Network Working Group
Request for Comments: 3474
Category: Informational
Z. Lin
New York City Transit
D. Pendarakis
Tellium
March 2003

Documentation of IANA assignments for Generalized MultiProtocol Label Switching (GMPLS) Resource Reservation Protocol - Traffic Engineering (RSVP-TE) Usage and Extensions for Automatically Switched Optical Network (ASON)

Status of this Memo

Copyright Notice

Abstract

The Generalized MultiProtocol Label Switching (GMPLS) suite of protocol specifications has been defined to provide support for different technologies as well as different applications. These include support for requesting TDM connections based on Synchronous Optical NETwork/Synchronous Digital Hierarchy (SONET/SDH) as well as Optical Transport Networks (OTNs).

This document concentrates on the signaling aspects of the GMPLS suite of protocols, specifically GMPLS signaling using Resource Reservation Protocol - Traffic Engineering (RSVP-TE). It proposes additional extensions to these signaling protocols to support the capabilities of an ASON network.

This document proposes appropriate extensions towards the resolution of additional requirements identified and communicated by the ITU-T Study Group 15 in support of ITU's ASON standardization effort.

Table of Contents

1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14, RFC 2119 [RFC2119].

2. Introduction

This document contains the extensions to GMPLS for ASON-compliant networks [G7713.2]. The requirements describing the need for these extensions are provided in [GMPLS-ASON] as well as [ASON-RQTS]. These include:

- Support for call and connection separation
- Support for soft permanent connection
- Support for extended restart capabilities
- Additional error codes/values to support these extensions

This document concentrates on the signaling aspects of the GMPLS suite of protocols, specifically GMPLS signaling using RSVP-TE. It introduces extensions to GMPLS RSVP-TE to support the capabilities as specified in the above referenced documents. Specifically, this document uses the messages and objects defined by [RFC2205], [RFC2961], [RFC3209], [RFC3471], [RFC3473], [OIF-UNI1] and [RFC3476] as the basis for the GMPLS RSVP-TE protocol, with additional extensions defined in this document.

3. Support for Soft Permanent Connection

In the scope of ASON, to support soft permanent connection (SPC) for RSVP-TE, one new sub-type for the GENERALIZED_UNI object is defined. The GENERALIZED_UNI object is defined in [RFC3476] and [OIF-UNI1]. This new sub-type has the same format and structure as the EGRESS_LABEL (the sub-type is the suggested value for the new sub-object):

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

The label association of the permanent ingress segment with the switched segment at the switched connection ingress node is a local policy matter (i.e., between the management system and the switched connection ingress node) and is thus beyond the scope of this document.

The processing of the SPC_LABEL sub-object follows that of the EGRESS_LABEL sub-object [OIF-UNI1]. Note that although the explicit label control described in [RFC3471] and [RFC3473] may provide a mechanism to support specifying the egress label in the context of supporting the GMPLS application, the SPC services support for the ASON model uses the GENERALIZED_UNI object for this extension to help align the model for both switched connection and soft permanent connection, as well as to use the service level and diversity attributes of the GENERALIZED_UNI object.

4. Support for Call

To support basic call capability (logical call/connection separation), a call identifier is introduced to the RSVP-TE message sets. This basic call capability helps introduce the call model; however, additional extensions may be needed to fully support the canonical call model (complete call/connection separation).

4.1 Call Identifier and Call Capability

A Call identifier object is used in logical call/connection separation while both the Call identifier object and a Call capability object are used in complete call/connection separation.

4.1.1 Call Identifier

To identify a call, a new object is defined for RSVP-TE. The CALL_ID object carries the call identifier. The value is globally unique (the Class-num is the suggested value for the new object):

CALL_ID (Class-num = 230)

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Length            |Class-Num (230)|    C-Type     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Call identifier                        |
~                              ...                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Where the following C-types are defined:

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

- C-Type = 2 (globally unique): The call identifier contains a 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 a call identifier must be multiples of 32 bits, with a 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 LSR 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 LSR 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 is required 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 Recommendation 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).

National Segment (NS):
The National Segment (NS) field consists of two sub-fields:

- the first subfield contains the ITU Carrier Code
- the second subfield contains 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 Service Bureauin association with Recommendation M.1400. This code consists of 1-6 left-justified alphabetic, or leading alphabetic followed by numeric, characters (bytes). If the code is less than 6 characters (bytes), it is padded with a trailing NULL to fill the subfield.

The Unique Access Point Code is a matter for the organization to which the country code and ITU carrier code have been assigned, provided that uniqueness is guaranteed. This code consists of 1-6 characters (bytes), trailing NULL, completing the 12-character National Segment. If the code is less than 6 characters, it is padded by a trailing NULL to fill the subfield.

Format of the National Segment

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       ITU carrier code                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ITU carrie dode (cont)        |  Unique access point code     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|               Unique access point code (continued)            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |Class-Num (230)|  C-Type  (1)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |                     Resv                      |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Source LSR address                       |
~                              ...                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Local identifier  (continued)               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|            Length             |Class-Num (230)|  C-Type  (2)  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |                 IS (3 bytes)                  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         NS (12 bytes)                         |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Source LSR address                       |
~                              ...                              ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                       Local identifier                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  Local identifier  (continued)                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

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

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

Source LSR address:
An address of the LSR 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 LSR 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 the source LSR address may be assigned from an operator-specific address space.

4.1.2 Call Capability

The call capability feature is provided in Section 10. This is an optional capability. If supported, section 10 extensions must be followed.

4.2 What Does Current GMPLS Provide

The signaling mechanism defined in [RFC2961], [RFC3209] and [RFC3471] supports automatic connection management; however it does not provide capability to support the call model. A call may be viewed as a special purpose connection that requires a different subset of information to be carried by the messages. This information is targeted to the call controller for the purpose of setting up a call/connection association.

4.3 Support for Call and Connection

Within the context of this document, every call (during steady state) may have one (or more) associated connections. A zero connection call is defined as a transient state, e.g., during a break-before-make restoration event.

In the scope of ASON, to support a logical call/connection separation, a new call identifier is needed as described above. The new GENERALIZED_UNI object is carried by the Path message. The new CALL_ID object is carried by the Path, Resv, PathTear, PathErr, and Notify messages. The ResvConf message is left unmodified. The CALL_ID object (along with other objects associated with a call, e.g., GENERALIZED_UNI) is processed by the call controller, while other objects included in these messages are processed by the connection controller as described in [RFC3473]. Processing of the CALL_ID (and related) object is described in this document.

Note: unmodified RSVP message formats are not listed below.

4.3.1 Processing Rules

The following processing rules are applicable for call capability:

- For initial calls, the source user MUST set the CALL_ID's C-Type and call identifier value to all-zeros.
- For a new call request, the first network node sets the appropriate C-type and value for the CALL_ID.
- For an existing call (in case CALL_ID is non-zero) the first network node verifies existence of the call.

- The CALL_ID object on all messages MUST be sent from the ingress call controller to egress call controller by all other (intermediate) controllers without alteration. Indeed, the Class-Num is chosen such that a node which does not support ASON extensions to GMPLS forwards the object unmodified (value in the range 11bbbbbb).
- The destination user/client receiving the request uses the CALL_ID value as a 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.

4.3.2 Modification to Path Message

```
<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 descriptor for unidirectional LSPs is not modified by this document.

The format of the sender descriptor for bidirectional LSPs is not modified by this document.

Note that although the GENERALIZED_UNI and CALL_ID objects are optional for GMPLS signaling, these objects are mandatory for ASON-compliant networks, i.e., the Path message MUST include both GENERALIZED_UNI and CALL_ID objects.

4.3.3 Modification to Resv Message

```
<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> is not modified by this document.

Note that although the CALL_ID object is optional for GMPLS signaling, this object is mandatory for ASON-compliant networks, i.e., the Resv message MUST include the CALL_ID object.

4.3.4 Modification to PathTear Message

```
<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)

Note that although the CALL_ID object is optional for GMPLS signaling, this object is mandatory for ASON-compliant networks, i.e., the PathTear message MUST include the CALL_ID object.

4.3.5 Modification to PathErr Message

```
<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>

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

Note that although the CALL_ID object is optional for GMPLS signaling, this object is mandatory for ASON-compliant networks, i.e., the PathErr message MUST include the CALL_ID object.

4.3.6 Modification to Notify Message

Note that this message may include sessions belonging to several calls.

```
<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>
```

Note that although the CALL_ID object is optional for GMPLS signaling, this object is mandatory for ASON-compliant networks, i.e., the Notify message MUST include the CALL_ID object.

5. Support For Behaviors during Control Plane Failures

Various types of control plane failures may occur within the network. These failures may impact the control plane as well as the data plane (e.g., in a SDH/SONET network if the control plane transport uses the DCC and a fiber cut occurs, then both the control plane signaling channel and the transport plane connection fails). As described in [RFC3473], current GMPLS restart mechanisms allows support to handle all of these different scenarios, and thus no additional extensions are required.

In the scope of the ASON model, several procedures may take place in order to support the following control plane behaviors (as per [G7713] and [IPO-RQTS]):

- A control plane node SHOULD provide capability for persistent storage of call and connection state information. This capability allows each control plane node to recover the states of calls/connections after recovery from a signaling controller entity failure/reboot (or loss of local FSM state). Note that although the restart mechanism allows neighbor control plane nodes to automatically recover (and thus infer) the states of calls/connections, this mechanism can also be used for verification of neighbor states, while the persistent storage provides the local recovery of lost state. In this case, per [RFC3473], if during the Hello synchronization the restarting node determines that a neighbor does not support state recovery (i.e., local state recovery only), and the restarting node maintains its state on a per neighbor basis, the restarting node should immediately consider the Recovery as completed.
- A control plane node detecting a failure of all control channels between a pair of nodes SHOULD request an external controller (e.g., the management system) for further instructions. The default behavior is to enter into self-refresh mode (i.e., preservation) for the local call/connection states. As an example, possible external instructions may be to remain in self-refresh mode, or to release local states for certain connections. Specific details are beyond the scope of this document.
- A control plane node detecting that one (or more) connections cannot be re-synchronized with its neighbor (e.g., due to different states for the call/connection) SHOULD request an external controller (e.g., the management system) for further instructions on how to handle the non-synchronized connection. As an example, possible instructions may be to maintain the current

local connection states. Specifics of the interactions between the control plane and management plane are beyond the scope of this document.

- A control plane node (after recovering from node failure) may lose information on forwarding adjacencies. In this case the control plane node SHOULD request an external controller (e.g., the management system) for information to recover the forwarding adjacency information. Specifics of the interactions between the control plane and management plane are beyond the scope of this document.

6. Support For Label Usage

Labels are defined in GMPLS to provide information on the resources used for a particular connection. The labels may range from specifying a particular timeslot, or a particular wavelength, to a particular port/fiber. In the context of the automatic switched optical network, the value of a label may not be consistently the same across a link. For example, the figure below illustrates the case where two GMPLS/ASON-enabled nodes (A and Z) are interconnected across two non-GMPLS/ASON-enabled nodes (B and C; i.e., nodes B and C do not support the ASON capability), where these nodes are all SDH/SONET nodes providing, e.g., a VC-4 service.

```
+-----+                          +-----+
|     |    +---+    +---+    |     |
|  A  |---| B |---| C |---|  Z  |
|     |    +---+    +---+    |     |
+-----+                          +-----+
```

Labels have an associated structure imposed on them for local use based on [GMPLS-SDHSONET] and [GMPLS-OTN]. Once the local label is transmitted across an interface to its neighboring control plane node, the structure of the local label may not be significant to the neighbor node since the association between the local and the remote label may not necessarily be the same. This issue does not present a problem in a simple point-to-point connection between two control plane-enabled nodes where the timeslots are mapped 1:1 across the interface. However, in the scope of the ASON, once a non-GMPLS capable sub-network is introduced between these nodes (as in the above figure, where the sub-network provides re-arrangement capability for the timeslots) label scoping may become an issue.

In this context, there is an implicit assumption that the data plane connections between the GMPLS capable edges already exist prior to any connection request. For instance node A's outgoing VC-4's timeslot #1 (with SUKLM label=[1,0,0,0,0]) as defined in [GMPLS-SONETSDH]) may be mapped onto node B's outgoing VC-4's timeslot #6 (label=[6,0,0,0,0]) may be mapped onto node C's outgoing VC-4's timeslot #4 (label=[4,0,0,0,0]). Thus by the time node Z receives the request from node A with label=[1,0,0,0,0], the node Z's local label and the timeslot no longer corresponds to the received label and timeslot information.

As such to support the general case, the scope of the label value is considered local to a control plane node. A label association function can be used by the control plane node to map the received (remote) label into a locally significant label. The information necessary to allow mapping from a received label value to a locally significant label value may be derived in several ways:

- Via manual provisioning of the label association
- Via discovery of the label association

Either method may be used. In the case of dynamic association, this implies that the discovery mechanism operates at the timeslot/label level before the connection request is processed at the ingress node. Note that in the simple case where two nodes are directly connected, no association may be necessary. In such instances, the label association function provides a one-to-one mapping of the received local label values.

7. Support for UNI and E-NNI Signaling Session

[RFC3476] defines a UNI IPv4 SESSION object used to support the UNI signaling when IPv4 addressing is used. This document introduces three new extensions. These extensions specify support for the IPv4 and IPv6 E-NNI session and IPv6 UNI session. These C-Types are introduced to allow for easier identification of messages as regular GMPLS messages, UNI messages, or E-NNI messages. This is particularly useful if a single interface is used to support multiple service requests.

Extensions to SESSION object (Class-num = 1):
- C-Type = 12: UNI_IPv6 SESSION object
- C-TYPE = 15: ENNI_IPv4 SESSION object
- C-Type = 16: ENNI_IPv6 SESSION object

The format of the SESSION object with C-Type = 15 is the same as that for the SESSION object with C-Type = 7. The format of the SESSION object with C-Type = 12 and 16 is the same as that for the SESSION object with C-Type = 8.

The destination address field contains the address of the downstream controller processing the message. For the case of the E-NNI signaling interface (where eNNI-U represents the upstream controller and eNNI-D represents the downstream controller) the destination address contains the address of eNNI-D. [OIF-UNI1] and [RFC3476] describes the content of the address for UNI_IPv4 SESSION object, which is also applicable for UNI_IPv6 SESSION object.

8. Additional Error Cases

In the scope of ASON, the following additional error cases are defined:

- Policy control failure: unauthorized source; this error is generated when the receiving node determines that a source user/client initiated request for service is unauthorized based on verification of the request (e.g., not part of a contracted service). This is defined in [RFC3476].
- Policy control failure: unauthorized destination; this error is generated when the receiving node determines that a destination user/client is unauthorized to be connected with a source user/client. This is defined in [RFC3476].
- Routing problem: diversity not available; this error is generated when a receiving node determines that the requested diversity cannot be provided (e.g., due to resource constraints). This is defined in [RFC3476].
- Routing problem: service level not available; this error is generated when a receiving node determines that the requested service level cannot be provided (e.g., due to resource constraints). This is defined in [RFC3476].
- Routing problem: invalid/unknown connection ID; this error is generated when a receiving node determines that the connection ID generated by the upstream node is not valid/unknown (e.g., does not meet the uniqueness property). Connection ID is defined in [OIF-UNI1].
- Routing problem: no route available toward source; this error is generated when a receiving node determines that there is no available route towards the source node (e.g., due to unavailability of resources).
- Routing problem: unacceptable interface ID; this error is generated when a receiving node determines that the interface ID specified by the upstream node is unacceptable (e.g., due to resource contention).
- Routing problem: invalid/unknown call ID; this error is generated when a receiving node determines that the call ID generated by the source user/client is invalid/unknown (e.g., does not meet the uniqueness property).
- Routing problem: invalid SPC interface ID/label; this error is generated when a receiving node determines that the SPC interface ID (or label, or both interface ID and label) specified by the upstream node is unacceptable (e.g., due to resource contention).

9. Optional Extensions for Supporting Complete Separation of Call and Connection

This section describes the optional and non-normative capability to support complete separation of call and connection. To support complete separation of call and connection, a call capability object is introduced. The capability described in this Appendix is meant to be an optional capability within the scope of the ASON specification. An implementation of the extensions defined in this document include support for complete separation of call and connection, defined in this section.

9.1 Call Capability

A call capability is used to specify the capabilities supported for a call. For RSVP-TE a new CALL_OPS object is defined to be carried by the Path, Resv, PathTear, PathErr, and Notify messages. 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                   3
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |             Length            |Class-Num (228)|  C-Type (1)   |
 +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 |                 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)

9.2 Complete Separation of Call and Connection (RSVP-TE Extensions)

A complete separation of call and connection implies that a call (during steady state) may have zero (or more) associated connections. A zero connection call is a steady state, e.g., simply setting up the user end-point relationship prior to connection setup. The following modified messages are used to set up a call. Set up of a connection uses the messages defined in Section 5 above.

9.2.1 Modification to Path

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

The format of the sender descriptor for unidirectional LSPs is:

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

The format of the sender descriptor for bidirectional LSPs is:

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

Note that LABEL_REQUEST, SENDER_TSPEC and UPSTREAM_LABEL are not required for a call; however these are mandatory objects. As such, for backwards compatibility purposes, processing of these objects for a call follows the following rules:

- These objects are ignored upon receipt; however, a valid-formatted object (e.g., by using the received valid object in the transmitted message) must be included in the generated message.

9.2.2 Modification to Resv

```
<Resv Message> ::=       <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> |
             <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <SESSION>
     <RSVP_HOP>
     <TIME_VALUES>
     <CALL_OPS>
     <CALL_ID>
     [ <RESV_CONFIRM> ]
     [ <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>
     [ <RECORD_ROUTE> ]
     | <FF flow descriptor list>
     <FF flow descriptor>
<FF flow descriptor> ::=
     [ <FLOWSPEC> ]
     <FILTER_SPEC>
     [ <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>
     [ <RECORD_ROUTE> ]
```

Note that FILTER_SPEC and LABEL are not required for a call; however these are mandatory objects. As such, for backwards compatibility purposes, processing of these objects for a call follows the following rules:

- These objects are ignored upon receipt; however, a valid-formatted object (e.g., by using the received valid object in the transmitted message) must be included in the generated message.

9.2.3 Modification to PathTear

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

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

9.2.4 Modification to PathErr

```
<PathErr Message> ::=    <Common Header>
     [ <INTEGRITY> ]
     [ [<MESSAGE_ID_ACK> |
             <MESSAGE_ID_NACK>] ... ]
     [ <MESSAGE_ID> ]
     <SESSION>
     <CALL_OPS>
     <CALL_ID>
     <ERROR_SPEC>
     [ <POLICY_DATA> ... ]
     <sender descriptor>
<sender descriptor> ::= (see earlier definition in this section)
```

9.2.5 Modification to Notify

```
<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>
```

10. Security Considerations

This document introduces no new security considerations.

11. IANA Considerations

There are multiple fields and values defined within this document. IANA administers the assignment of these class numbers in the FCFS space as shown in [see: http://www.iana.org/assignments/rsvp-parameters].

11.1 Assignment of New Messages

No new messages are defined to support the functions discussed in this document.

11.2 Assignment of New Objects and Sub-Objects

Two new objects are defined:

- CALL_ID (ASON); this object should be assigned an object identifier of the form 11bbbbbb. A suggested value is 230. Two C-types are defined for this object
- C-Type = 1: Operator specific
- C-Type = 2: Globally unique

For the Type field, there is no range restriction, and the entire range 0x00 to 0xff is open for assignment, with 0x00 to 0x7f assignment based on FCFS and 0x80 to 0xff assignment reserved for Private Use. The assignments are defined in this document:

- Type = 0x01: Source LSR address is 4-bytes
- Type = 0x02: Source LSR address is 16-bytes
- Type = 0x03: Source LSR address is 20-bytes
- Type = 0x04: Source LSR address is 6-bytes
- Type = 0x7f: the source LSR address has the length defined by the vendor
- CALL_OPS (ASON); this object should be assigned an object identifier of the form 11bbbbbb. The value is 228. One C-type is defined for this object; the value is 1.

One new sub-object is defined under the GENERALIZED_UNI object:

- SPC_LABEL; this sub-object is part of the GENERALIZED_UNI object, as a sub-type of the EGRESS_LABEL sub-object (which is Type=4). The value is sub-type=2.

11.3 Assignment of New C-Types

Three new C-Types are defined for the SESSION object (Class-num = 1):

- C-Type = 12 (ASON): UNI_IPv6 SESSION object
- C-Type = 15 (ASON): ENNI_IPv4 SESSION object
- C-Type = 16 (ASON): ENNI_IPv6 SESSION object

11.4 Assignment of New Error Code/Values

No new error codes are required. The following new error values are defined. Error code 24 is defined in [RFC3209].

24/103 (ASON): No route available toward source
24/104 (ASON): Unacceptable interface ID
24/105 (ASON): Invalid/unknown call ID
24/106 (ASON): Invalid SPC interface ID/label

12. Acknowledgements

The authors would like to thank Osama Aboul-Magd, Jerry Ash, Sergio Belotti, Greg Bernstein, Adrian Farrel, Nic Larkin, Lyndon Ong, Dimitri Papadimitriou, Bala Rajagopalan, and Yangguang Xu for their comments and contributions to the document.

13. References

13.1 Normative References

[G8080] ITU-T Rec. G.8080/Y.1304, Architecture for the Automatically Switched Optical Network (ASON), November 2001.

[G7713] ITU-T Rec. G.7713/Y.1704, Distributed Call and Connection Management (DCM), November 2001.

[G7713.2] DCM Signalling Mechanisms Using GMPLS RSVP-TE, ITU-T G.7713.2, January 2003.

[OIF-UNI1] UNI 1.0 Signaling Specification, The Optical Internetworking Forum, http://www.oiforum.com/public/UNI_1.0_ia.html

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2205] Braden, R., Editor, Zhang, L., Berson, S., Herzog, S. and S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F. and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", RFC 2961, April 2001.

[RFC3209] Awaduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V. and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", RFC 3209, December 2001.

[RFC3471] Berger, L., Editor, "Generalized Multi-Protocol Label Switching (MPLS) - Signaling Functional Description", RFC 3471, January 2003.

[RFC3473] Berger, L., Editor, "Generalized Multi-Protocol Label Switching (MPLS) Signaling - Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions", RFC 3473, January 2003.

[RFC3476] Rajagopalan, R., "Label Distribution Protocol (LDP) and Resource ReserVation Protocol (RSVP) Extensions for Optical UNI Signaling", RFC 3476, March 2003.

[ITU-LIAISE] http://www.ietf.org/IESG/LIAISON/ITU-OIF.html

13.2 Informative References

[G807] ITU-T Rec. G.807/Y.1301, Requirements For Automatic Switched Transport Networks (ASTN), July 2001

[IPO-RQTS] Aboul-Magd, O., "Automatic Switched Optical Network (ASON) Architecture and Its Related Protocols", Work in Progress.

[GMPLS-ASON] Lin, Z., "Requirements for Generalized MPLS (GMPLS) Usage and Extensions For Automatically Switched Optical Network (ASON)", Work in Progress.

[ASON-RQTS] Xue, Y., "Carrier Optical Services Requirements", Work in Progress.

[GMPLS-SDHSONET] Mannie, E., "GMPLS Extensions for SONET and SDH Control", Work in Progress.

14. Intellectual Property

The IETF takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on the IETF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC 2028. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementors or users of this specification can be obtained from the IETF Secretariat.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the IETF Executive Director.

15. Contributors Contact Information

Sergio Belotti
Alcatel
Via Trento 30,
I-20059 Vimercate, Italy
EMail: sergio.belotti@netit.alcatel.it

Nic Larkin
Data Connection
100 Church Street,
Enfield
Middlesex EN2 6BQ, UK
EMail: npl@dataconnection.com

Dimitri Papadimitriou
Alcatel
Francis Wellesplein 1,
B-2018 Antwerpen, Belgium
EMail: Dimitri.Papadimitriou@alcatel.be

Yangguang Xu
Lucent
1600 Osgood St, Room 21-2A41
North Andover, MA 01845-1043
EMail: xuyg@lucent.com

16. Authors' Addresses

Zhi-Wei Lin
New York City Transit
2 Broadway, Room C3.25
New York, NY 10004

EMail: zhiwlin@nyct.com

Dimitrios Pendarakis
Tellium
2 Crescent Place
Oceanport, NJ 07757-0901

EMail: dpendarakis@tellium.com

17. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.