

CCAMP Working Group  
Internet Draft  
Updates [RFC 3473](#)  
Proposed Category: Standard Track  
Expiration Date: May 2006

Editors  
D. Papadimitriou (Alcatel)  
A. Farrel (Old Dog Consulting)

November 2005

**Generalized MPLS (GMPLS) RSVP-TE Signaling Extensions  
in support of Calls**

[draft-papadimitriou-ccamp-gmpls-rsvp-te-call-00.txt](#)

Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with [Section 6 of BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at  
<http://www.ietf.org/ietf/1id-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at  
<http://www.ietf.org/shadow.html>.

Abstract

In certain networking topologies it may be advantageous to maintain associations between endpoints and key transit points to support an instance of a service. Such associations are known as Calls.

A Call does not provide the actual connectivity for transmitting user traffic, but only builds a relationship by which subsequent connections may be made. In Generalized MPLS (GMPLS) such connections are known as Label Switched Paths (LSPs).

This document specifies how GMPLS RSVP-TE signaling may be used and extended to support Calls. These mechanisms provide full and logical Call/Connection separation.

The mechanisms proposed in this 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.



## Table of Content

<a href="#">1. Conventions used in this document .....</a>	<a href="#">3</a>
<a href="#">2. Introduction .....</a>	<a href="#">3</a>
<a href="#">2.1 Applicability to ASON .....</a>	<a href="#">4</a>
<a href="#">3. Requirements .....</a>	<a href="#">4</a>
<a href="#">3.1 Basic Call Function .....</a>	<a href="#">4</a>
<a href="#">3.2 Call/Connection Separation .....</a>	<a href="#">4</a>
<a href="#">3.3 Call Segments .....</a>	<a href="#">5</a>
<a href="#">4. Concepts and Terms .....</a>	<a href="#">5</a>
<a href="#">4.1 What is a Call? .....</a>	<a href="#">5</a>
<a href="#">4.2 A Hierarchy of Calls, Connections, Tunnels and LSPs .....</a>	<a href="#">5</a>
<a href="#">4.3 Exchanging Access Link Capabilities .....</a>	<a href="#">6</a>
<a href="#">4.3.1 Network-initiated Calls .....</a>	<a href="#">6</a>
<a href="#">4.3.2 User-initiated Calls .....</a>	<a href="#">7</a>
<a href="#">4.3.3 Utilizing Call Setup .....</a>	<a href="#">7</a>
<a href="#">5. Protocol Extensions for Calls and Connections .....</a>	<a href="#">7</a>
<a href="#">5.1 Call Setup and Teardown .....</a>	<a href="#">7</a>
<a href="#">5.2 Call Identification .....</a>	<a href="#">8</a>
<a href="#">5.2.1 Long Form Call Identification .....</a>	<a href="#">8</a>
<a href="#">5.2.2 Short Form Call Identification .....</a>	<a href="#">8</a>
<a href="#">5.2.3 Short Form Call ID Encoding .....</a>	<a href="#">9</a>
<a href="#">5.3 LINK_CAPABILITY object .....</a>	<a href="#">10</a>
<a href="#">5.4 Revised Message Formats .....</a>	<a href="#">11</a>
<a href="#">5.4.1 Notify Message .....</a>	<a href="#">11</a>
<a href="#">5.5 ADMIN_STATUS Object .....</a>	<a href="#">11</a>
<a href="#">6. Procedures in Support of Calls and Connections .....</a>	<a href="#">12</a>
<a href="#">6.1 Call/Connection Setup Procedures .....</a>	<a href="#">12</a>
<a href="#">6.2 Call Setup .....</a>	<a href="#">12</a>
<a href="#">6.2.1 Accepting Call Setup .....</a>	<a href="#">14</a>
<a href="#">6.2.2 Call Setup Failure and Rejection .....</a>	<a href="#">15</a>
<a href="#">6.3 Adding a Connections to a Call .....</a>	<a href="#">15</a>
<a href="#">6.3.1 Adding a Reverse Direction LSP to a Call .....</a>	<a href="#">16</a>
<a href="#">6.4 Call-Free Connection Setup .....</a>	<a href="#">16</a>
<a href="#">6.5 Call Collision .....</a>	<a href="#">16</a>
<a href="#">6.6 Call/Connection Teardown .....</a>	<a href="#">17</a>
<a href="#">6.6.1 Removal of a Connection from a Call .....</a>	<a href="#">18</a>
<a href="#">6.6.2 Removal of the Last Connection from a Call .....</a>	<a href="#">18</a>
<a href="#">6.6.3 Teardown of an "Empty" Call .....</a>	<a href="#">18</a>
<a href="#">6.6.4 Attempted Teardown of a Call with Existing Connections ....</a>	<a href="#">18</a>
<a href="#">6.6.5 Teardown of a Call from the Egress .....</a>	<a href="#">19</a>
<a href="#">6.7 Control Plane Survivability .....</a>	<a href="#">19</a>
<a href="#">7. Applicability of Call and Connection Procedures .....</a>	<a href="#">20</a>
<a href="#">7.1 Network-initiated Calls .....</a>	<a href="#">20</a>
<a href="#">7.2 User-initiated Calls .....</a>	<a href="#">20</a>
<a href="#">7.3 External Call Managers .....</a>	<a href="#">21</a>
<a href="#">7.3.1 Call Segments .....</a>	<a href="#">21</a>
<a href="#">8. Non-support of Call ID .....</a>	<a href="#">21</a>
<a href="#">8.1 Non-Support by External Call Managers .....</a>	<a href="#">22</a>

<a href="#">8.2</a> Non-Support by Transit Node .....	<a href="#">22</a>
---	--------------------

<a href="#">8.3</a>	Non-Support by Egress Node .....	<a href="#">23</a>
<a href="#">9.</a>	Security Considerations .....	<a href="#">23</a>
<a href="#">9.1</a>	Call and Connection Security Considerations .....	<a href="#">23</a>
<a href="#">10.</a>	IANA Considerations .....	<a href="#">23</a>
<a href="#">10.1</a>	RSVP Objects .....	<a href="#">23</a>
<a href="#">10.2</a>	RSVP Error Codes and Error Values .....	<a href="#">24</a>
<a href="#">10.3</a>	RSVP ADMIN_STATUS object Bits .....	<a href="#">24</a>
<a href="#">11.</a>	Acknowledgements .....	<a href="#">24</a>
<a href="#">12.</a>	References .....	<a href="#">24</a>
<a href="#">12.1</a>	Normative References .....	<a href="#">24</a>
<a href="#">12.2</a>	Informative References .....	<a href="#">26</a>
<a href="#">13.</a>	Authors' Addresses .....	<a href="#">26</a>

## **[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 [[RFC2119](#)].

In addition, the reader is assumed to be familiar with the terminology used in [[RFC3471](#)], [[RFC3473](#)], [[RFC3477](#)] and [[RFC3945](#)].

## **[2.](#) Introduction**

This document defines protocol procedures and extensions to support Calls within Generalized MPLS (GMPLS).

A Call is an association between endpoints and possibly between key transit points (such as network boundaries) in support of an instance of a service. The end-to-end association is termed a "Call," and the association between two transit points or between an endpoint and a transit point is termed a "Call Segment." An entity that processes a Call or Call Segment is called a "Call Manager."

A Call does not provide the actual connectivity for transmitting user traffic, but only builds a relationship by which subsequent connections may be made. In GMPLS such connections are known as Label Switched Paths (LSPs).

A Call may be associated with zero, one or more connections, and a connection may be associated with zero or one Call. Thus full and logical Call/Connection separation is needed.

An example of the requirement for Calls can be found in the ITU-T's Automatically Switched Optical Network (ASON) architecture [[G.8080](#)] and specific requirements for support of Calls in this context can be found in [[RFC4139](#)]. Note, however, that while the mechanisms described in this document meet the requirements stated in [[RFC4139](#)] they have wider applicability.



The mechanisms defined in this document are equally applicable to any packet (PSC) interface, layer-2 interfaces (L2SC), TDM capable interfaces, LSC interfaces or FSC interfaces. The mechanisms and protocol extensions are backward compatible, and can be used for Call management where only the Call Managers need to be aware of the protocol extensions.

## **2.1 Applicability to ASON**

[RFC4139] details the requirements on GMPLS signaling to satisfy the ASON architecture described in [G.8080]. The mechanisms described in this document meet the requirements for Calls as described in Sections 4.2 and 4.3 of [RFC4139] and the additional Call-related requirements in Sections 4.4, 4.7, 5 and 6 of [RFC4139].

[ASON-APPL] describes the applicability of GMPLS protocols to the ASON architecture.

## **3. Requirements**

### **3.1 Basic Call Function**

The Call concept 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.
- Association of multiple LSPs with a single Call (note aspects related to recovery are detailed in [GMPLS-FUNCT] and [GMPLS-E2E]).
- Facilitation of 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 [Section 5](#) and onwards of this document.

### **3.2 Call/Connection Separation**

Full and logical Call and Connection separation is required. That is:

- It MUST be possible to establish a Connection without dependence on a Call.
- It MUST be possible to establish a Call without any associated Connections.





- It MUST be possible to associate more than one Connection with a Call.
- Removal of the last Connection associated with a Call SHOULD NOT result in the automatic removal of the Call except as a matter of local policy at the ingress of the Call.
- Signaling of a Connection associated with a Call MUST NOT require the distribution or retention of Call-related information (state) within the network.

### **3.3 Call Segments**

Call Segments capabilities MUST be supported.

Procedures and (GMPLS) RSVP-TE signaling protocol extensions to support Call Segments are described in [Section 7.3.1](#) of this document.

## **4. Concepts and Terms**

The concept of a Call and a Connection are also discussed in the ASON architecture [[G.8080](#)] and [[RFC4139](#)]. This section is not intended as a substitute for those documents, but is a brief summary of the key terms and concepts.

### **4.1 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.

### **4.2 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 with a Call. A Connection may not be part of more than one Call. A Connection may, however, exist without a Call.

In GMPLS RSVP-TE [[RFC3473](#)], 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.

### **4.3 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.

#### **4.3.1 Network-initiated Calls**

In this case, 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.



#### **4.3.2 User-initiated Calls**

In this case, 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.

#### **4.3.3 Utilizing Call Setup**

When IGP and EGP solutions are not available at the UNI, there is still a requirement to have, at the local edge nodes, the knowledge of the remote 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 [[RFC4202](#)]).

### **5. 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 6](#).

#### **5.1 Call Setup and Teardown**

Calls are established independently of connections through the use of the Notify message. The Notify message is a targeted message and does not follow the path of LSPs through the network.

Simultaneous Call and connection establishment (sometimes referred to as piggybacking) is not supported.



## **5.2 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.

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 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.

Further, the long Call ID is not required as part of the connection (LSP) state even at the sender and receiver nodes so long as some form of correlation is available. This correlation is provided through the short Call ID.

### **5.2.1 Long Form Call Identification**

The long Call ID is only required on the Notify message used to establish the Call. It is carried in the "Session Name" field of the SESSION\_ATTRIBUTE Object on the Notify message.

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.

### **5.2.2 Short Form Call Identification**

The connections (LSPs) associated with a Call need to carry a reference to the Call - the short Call ID. 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





In the unlikely case of short Call\_ID exhaustion, local node policy decides upon specific actions to be taken, but might include the use of second Sender Address. Local policy details are outside of the scope of this document.

Extended Tunnel ID: 32 bits/128 bits (see [\[RFC3209\]](#))



### 5.3 LINK\_CAPABILITY object

The LINK CAPABILITY object is introduced to support link capability exchange during Call setup and MAY be included in a Notify message used for Call setup. This optional 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

```

      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
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     |
//                               (Subobjects)                               //
|                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The contents of the LINK\_CAPABILITY object is defined as series of variable-length data items called subobjects. The subobject format is defined in [\[RFC3209\]](#).

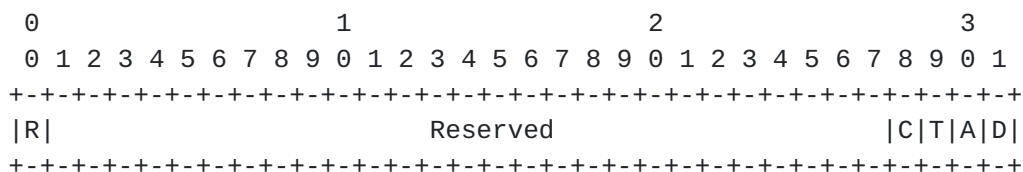
The following subobjects are currently defined:

- 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 [\[RFC4201\]](#))
- Type 65: the interface switching capability descriptor (see [\[RFC4202\]](#)) corresponding to this bundle (see also [\[RFC4201\]](#)).

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.







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 6](#).

## **[6](#). Procedures in Support of Calls and Connections**

### **[6.1](#) Call/Connection Setup Procedures**

This section describes the processing steps for Call and connection setup.

There are three cases considered:

- A Call is set up 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 (see [Section 6.4](#)). 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 6.6](#).

### **[6.2](#) Call Setup**

A Call is set up before, and independent of, LSP (i.e. Connection) setup.

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 in the ADMIN\_STATUS object is used to indicate that this Notify is managing a Call. The Notify message is sent with source and destination IPv4/IPv6 addresses 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 the Notify 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). This value will be used as the short Call ID carried on all messages for LSPs associated with this Call.

Note that the Call\_ID value of zero is reserved and MUST NOT be used since it will be present in SESSION objects of LSPs that are not associated with Calls. 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 is relevant to this process and

this field SHOULD be set to zero. The Session Name field is used

[draft-papadimitriou-ccamp-gmpls-rsvp-te-call-00.txt](#)

[Page 13]

to carry the long Call Id as described in [Section 5](#).

- 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\_ATTRIBUTE 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 Call setup.

#### **[6.2.1](#) Accepting 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 CAPABILITY 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 MUST 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 liveness procedures are covered in [Section 6.7](#).



### **6.2.2 Call Setup Failure and Rejection**

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 [RFC2961]), the initiator SHOULD declare the Call failed and SHOULD send a Call teardown request (see [Section 6.6](#)).

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 6.7](#)) before declaring that the Call has failed. At this point the initiator MUST send a Call teardown request (see [Section 6.6](#)).

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") if appropriate to the error detected.

The Call setup MAY be rejected by the receiver because of security, authorization or policy reasons. Suitable error codes already exist [RFC2205] 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.

### **6.3 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, and the Notify message used to establish the Call carried the same Call ID in its SESSION object.

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, and MUST NOT be equal to zero for a valid Call ID.

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



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.

The C bit of the ADMIN STATUS object MUST NOT be set on LSP messages including on Notify messages that pertain to the LSP and MUST be ignored.

#### **6.3.1 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.

Note that since Calls are defined here to be symmetrical the issue of potential Call ID collision arises. This is discussed in [Section 6.5](#).

#### **6.4 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 MUST be interpreted simply as the name of the session (see [\[RFC3209\]](#)).

The C bit of the ADMIN STATUS object MUST NOT be set on messages for LSP control, including on Notify messages that pertain to LSPs, and MUST be ignored when received on such messages.

#### **6.5 Call Collision**

Since Calls are symmetrical, it is possible that both ends of a Call will attempt to establish Calls with the same long Call IDs at the same time. This is only an issue if the source and destination address pairs match. This situation can be avoided by applying some rules to the contents of the long Call ID, but such mechanisms are 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.

If a node receives a Call setup request carrying an address pair and long Call ID that match an existing Call, the node MUST return an error message (Notify message) with the new Error Code "Call Management" and the new Error Value "Duplicate Call" in response to the new Call request, and MUST NOT make any changes to the existing Call.

A further possibility for contention arises when short Call IDs are assigned by a pair of nodes for two distinct Calls that are set up simultaneously using different long Call IDs. 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) with the new Error Code "Call Management" and the new Error Value "Call ID Contention".
- 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.

## **6.6 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](#)].



### **6.6.1 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.

### **6.6.2 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.

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.

### **6.6.3 Teardown of an "Empty" Call**

When all LSPs have been removed from a Call, the Call may be torn down or left for use by future LSPs.

Deletion of 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.

### **6.6.4 Attempted Teardown of 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" and Error Value "Connections Still Exist". The state of the Call MUST NOT be changed.



#### **6.6.5 Teardown of a Call from the Egress**

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

When the Call is "empty" (has no associated LSPs) it may be deleted by the egress sending a Notify message just as described above.

Note that there is a possibility that both ends of a Call initiate Call deletion at the same time. In this case, the Notify message acting as teardown request MAY be interpreted by its recipient as a teardown response. But since the Notify messages acting as teardown requests carry the R bit in the ADMIN STATUS object, they MUST be responded to anyway. If a teardown request Notify message is received for an unknown Call ID it is, nevertheless, responded to in the affirmative.

#### **6.7 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. Notify messages 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 shortest refresh period of any LSP associated with the Call. When there are no LSPs associated with a Call, an LSR is RECOMMENDED to use a refresh period of no less than one minute. The Notify messages are identical to those sent as if establishing the Call for the first time, except for the LINK CAPABILITY object, which may have changed since the Call was first established, due to, e.g., the establishment of connections, link failures, and the addition of new component links. The current link information is useful for the establishment of subsequent connections. A node that receives a refresh Notify message carrying the R bit in the ADMIN STATUS object 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 (sender or receiver) 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 specific meaning. A node that fails to receive a refresh Notify request MAY send its own refresh Notify request to establish the status of the call. If an LSR 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 message) with the error code "Call Management" and Error Value "Unknown Call ID".

## **7. Applicability of Call and Connection Procedures**

This section considers the applicability of the different Call establishment procedures at the NNI and UNI reference points. This section is informative and is not intended to prescribe or prevent other options.

### **7.1 Network-initiated Calls**

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

In multi-domain networks it is possible that access link properties and other traffic-engineering attributes are not known since the domains do not share this sort of information. In this case, the Call setup mechanism may include the LINK CAPABILITY object.

### **7.2 User-initiated Calls**

It is possible that the access link properties and other traffic-engineering attributes are not shared across the core network. In this case, the Call setup mechanism may include the LINK CAPABILITY object.





Further, the first node within 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 by the user network edge node 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).

Further, when used in an overlay context, the first core node is allowed (see [[RFC4208](#)]) 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:

- 1) MAY insert a long Call ID in the Session Name of a Path message
- 2) MUST replace the Session Name with that originally issued by the user edge node when it returns the Resv message to the ingress node.

### **7.3 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 is a particular case of this, but the process and procedures are identical.

#### **7.3.1 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.

An External Call Manager uses normal IP routing to route the Notify message to the next External Call Manager. Notify messages (requests and responses) are therefore encapsulated in IP packets that identify the sending and receiving External Call Managers, but the addresses used to identify the Call (the Sender Address in the SENDER TEMPLATE object and the Tunnel Endpoint Address in the SESSION object) continue to identify the endpoints of the Call.

### **8. 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.

### **8.1 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 supported.

### **8.2 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.

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 will not be possible. If either of the fields is zeroed either on the Path or the Resv message, the Resv message 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. Ultimately, however, it may be necessary to upgrade the offending nodes to handle these protocol extensions.



### **8.3 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 8.1](#)) in the same way as an External Call Manager.

## **9. Security Considerations**

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

### **9.1 Call and Connection Security Considerations**

Call setup is vulnerable to attacks both of spoofing and denial of service. Since Call setup uses Notify messages, the process can be protected by the measures applicable to securing those messages as described in [[RFC3473](#)].

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](#)]).

## **10. IANA Considerations**

### **10.1 RSVP Objects**

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 drop it silently. That is, the top bit is set and the next bit is cleared.

C-Type = 1 (TE Link Capabilities)

The LINK CAPABILITY object is only defined for inclusion on Notify messages.

Refer to [Section 5.3](#) of this document.



## **10.2 RSVP Error Codes and Error Values**

New RSVP Error Codes and Error Values are introduced

o Error Codes:

- Call Management (value TBA)

o Error Values:

- Call Management/Call ID Contention (value TBA)
- Call Management/Connections still Exist (value TBA)
- Call Management/Unknown Call ID (value TBA)
- Call Management/Duplicate Call (value TBA)

## **10.3 RSVP ADMIN\_STATUS object Bits**

[GMPLS-E2E] requests IANA to manage the bits of the RSVP ADMIN\_STATUS object.

One new bit, the C bit, is defined in this document. Bit number 28 is suggested.

See [Section 5.5](#) of this document.

## **11. Acknowledgements**

The authors would like to thank George Swallow, Yakov Rekhter, Lou Berger, Jerry Ash and Kireeti Kompella for their very useful input to and comments on an earlier revision of this document.

## **12. References**

### **12.1 Normative References**

- [GMPLS-E2E] Lang, J.P., Rekhter, Y., and D. Papadimitriou, "RSVP-TE Extensions in support of End-to-End Generalized Multi-Protocol Label Switching (GMPLS)-based Recovery," [draft-ietf-ccamp-gmpls-recovery-e2e-signaling](#), work in progress.
- [GMPLS-FUNCT] Lang, J.P., and B. Rajagopalan (Editors) et al., "Generalized MPLS Recovery Functional Specification," work in progress.
- [RFC2119] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," [BCP 14](#), [RFC 2119](#), March 1997.





- [RFC2205] R. Braden et al., "Resource ReSerVation Protocol (RSVP)- Version 1 Functional Specification," [RFC 2205](#), September 1997.
- [RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header," [RFC 2402](#), November 1998.
- [RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Payload (ESP)," [RFC 2406](#), November 1998.
- [RFC2961] Berger, L., Gan, D., Swallow, G., Pan, P., Tommasi, F. and S. Molendini, "RSVP Refresh Overhead Reduction Extensions", [RFC 2961](#), April 2001.
- [RFC3209] D. Awduche et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels," [RFC 3209](#), December 2001.
- [RFC3471] L. Berger (Editor) et al., "Generalized MPLS - Signaling Functional Description," [RFC 3471](#), January 2003.
- [RFC3473] L. Berger (Editor) et al., "Generalized MPLS Signaling - RSVP-TE Extensions," [RFC 3473](#), January 2003.
- [RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic Engineering (RSVP-TE)," [RFC 3477](#), January 2003.
- [RFC3945] E. Mannie, Ed., "Generalized Multi-Protocol Label Switching (GMPLS) Architecture", [RFC 3945](#), October 2004.
- [RFC4139] D. Papadimitriou, et al., "Requirements for Generalized MPLS (GMPLS) Signaling Usage and Extensions for Automatically Switched Optical Network (ASON)," [RFC 4139](#), July 2005.
- [RFC4201] Kompella K., Rekhter Y., and L. Berger, "Link Bundling in MPLS Traffic Engineering," [RFC 4201](#), October 2005.
- [RFC4202] Kompella, K. and Y. Rekhter (Editors) et al., "Routing Extensions in Support of Generalized MPLS," [RFC 4202](#), October 2005.
- [RFC4208] G. Swallow et al., "GMPLS RSVP Support for the Overlay Model," [RFC 4208](#), October 2005.



## **12.2 Informative References**

[ASON-APPL] D. Papadimitriou et. al., "Generalized MPLS (GMPLS) RSVP-TE signaling usage in support of Automatically Switched Optical Network (ASON)," [draft-ietf-ccamp-gmpls-rsvp-te-ason](#), work in progress.

For information on the availability of the following document, please see <http://www.itu.int>.

[G.8080] ITU-T, "Architecture for the Automatically Switched Optical Network (ASON)," Recommendation G.8080/Y.1304, November 2001 (and Revision, January 2003).

## **13. Authors' Addresses**

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

John Drake  
Boeing Satellite Systems  
2300 East Imperial Highway  
El Segundo, CA 90245  
EMail: John.E.Drake2@boeing.com

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

Arthi Ayyangar (Juniper)  
1194 N.Mathilda Ave  
Sunnyvale, CA 94089, USA  
EMail: arthi@juniper.net



Don Fedyk (Nortel Networks)  
600 Technology Park Drive  
Billerica, MA, 01821, USA  
Email: dwfedyk@nortel.com

#### Intellectual Property Statement

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights 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; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat 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 on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

#### Disclaimer of Validity

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

#### Copyright Statement

Copyright (C) The Internet Society (2005). This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

