CCAMP Working Group
Internet Draft
Category: Standard Track

J. Drake (Calient)
D. Papadimitriou (Alcatel)
A. Farrel (Old Dog Consulting)
D. Brungard (ATT)
Z. Ali (Cisco)

Expiration Date: April 2004

October 2003

Generalized MPLS (GMPLS) RSVP-TE Signalling in support of Automatically Switched Optical Network (ASON)

draft-dimitri-ccamp-gmpls-rsvp-te-ason-01.txt

Status of this Memo

1. Abstract

This document specifies how Generalized MPLS (GMPLS) RSVP-TE signaling may be used and extended to satisfy the requirements of the Automatically Switched Optical Network (ASON) architecture specified by the ITU-T. The requirements are in a companion document "Requirements for Generalized MPLS (GMPLS) Usage and Extensions for Automatically Switched Optical Network (ASON)."

In particular, this document details the mechanisms for setting up Soft Permanent Connections (SPC), the necessary extensions in delivering full and logical call/connection separation support, the extended restart capabilities during control plane failures, extended label usage and crankback signalling capability.

The mechanisms proposed in the present document are applicable to any environment (including multi-area) and for any type of

interface: packet, layer-2, time-division multiplexed, lambda or fiber switching.

2. 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 [RFC2119].

In addition, the reader is assumed to be familiar with the terminology used in [RFC3471] and [RFC3473].

3. Introduction

This document describes how GMPLS RSVP-TE signaling [RFC-3473] can be used and extended in support of Automatically Optical Switched Networks (ASON) as specified in the ITU-T G.8080 recommendation [G.8080]. Note, however, that the mechanisms that it describes and references have a larger scope than the one described in this document.

[ASON-REQ] identifies the requirements to be covered by the extensions to the GMPLS signaling protocols to support the capabilities of an ASON network.

The following are expected from the GMPLS protocol suite to realize the needed ASON functionality:
a) support for soft permanent connection functionality
b) support for call and connection separation
c) support for call segments
d) support for extended restart capabilities during control plane failures
e) support for extended label association
f) support for crankback capability.

This document is aligned with the [RSVP-CHANGE] process, which requires evaluation of the existing protocol functionality for achieving the requested functionality and justification for any requested changes or new extensions. In this context, the following summarizes the evaluation and assumptions made:

1. The requirements for LSP setup can be achieved using the peer model described in [RFC3473] or the overlay model described in [GMPLS-OVERLAY]. Thus, the processing of standard objects and functions (such as Explicit Route object and Record Route object) are exactly as described in those documents.

2. The second is that any GMPLS RSVP object, message or procedure not defined in this document or in a directly referenced document is handled exactly as described in [RFC3473], [RFC3209] and [RFC2205]. An important consideration is that the procedures introduced by this document do not introduce any forward or backward compatibility issue.

3. The mechanisms proposed in this document are not restricted to LSC or TDM capable interfaces, but are equally applicable to any

packet or layer-2 interfaces. As a consequence, the present document proposes ubiquitously applicable RSVP extensions.

3.1 Comparison with Previous Work

Informational RFCs [RFC3474] and [RFC3476] document extensions to and uses of GMPLS signaling to meet the requirements of ASON Distributed Call and Connection Management (DCM) as specified in [G.7713] and [OIF-UNI] implementation agreement, respectively.

While both RFCs make use of GMPLS RSVP-TE signaling, there are key differences from the problem statement in [ASON-REQ] and the solution provided by these Informational RFCs. These differences result from the development of a fuller and clearer set of requirements in [G.8080] after the time that [RFC3474] was published and [ASON-REQ] considerations for compatibility aspects with GMPLS [RFC3473]. These differences are enumerated below and detailed in Appendix 1.

1. As described in [G.8080], there are various models and multiple methods of achieving connections across multiple domains. [RFC3474] is similar to a cooperative connection model between domains, that is, there is no overall coordination, and it uses point-to-point E-NNI signaling between inter-domain border controllers (i.e. single-hop LSP). Additionally, it requires address resolution at both border controllers regardless of the address space used. Recent enhancements to [G.8080] include end-to-end network capabilities based on flexible path selection (end-to-end) to support optimal route selection i.e. source-based rerouting and crankback.
To provide for these enhancements and future capabilities (e.g., VPNs), [ASON-REQ] is based on an inter-domain model using an end to end call model, modeling multiple domains as one virtual network and optional one-time (ingress) address resolution (optional, if multiple address families are needed). Note that this model is same model used by [RFC3471] and [RFC3473].

2. [RFC3474] distinguishes between use of [RFC3474] for ASON

networks and use of [RFC3473] for GMPLS networks; no compatibility aspects are addressed, whereas, [ASON-REQ] addresses ASON requirements for GMPLS networks. Backward compatibility allows for a migration or coexistence with GMPLS RSVP-TE [RFC3473] use. [ASON-REQ] requires that for any new and existing GMPLS features, [RFC3473] transit nodes do not need to be updated and do not need to modify their behavior to support the end-to-end features of ASON. The solution provided by [RFC3474] is not backward compatible with [RFC3473], and [RFC3474] can not be used in a network with [RFC3473], as incorrect network behavior will result.

3. While existing GMPLS signalling [RFC3473] supports Soft Permanent Connections (SPCs), [RFC3474] defines a new mechanism to support SPCs, and this new mechanism is incompatible with [RFC3473].

4. [RFC3474] does not support full call and connection separation, multiple connections per call, or ingress/egress node capability negotiation prior to connection establishment.

5. [RFC3474] does not support call segment signaling mechanisms, as required in [G.8080] and [G.7713].

6. [RFC3474] defines control plane restart capabilities that are incompatible with those described in [RFC3473].

7. [RFC3474] does not support crankback signaling mechanisms [GMPLS-CRANK], as required in [G.8080] and [G.7713].

3.2 Applicability

The requirements placed on the signaling plane of an optical network to support the capabilities of an Automatically Switched Optical Network (see [ASON-REQ]) can be met by both the peer model and the overlay model as described below.

Some extensions to the core signaling features (see [RFC3473]) are required in support of some of the requirements. [GMPLS-OVERLAY] defines a common set of standard procedures for the overlay model. Other documents referenced in specific subsections of this document define specific protocol extensions in support of specific ASON requirements.

3.2.1 Peer Model

In the peer model, the ingress and egress nodes play a full part in

the GMPLS network from a signaling point of view. Routing information may be fully or partially distributed to the ingress and egress nodes. This behavior is described [RFC3471] and [RFC3473].

Note that this model supports a User to Network Interface (UNI) separation. The ingress node may make an LSP setup request to the network using standard GMPLS procedures.

3.2.2 Overlay Model

In the overlay model, the ingress and/or the egress nodes are not full players in the GMPLS network. Routing information leaked to the edge nodes is very limited. Signaling information may be filtered and substituted by the network. This process is described in [GMPLS-OVERLAY].

Note that this model supports a UNI separation. The ingress node may initiate an LSP setup request to the network using standard GMPLS procedures. The modifications to behavior described in [GMPLS-OVERLAY] apply to the nodes within the network and not ingress or egress nodes.

4. Soft Permanent Connection (SPC)

A Soft Permanent Connection (SPC) is defined as a permanent connection at the network edges and as a switched connection within the network.

SPC setup is provided using Explicit Label Control as specified in [RFC3473]. This solution is applicable in both the peer and overlay models. For the overlay model, [GMPLS-OVERLAY] describes the procedure for unambiguous identification of both the egress link and label.

5. Call/Connection Separation

The call concept for optical networks is defined in [G.8080] where it is used to deliver the following capabilities:

- Verification and identification of the call initiator (prior to LSP setup)

- Support of virtual concatenation with diverse path component LSPs

- Multiple LSP association with a single call (note aspects related to recovery are covered in [GMPLS-FUNCT] and [GMPLS-E2E])

- Facilitate control plane operations by allowing operational status change of the associated LSP.

Procedures and protocol extensions to support Call setup, and the association of Calls with Connections are described in sections 10 and onwards of this document.

6. Control Plane Restart Capabilities

Restart capabilities are provided by GMPLS RSVP-TE signaling in case of control plane failure including nodal and control channel faults. The handling of node and control channels faults is described in [RFC3473] Section 9. No additional RSVP mechanisms or objects are required to fulfill the ASON control plane restart capabilities.

However, it should be noted that restart considerations must form part of each of the procedures referenced from or described in this document.

7. Extended Label Association

Dynamic discovery of label associations as described in [ASON-REQ] can be either performed through manual provisioning or using the Link Management Protocol [LMP] capabilities.

8. Crankback Signaling

Crankback signaling allows a connection setup request to be retried on an alternate path that detours around a blocked link or node upon

a setup failure, for instance, because a link or a node along the selected path has insufficient resources. Crankback mechanisms may also be applied during connection recovery by indicating the location of the failed link or node. This would significantly improve the successful recovery ratio for failed connections, especially in situations where a large number of setup requests are simultaneously triggered.

Crankback mechanisms for (GMPLS) RSVP-TE signaling are covered in a dedicated companion document [GMPLS-CRANK]. That document is intended to fulfill all the corresponding ASON requirements as well as satisfying any other crankback needs.

9. Call Segments

Call segments capabilities MUST be supported by both independent

call setup and simultaneous call/connection setup.

Procedures and (GMPLS) RSVP-TE signaling protocol extensions to support call segments are described in sections 13.4.1 of this document.

10. Concepts and Terms

The concept of a Call and a Connection are discussed in the ASON architecture [G.8080]. This section is not intended as a substitute for that document, but is a brief summary of the key terms and concepts.

10.1 What is a Call?

A Call is an agreement between endpoints possibly in cooperation with the nodes that provide access to the network. Call setup may include capability exchange, policy, authorization and security.

A Call is used to facilitate and manage a set of Connections that provide end to end data services. While Connections require state to be maintained at nodes along the data path within the network, Calls do not involve the participation of transit nodes except to forward the Call management requests as transparent messages.

A Call may be established and maintained independently of the Connections that it supports.

10.2 A Hierarchy of Calls, Connections, Tunnels and LSPs

Clearly there is a hierarchical relationship between Calls and Connections. One or more Connections may be associated to a Call. A Connection may not be part of more than one call. A Connection may, however, exist without a Call.

In GMPLS, a Connection is identified with a GMPLS TE Tunnel. Commonly a Tunnel is identified with a single LSP, but it should be

noted that for protection, load balancing and many other functions, a Tunnel may be supported by multiple parallel LSPs. The following identification reproduces this hierarchy:

Call IDs are unique within the context of the pair of addresses that are the source and destination of the Call.

Tunnel IDs are unique within the context of the Session (that is the destination of the Tunnel). Applications may also find it convenient

to keep the Tunnel ID unique within the context of a Call.

LSP IDs are unique within the context of a Tunnel.

Note that the Call_ID value of zero is reserved and MUST NOT be used during LSP-independent call establishment.

Throughout the remainder of this document, the terms LSP and Tunnel are used interchangeably with the term Connection. The case of a Tunnel that is supported by more than one LSP is covered implicitly.

10.3 Exchanging Access Link Capabilities

It is useful for the ingress node of an LSP to know the link capabilities of the link between the network and the egress node. This information may allow the ingress node to tailor its LSP request to fit those capabilities and to better utilize network resources with regard to those capabilities.

In particular, this may be used to achieve end-to-end spectral routing attribute negotiation for signal quality negotiation (such as BER) in photonic environments where network edges are signal regeneration capable. Similarly, it may be used to provide end-to-end spatial routing attribute negotiation in multi-area routing environments, in particular, when TE links have been bundled based on technology specific attributes.

Call setup may provide a suitable mechanism to exchange information for this purpose, although several other possibilities exist.

10.3.1 Peer Networks

In peer networks, there may be no need to distribute additional link capability information over and above the information distributed by the TE and GMPLS extensions to the IGP. Further, it is possible that future extensions to these IGPs will allow the distribution of more detailed information including optical impairments.

10.3.2 Overlay Networks

In overlay networks, edge link information may not be visible within the core network, nor (and specifically) at other edge nodes. This may prevent an ingress from requesting suitable LSP characteristics to ensure successful LSP setup.

Various solutions to this problem exist including the definition of

static TE links (that is, not advertised by a routing protocol) between the core network and the edge nodes. Nevertheless, special procedures may be necessary to advertise edge TE link information to the edge nodes outside of the network without advertising the information specific to the contents of the network.

In the future, when the requirements are understood on the information that needs to be supported, TE extensions to EGPs may be defined that provide this function.

10.3.3 Utilizing Call Setup

In the event that IGP and EGP solutions are not available in overlay networks, there is still a requirement to advertise edge link capabilities.

The Call setup procedure provides an opportunity to discover edge link capabilities of remote edge nodes before LSP setup is attempted. The LINK CAPABILITY object is defined to allow this information to be exchanged. The information that is included in this object is similar to that distributed by GMPLS-capable IGPs (see [GMPLS-RTG]).

11. Protocol Extensions for Calls and Connections

This section describes the protocol extensions needed in support of Call identification and management of Calls and Connections. Procedures for the use of these protocol extensions are described in section 12.

11.1 Call Identification

As soon as the concept of a call is introduced, it is necessary to support some means of identifying the call. This becomes particularly important when calls and connections are separated and connections must contain some reference to the call.

According to [ASON-REQ], a call may be identified by a sequence of bytes that may have considerable (but not arbitrary) length. A Call ID of 40 bytes would not be unreasonable. It is not the place of this document to supply rules for encoding or parsing Call IDs, but it must provide a suitable means to communicate Call IDs within the protocol. The full call identification as required by ASON is referred to as the long Call ID.

The Call_ID is only relevant at the sender and receiver nodes. Maintenance of this information in the signaling state is not mandated at any intermediate node. Thus no change in [RFC3473] transit implementations is required and there are no backward compatibility issues. Forward compatibility is maintained by using

the existing default values to indicate that no Call processing is required.

11.1.1 Long Form Call Identification

The "Session Name" attribute of the SESSION_ATTRIBUTE Object is used to carry the long form of the Call ID.

A unique value per call is inserted in the "Session Name" field by the initiator of the call. Subsequent network nodes MAY inspect this object and MUST forward this object transparently across network interfaces until reaching the egress node. Note that the structure of this field MAY be the object of further formatting depending on the naming convention(s). However, [RFC3209] defines the "Session Name" field as a Null padded display string, and that any formatting conventions for the Call ID must be limited to this scope.

11.1.2 Short Form Call Identification

The connections (LSPs) associated with a call need to carry a reference to the call - the Call ID. Each LSP MAY carry the full long Call ID in the "Session Name" of the SESSION ATTRIBUTE object to achieve this purpose. However, existing (and future) implementations may need to place other strings in this field (in particular, the field is currently intended to provide the Session Name). To allow for this possibility a new field is added to the signaling protocol to identify an individual LSP with the Call to which it belongs.

The new field is a 16-bit identifier (unique within the context of the address pairing provided by the Tunnel_End_Point_Address and the Sender_Address) that MUST be exchanged during Call initialization and is used on all subsequent LSP setups that are associated with the Call. This identifier is known as the short Call ID and is encoded as described in Section 11.1.3. When relevant, the Call Id MUST NOT be used as part of the processing to determine the session to which an RSVP signaling message applies. This does not generate any backward compatibility issue since the reserved field of the SESSION object defined in [RFC3209] MUST NOT be examined on receipt.

In the unlikely case of short Call_ID exhaustion, local node policy decides upon specific actions to be taken. Local policy details are outside of the scope of this document.

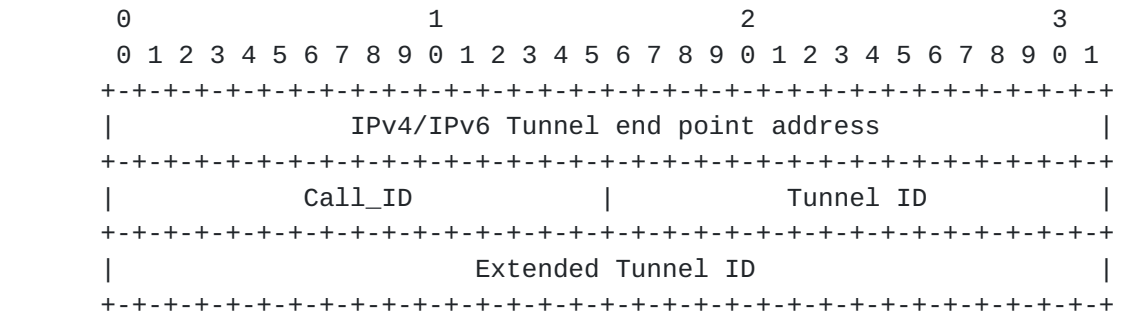
11.1.3 Short Form Call ID Encoding

The short Call ID is carried in a 16-bit field in the SESSION Object

used during Call and LSP setup. The field used was previously reserved (MUST be set to zero on transmission and ignored on receipt). This ensures backward compatibility with nodes that do not utilize calls.

The figure below shows the new version of the object.

Class = SESSION, Class-Num = 1, C-Type = 7(IPv4)/8(IPv6)



IPv4/IPv6 Tunnel End Point Address: 32 bits/128 bits - see [RFC3209]

Call_ID: 16 bits

A 16-bit identifier used in the SESSION object that remains constant over the life of the call. The Call_ID value MUST be set to zero when there is no corresponding call.

Tunnel ID: 16 bits - see [RFC3209]

Extended Tunnel ID: 32 bits/128 bits - see [RFC3209]

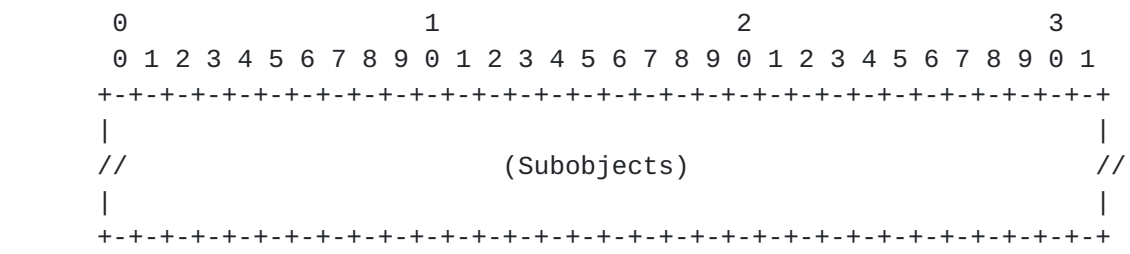
11.2 LINK_CAPABILITY object

The LINK CAPABILITY object is introduced to support link capability exchange during Call setup. The object includes the bundled link local capabilities of the call initiating node (or terminating node) indicated by the source address of the Notify message.

The Class Number is selected so that the nodes that do not recognize this object drop it silently. That is, the top bit is set and the next bit is clear.

This object has the following format:

Class-Num = TBA (form 10bbbbbb), C_Type = 1



The contents of the LINK_CAPABILITY object is defined as series of variable-length data items called subobjects. The subobject format is defined in [RFC3209].

The following subobjects are currently defined:

- Type 1: the link local IPv4 address (numbered bundle) using the format defined in [RFC3209]
- Type 2: the link local IPv6 address (numbered bundle) using the format defined in [RFC3209]
- Type 4: the link local identifier (unnumbered links and bundles) using the format defined in [RFC3477]
- Type 64: the Maximum Reservable Bandwidth corresponding to this bundle (see [BUNDLE])
- Type 65: the Interface Switching Capability Descriptor (see [GMPLS-RTG]) corresponding to this bundle (see also [BUNDLE]).

Note: future revisions of this document may extend the above list.

This object MAY also be used to exchange more than one bundled link capability. In this case, the following ordering MUST be followed: one identifier subobject (Type 1, 2 or 4) MUST be inserted before any capability subobject (Type 64 or 65) to which it refers.

11.3 Revised Message Formats

One message (the Notify message) is enhanced to support Call establishment and teardown of Calls that operate independent of LSPs. See section 12 for a description of the procedures.

11.3.1 Notify Message

The Notify message is modified in support of Call establishment by the optional addition of the LINK CAPABILTY object. Further, the SESSION ATTRIBUTE object is added to the <notify session> sequence to carry the long Call ID. The presence of the SESSION ATTIBUTE object may be used to distinguish a Notify message used for Call management

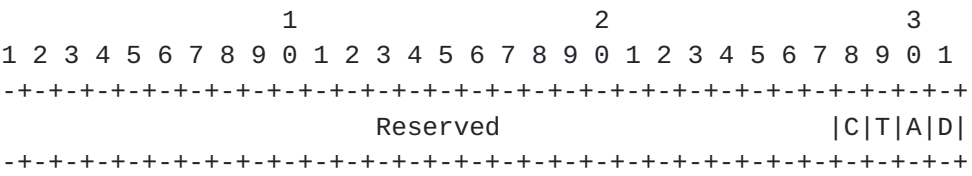
The format of the Notify Message is as follows:

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

<notify session list> ::= [ <notify session list> ] <notify session>

<notify session>  ::= <SESSION> [ <ADMIN_STATUS> ]
                      [ <POLICY_DATA>...]
                      [ <LINK_CAPABILITY> ]
                      | <SESSION_ATTRIBUTE> ]
                      [ <sender descriptor> | <flow descriptor> ]

<sender descriptor> ::= see [RFC3473]

<flow descriptor> ::= see [RFC3473]
```


11.4 Administrative Status Object

Messages (such as Notifys, Paths, etc.) exchanged for Call control and management purposes carry a specific new bit (the Call Management or C bit) in the ADMIN STATUS object.

The format of the contents of the ADMIN_STATUS object are both dictated by [RFC3473] in favor of [RFC3471]. The new 'C' bit is added as shown below.



Reflect (R): 1 bit - see [RFC3471]
Testing (T): 1 bit - see [RFC3471]
Administratively down (A): 1 bit - see [RFC3471]
Deletion in progress (D): 1 bit - see [RFC3471]

Call Management (C): 1 bit
This bit is set when the message is being used to control and manage a Call.

The procedures for the use of the C bit are described in section 12.

Note that the use of the C bit may appear as redundant since Call setup can be distinguished by the presence of the SESSION ATTRIBUTE object in a Notify message or an non-zero short Call Id in a Path message. However, in the case of lost messages and node restart, this further distinction is useful to distinguish Path messages that set up Calls from Path messages that belong to calls.

12. Procedures in Support of Calls and Connections

12.1 Call/Connection Setup Procedures

This section describes the processing steps for call and connection setup. There are four cases considered:

- A Call and Connection may be established simultaneously. That is, a Connection may be established and designated as belonging to a new Call. It is an implementation decision how the work a the ingress and egress points is split and whether the qualities of the Call are policed before, after or at the same time as those of the Connection. In the event that the establishment of either the Call or the Connection fails, an error is returned as described in section 12.4.2 and neither is set up.

- A Call can be set up on its own. That is, without any associated Connection. It is assumed that Connections will be added to the Call at a later time, but this is neither a requirement nor

a constraint.

- A Connection may be added to an existing Call. This may happen if the Call was set up without any associated Connections, or if a further Connection is added to a Call that already has one or more associated Connections.

- A Connection may be established without any reference to a Call. This encompasses the previous LSP setup procedure.

Note that a Call MAY NOT be imposed upon a Connection that is already established. To do so would require changing the short Call Id in the Session Object of the existing LSPs and this would constitute a change in the Session Identifier. This is not allowed by existing protocol specifications.

Call and Connection teardown procedures are described later in Section 12.7.

12.2 Independent Call Setup

It is possible to set up a Call before, and independent of, LSP setup. A Call setup without LSPs MUST follow the procedure described in this section.

Prior to the LSP establishment, Call setup MAY necessitate verification of the link status and link capability negotiation between the Call ingress node and the Call egress node. The procedure described below is applied only once for a Call and hence only once for the set of LSPs associated with a Call.

The Notify message (see [RFC3473]) is used to signal the Call setup request and response. The new Call Management (C) bit is used to indicate that this Notify is managing a Call. The Notify message is sent with source and destination IPv4/IPv6 address set to any of the routable ingress/egress node addresses respectively.

At least one session MUST be listed in the <notify session list> of the Notify message. In order to allow for long identification of the Call the SESSION_ATTRIBUTE object is added as part of the <notify session list>. Note that the ERROR SPEC object is not relevant in Call setup and MUST carry the Error Code zero ('Confirmation') to indicate that there is no error.

During Call setup, the ADMIN STATUS object is sent with the following bits set. Bits not listed MUST be set to zero.

R - to cause the egress to respond
C - to indicate that this message is managing a Call.

The SESSION, SESSION ATTRIBUTE, SENDER_TEMPLATE, SENDER_TSPEC objects included in the <notify session> of the Notify message are built as follows:

- The SESSION object includes as Tunnel_End_Point_Address any of the call terminating (egress) node's IPv4/IPv6 routable addresses. The Call_ID is set to a non-zero value unique within the context of the address pairing provided by the Tunnel_End_Point_Address and the Sender_Address from the SENDER TEMPLATE object (see below). Note that the Call_ID value of zero is reserved and MUST NOT be used during LSP-independent call establishment. The Tunnel_ID of the SESSION object is not relevant for this procedure and SHOULD be set to zero. The Extended_Tunnel_ID of the SESSION object is not relevant for this procedure and MAY be set to zero or to an address of the ingress node.

- The SESSION ATTRIBUTE object contains priority flags. Currently no use of these flags is envisioned, however, future work may identify value is assigning priorities to Calls; accordingly the Priority fields MAY be set to non-zero values. None of the Flags in the SESSION ATTRIBUTE object are relevant to this process and this field SHOULD be set to zero. The Session Name field is used to carry the long Call Id as described in Section 11.

- The SENDER_TEMPLATE object includes as Sender Address any of the call initiating (ingress) node's IPv4/IPv6 routable addresses. The LSP_ID is not relevant and SHOULD be set to zero.

- The bandwidth value inserted in the SENDER_TSPEC and FLOWSPEC objects MUST be ignored upon receipt and SHOULD be set to zero when sent.

Additionally, ingress/egress nodes that need to communicate their respective link local capabilities may include a LINK_CAPABILITY object in the Notify message.

The receiver of a Notify message may identify whether it is part of Call management or reporting an error by the presence or absence of the SESSION ATTRIUBTE object in the <notify session list>. Full clarity, however, may be achieved by inspection of the new Call Management (C) bit in the ADMIN STATUS object.

Note that the POLICY_DATA object may be included in the <notify session list> and may be used to identify requestor credentials, account numbers, limits, quotas, etc. This object is opaque to RSVP, which simply passes it to policy control when required.

Message IDs MUST be used during independent Call setup.

12.2.1 Accepting Independent Call Setup

A node that receives a Notify message carrying the ADMIN STATUS object with the R and C bits set is being requested to set up a Call. The receiver may perform authorization and policy according to local requirements.

If the Call is acceptable, the receiver responds with a Notify message reflecting the information from the Call request with two exceptions.

- The responder removes any LINK CAPABLITY object that was received

and MAY insert a LINK CAPABILITY object that describes its own access link.

- The ADMIN STATUS object is sent with only the C bit set. All other bits MUST be set to zero.

The responder MAY use the Message ID object to ensure reliable delivery of the response. If no Message ID Acknowledgement is received after the configured number of retries, the responder should continue to assume that the Call was successfully established. Call liveliness procedures are covered in section 12.8.

12.2.2 Rejecting Independent Call Setup

Call setup may fail or be rejected.

If the Notify message can not be delivered, no Message ID acknowledgement will be received by the sender. In the event that the sender has retransmitted the Notify message a configurable number of times without receiving a Message ID Acknowledgement (as described in [RFC3473]), the initiator SHOULD declare the Call failed and SHOULD send a Call teardown request (see section 12.7).

It is also possible that a Message ID Acknowledgement is received but no Call response Notify message is received. In this case, the initiator MAY re-send the Call setup request a configurable number of times (see Section 12.8) before declaring the Call has failed. At this point the initiator MUST send a Call teardown request (see Section 12.7).

If the Notify message cannot be parsed or is in error it MAY be responded to with a Notify message carrying the error code 13 ('Unknown object class') or 14 ('Unknown object C-Type').

The Call setup may be rejected by the receiver because of security, authorization or policy reasons. Suitable error codes already exist and can be used in the ERROR SPEC object included in the Notify message sent in response.

Error response Notify messages SHOULD also use the Message ID object to achieve reliable delivery. No action should be taken on the failure to receive a Message ID Acknowledgement after the configured number of retries.

12.3 Adding a Connections to a Call

Once a Call has been established, LSPs can be added to the Call. Since the short Call ID is part of the SESSION Object, any LSP that

has the same Call ID value in the SESSION Object belongs to the same Call. There will be no confusion between LSPs that are associated with a Call and those which are not since the Call ID value MUST be equal to zero for LSPs which are not associated with a Call.

LSPs for different Calls can be distinguished because the Call ID is unique within the context of the source address (in the SENDER TEMPLATE) and the destination address (in the SESSION).

Ingress and egress nodes may group together LSPs associated with the same call and process them as a group according to implementation requirements. Transit nodes need not be aware of the association of multiple LSPs with the same Call.

The ingress node MAY choose to set the "Session Name" of an LSP to match the long Call ID of the associated Call and the "Session Name" MAY still be used to distinguish between virtually concatenated LSPs belonging to the same Call. Thus, there is not necessarily a one-to-one mapping between the "Session Name" of an LSP and the short Call_ID.

The C bit of the ADMIN STATUS object MUST NOT be set on LSP messages.

12.3.1 Adding a Reverse Direction LSP to a Call

Note that once a Call has been established it is symmetric. That is, either end of the Call may add LSPs to the Call.

Special care is needed when managing LSPs in the reverse direction since the addresses in the SESSION and SENDER TEMPLATE are reversed. However, since the short Call ID is unique in the context of a given ingress-egress address pair it may safely be used to associate the LSP with the Call.

12.4 Simultaneous Call/Connection Setup

It is not always necessary to establish a Call before adding Connections to the Call. Where the features made available by independent Call setup are not required, a simplification can be made by establish a Call at the same time as the first Connection associated to the Call.

Simultaneous Call and LSP setup requires the usage of Call identification and an indication that a Call is being managed. No other protocol mechanisms beyond those described in [RFC3473] are needed. Normal RSVP-TE GMPLS processing takes place.

The Path message used to simultaneously set up the Call and LSP MUST carry the ADMIN STATUS object with the R and C bits set. Other bits

may be set or cleared according to the requirements of LSP setup. The D bit MUST NOT be set.

The Path message MUST also carry the long Call ID in the Session Name field of the SESSION ATTRIBUTE Object as described above. This field is not available to contain a Session Name distinct from the Call ID.

A non-zero short Call ID MUST be placed in the new Call ID field of the SESSION Object as described above. The reserved value of zero is used when the LSP is being set up with no association to a Call.

12.4.1 Accepting Simultaneous Call/Connection Setup

A Path message that requests simultaneous Call and Connection setup is subject to local authorization and policy procedures applicable to Call establishment in addition to the standard procedures associated with LSP setup described in [RFC3473].

If the Call and LSP setup is to be accepted, a Resv message is returned. The Resv message MUST carry the ADMIN STATUS object with the R bit clear and the C bit set. Other bits may be set or cleared according to the requirements of LSP setup. The D bit MUST NOT be set.

The Call ID must be reflected in the SESSION object.

12.4.2 Rejecting Simultaneous Call/Connection Setup

The Path message that is sent to set up a Call and Connection simultaneously may fail or be rejected.

Failures may include all those reasons described in [RFC3473]. Additionally, policy and authorization reasons specifically associated with Call setup may cause the Path message to be rejected.

The PathErr message is issued to signal such failures and no new error codes are required. It is RECOMMENDED that the procedures for PathErr with state removal described in [RFC3473] is used during Call setup failure processing.

12.5 Call-Free Connection Setup

It continues to be possible to set up LSPs as per [RFC3473] without associating them with a Call. If the short Call ID in the SESSION

Object is set to zero, there is no associated Call and the Session Name field in the SESSION ATTRIBUTE Object SHOULD be interpreted simply as the name of the session (see [RFC3209]).

The new C bit in the ADMIN STATUS SHOULD be set to zero in such messages and MUST be ignored if the Call ID is zero.

12.6 Call Collision

Since Calls are symmetrical, it is possible that both ends of a call will attempt to establish a Call with the same long Call ID at the same time. This is only an issue if the source and destination address pair matches. This situation can be avoided by applying some rules to the contents of the long Call ID, but that is outside the scope of this document.

If a node that has sent a Call setup request and has not yet received a response, itself receives a Call setup request with the same long Call ID and matching source/destination addresses it should process as follows.

- If its source address is numerically greater than the remote source address, it MUST discard the received message and continue to wait for a response to its setup request.

- If its source address is numerically smaller than the remote source address, it MUST discard state associated with the Call setup that it initiated, and MUST respond to the received Call setup.

In the second case, special processing is necessary if simultaneous Call and Connection establishment was being used. Firstly, the node that is discarding the Call that it initiated MUST send a PathTear message to remove state from transit nodes. Secondly, this node may want to hold onto the Connection request and establish an LSP once the Call is in place since only the Call that it was trying to establish has been set up by the destination - the Connection may still be required.

A further possibility for contention arises when Call IDs are assigned by a pair of nodes for two distinct Calls that are set up simultaneously. In this event a node receives a Call setup request carrying a short Call ID that matches one that it previously sent for the same address pair. The following processing MUST be followed.

- If the receiver's source address is numerically greater than the remote source address, the receiver returns an error (Notify message or PathErr as appropriate) with the new Error Code 'Call Management' (TBD) and the new Error Value 'Call ID Contention' (TBD).

- If the receiver's source address is numerically less than the remote source address, the receiver accepts and processes the Call request. It will receive an error message sent as described above, and at that point it selects a new short Call ID and re-sends the Call setup request.

12.7 Call/Connection Teardown

As with Call/Connection setup, there are several cases to consider.

- Removal of a Connection from a Call
- Removal of the last Connection from a Call
- Teardown of an 'empty' Call

The case of tearing down an LSP that is not associated with a Call does not need to be examined as it follows exactly the procedures described in [RFC3473].

12.7.1 Removal of a Connection from a Call

An LSP that is associated with a Call may be deleted using the standard procedures described in [RFC3743]. No special procedures are required.

Note that it is not possible to remove an LSP from a Call without deleting the LSP. It is not valid to change the short Call ID from non-zero to zero since this involves a change to the SESSION object, which is not allowed.

12.7.2 Removal of the Last Connection from a Call

When the last LSP associated with a Call is deleted the question arises as to what happens to the Call. Since a Call may exist independently of Connections, it is not always acceptable to say that the removal of the last LSP from a Call removes the Call.

If the Call was set up using independent Call setup (that is, using a Notify message) the removal of the last LSP does not remove the Call and the procedures described in the next section MUST be used

to delete the Call.

If the Call was set up using simultaneous Call/Connection establishment, the removal of the last LSP does remove the Call and the Call ID becomes invalid.

12.7.3 Teardown of an 'Empty' Call

When all LSPs have been removed from a Call that was set up independent of Connections, the Call may be torn down or left for use by future LSPs.

Deletion of such Calls is achieved by sending a Notify message just as for Call setup, but the ADMIN STATUS object carries the R, D and C bits on the teardown request and the D and C bits on the teardown response. Other bits MUST be set to zero.

When a Notify message is sent for deleting a call and the initiator does not receive the corresponding reflected Notify message (or possibly even the Message ID Ack), the initiator MAY retry the deletion request using the same retry procedures as used during Call establishment. If no response is received after full retry, the node deleting the Call MAY declare the Call deleted, but under such

circumstances the node SHOULD avoid re-using the long or short Call IDs for at least the five times the Notify refresh period.

12.7.4 Tearing a Call with Existing Connections

If a Notify request with the D bit of the ADMIN STATUS object set is received for a Call for which LSPs still exist, the request MUST be rejected with the Error Code 'Call Management' (TBD) and Error Value 'Connection Still Exists' (TBD).

12.7.5 Tearing a Call from the Egress

Since Calls are symmetric they may be torn down from the ingress or egress.

If the Call was established using simultaneous Call/Connection setup the removal of the last LSP deletes the Call. This, regardless of whether the LSP is torn down by using a PathTear message (for an egress-initiated LSP) or by using a PathErr message with the Path_State_Removed flag set (for an ingress-initiated LSP).

If the Call was established using independent Call/Connection setup

and the Call is 'empty' it may be deleted by the egress sending a Notify message just as described above.

Note that there is still a possibility that both ends of a Call initiate a simultaneous Call deletion. In this case, the Notify message acting as teardown request is interpreted by its recipient as a teardown response. Since the Notify messages carry the R bit in the ADMIN STATUS object, they are responded to anyway. If a teardown request Notify message is received for an unknown Call ID it is, nevertheless, responded to in the affirmative.

12.8 Control Plane Survivability

Delivery of Notify messages is secured using message ID acknowledgements as described in previous sections.

Notify messages provide end-to-end communication that does not rely on constant paths through the network but are routed according to IGP routing information. No consideration is, therefore, required for network resilience (for example, make-before-break, protection, fast re-route), although end-to-end resilience is of interest for node restart and completely disjoint networks.

Periodic Notify messages SHOULD be sent by the initiator and terminator of the Call to keep the Call alive and to handle ingress or egress node restart. The time period for these retransmissions is a local matter, but it is RECOMMENDED that this period should be twice the refresh period of the LSPs associated with the Call. The Notify messages are identical to those sent as if establishing the Call for the first time. A node that receives a refresh Notify message MUST respond with a Notify response. A node that receives a

refresh Notify message (response or request) MAY reset its timer - thus, in normal processing, Notify refreshes involve a single exchange once per time period.

A node that is unsure of the status of a Call MAY immediately send a Notify message as if establishing the Call for the first time.

Failure to receive a refresh Notify request has no meaning. If it receives no response to a refresh Notify request (including no Message ID Acknowledgement) a node MAY assume that the remote node is unreachable or unavailable. It is a local policy matter whether this causes the local node to teardown associated LSPs and delete the Call.

In the event that an edge node restarts without preserved state, it

MAY relearn LSP state from adjacent nodes and Call state from remote nodes. If a Path or Resv message is received with a non-zero Call ID but without the C bit in the ADMIN STATUS, and for a Call ID that is not recognized, the receiver is RECOMMENDED to assume that the Call establishment is delayed and ignore the received message. If the Call setup never materializes the failure by the restarting node to refresh state will cause the LSPs to be torn down. Optionally, the receiver of such an LSP message for an unknown Call ID may return an error (PathErr or ResvErr) with the error code 'Call Management' (TBD) and Error Value 'Unknown Call ID' (TBD).

13. Applicability of Call and Connection Procedures

This section considers the applicability of the different Call establishment procedures to different network models. This section is informative and is not intended to prescribe or prevent other options.

13.1 Peer Model

Both independent and simultaneous Call/Connection setup are appropriate in this model.

Since the access link properties and other traffic-engineering attributes are likely known through the IGP, the LINK CAPABILITY object is not usually required.

13.2 Multi-Area Networks

Both independent and simultaneous Call/Connection setup are appropriate in this model.

Possibly, access link properties and other traffic-engineering attributes are not known since the areas do not leak this sort of information. In this case, the independent Call setup mechanism may be preferred to allow the inclusion of the LINK CAPABILITY object.

13.3 Overlay Model

Both independent and simultaneous Call/Connection setup are appropriate in this model.

It is possible in this model that access link properties and other traffic-engineering attributes are not shared across the core network. In this case, the independent Call setup mechanism may be

preferred to allow the inclusion of the LINK CAPABILITY object.

Further, the first node in the network may be responsible for managing the Call. In this case, the Notify message that is used to set up the Call is addressed to the first node of the core network. Moreover, neither the long Call ID nor the short Call ID is supplied (the Session Name Length is set to zero and the Call ID value is set to zero). The Notify message is passed to the first network node which is responsible for generating the long and short Call IDs before dispatching the message to the remote Call end point (which is known from the SESSION object). Similarly, the first network node may be responsible for generating the long and short Call IDs for inclusion in Path messages that have the C bit set in the ADMIN STATUS object.

Further, when used in an overlay context, the first core node is allowed (see [GMPLS-OVERLAY]) to replace the Session Name assigned by the ingress node and passed in the Path message. In the case of Call management, the first network node MUST in addition 1) be aware that the name it inserts MUST be a long Call ID and 2) replace the long Call ID when it returns the Resv message to the ingress node.

13.4 External Call Managers

Third party Call management agents may be used to apply policy and authorization at a point that is neither the initiator nor terminator of the Call. The previous example in the overlay model is a special example of this, but the process and procedures are identical.

13.4.1 Call Segments

Call segments exist between a set of default and configured External Call Managers along a path between the ingress and egress nodes, and use the protocols described in this document.

The techniques that are used by a given service provider to identify which External Call Managers within its network should process a given call are beyond the scope of this document.

For independent call setup, an External Call manager uses normal IP routing to route the Notify message to the next External Call Manager. For simultaneous call/connection setup, an External Call

Manager expands the EXPLICIT_ROUTE Object to route the Path message to the next External Call Manager.

14. Non-support of Call ID

It is important that the procedures described above operate as seamlessly as possible with legacy nodes that do not support the extensions described.

Clearly there is no need to consider the case where the Call initiator does not support Call setup initiation.

14.1 Non-Support by External Call Managers

It is unlikely that a Call initiator will be configured to send Call establishment Notify requests to an external Call manager including the first network node, if that node does not support Call setup.

A node that receives an unexpected Call setup request will fall into one of the following categories.

- Node does not support RSVP. The message will fail to be delivered or responded. No Message ID Acknowledgement will be sent. The initiator will retry and then give up.

- Node supports RSVP or RSVP-TE but not GMPLS. The message will be delivered but not understood. It will be discarded. No Message ID Acknowledgement will be sent. The initiator will retry and then give up.

- Node supports GMPLS but not Call management. The message will be delivered, but parsing will fail because of the presence of the SESSION ATTRIBUTE object. A Message ID Acknowledgement may be sent before the parse fails. When the parse fails the Notify message may be discarded in which case the initiator will retry and then give up, alternatively a parse error may be generated and returned in a Notify message which will indicate to the initiator that Call management is not set up.

14.2 Non-Support by Transit Node

Transit nodes SHOULD not examine Notify messages that are not addressed to them. However, they will see short Call IDs in all LSPs associated with Calls. Further, they will see the C bit in the ADMIN STATUS object of Path and Resv messages that are used to establish Calls.

Previous specifications state that these fields SHOULD be ignored on receipt and MUST be transmitted as zero. This is interpreted by some implementations as meaning that the fields should be zeroed before the objects are forwarded. If this happens, LSP setup (and so

possibly Call setup if simultaneous establishment is used) will not be possible. If either of the fields is zeroed either on the Path or

the Resv message, the Resv will reach the initiator with the field set to zero - this is indication to the initiator that some node in the network is preventing Call management. Use of Explicit Routes may help to mitigate this issue by avoiding such nodes. The use of independent Call setup may also help since it removes the need for the C bit in the Path and Resv messages. Ultimately, however, it may be necessary to upgrade the offending nodes to handle these protocol extensions.

14.3 Non-Support by Egress Node

It is unlikely that an attempt will be made to set up a Call to remote node that does not support Calls.

If the egress node does not support Call management through the Notify message it will react (as described in Section 14.1) in the same way as an external Call manager.

If the egress node does not support the use of the C bit in the ADMIN STATUS object or the Call ID in the SESSION object, it MAY respond with the fields zeroed in which case the initiator will know that the Call setup has failed.

On the other hand, it is possible that the egress will respond copying the fields from the Path message without understanding or acting on the fields. In this case the initiator will believe that the Call has been set up when it has not. This occurrence can be prevented using the independent Call setup procedures, but is, in any case, detected when a Notify message is sent to keep the Call alive.

15. Security Considerations

Please refer to each of the referenced documents for a description of the security considerations applicable to the features that they provide.

15.1 Call and Connection Security Considerations

Call setup is vulnerable to attacks both of spoofing and denial of service. Since Call setup uses either Path messages or Notify messages, the process can be protected by the measures applicable to securing those messages as described in [RFC3471], [RFC3209] and [RFC2205].

Note, additionally, that the process of Call establishment independent of LSP setup may be used to apply an extra level of authentication and policy to hop-by-hop LSP setup. It may be possible to protect the Call setup exchange using end-to-end security mechanisms such as those provided by Insect (see [RFC2402] and [RFC2406]).

16. IANA Considerations

A new RSVP object is introduced:

o LINK CAPABILITY object

Class-Num = TBA (form 10bbbbbb)

The Class Number is selected so that nodes not recognizing this object fail drop it silently. That is, the top bit is set and the next bit is clear.

C-Type = 1 (TE Link Capabilities)

New RSVP Error Codes and Error Values are introduced

o Error Codes:

- 'Call Management' (TBD)

o Error Values:

- 'Call Management/Call ID Contention' (TBD)
- 'Call Management/Connections Still Exist' (TBD)
- 'Call Management/Unknown Call ID' (TBD)

17. Acknowledgements

The authors would like to thank George Swallow, Yakov Rekhter, Lou Berger and Jerry Ash for their very useful input and comments to this document.

18. Intellectual Property Considerations

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 BCP-11. 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 implementers 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.

19. References

19.1 Normative References

[ASON-REQ] D.Papadimitriou, et al., "Requirements for Generalized MPLS (GMPLS) Usage and Extensions for Automatically Switched Optical Network (ASON)," Work in progress, Oct'03.

[BUNDLE] K.Kompella, Y.Rekhter and L.Berger, "Link Bundling in MPLS Traffic Engineering," Work in Progress.

[GMPLS-CRANK] A.Farrel (Editor) et al., "Crankback Routing Extensions for MPLS Signaling," Work in progress, Jun'03.

[GMPLS-FUNCT] J.P.Lang and B.Rajagopalan (Editors) et al., "Generalized MPLS Recovery Functional Specification," Work in Progress, Sep'03.

[GMPLS-OVERLAY] G.Swallow et al., "GMPLS RSVP Support for the Overlay Model," Work in Progress, Feb'03.

[GMPLS-ROUTING] K.Kompella and Y.Rekhter (Editors) et al., "Routing Extensions in Support of Generalized MPLS," Work in Progress, Oct'03.

[LMP] J.P.Lang (Editor) et al. "Link Management Protocol (LMP) - Version 1," Work in progress, Oct'03.

[RFC2026] S.Bradner, "The Internet Standards Process --

Revision 3," BCP 9, RFC 2026, Oct'96.

[RFC2119] S.Bradner, "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, Mar'97.

[RFC2205] R.Braden et al., "Resource ReSerVation Protocol (RSVP)- Version 1 Functional Specification," RFC 2205, Sep'97

[RFC2402] S.Kent and R.Atkinson, "IP Authentication Header," RFC 2402, Nov'98.

[RFC2406] S.Kent and R.Atkinson, "IP Encapsulating Payload (ESP)," RFC 2406, Nov'98.

[RFC3209] D.Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, Dec'01.

[RFC3471] L.Berger (Editor) et al., "Generalized MPLS - Signaling Functional Description," RFC 3471, Jan'03.

[RFC3473] L.Berger (Editor) et al., "Generalized MPLS

Signaling - RSVP-TE Extensions," RFC 3473, Jan'03.

[RFC3477] K.Kompella and Y.Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)," RFC 3477, Jan'03.

[RSVP-CHANGE] K.Kompella and J.P.Lang, "Procedures for Modifying RSVP," Work in Progress, draft-kompella-rsvp-change-01.txt, Jun'03.

19.2 Informative References

[RFC3474] Z.Lin (Editor), " 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)," RFC 3474, Mar'03.

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

3476, Mar'03.

[G.7713] ITU-T, "Distributed Call and Connection Management," Recommendation G.7713/Y.1304, Nov'01.

[G.8080] ITU-T, "Architecture for the Automatically Switched Optical Network (ASON)," Recommendation G.8080/ Y.1304, Nov'01 (and Revision, Jan'03).

20. Author's Addresses

Dimitri Papadimitriou (Alcatel)
Fr. Wellesplein 1,
B-2018 Antwerpen, Belgium
Phone: +32 3 240-8491
EMail: dimitri.papadimitriou@alcatel.be

John Drake (Calient)
5853 Rue Ferrari,
San Jose, CA 95138, USA
Phone: +1 408 972-3720
EMail: jdrake@calient.net

Adrian Farrel
Old Dog Consulting
Phone: +44 (0) 1978 860944
EMail: adrian@olddog.co.uk

Deborah Brungard (AT&T)

Rm. D1-3C22 - 200 S. Laurel Ave.
Middletown, NJ 07748, USA
EMail: dbrungard@att.com

Zafar Ali (Cisco)
100 South Main St. #200
Ann Arbor, MI 48104, USA
EMail: zali@cisco.com

Appendix 1: Analysis of RFC 3474 (and RFC 3476) against GMPLS RSVP-TE Signaling Requirements in support of ASON

This appendix analyzes the rationale and the relevance of the informational IANA code-point assignments RFCs [RFC3474] and [RFC3476] against the ASON/GMPLS requirements specified in [ASON-REQ]. [ASON-REQ] identifies the requirements to be covered by the extensions to the GMPLS signaling protocols (see [RFC3471] and [RFC3473]) to support the capabilities of an ASON network. The following are expected from the GMPLS protocol suite to realize the needed ASON functionality:

- o soft permanent connection capability
- o call and connection separation

o call segments
o extended restart capabilities during control plane failures
o extended label usage
o crankback capability

Informational RFCs [RFC3474] and [RFC3476] document extensions to and uses of GMPLS signaling to meet the requirements of ASON Distributed Call and Connection Management (DCM) as specified in [G.7713] and [OIF-UNI] implementation agreement, respectively. Both RFCs make use of GMPLS RSVP-TE signaling. However, there are key differences from the problem statement in [ASON-REQ] and the solution provided by these Informational RFCs. These differences result from the development of a fuller and clearer set of requirements in [G.8080] after the time that [RFC3474] was published and [ASON-REQ] considerations for compatibility issues with GMPLS [RFC3473] (see also [RSVP-CHANGE]). These differences lead to a substantially different protocol solution and implementation.

1. Support for UNI and E-NNI Signaling Session

In GMPLS (see [RFC3473] and related), a connection is identified with a GMPLS tunnel. A tunnel is generally identified with a single LSP but may be supported by multiple LSPs.

LSP tunnels are identified by a combination of the SESSION and SENDER_TEMPLATE objects. The relevant fields are as follows.

IPv4 (or IPv6) tunnel end point address

IPv4 (or IPv6) address of the egress node for the tunnel.

Tunnel ID

A 16-bit identifier used in the SESSION that remains constant over the life of the tunnel.

Extended Tunnel ID

A 32-bit (IPv4) or 128-bit (IPv6) identifier used in the SESSION that remains constant over the life of the tunnel.

Normally set to all zeros. Ingress nodes that wish to narrow the scope of a SESSION to the ingress-egress pair may place their IP address here as a globally unique identifier.

IPv4 (or IPv6) tunnel sender address

IPv4 (or IPv6) address for a sender node

LSP ID

A 16-bit identifier used in the SENDER_TEMPLATE and the FILTER_SPEC that can be changed to allow a sender to share resources with itself.

The first three of these are in the SESSION object and are the basic identification of the tunnel. The "Extended Tunnel ID" MAY be set to an IP address of the head-end LSR allowing the scope of the SESSION to be narrowed to only LSPs sent by that node. The last two are in the SENDER_TEMPLATE. Multiple LSPs may belong to the same tunnel (and thus SESSION) and in this case they are uniquely identified by their LSP IDs.

In contrast, [RFC3474] (and [RFC3476]) define an E-NNI IPv4 and IPv6 SESSION object (UNI IPv4 and IPv6 SESSION object, respectively). [RFC3474] mandates the use of these objects to support the E-NNI (UNI, respectively) signaling session when IPv4 and IPv6 addressing is used. The "Tunnel End-point Address" field contains the IPv4 or IPv6 address of the downstream controller. In addition, [RFC3476] mandates that the "Extended Tunnel ID" field to be set to the IPv4 or IPv6 of the upstream controller. It also mandates that the tunnel sender address field of the SENDER_TEMPLATE be set to the IPv4 or the IPv6 address of the upstream controller.

Thus, these RFCs define a point-to-point signaling interface allowing for LSP tunnel provisioning between adjacent controllers only. This approach mandates the introduction of an additional object and sub-objects for connection identification purposes (see [RFC3476]): the GENERALIZED_UNI object and its connection end-point address sub-objects (IPv4/IPv6/NSAP) referred to as "TNA or Transport Network Address" as defined by the [OIF-UNI] implementation agreement.

The situation is summarized in the following table, using the following example and a connection set from node A to Z:

```
   UNI                E-NNI              E-NNI                UNI
A ----- B -- ... -- I ----- J -- .. -- M ----- N -- ... -- Y ----- Z
```

At node I, the GMPLS compliant [RFC3473] Path message would include the following information in the IP header, the SESSION and SENDER_TEMPLATE objects:

Source IP address (Header): Node I IP Controller Address

```
Dest. IP address (Header):   Node J IP Controller Address
Tunnel End-point Address:    Routable Node Z IP Address
Tunnel ID:                   16 bit (selected by the sender)
Extended Tunnel ID:          optionally set to Node A IP Address
Tunnel Sender Address:       Routable Node A IP Address
LSP ID:                      16 bit (selected by the sender)
```

At node I, the [RFC3474] Path message would include the following:

```
Source IP address (Header):  Node I IP Controller Address
Dest. IP address (Header):   Node J IP Controller Address
Tunnel End-point Address:    Node J IP Controller Address
Tunnel ID:                   16 bit (selected by the sender)
Extended Tunnel ID:          Node I IP Controller Address
Tunnel Sender Address:       Node I IP Controller Address
LSP ID:                      16 bit (selected by the sender)
GENERALIZED_UNI object:
- Source Address (Connection): End-point A Address (IPv4/IPv6/etc.)
- Dest. Address (Connection):  End-point Z Address (IPv4/IPv6/etc.)
```

The same observation would apply at node M, by replacing I by M and J by N.

The following can be thus deduced from the above example:

1. For a given connection, the [RFC3474] point-to-point signaling interface leads to a sequence of at least N different identifications of the same connection when crossing N signaling interfaces (due to the setup and maintenance of N distinct LSP tunnels).

2. The information included in the RSVP message header and in the SESSION/SENDER_TEMPLATE objects, is redundant in [RFC3474].

3. [RFC3474] allows only for single-hop LSP tunnels and mandates the processing of a new object (i.e. the GENERALIZED_UNI object) for the definition of the source and destination connection end-point addresses (A and Z in the above example).

4. The processing of the signaling Path message (in particular, the EXPLICIT ROUTE object) mandates the processing of the GENERALIZED_UNI object at E-NNI reference points and at UNI reference points, for the connection end-point addresses (A and Z, in the above example).

5. Connection end-point addresses A and Z are allowed by [RFC3474] and [RFC3476] to be from different address spaces (for instance, IPv4 source and IPv6 destination or an IPv4 source and NSAP destination). Address resolution, management of addresses (e.g. for uniqueness), and impact evaluation on processing performance, are not provided in these RFCs (considered out of scope).

Note: the [ASON-REQ] addressing model of supporting only IP

addressing is aligned with ITU-T G.7713.1 (PNNI) which only uses NSAP addresses, multiple address families are not supported.

Conclusion: The solution proposed by [RFC3474] and [RFC3476] is not backward compatible with [RFC3473]. A GMPLS-compliant node [RFC3473] is not interoperable with a [RFC3474] or [RFC3476] node. Also, the "RSVP paradigm" is broken because the solution requires that all the UNI reference points (A, B and Y, Z, in the above example) and the E-NNI reference points (I, J and M, N, in the above example) support the GENERALIZED_UNI object. Additionally, the management of the network requires maintaining multiple LSP tunnels per single connection, with no end-to-end view provided for expedient fault notification or recovery operations.

The solution also introduces processing overhead for address resolution that during time critical operations (such as recovery) will severely impact performance and scalability. Whereas the ITU-T G.7713.1 (PNNI) and [ASON-REQ] by using a single address family (with address mapping provided at edge nodes if needed) supports a scalable model for inter-domain interworking applications.

2. Support for Soft Permanent Connections (SPC)

A Soft Permanent Connection (SPC) is defined as a permanent connection at the network edges and as a switched connection within the network.

[RFC3474] mandates the use of the GENERALIZED_UNI subobjects (End-point Connection Address and SPC_LABEL) to support SPC capability. Thus, in addition to suffering from the problem described in Section 4, it mandates the processing of an object where it should never occur. This is because SPCs do not assume the existence of a UNI signaling interface between the source and the destination user-to-network interface. Note also that the SPC_LABEL as defined in [RFC3474] does not comply with the generalized label C-Type definition of [RFC3473] meaning that an implementation adhering to [RFC3473] cannot be the "soft" side of the connection.

This requires the mandatory usage of dedicated connection end-point addresses by the ingress and egress nodes for SPC capability support. The existing RECORD_ROUTE object and its capabilities get corrupted by the use of the dedicated end-point address subobjects falling outside of the corresponding EXPLICIT_ROUTE object.

SPC support is already provided by [RFC3473] using Explicit Label Control and its application to the overlay model in [GMPLS-OVERLAY]. Therefore, [RFC3474] defines a new method for an existing capability of GMPLS signaling.

3. Call/Connection Separation

The call concept for optical networks is defined in [G.8080]. It is used to deliver the following capabilities:

- Verification and identification of the call initiator (prior to LSP setup) including negotiation between call ingress/egress nodes
- Support of multiple connections can be associated with a single call.
- Facilitate control plane operations by allowing operational status change of the associated LSP.

A call is an agreement between end-points (possibly in cooperation with the nodes that provide access to the network) used to manage a set of connections that provide end to end services. While connections require state to be maintained at nodes along the data path within the network, *** calls do not involve the participation of transit nodes except to forward the call management requests as transparent messages ***. Moreover, a call may be established and maintained independently of the connections that it supports.

Also, there is a hierarchical relationship between calls and connections. One or more (or even no) connections may be associated with a given call but a connection can not be part of more than one call. A connection may, however, exist without a call. Moreover, the establishment of a call can be independent ("full call/connection separation") or simultaneous ("logical call/connection separation") from the connection setup (i.e. establishing a call before adding connections to the call or perform these operations simultaneously).

Thus, with the introduction of the call concept, it is necessary to support a means of identifying the call. This becomes important when calls and connections are separated and a connection must contain a reference to its associated call. The following identification enables this hierarchy:

- Call IDs are unique within the context of the pair of addresses that are the source and destination of the call.
- Tunnel IDs are unique within the context of the Session (that is the destination of the Tunnel) and Tunnel IDs may be unique within the context of a Call.
- LSP IDs are unique within the context of a Tunnel.

For this purpose, [RFC3474] introduces two new objects: a CALL_ID and a CALL_OPS object to be used in the Path, Resv, PathTear, PathErr, and Notify messages (note: additional requirements for ResvErr and ResvTear messages' support are not addressed). The CALL_OPS object is also referred to as a "call capability" object, since it specifies the capability of the call. These objects belongs to the range 224-255 defined as "RSVP will silently ignore, but FORWARD an object with a Class Number in this range that it does not understand."

However, the solution described in [RFC3474]:

- Does not provide backward compatible extensions in support of full call/connection separation and thus only supports logical call/ connection separation (i.e. a call with zero connections is not supported). This because node that does not implement [RFC3474],

will not process the CALL_OPS object, though it will establish the *connection* (while forwarding the "Call Setup" message), i.e. allocating labels and possibly attempting to reserve bandwidth. [RFC3474] forbids this behavior by a transit node, but only a node implementing [RFC3474] will know the difference between a call and a connection.

In turn, the required signaling protocol independence between intra- and inter-domain reference points is broken: an operator does not have the possibility to use GMPLS [RFC3473] and must exclusively use [RFC3474].

- Does not describe how to support multiple connections per call but limits the description to a single connection per call. Further, in the case of complete call/connection separation, it does not describe how to add the first connection to the call.

- Does not describe how to support multiple connections per call and limits the description to a single connection per call. Further, it does not describe how to add the first connection to the call when to support call/connection separation.

- Does not specify any procedure for negotiating call ingress/egress node capabilities during call setup.

- Does not allow for call support *independently* of the initiating/ terminating nodes (the CALL_ID is attached to the ingress node) thus restricting the flexibility in terms of call identifiers.

- Requires the inclusion of the CALL ID and CALL OPS objects in

PathErr messages that may be generated at transit nodes, which do not implement [RFC3474] and so do not support these objects.

4. Call Segments

The RFCs [RFC3474] and [RFC3476] cannot, by definition, support call segments signaling mechanisms, as required in [G.8080] and [G.7713], since [RFC3474] does not support full call/connection separation.

5. Control Plane Restart Capabilities

Restart capabilities are provided by GMPLS RSVP-TE signaling in case of control plane failure including nodal and control channel faults. The handling of node and control channels faults is described in [RFC3473] Section 9. No additional RSVP mechanisms or objects are required to fulfill the ASON control plane restart capabilities.

However, [RFC3474] defines additional procedures for control plane recovery, three of them being considered in the context of an interaction with the management plane and thus outside the scope of the present document. The last one expects persistent state storage and the restart mechanism defined in [RFC3473] is to be used for verification of neighbor states, while the persistent storage

provides the local recovery of lost state. However, per [RFC3473], if during the Hello synchronization the restarting node determines that a neighbor does not support state recovery and the restarting node maintains its local state on a per neighbor basis, the restarting node should immediately consider the Recovery as completed. Therefore, the procedure described in [RFC3474] requires disabling state recovery on each neighboring node leading also to an unspecified verification procedure.

6. Extended Label Usage

No specific GMPLS RSVP-TE extensions have been proposed in [RFC3474] for extended label usage.

7. Crankback Signaling

The RFCs [RFC3474] and [RFC3476] do not support crankback signaling mechanisms, as required in [G.8080] and [G.7713].

8. Security Considerations

This is an informational draft and does not introduce any new security considerations.

Please refer to each of the referenced documents for a description of the security considerations applicable to the features that they provide.

Note that although [RFC3474] is an informational RFC it does document new protocol elements and functional behavior and as such introduces new security considerations. In particular, the ability to place authentication and policy details within the context of Call establishment may strengthen the options for security and may weaken the security of subsequent Connection establishment. Also the potential to subvert accidentally or deliberately a soft permanent connection by establishing the soft part of the connection from a false remote node needs to be examined. However, [RFC3474] has a minimal security considerations section.

Full Copyright Statement