local state and forward the message. This means that the message is informational. r: Reserved bit, should be zero. cccc cccc cccc: 12-bit code. The following globally-defined sub-codes may appear in the low-order 12 bits when ssur = 0000: – Sub-code = 1: Delay bound cannot be met – Sub-code = 2: Requested bandwidth unavailable – Sub-code = 3: MTU in flowspec larger than interface MTU.

Table 4/G.7713.2/Y.1704.2 – List of error codes and values defined by RFC 2205 and RFC 3209, modified for distributed connection management

Error code	Error value
02: Policy Control failure	(from RFC 2750): 0 = ERR_INFO : Information reporting 1 = ERR_WARN : Warning 2 = ERR_UNKNOWN : Reason unknown 3 = ERR_REJECT : Generic Policy Rejection 4 = ERR_EXCEED : Quota or Accounting violation 5 = ERR_PREEMPT : Flow was pre-empted 6 = ERR_EXPIRED : Previously installed policy expired (not refreshed) 7 = ERR_REPLACED: Previous policy data was replaced and caused rejection 8 = ERR_MERGE : Policies could not be merged (multicast) 9 = ERR_PDP : PDP down or non functioning 10 = ERR_SERVER : Third Party Server (e.g., Kerberos) unavailable 11 = ERR_PD_SYNTX: POLICY_DATA object has bad syntax 12 = ERR_PD_INTGR: POLICY_DATA object failed Integrity Check 13 = ERR_PE_BAD : POLICY_ELEMENT object has bad syntax 14 = ERR_PD_MISS : Mandatory PE Missing (Empty PE is in the PD object) 15 = ERR_NO_RSC : PEP Out of resources to handle policies 16 = ERR_RSVP : PDP encountered bad RSVP objects or syntax 17 = ERR_SERVICE : Service type was rejected 18 = ERR_STYLE : Reservation Style was rejected 19 = ERR_FL_SPEC : FlowSpec was rejected (too large) 100 = Unauthorized sender 101 = Unauthorized receiver Values between 2^15 and 2^16 – 1 can be used for site and/or vendor error values.
03: No path information for this Resv message	
04: No sender information for this Resv message	
05: Conflicting reservation style	The Error Value field contains the low-order 16 bits of the Option Vector of the existing style with which the conflict occurred. This Resv message cannot be forwarded.
06: Unknown reservation style	
07: Conflicting dest ports	
08: Conflicting sender ports	
09: (reserved)	
10: (reserved)	
11: (reserved)	

Table 4/G.7713.2/Y.1704.2 – List of error codes and values defined by RFC 2205 and RFC 3209, modified for distributed connection management

Error code	Error value
12: Service preempted	Bits in this format: ssur cccc cccc cccc Here the high-order bits ssur are as defined under Error Code 01. The globally-defined sub-codes that may appear in the low-order 12 bits when ssur = 0000 are to be defined in the future.
13: Unknown object class	Error Value contains 16-bit value composed of (Class-Num, C-Type) of unknown object. This error should be sent only if RSVP is going to reject the message, as determined by the high-order bits of the Class-Num.
14: Unknown object C-Type	Error Value contains 16-bit value composed of (Class-Num, C-Type) of object.
15: (reserved)	
16: (reserved)	
17: (reserved)	
18: (reserved)	
19: (reserved)	
20: Reserved for API	Error Value field contains an API error code, for an API error that was detected asynchronously and must be reported via an upcall.
21: Traffic Control Error	Bits in this format: ss00 cccc cccc cccc Here the high-order bits ss are as defined under Error Code 01. The following globally-defined sub-codes may appear in the low order 12 bits (cccc cccc cccc) when ss = 00: – Sub-code = 01: Service conflict Trying to merge two incompatible service requests. – Sub-code = 02: Service unsupported Traffic control can provide neither the requested service nor an acceptable replacement. – Sub-code = 03: Bad Flowspec value Malformed or unreasonable request. – Sub-code = 04: Bad Tspec value Malformed or unreasonable request. – Sub-code = 05: Bad Adspec value Malformed or unreasonable request.
22: Traffic Control System error	The Error Value will contain a system-specific value giving more information about the error. RSVP is not expected to be able to interpret this value.
23: RSVP System error	The Error Value field will provide implementation-dependent information on the error. RSVP is not expected to be able to interpret this value.

Table 4/G.7713.2/Y.1704.2 – List of error codes and values defined by RFC 2205 and RFC 3209, modified for distributed connection management

Error code	Error value	
24: Routing Problem	1	Bad EXPLICIT_ROUTE object
	2	Bad strict node
	3	Bad loose node
	4	Bad initial subobject
	5	No route available toward destination
	6	Unacceptable label value
	7	RRO indicated routing loops
	8	MPLS being negotiated, but a non-RSVP-capable router stands in the path
	9	MPLS label allocation failure
	10	Unsupported L3PID
	11	Label Set
	12	Switching Type
	13	reserved
	14	Unsupported Encoding
	15	Unsupported Link Protection
	100	Diversity not available
	101	Service level not available
	102	Invalid/unknown connection ID
	103	No route available toward source
	104	Unacceptable Interface ID
	105	Invalid/unknown call ID
	106	Invalid SPC Interface ID/Label
25: Notify Error	1	RRO too large for MTU
	2	RRO Notification
	3	Tunnel locally repaired
	4	Control Channel Active State
	5	Control Channel Degraded State

Annex A

Summary of GENERALIZED_UNI object

The GENERALIZED_UNI object has the following format:

0 1 2 7 8 15	16 23	24 31
Length	Class-Num	C-type
… Sub-objects …		

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:

<table>
<tr><td>0 1 2 7 8 15</td><td>16 23</td><td>24 31</td></tr>
<tr><td>Length</td><td>Type</td><td>Sub-type</td></tr>
<tr><td colspan="3">…
Value
…</td></tr>
</table>

The following sub-objects are defined. These sub-objects are all defined as sub-objects under the common C-Type = 1. The Type field distinguishes the sub-objects, while the Sub-type field distinguishes different uses of the sub-object. The contents of these sub-objects are described in OIF UNI-01.0:

- SOURCE_TNA Address sub-object: Type = 1. The following sub-types are defined:
  - IPv4 (Sub-type = 1);
  - IPv6 (Sub-type = 2);
  - NSAP (Sub-type = 3).
- DESTINATION_TNA Address sub-object: Type = 2. The following sub-types are defined:
  - IPv4 (Sub-type = 1);
  - IPv6 (Sub-type = 2);
  - NSAP (Sub-type = 3).
- DIVERSITY sub-object: Type = 3, Sub-type = 1
- EGRESS_LABEL sub-object: Type = 4, Sub-type = 1
  - SPC_LABEL sub-object: Type = 4, Sub-type = 2
- SERVICE_LEVEL[1] sub-object: Type = 5, Sub-type = 1

Annex B

Label scope

B.1 Scope of the label

Labels provide information that are useful only to the CC/LRM using them. Labels may have an associated structure imposed on them for local use. Once the labels are transmitted to another CC or LRM, the structure of a label should no longer be important. This issue does not present a problem in a simple point-to-point connection between two control plane-enabled nodes. However, once a subnetwork is introduced between these nodes (where the subnetwork provides rearrangement capability for the signals) label scoping becomes an issue. Figure B.1 illustrates the case of a connection traversing a non-control plane rearrangeable subnetwork (e.g., label rearrangement may be performed via a management system). There is an implicit assumption that the non-control plane connections already exist prior to any connection request.

[1] The service level sub-object can be used to identify specific levels of Class of Service to be provided to the call/connection requested. The value and interpretation of specific classes of service is defined by carriers, in agreement with clients in the case of switched connections.

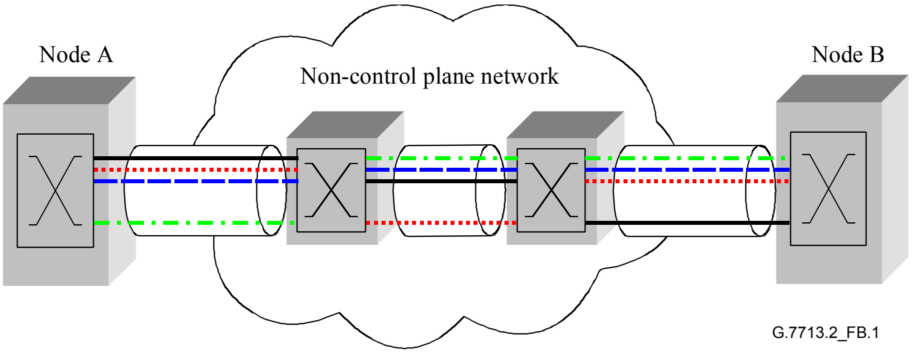


Figure B.1/G.7713.2/Y.1704.2 – Example link where labels are rearranged via non-control plane network

The only characteristic of a label that is important once it is transmitted is the format of the label and the uniqueness of the label values. Characteristics such as the structure of the label are no longer important or useful. In fact, imposing structure of a label outside of the local space may result in restrictions to the architecture of a network.

B.2 Label association function

In order to support the capability to map a received label value to a locally significant label value, an additional function is needed as part of the local process: that of label association. This function takes as input a received label value and provides as output a locally significant label value. As such, this function may be considered generally to provide a table lookup function.

The information necessary to allow mapping from received label value to a locally significant label value may be derived in several ways:

– Via manual provisioning of the association.

– Via automatic discovery of the association.

Either method may be used. In the case of automatic discovery of the association, this implies that the discovery mechanism operates at the SNP level, as per ITU-T Rec. G.7714/Y.1705. Note that in the simple case where two NEs may be directly connected, no association may be necessary. In such instances, the label association function provides a one-to-one mapping of the input to output label values.

Annex C

Technology-specific terminology updates

The terminology used in [RFC 3471 GMPLS-SIG] for the GENERALIZED_LABEL_REQUEST object is updated to align with the ITU-T transport terminology. Note that no technical or procedural modifications are made. Tables C.1 and C.2 provide the updated terminology for the relevant fields applicable for ASON (LSP Encoding Type, and Generalized Payload ID):

Table C.1/G.7713.2/Y.1704.2 – Terminology update for LSP encoding type within GENERALIZED_LABEL_REQUEST object

Value	Type (in the RFC)	Updated type terminology
5	SDH ITU-T G.707/Y.1322/SONET ANSI T1.105	SDH ITU-T G.707/Y.1322
7	Digital Wrapper	OTN ITU-T G.709/Y.1331 ODUx
8	Lambda (photonic)	OTN ITU-T G.709/Y.1331 OCh

Table C.2/G.7713.2/Y.1704.2 – Terminology update for generalized payload ID within GENERALIZED_LABEL_REQUEST Object

Value	Type
0	Unknown
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Asynchronous mapping of 139 264 kbit/s (P4x) into VC-4
6	Asynchronous mapping of 44 736 kbit/s (P32x) into VC-3
7	Asynchronous mapping of 34 368 kbit/s (P31x) into VC-3
10	Asynchronous mapping of 6 312 kbit/s (P21x) into VC-2
11	Bit synchronous mapping of 6 312 kbit/s (P21x) into VC-2
13	Asynchronous mapping of 2 048 kbit/s (P12x) into VC-12
14	Byte synchronous mapping of 2 048 kbit/s (P12s) into VC-12
15	Byte synchronous mapping of 31 * 64 kbit/s (P0) into VC-12
16	Asynchronous mapping of 1 544 kbit/s (P11x) into VC-11
17	Bit synchronous mapping of 1 544 kbit/s (P11x-bit) into VC-11
18	Byte synchronous mapping of 1 544 kbit/s (P11s) into VC-11
25	Multiplexing of SDH LOVC via TUG-2 into a VC-3
26	Multiplexing of SDH LOVC via TUG-3s into a VC-4
27	Multiplexing of SDH HOVC into STM-N
28	POS – No Scrambling, 16-bit CRC
29	POS – No Scrambling, 32-bit CRC
30	POS – Scrambling, 16-bit CRC
31	POS – Scrambling, 32-bit CRC
41	FDDI mapping into VC-4
42	DQDB mapping into VC-4
NOTE – The reference to the particular mapping schemes may be found in ITU-T Rec. G.707/Y.1322.	

Appendix I

Mapping of messages

Table I.1/G.7713.2/Y.1704.2 – Mapping of DCM UNI messages to GMPLS RSVP-TE messages

	UNI messages	GMPLS RSVP-TE
Call Setup messages	CallSetupRequest	Path
	CallSetupIndication	Resv, PathErr
	CallSetupConfirm	ResvConf
Call Release messages	CallReleaseRequest	Path or Resv (with D & R bit) or Path or Resv (with A & R bit)
	CallReleaseIndication	PathErr (Path_State_Removed flag) or PathTear
Call Query messages	CallQueryRequest	Path (implicit in RSVP-TE via periodic refreshes)
	CallQueryIndication	Resv (implicit in RSVP-TE via periodic refreshes)
Call notification message	CallNotify	Notify, also PathErr

Table I.2/G.7713.2/Y.1704.2 – E-NNI messages

	E-NNI messages	GMPLS RSVP-TE
Connection Setup messages	ConnectionSetupRequest	Path
	ConnectionSetupIndication	Resv, PathErr
	ConnectionSetupConfirm	ResvConf
Connection Release messages	ConnectionReleaseRequest	Path or Resv (with D & R bit) or Path or Resv (with A & R bit)
	ConnectionReleaseIndication	PathErr (Path_State_Removed flag) or PathTear
Connection Query messages	ConnectionQueryRequest	Path (implicit in RSVP-TE via periodic refreshes)
	ConnectionQueryIndication	Resv (implicit in RSVP-TE via periodic refreshes)
Connection Notification message	ConnectionNotify	Notify, also PathErr

Appendix II

Mapping of attributes

Table II.1/G.7713.2/Y.1704.2 – Mapping of DCM attributes to GMPLS RSVP-TE objects

	Attributes	Scope	GMPLS RSVP-TE
Identity attributes	A-end user name	End-to-end	SOURCE_TNA
	Z-end user name	End-to-end	DESTINATION_TNA
	Initiating CC/CallC name	Local	Source Node ID (in the IP header), also SENDER_TEMPLATE/FILTER_SPEC
	Terminating CC/CallC name	Local	Destination Node ID (in the IP header), also SESSION
	Connection name	Local	SESSION + SENDER_TEMPLATE
	Call name	End-to-end	CALL_ID
Service attributes	SNP ID	Local	GENERALIZED_LABEL, UPSTREAM_LABEL, EGRESS_LABEL, SUGGESTED_LABEL, SPC_LABEL
	SNPP ID	Local	Source/destination TNA, RSVP_HOP, LABEL_SET
	Directionality	Local	(implied by UPSTREAM_LABEL)
Policy attributes	CoS	End-to-end	DIVERSITY, SERVICE_LEVEL, POLICY_DATA (available as part of OIF UNI-01.0 extensions), SESSION_ATTRIBUTE
	GoS	End-to-end	Same as CoS above
	Explicit resource list	Local	EXPLICIT_ROUTE, ROUTE_RECORD
	Recovery	Local	PROTECTION
	Security	Local	INTEGRITY (also implicit lower layer security via, e.g., IPsec)
Additional attributes of GMPLS	implied layer info		GENERALIZED_LABEL_REQUEST, SENDER_TSPEC/FLOWSPEC, RSVP_HOP
	For disabling monitoring (see 6.1.1.2/G.7713/Y.1704)		ADMIN_STATUS
	For protocol robustness		HELLO_REQUEST, HELLO_ACK
	For status/error codes		ERROR_SPEC
	For optional confirmation		RESV_CONFIRM
	For protocol robustness		MESSAGE_ID, MESSAGE_ID_ACK, MESSAGE_ID_NACK, MESSAGE_ID_LIST
	For protocol robustness		RESTART_CAP, RECOVERY_LABEL
	Protocol specific attribute		STYLE
	Protocol specific attribute		TIME_VALUES

Appendix III

Protocol elements not used

III.1 Messages not used

The following messages are not used in DCM GMPLS RSVP-TE:

– ResvTear.

This message is modified from definitions in RFC 2205 by RFC 2961. No additional modifications are necessary to support distributed connection management. For clarity, this message format is re-produced below:

This message is not used as part of connection-oriented release procedures. The PathErr (with Path_State_Removed flag) is used to support destination-initiated release.

```
<ResvTear Message> ::=
     <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <SESSION>
     <RSVP_HOP>
     [ <SCOPE> ]
     <STYLE>
     <flow descriptor list>

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

– ResvErr

This message is modified from definitions in RFC 2205 and RFC 2961, with further extensions to support distributed connection management.

This message is used to:

Respond to a Resv connection setup request (when encountering problems with setup); however, note that in the GMPLS implementation where a connection setup error requires releasing the connection, and since ResvErr does not remove Path states, the PathTear is used for GMPLS connection-oriented network to remove Path states, i.e., ResvErr is not used during setup and release.

```
<ResvErr Message> ::=
     <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> | <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <SESSION>
     <RSVP_HOP>
     <ERROR_SPEC>
     [ <SCOPE> ]
     [ <ACCEPTABLE_LABEL_SET> ... ]
     [ <POLICY_DATA> ... ]
     <STYLE>
     <error flow descriptor>
```

III.2 Objects not used

The following objects and C-types are not used in DCM GMPLS RSVP-TE, see Table III.1.

Table III.1/G.7713.2/Y.1704.2 – Objects and C-types not used

Object	Object format
SESSION	1 IPv4 2 IPv6
FLOWSPEC	1 Reserved
FILTER_SPEC	1 IPv4 2 IPv6 3 IPv6 Flow Label
SENDER_TEMPLATE	1 IPv4 2 IPv6 3 IPv6 Flow Label
ADSPEC	2 Int-serv

Appendix IV

Support for call capability

IV.1 Call capability object

To support call capability an additional object is defined. A call capability is used to specify the capabilities supported for a call. These may include specifying supplementary services. For RSVP-TE a new CALL_OPS object is defined to be carried by the Path, Resv, PathTear, PathErr, and Notify message. The CALL_OPS object also serves to differentiate the messages to indicate a "call-only" call. In the case for logical separation of call and connection, the CALL_OPS object is not needed.

The CALL_OPS object is defined as follows (the Class-num is the suggested value for the new object):

– CALL_OPS (Class-num = 228, C-type = 1)

0 1 2 7 8 15	16 23	24 31
Length	Class-Num	C-type
Reserved		Call ops flag

Two flags are currently defined for the "call ops flag":

– 0x01: call without connection.

– 0x02: synchronizing a call (for restart mechanism).

Appendix V

Example multiple control plane node failures

This appendix illustrates how the control plane handles multiple control plane node failures. Note that for any multiple failure scenarios, there is no guarantee of full recovery as information may not be recoverable. This appendix describes a mechanism that attempts a "best effort" recovery due to multiple control plane node failures. Figure V.1 shows the scenario of multiple control plane node failures. There are two sub-cases:

– Nodes B and C do not recover simultaneously, i.e., either node B recovers first or node C recovers first; as such this case may be treated as two instances of the above case. For example, if node B recovers first, it synchronizes with node A for connections #1, #4 and #6, while entering self-refresh for communications with node C. Once node C recovers, it synchronizes with node B for connections #2, #4, #5 and #6, and synchronizes with node D for connections #3, #5 and #6. Any out-of-sync connections are resolved by communications with the management plane as per the above behaviours.

– Nodes B and C recover simultaneously; in this case node B's initial recovery should be with node A and not node C, while node C's initial recovery should be with node D and not node B (this reduces problem of synchronizing against incorrect information). Once these states are synchronized (according to above), then nodes B and C may synchronize with each other (again according to above). Note that in this case connection #2 which starts and ends at the two failed nodes, may require that management system restore this connection state (depending on how much state was restored from backup information). Note that to support this behaviour a recovered node does not immediately send Hello messages. The following behaviour is needed:

  - When a recovery node receives a Hello message from its neighbour, it may respond by sending a Hello message, i.e., it should not initiate any Hello messages, but only respond to received Hello messages. This initiates synchronization of connection states with the neighbour.
  - Once the recovery node has recovered all possible states from these neighbours, it may initiate sending Hello messages to all known neighbours (information about known neighbours may be recovered from local persistent storage or from an external component).
  - These procedures together ensure meeting the multiple control plane node failure scenario, by allowing recovery nodes to synchronize with non-failed-nodes prior to synchronizing with each other (i.e., there is an implicit assumption that there exists at least one node that has not failed – this non-failed node will thus serve as the trigger for recovered nodes to synchronize states with neighbours).

Note that if local persistent storage and external component does not provide connection state information, then connection #2's state may not be recoverable. In this case, the above behaviour may result that the management plane instructs the control plane to retain the connection even with non-synchronized information (or it may instruct the control plane to release the connection).

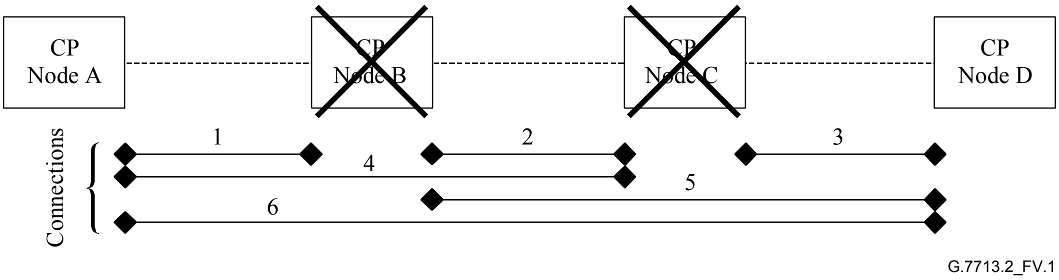


Figure V.1/G.7713.2/Y.1704.2 – Multiple control plane node failures

ITU-T Y-SERIES RECOMMENDATIONS

GLOBAL INFORMATION INFRASTRUCTURE AND INTERNET PROTOCOL ASPECTS

GLOBAL INFORMATION INFRASTRUCTURE	
General	Y.100–Y.199
Services, applications and middleware	Y.200–Y.299
Network aspects	Y.300–Y.399
Interfaces and protocols	Y.400–Y.499
Numbering, addressing and naming	Y.500–Y.599
Operation, administration and maintenance	Y.600–Y.699
Security	Y.700–Y.799
Performances	Y.800–Y.899
INTERNET PROTOCOL ASPECTS	
General	Y.1000–Y.1099
Services and applications	Y.1100–Y.1199
Architecture, access, network capabilities and resource management	Y.1200–Y.1299
Transport	Y.1300–Y.1399
Interworking	Y.1400–Y.1499
Quality of service and network performance	Y.1500–Y.1599
Signalling	Y.1600–Y.1699
Operation, administration and maintenance	**Y.1700–Y.1799**
Charging	Y.1800–Y.1899

For further details, please refer to the list of ITU-T Recommendations.

SERIES OF ITU-T RECOMMENDATIONS

Series A	Organization of the work of ITU-T
Series B	Means of expression: definitions, symbols, classification
Series C	General telecommunication statistics
Series D	General tariff principles
Series E	Overall network operation, telephone service, service operation and human factors
Series F	Non-telephone telecommunication services
Series G	**Transmission systems and media, digital systems and networks**
Series H	Audiovisual and multimedia systems
Series I	Integrated services digital network
Series J	Cable networks and transmission of television, sound programme and other multimedia signals
Series K	Protection against interference
Series L	Construction, installation and protection of cables and other elements of outside plant
Series M	TMN and network maintenance: international transmission systems, telephone circuits, telegraphy, facsimile and leased circuits
Series N	Maintenance: international sound programme and television transmission circuits
Series O	Specifications of measuring equipment
Series P	Telephone transmission quality, telephone installations, local line networks
Series Q	Switching and signalling
Series R	Telegraph transmission
Series S	Telegraph services terminal equipment
Series T	Terminals for telematic services
Series U	Telegraph switching
Series V	Data communication over the telephone network
Series X	Data networks and open system communications
Series Y	**Global information infrastructure and Internet protocol aspects**
Series Z	Languages and general software aspects for telecommunication systems

Printed in Switzerland
Geneva, 2003