                 GMPLS - Communication of Alarm Information

Status of This Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Abstract

   This document describes an extension to Generalized MPLS (Multi-
   Protocol Label Switching) signaling to support communication of alarm
   information. GMPLS signaling already supports the control of alarm
   reporting, but not the communication of alarm information. This
   document presents both a functional description and GMPLS-RSVP
   specifics of such an extension. This document also proposes
   modification of the RSVP ERROR_SPEC object.

   This document updates RFC 3473, "Generalized Multi-Protocol Label
   Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic
   Engineering (RSVP-TE) Extensions", through the addition of new,
   optional protocol elements. It does not change, and is fully
   backward compatible with, the procedures specified in RFC 3473.

Table of Contents

1. Introduction

   GMPLS signaling provides mechanisms that can be used to control the
   reporting of alarms associated with a label switched path (LSP).
   This support is provided via Administrative Status Information
   [RFC3471] and the Admin_Status object [RFC3473]. These mechanisms
   only control if alarm reporting is inhibited. No provision is made
   for communication of alarm information within GMPLS.

   The extension described in this document defines how the alarm
   information associated with a GMPLS LSP may be communicated along the
   path of the LSP. Communication both upstream and downstream is
   supported. The value in communicating such alarm information is that
   this information is then available at every node along the LSP for
   display and diagnostic purposes. Alarm information may also be
   useful in certain traffic protection scenarios, but such uses are out
   of the scope of this document. Alarm communication is supported via
   a new object, new error/alarm information TLVs, and a new
   Administrative Status Information bit.

   The communication of alarms, as described in this document, is
   controllable on a per-LSP basis. Such communication may be useful
   within network configurations where not all nodes support
   communication to a user for reporting of alarms and/or communication
   is needed to support specific applications. The support of this
   functionality is optional.

   The communication of alarms within GMPLS does not imply any
   modification in behavior of processing of alarms, or for the
   communication of alarms outside of GMPLS. Additionally, the
   extension described in this document is not intended to replace any
   (existing) data plane fault propagation techniques.

   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].

1.1. Background

   Problems with data plane state can often be detected by associated
   data plane hardware components. Such data plane problems are
   typically filtered based on elapsed time and local policy. Problems
   that pass the filtering process are normally raised as alarms. These
   alarms are available for display to operators. They also may be
   collected centrally through means that are out of the scope of this
   document.

Not all data plane problems cause the LSP to be immediately torn
down. Further, there may be a desire, particularly in optical
transport networks, to retain an LSP and communicate relevant alarm
information even when the data plane state has failed completely.

Although error information can be reported using PathErr, ResvErr,
and Notify messages, these messages typically indicate a problem in
signaling state and can only report one problem at a time. This
makes it hard to correlate all of the problems that may be associated
with a single LSP and to allow an operator examining the status of an
LSP to view a full list of current problems. This situation is
exacerbated by the absence of any way to communicate that a problem
has been resolved and a corresponding alarm cleared.

The extensions defined in this document allow an operator or a
software component to obtain a full list of current alarms associated
with all of the resources used to support an LSP. The extensions
also ensure that this list is kept up-to-date and synchronized with
the real alarm status in the network. Finally, the extensions make
the list available at every node traversed by an LSP.

2. Alarm Information Communication

A new object is introduced to carry alarm information details. The
details of alarm information are much like the error information
carried in the existing ERROR_SPEC objects. For this reason the
communication of alarm information uses a format that is based on the
communication of error information.

The new object introduced to carry alarm information details is
called an ALARM_SPEC object. This object has the same format as the
ERROR_SPEC object, but uses a new C-Num to avoid the semantics of
error processing. Also, additional TLVs are defined for the IF_ID
ALARM_SPEC objects to support the communication of information
related to a specific alarm. These TLVs may also be useful when
included in ERROR_SPEC objects, e.g., when the ERROR_SPEC object is
carried within a Notify message.

While the details of alarm information are like the details of
existing error communication, the semantics of processing differ.
Alarm information will typically relate to changes in data plane
state, without changes in control state. Alarm information will
always be associated with in-place LSPs. Such information will also
typically be most useful to operators and applications other than
control plane protocol processing. Finally, while error information
is communicated within PathErr, ResvErr, and Notify messages
[RFC3473], alarm information will be carried within Path and Resv
messages.

Path messages are used to carry alarm information to downstream
nodes, and Resv messages are used to carry alarm information to
upstream nodes. The intent of sending alarm information both
upstream and downstream is to provide the same visibility to alarm
information at any point along an LSP. The communication of multiple
alarms associated with an LSP is supported. In this case, multiple
ALARM_SPEC objects will be carried in the Path or Resv messages.

The addition of alarm information to Path and Resv messages is
controlled via a new Administrative Status Information bit.
Administrative Status Information is carried in the Admin_Status
object.

3. GMPLS-RSVP Details

   This section provides the GMPLS-RSVP [RFC3473] specification for
   communication of alarm information. The communication of alarm
   information is OPTIONAL. This section applies to nodes that support
   communication of alarm information.

3.1. ALARM_SPEC Objects

   The ALARM_SPEC objects use the same format as the ERROR_SPEC object,
   but with class number of 198 (assigned by IANA in the form 11bbbbbb,
   per Section 3.1.4).

   o Class = 198, C-Type = 1
      Reserved. (C-Type value defined for ERROR_SPEC, but is not
      defined for use with ALARM_SPEC.)

   o Class = 198, C-Type = 2
      Reserved. (C-Type value defined for ERROR_SPEC, but is not
      defined for use with ALARM_SPEC.)

   o IPv4 IF_ID ALARM_SPEC object: Class = 198, C-Type = 3
      Definition same as IPv4 IF_ID ERROR_SPEC [RFC3473].

   o IPv6 IF_ID ALARM_SPEC object: Class = 198, C-Type = 4
      Definition same as IPv6 IF_ID ERROR_SPEC [RFC3473].

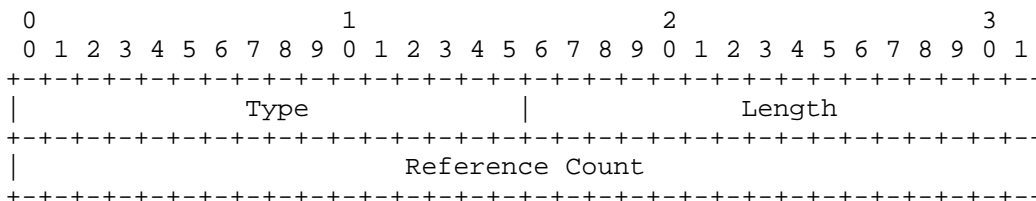3.1.1. IF_ID ALARM_SPEC (and ERROR_SPEC) TLVs

   The following new TLVs are defined for use with the IPv4 and IPv6
   IF_ID ALARM_SPEC objects. They may also be used with the IPv4 and
   IPv6 IF_ID ERROR_SPEC objects. See [RFC3471] Section 9.1.1 for the
   original definition of these values. Note the length provided below
   is for the total TLV. All TLVs defined in this section are OPTIONAL.

The defined TLVs MUST follow any interface identifying TLVs. No
rules apply to the relative ordering of the TLVs defined in this
section.

```
   Type      Length     Description
   --------------------------------
   512        8         REFERENCE_COUNT
   513        8         SEVERITY
   514        8         GLOBAL_TIMESTAMP
   515        8         LOCAL_TIMESTAMP
   516     variable     ERROR_STRING
```

The Reference Count TLV has the following format:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Type             |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        Reference Count                       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Reference Count: 32 bits

   The number of times this alarm has been repeated as determined
   by the reporting node. This field MUST NOT be set to zero, and
   TLVs received with this field set to zero MUST be ignored.

Only one Reference Count TLV may be included in an object.

The Severity TLV has the following format:

```
    0                   1                   2                   3
    0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Type             |             Length            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |              Reserved                  |Impact |  Severity    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
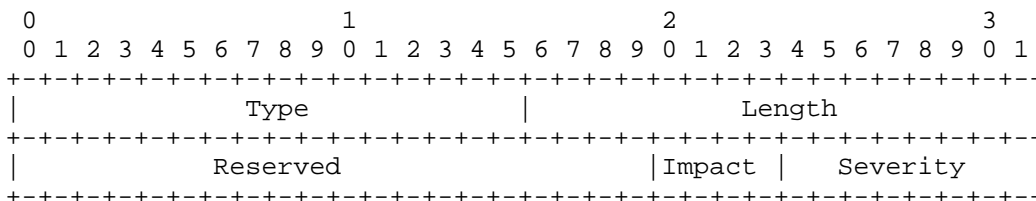
Reserved: 20 bits

   This field is reserved. It MUST be set to zero on generation,
   MUST be ignored on receipt, and MUST be forwarded unchanged and
   unexamined by transit nodes.

Impact: 4 bits

> Indicates the impact of the alarm indicated in the TLV. See
> [M.20] for a general discussion on classification of failures.
> The following values are defined in this document. The details
> of the semantics may be found in [M.20].

```
   Value     Definition
   -----     --------------------
     0       Unspecified impact
     1       Non-Service Affecting (Data traffic not interrupted)
     2       Service Affecting (Data traffic is interrupted)
```
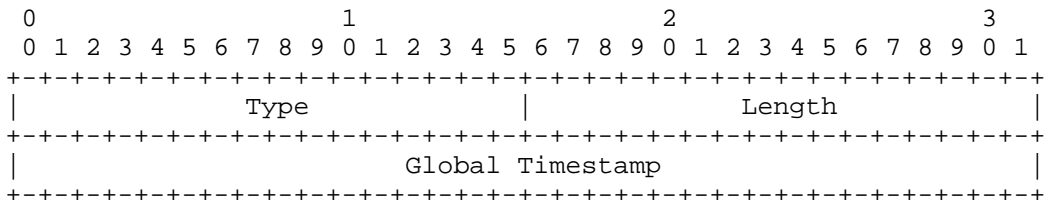
Severity: 8 bits

> Indicates the impact of the alarm indicated in the TLV. See
> [RFC3877] and [M.3100] for more information on severity. The
> following values are defined in this document. The details of
> the semantics may be found in [RFC3877] and [M.3100]:

```
   Value     Definition
   -----     ----------
     0       Cleared
     1       Indeterminate
     2       Critical
     3       Major
     4       Minor
     5       Warning
```

> Only one Severity TLV may be included in an object.

The Global Timestamp TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              Type             |             Length            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Global Timestamp                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
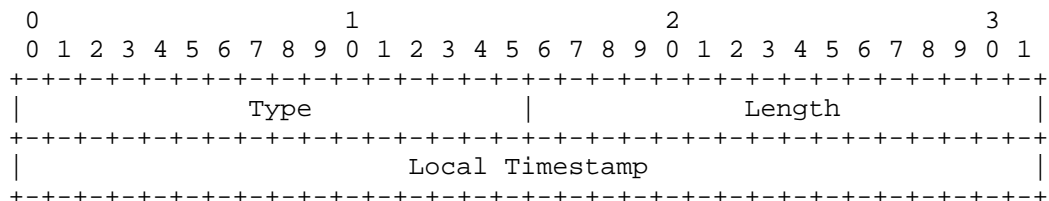
   Global Timestamp: 32 bits

      An unsigned fixed-point integer that indicates the number of
      seconds since 00:00:00 UT on 1 January 1970 according to the
      clock on the node that originates this TLV. This time MAY
      include leap seconds if they are used by the local clock and
      SHOULD contain the same time value as used by the node when the
      alarm is reported through other systems (such as within the
      Management Plane) if global time is used in those reports.

   Only one Global Timestamp TLV may be included in an object.

The Local Timestamp TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Local Timestamp                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
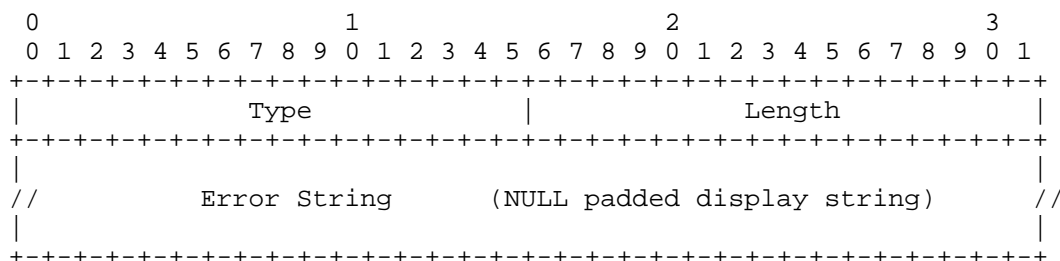
   Local Timestamp: 32 bits

      Number of seconds reported by the local system clock at the
      time the associated alarm was detected on the node that
      originates this TLV. This number is expected to be meaningful
      in the context of the originating node. For example, it may
      indicate the number of seconds since the node rebooted or may
      be a local representation of an unsynchronized real-time clock.

   Only one Local Timestamp TLV may be included in an object.

The Error String TLV has the following format:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|             Type              |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//         Error String     (NULL padded display string)      //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Error String: 32 bits minimum (variable)

A string of characters in US-ASCII, representing the type of
error/alarm. This string is padded to the next largest 4-byte
boundary using null characters. Null padding is not required
when the string is 32-bit aligned. The contents of error
string are implementation dependent. See the condition types
listed in Appendices of [GR833] for a list of example strings.
Note length includes padding.

Multiple Error String TLVs may be included in an object.

3.1.2. Procedures

This section applies to nodes that support the communication of alarm
information. ALARM_SPEC objects are carried in Path and Resv
messages. Multiple ALARM_SPEC objects MAY be present.

Nodes that support the extensions defined in this document SHOULD
store any alarm information from received ALARM_SPEC objects for
future use. All ALARM_SPEC objects received in Path messages SHOULD
be passed unmodified downstream in the corresponding Path messages.
All ALARM_SPEC objects received in Resv messages SHOULD be passed
unmodified upstream in the corresponding Resv messages. ALARM_SPEC
objects are merged in transmitted Resv messages by including a copy
of all ALARM_SPEC objects received in corresponding Resv Messages.

To advertise local alarm information, a node generates an ALARM_SPEC
object for each alarm and adds it to both the Path and Resv messages
for the impacted LSP.

In all cases, appropriate Error Node Address, Error Code, and Error
Values MUST be set (see below for a discussion on Error Code and
Error Values). As the InPlace and NotGuilty flags only have meaning
in ERROR_SPEC objects, they SHOULD NOT be set. TLVs SHOULD be
included in the ALARM_SPEC object to identify the interface, if any,
associated with the alarm. The TLVs defined in [RFC3471] for
identifying interfaces in the IF_ID ERROR_SPEC object [RFC3473]
SHOULD be used for this purpose, but note that TLVs type 4 and 5
(component interfaces) are deprecated by [RFC4201] and SHOULD NOT be
used. TLVs SHOULD also be included to indicate the severity
(Severity TLV), the time (Global Timestamp and/or Local Timestamp
TLVs), and a (brief) string (Error String TLV) associated with the
alarm. The reference count TLV MAY also be included to indicate the
number of times an alarm has been repeated at the reporting node.
ALARM_SPEC objects received from other nodes are not impacted by the
addition of local ALARM_SPEC objects, i.e., they continue to be
processed as described above. The choice of which alarm or alarms to

advertise and which to omit is a local policy matter, and may be
configurable by the user.

There are two ways to indicate time. A global timestamp TLV is used
to provide an absolute time reference for the occurrence of an alarm.
The local timestamp TLV is used to provide time reference for the
occurrence of an alarm that is relative to other information
advertised by the node. The global timestamp SHOULD be used on nodes
that maintain an absolute time reference. Both timestamp TLVs MAY be
used simultaneously.

Note, ALARM_SPEC objects SHOULD NOT be added to the Path and Resv
states of LSPs that are in "alarm communication inhibited" state.
ALARM_SPEC objects MAY be added to the state of LSPs that are in an
"administratively down" state. These states are indicated by the I
and A bits of the Admin_Status object; see Section 3.2.

To remove local alarm information, a node simply removes the matching
locally generated ALARM_SPEC objects from the outgoing Path and Resv
messages. A node MAY modify a locally generated ALARM_SPEC object.

Normal refresh and trigger message processing applies to Path or Resv
messages that contain ALARM_SPEC objects. Note that changes in
ALARM_SPEC objects from one message to the next may include a
modification in the contents of a specific ALARM_SPEC object, or a
change in the number of ALARM_SPEC objects present. All changes in
ALARM_SPEC objects SHOULD be processed as trigger messages.

Failure to follow the above directives, in particular the ones
labeled "SHOULD" and "SHOULD NOT", may result in the alarm
information not being properly or fully communicated.

3.1.3. Error Codes and Values

The Error Codes and Values used in ALARM_SPEC objects are the same as
those used in ERROR_SPEC objects. New Error Code values for use with
both ERROR_SPEC and ALARM_SPEC objects may be assigned to support
alarm types defined by other standards.

In this document we define one new Error Code. The Error Code uses
the value 31 and is referred to as "Alarms". The values used in the
Error Values field when the Error Code is "Alarms" are the same as
the values defined in the IANAItuProbableCause Textual Convention of
IANA-ITU-ALARM-TC-MIB in the Alarm MIB [RFC3877]. Note that these
values are managed by IANA; see http://www.iana.org.

3.1.4. Backwards Compatibility

   The support of ALARM_SPEC objects is OPTIONAL. Non-supporting nodes
   will (according to the rules defined in [RFC2205]) pass the objects
   through the node unmodified, because the ALARM_SPEC object has a
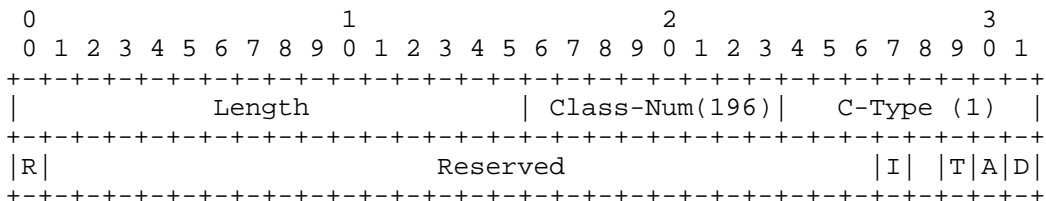   C-Num of the form 11bbbbbb.

   This allows alarm information to be collected and examined in a
   network built from a collection of nodes some of which support the
   communication of alarm information, and some of which do not.

3.2. Controlling Alarm Communication

   Alarm information communication is controlled via Administrative
   Status Information as carried in the Admin_Status object. A new bit
   is defined, called the I bit, that indicates when alarm communication
   is to be inhibited. The definition of this bit does not modify the
   procedures defined in Section 7 of [RFC3473].

3.2.1. Updated Admin_Status Object

   The format of the Admin_Status object is updated to include the I
   bit:

```
      0                   1                   2                   3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |          Length               | Class-Num(196)|   C-Type (1)  |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
     |R|                    Reserved                 |I|  |T|A|D|
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   Inhibit Alarm Communication (I): 1 bit
        When set, indicates that alarm communication is disabled for
        the LSP and that nodes SHOULD NOT add local alarm information.

   See Section 7.1 of [RFC3473] for the definition of the remaining
   bits.

3.2.2. Procedures

   The I bit may be set and cleared using the procedures defined in
   Sections 7.2 and 7.3 of [RFC3473]. A node that receives (or
   generates) an Admin_Status object with the A or I bits set (1),
   SHOULD remove all locally generated alarm information from the
   matching LSP's outgoing Path and Resv messages. When a node receives
   (or generates) an Admin_Status object with the A and I bits clear (0)
   and there is local alarm information present, it SHOULD add the local

   alarm information to the matching LSP's outgoing Path and Resv
   messages.

   The processing of non-locally generated ALARM_SPEC objects MUST NOT
   be impacted by the contents of the Admin_Status object; that is,
   received ALARM_SPEC objects MUST be forwarded unchanged regardless of
   the received or transmitted settings of the I and A bits. Note that,
   per [RFC3473], the absence of the Admin_Status object is equivalent
   to receiving an object containing values all set to zero (0).

   I bit related processing behavior MAY be overridden locally based on
   configuration.

   When generating Notify messages for LSPs with the I bit set, the TLVs
   described in this document MAY be added to the ERROR_SPEC object sent
   in the Notify message.

3.3. Message Formats

   This section presents the RSVP message-related formats as modified by
   this document. The formats specified in [RFC3473] served as the
   basis of these formats. The objects are listed in suggested
   ordering.

   The format of a Path message is as follows:

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

 <sender descriptor> is not modified by this document.

The format of a Resv message is as follows:

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

<flow descriptor list> is not modified by this document.

3.4. Relationship to GMPLS UNI

   [RFC4208] defines how GMPLS may be used in an overlay model to
   provide a user-to-network interface (UNI). In this model,
   restrictions may be applied to the information that is signaled
   between an edge-node and a core-node. This restriction allows the
   core network to limit the information that is visible outside of the
   core. This restriction may be made for confidentiality, privacy, or
   security reasons. It may also be made for operational reasons, for
   example, if the information is only applicable within the core
   network.

   The extensions described in this document are candidates for
   filtering as described in [RFC4208]. In particular, the following
   observations apply.

   o An ingress or egress core-node MAY filter alarms from the GMPLS
      core to a client-node UNI LSP. This may be to protect information
      about the core network, or to indicate that the core network is
      performing or has completed recovery actions for the GMPLS LSP.

   o An ingress or egress core-node MAY modify alarms from the GMPLS
      core when sending to a client-node UNI LSP. This may facilitate
      the UNI client's ability to understand the failure and its effect
      on the data plane, and enable the UNI client to take corrective
      actions in a more appropriate manner.

   o Similarly, an egress core-node MAY choose not to request alarm
      reporting on Path messages that it sends downstream to the overlay
      network.

3.5. Relationship to GMPLS E-NNI

   GMPLS may be used at the external network-to-network interface
   (E-NNI); see [ASON-APPL]. At this interface, restrictions may be
   applied to the information that is signaled between an egress and an
   ingress core-node.

   This restriction allows the ingress core network to limit the
   information that is visible outside of its core network. This
   restriction may be made for confidentiality, privacy, or security
   reasons. It may also be made for operational reasons, for example,
   if the information is only applicable within the core network.

   The extensions described in this document are candidates for
   filtering as described in [ASON-APPL]. In particular, the following
   observations apply.

   o An ingress or egress core-node MAY filter internal core network
      alarms. This may be to protect information about the internal
      network or to indicate that the core network is performing or has
      completed recovery actions for this LSP.

   o An ingress or egress core-node MAY modify internal core network
      alarms. This may facilitate the peering E-NNI (i.e., the egress
      core-node) to understand the failure and its effect on the data
      plane, and take corrective actions in a more appropriate manner or
      prolong the generated alarms upstream/downstream as appropriated.

   o Similarly, an egress/ingress core-node MAY choose not to request
      alarm reporting on Path messages that it sends downstream.

4. Security Considerations

   Some operators may consider alarm information as sensitive. To
   support environments where this is the case, implementations SHOULD
   allow the user to disable the generation of ALARM_SPEC objects, or to
   filter or correlate them at domain boundaries.

   This document introduces no additional security considerations. See
   [RFC3473] for relevant security considerations.

   It may be noted that if the security considerations of [RFC3473] are
   breached, alarm information may be spoofed. Such spoofing would be
   at most annoying and cause slight degradation of control plane
   performance since the details are provided for information only and
   do not result in protocol actions beyond the exchange of messages to
   convey the information. If the protocol security is able to be
   breached sufficiently to allow spoofing of alarm information then

considerably more interesting and exciting damage can be caused by
spoofing other elements of the protocol messages.

5. IANA Considerations

   IANA administered assignment of new values for namespaces defined in
   this document and reviewed in this section.

5.1. New RSVP Object

   IANA made the following assignments in the "Class Names, Class
   Numbers, and Class Types" section of the "RSVP PARAMETERS" registry
   located at http://www.iana.org/assignments/rsvp-parameters.

   A new class named ALARM_SPEC (198) was created in the 11bbbbbb range
   with following values

   o Class = 198, C-Type = 1
      RFC 4783
      Reserved. (C-Type value defined for ERROR_SPEC, but is not
      defined for use with ALARM_SPEC.)

   o Class = 198, C-Type = 2
      RFC 4783
      Reserved. (C-Type value defined for ERROR_SPEC, but is not
      defined for use with ALARM_SPEC.)

   o IPv4 IF_ID ALARM_SPEC object: Class = 198, C-Type = 3
      RFC 4783
      Definition same as IPv4 IF_ID ERROR_SPEC [RFC3473].

   o IPv6 IF_ID ALARM_SPEC object: Class = 198, C-Type = 4
      RFC 4783
      Definition same as IPv6 IF_ID ERROR_SPEC [RFC3473].

   The ALARM_SPEC object uses the Error Code and Error Values from the
   ERROR_SPEC object.

5.2. New Interface ID Types

   IANA made the following assignments in the "Interface_ID Types"
   section of the "GMPLS Signaling Parameters" registry located at
   http://www.iana.org/assignments/gmpls-sig-parameters.

      512 8 REFERENCE_COUNT   RFC 4783
      513 8 SEVERITY   RFC 4783
      514 8 GLOBAL_TIMESTAMP   RFC 4783
      515 8 LOCAL_TIMESTAMP   RFC 4783
      516 variable ERROR_STRING RFC 4783

5.3. New Registry for Admin-Status Object Bit Fields

   IANA created a new section titled "Administrative Status Information
   Flags" in the "GMPLS Signaling Parameters" registry located at
   http://www.iana.org/assignments/gmpls-sig-parameters and made the
   following assignments:

   Value   Name   Reference
   ----------- ------------------------------- -----------------
   0x80000000 Reflect (R)   [RFC3473/RFC3471]
   0x00000010 Inhibit Alarm Communication (I) RFC 4783
   0x00000004 Testing (T)   [RFC3473/RFC3471]
   0x00000002 Administratively down (A)   [RFC3473/RFC3471]
   0x00000001 Deletion in progress (D)   [RFC3473/RFC3471]

5.4. New RSVP Error Code

   IANA made the following assignments in the "Error Codes and Values"
   section of the "RSVP PARAMETERS" registry located at
   http://www.iana.org/assignments/rsvp-parameters.

   31 Alarms   RFC 4783

      The Error Value sub-codes for this Error Code have values and
      meanings identical to the values and meanings defined in the
      IANAItuProbableCause Textual Convention of IANA-ITU-ALARM-TC-MIB
      in the Alarm MIB [RFC3877]. Note that these values are already
      managed the IANA.

6. References

6.1. Normative References

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

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

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

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

   [RFC3877]   Chisholm, S. and D. Romascanu, "Alarm Management
                Information Base (MIB)", RFC 3877, September 2004.

   [M.3100]   ITU Recommendation M.3100, "Generic Network Information
                Model", 1995.

6.2. Informative References

   [RFC4201]   Kompella, K., Rekhter, Y., and L. Berger, "Link Bundling
                in MPLS Traffic Engineering (TE)", RFC 4201, October
                2005.

   [M.20]   ITU-T, "MAINTENANCE PHILOSOPHY FOR TELECOMMUNICATION
                NETWORKS", Recommendation M.20, October 1992.

   [GR833]   Bellcore, "Network Maintenance: Network Element and
                Transport Surveillance Messages" (GR-833-CORE), Issue 3,
                February 1999.

   [RFC4208]   Swallow, G., Drake, J., Ishimatsu, H., and Y. Rekhter,
                "Generalized Multiprotocol Label Switching (GMPLS) User-
                Network Interface (UNI): Resource ReserVation Protocol-
                Traffic Engineering (RSVP-TE) Support for the Overlay
                Model", RFC 4208, October 2005.

   [ASON-APPL] Papadimitriou, D., et al., "Generalized MPLS (GMPLS)
               RSVP-TE signaling usage in support of Automatically
               Switched Optical Network (ASON)", Work in Progress, July
               2005.

7. Acknowledgments

   Valuable comments and input were received from a number of people,
   including Wes Doonan, Bert Wijnen for the DISMAN reference, and Tom
   Petch for getting the DISMAN WG interactions started. We also thank
   David Black, Lars Eggert, Russ Housley, Dan Romascanu, and Magnus
   Westerlund for their valuable comments.

8. Contributors

   Contributors are listed in alphabetical order:

   Deborah Brungard
   AT&T Labs, Room MT D1-3C22
   200 Laurel Avenue
   Middletown, NJ 07748, USA
   Phone: (732) 420-1573
   EMail: dbrungard@att.com

   Igor Bryskin   Adrian Farrel
   Movaz Networks, Inc.   Old Dog Consulting
   7926 Jones Branch Drive
   Suite 615
   McLean VA, 22102, USA   Phone: +44 (0) 1978 860944
   EMail: ibryskin@movaz.com   EMail: adrian@olddog.co.uk

   Dimitri Papadimitriou (Alcatel)   Arun Satyanarayana
   Francis Wellesplein 1   Cisco Systems, Inc
   B-2018 Antwerpen, Belgium   170 West Tasman Dr.
                                         San Jose, CA 95134 USA
   Phone: +32 3 240-8491   Phone: +1 408 853-3206
   EMail: dimitri.papadimitriou@alcatel.be EMail: asatyana@cisco.com

Editor's Address

   Lou Berger
   LabN Consulting, L.L.C.

   Phone: +1 301-468-9228
   EMail: lberger@labn.net

Intellectual Property

   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.