INTERNET-DRAFT   Adrian Farrel (Editor)
Network Working Group   Old Dog Consulting
Obsoletes: RFC4420   Dimitri Papadimitriou
Updates: RFC3209, RFC3473   Alcatel
Category: Standards Track   Jean-Philippe Vasseur
Created: March 15, 2008   Cisco Systems, Inc.
Expires: September 15, 2008   Arthi Ayyangar
                                           Juniper Networks

    Encoding of Attributes for Multiprotocol Label Switching (MPLS)
         Label Switched Path (LSP) Establishment Using RSVP-TE

                    draft-ietf-ccamp-rfc4420bis-02.txt

Status of this Memo

Abstract

   Multiprotocol Label Switching (MPLS) Label Switched Paths (LSPs) may
   be established using the Resource Reservation Protocol Traffic
   Engineering (RSVP-TE) extensions. This protocol includes an object
   (the SESSION_ATTRIBUTE object) that carries a Flags field used to
   indicate options and attributes of the LSP. That Flags field has
   eight bits allowing for eight options to be set. Recent proposals in
   many documents that extend RSVP-TE have suggested uses for each of
   the previously unused bits.

   This document defines a new object for RSVP-TE messages that allows
   the signaling of further attribute bits and also the carriage of
   arbitrary attribute parameters to make RSVP-TE easily extensible to
   support new requirements. Additionally, this document defines a way
   to record the attributes applied to the LSP on a hop-by-hop basis.

   The object mechanisms defined in this document are equally applicable
   to Generalized MPLS (GMPLS) Packet Switch Capable (PSC) LSPs and to
   GMPLS non-PSC LSPs.

   This document replaces and obsoletes the previous version of this
   work published as [RFC 4420](RFC 4420). The only change is in the encoding of the
   Type-Length-Variable (TLV) data structures.

Contents

1. Introduction and Problem Statement

This document replaces and obsoletes the previous version of this
work published as RFC 4420. The only change is in the encoding of the
Type-Length-Variable (TLV) data structures presented in Section 3.
See Section 14 for a summary of changes.

Traffic-Engineered Multiprotocol Label Switching (MPLS) Label
Switched Paths (LSPs) [RFC3031] may be set up using the Path message
of the RSVP-TE signaling protocol [RFC3209]. The Path message
includes the SESSION_ATTRIBUTE object, which carries a Flags field
used to indicate desired options and attributes of the LSP.

The Flags field in the SESSION_ATTRIBUTE object has eight bits. Just
three of those bits are assigned in [RFC3209]. A further two bits
are assigned in [RFC4090] for fast re-reroute functionality leaving
only three bits available. Several recent proposals and Internet
Drafts have demonstrated that there is a high demand for the use of
the other three bits. Some, if not all, of those proposals are
likely to go forward as RFCs resulting in depletion or near depletion
of the Flags field and a consequent difficulty in signaling new
options and attributes that may be developed in the future.

This document defines a new object for RSVP-TE messages that allows
the signaling of further attributes bits. The new object is
constructed from TLVs, and a new TLV is defined to carry a variable
number of attributes bits.

The new RSVP-TE message object is quite flexible, due to the use of
the TLV format and allows:

- future specification of bit flags
- additional options and attribute parameters carried in TLV
  format.

Note that the LSP Attributes defined in this document are
specifically scoped to an LSP. They may be set differently on
separate LSPs with the same Tunnel ID between the same source and
destination (that is, within the same session).

It is noted that some options and attributes do not need to be acted
on by all Label Switched Routers (LSRs) along the path of the LSP.
In particular, these options and attributes may apply only to key
LSRs on the path such as the ingress LSR and egress LSR. Special
transit LSRs, such as Area or Autonomous System Border Routers (ABRs
or ASBRs), may also fall into this category. This means that the new
options and attributes should be signaled transparently, and only
examined at those points that need to act on them.

On the other hand, other options and attributes may require action at

all transit LSRs along the path of the LSP. Inability to support the
required attributes by one of those transit LSRs may require the LSR
to refuse the establishment of the LSP.

These considerations are particularly important in the context of
backward compatibility. In general, it should be possible to provide
new MPLS services across a legacy network without upgrading those
LSRs that do not need to participate actively in the new services.
Moreover, some features just require action on specific intermediate
hops, and not on every visited LSR.

Note that options already specified for the SESSION_ATTRIBUTE object
in preexisting RFCs are not migrated to the new mechanisms described
in this document.

RSVP includes a way for unrecognized objects to be transparently
forwarded by transit nodes without them refusing the incoming
protocol messages and without the objects being stripped from the
outgoing protocol message (see [RFC2205], Section 3.10). This
capability extends to RSVP-TE and provides a good way to ensure that
only those LSRs that understand a particular object examine it.

This document distinguishes between options and attributes that are
only required at key LSRs along the path of the LSP, and those that
must be acted on by every LSR along the LSP. Two LSP Attributes
objects are defined in this document: using the C-Num definition
rules inherited from [RFC2205], the first is passed transparently by
LSRs that do not recognize it, and the second causes LSP setup
failure with the generation of a PathErr message with an appropriate
Error Code if an LSR does not recognize it.

1.1. Applicability to Generalized MPLS

The RSVP-TE signaling protocol also forms the basis of a signaling
protocol for Generalized MPLS (GMPLS) as described in [RFC3471] and
[RFC3473]. The extensions described in this document are equally
applicable to MPLS and GMPLS.

1.2. A Rejected Alternate Solution

A rejected alternate solution was to define a new C-Type for the
existing SESSION_ATTRIBUTE object. This new C-Type could allow a
larger Flags field and address the immediate problem.

This solution was rejected because:

- A new C-Type is not backward compatible with deployed
  implementations that expect to see a C-Type of 1 or 7. It is
  important that any solution be capable of carrying new attributes
  transparently across legacy LSRs if those LSRs are not required to

   act on the attributes.

 - Support for arbitrary attributes parameters through TLVs would have
   meant a significant change of substance to the existing object.

2. Terminology

   This document uses terminology from the MPLS architecture document
   [RFC3031] and from the RSVP-TE protocol specification [RFC3209],
   which inherits from the RSVP specification [RFC2205]. It also makes
   use of the Generalized MPLS RSVP-TE terminology introduced in
   [RFC3471] and [RFC3473].

   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].

3. Attributes TLVs

   Attributes carried by the new objects defined in this document are
   encoded within TLVs. One or more TLVs may be present in each object.
   There are no ordering rules for TLVs, and no interpretation should be
   placed on the order in which TLVs are received.

   Each TLV is encoded as follows.

```
    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            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   //                             Value                           //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

     Type

         The identifier of the TLV.

     Length

         Indicates the total length of the TLV, i.e., 4 + the length of
         the value field in octets. A value field whose length is not a
         multiple of four MUST be zero-padded so that the TLV is four-
         octet aligned.

     Value

         The data for the TLV padded as described above.

3.1. Attributes Flags TLV

This document defines only one TLV type value. Type 1 indicates the
Attributes Flags TLV. Other TLV types may be defined in the future
with type values assigned by IANA (see Section 11.2).

The Attributes Flags TLV may be present in an LSP_ATTRIBUTES object
and/or an LSP_REQUIRED_ATTRIBUTES object defined in Sections 4 and 5.
The bits in the TLV represent the same attributes regardless of which
object carries the TLV. Documents that define individual bits MUST
specify whether the bit may be set in one object or the other, or
both. It is not expected that a bit will be set in both objects on a
single Path message at the same time, but this is not ruled out by
this document.

The Attribute Flags TLV Value field is an array of units of 32 flags
numbered from the most significant bit as bit zero. The Length field
for this TLV is therefore always a multiple of 4 bytes, regardless of
the number of bits carried and no padding is required.

Unassigned bits are considered as reserved and MUST be set to zero on
transmission by the originator of the object. Bits not contained in
the TLV MUST be assumed to be set to zero. If the TLV is absent
either because it is not contained in the LSP_ATTRIBUTES or
LSP_REQUIRED_ATTRIBUTES object, or because those objects are
themselves absent, all processing MUST be performed as though the
bits were present and set to zero. That is to say, assigned bits
that are not present either because the TLV is deliberately
foreshortened or because the TLV is not included MUST be treated as
though they are present and are set to zero.

No bits are defined in this document. The assignment of bits is
managed by IANA (see Section 11.3).

4. LSP_ATTRIBUTES Object

The LSP_ATTRIBUTES object is used to signal attributes required in
support of an LSP, or to indicate the nature or use of an LSP where
that information is not required to be acted on by all transit LSRs.
Specifically, if an LSR does not support the object, it forwards it
unexamined and unchanged. This facilitates the exchange of
attributes across legacy networks that do not support this new
object.

This object effectively extends the Flags field in the
SESSION_ATTRIBUTE object and allows for the future inclusion of more
complex objects through TLVs.

Note that some function may require an LSR to inspect both the
SESSION_ATTRIBUTE object and the LSP_ATTRIBUTES or
LSP_REQUIRED_ATTRIBUTES object.

The LSP_ATTRIBUTES object may also be used to report LSP operational
state on a Resv message even when no LSP_ATTRIBUTES or
LSP_REQUIRED_ATTRIBUTES object was carried on the corresponding Path
message. The object is added or updated by LSRs that support the
object. LSRs that do not understand the object or have nothing to
report do not add the object and forward it unchanged on Resv
messages that they generate.

The LSP_ATTRIBUTES object class is 197 of the form 11bbbbbb. This
C-Num value (see [RFC2205], Section 3.10) ensures that LSRs that do
not recognize the object pass it on transparently.

One C-Type is defined, C-Type = 1 for LSP Attributes.

This object is optional and may be placed on Path messages to convey
additional information about the desired attributes of the LSP, and
on Resv messages to report operational state.

4.1. Format

LSP_ATTRIBUTES class = 197, 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                       Attributes TLVs                       //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

The Attributes TLVs are encoded as described in Section 3.

4.2. Generic Processing Rules for Path Messages

An LSR that does not support this object is required to pass it on
unaltered as indicated by the C-Num and the rules defined in
[RFC2205].

An LSR that does support this object, but does not recognize a TLV
type code carried in this object, MUST pass the TLV on unaltered in
the LSP_ATTRIBUTES object that it places in the Path message that it
sends downstream.

An LSR that does support this object and recognizes a TLV, but does
not support the attribute defined by the TLV, MUST act as specified
in the document that defines the TLV.

An LSR that supports the Attributes Flags TLV, but does not recognize
a bit set in the Attributes Flags TLV, MUST forward the TLV
unchanged.

An LSR that supports the Attributes Flags TLV and recognizes a bit
that is set, but does not support the indicated attribute, MUST act
as specified in the document that defines the bit.

4.3. Generic Processing Rules for Resv Messages

An LSR that wishes to report operational status of an LSP may include
this object in a Resv message, or update the object that is already
carried in a Resv message.

Note that this usage reports the state of the entire LSP and not the
state of the LSP at an individual LSR. This latter function is
achieved using the LSP Attributes subobject of the Record Route
object (RRO) as described in Section 7.

The bits in the Attributes TLV may be used to report operational
status for the whole LSP. For example, an egress LSR may report a
particular status by setting a bit. LSRs within the network that
determine that this status has not been achieved may clear the bit as
they forward the Resv message.

Observe that LSRs that do not support the object or do not support
the function characterized by a particular bit in the Attributes TLV
will not clear the bit when forwarding the Resv. Thus, care must be
taken in defining the usage of this object on a Resv. The usage of
an individual bit in the Attributes TLV of the LSP_ATTRIBUTES object
on a Resv must be fully defined in the document that defines the bit.

Additional TLVs may also be defined to be carried in this object on a
Resv.

An LSR that does not support this object will pass it on unaltered
because of the C-Num.

5. **LSP_REQUIRED_ATTRIBUTES Object**

   The LSP_REQUIRED_ATTRIBUTES object is used to signal attributes
   required in support of an LSP, or to indicate the nature or use of an
   LSP where that information MUST be inspected at each transit LSR.
   Specifically, each transit LSR MUST examine the attributes in the
   LSP_REQUIRED_ATTRIBUTES object and MUST NOT forward the object
   without acting on its contents.

   This object effectively extends the Flags field in the
   SESSION_ATTRIBUTE object and allows for the future inclusion of more
   complex objects through TLVs. It complements the LSP_ATTRIBUTES
   object.

   The LSP_REQUIRED_ATTRIBUTES object class is 67 of the form 0bbbbbbb.
   This C-Num value ensures that LSRs that do not recognize the object
   reject the LSP setup effectively saying that they do not support the
   attributes requested. This means that this object SHOULD only be
   used for attributes that require support at some transit LSRs and so
   require examination at all transit LSRs. See Section 4 for how end-
   to-end and selective attributes are signaled.

   One C-Type is defined, C-Type = 1 for LSP Required Attributes.

   This object is optional and may be placed on Path messages to convey
   additional information about the desired attributes of the LSP.

5.1. **Format**

   LSP_REQUIRED_ATTRIBUTES class = 67, 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                      Attributes TLVs                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

   The Attributes TLVs are encoded as described in Section 3.

5.2. **Generic Processing Rules**

   An LSR that does not support this object will use a PathErr to reject
   the Path message based on the C-Num using the Error Code "Unknown
   Object Class".

   An LSR that does not recognize a TLV type code carried in this object
   MUST reject the Path message using a PathErr with Error Code "Unknown
   Attributes TLV" and Error Value set to the value of the unknown TLV
   type code.

   An LSR that does not recognize a bit set in the Attributes Flags TLV
   MUST reject the Path message using a PathErr with Error Code "Unknown
   Attributes Bit" and Error Value set to the bit number of the unknown
   bit in the Attributes Flags.

   An LSR that recognizes an attribute (however encoded), but that does
   not support that attribute, MUST act according to the behavior
   specified in the document that defines that specific attribute.

   Note that this object is not used on a Resv. In order to report the
   status of an LSP, either the LSP_ATTRIBUTES object on a Resv or the
   Attributes subobject in the Record Route object (see Section 7) must
   be used.

6. Inheritance Rules

   In certain circumstances, when reaching an LSP region boundary, a
   forwarding adjacency LSP (FA-LSP; see [RFC4206]) is initially set up
   to allow the establishment of the LSP carrying the LSP_ATTRIBUTES
   and/or LSP_REQUIRED_ATTRIBUTES objects. In this case, when the
   boundary LSR supports LSP_ATTRIBUTES and LSP_REQUIRED_ATTRIBUTES
   processing, the FA-LSP MAY upon local policy inherit a subset of the
   Attributes TLVs, in particular when the FA-LSP belongs to the same
   switching capability class as the triggering LSP.

   When these conditions are met, the LSP_ATTRIBUTES and/or
   LSP_REQUIRED_ATTRIBUTES objects are simply copied with the inherited
   Attributes TLVs in the Path message used to establish the FA-LSP. By
   default (and in order to simplify deployment), none of the incoming
   LSP Attributes TLVs are considered as inheritable. Note that when
   the FA-LSP establishment itself requires one or more Attributes TLVs,
   an 'OR' operation is performed with the inherited set of values.

   Documents that define individual bits for the LSP Attributes Flags
   TLV MUST specify whether or not these bits MAY be inherited
   (including the condition to be met in order for this inheritance to
   occur). The same applies for any other TLV that will be defined
   following the rules specified in Section 3.

7. **Recording Attributes Per LSP**

7.1. **Requirements**

   In some circumstances, it is useful to determine which of the
   requested LSP attributes have been applied at which LSRs along the
   path of the LSP. For example, an attribute may be requested in the
   LSP_ATTRIBUTES object such that LSRs that do not support the object
   are not required to support the attribute or provide the requested
   function. In this case, it may be useful to the ingress LSR to know
   which LSRs acted on the request and which ignored it.

   Additionally, there may be other qualities that need to be reported
   on a hop-by-hop basis. These are currently indicated in the Flags
   field of RRO subobjects. Since there are only eight bits available
   in this field, and since some are already assigned and there is also
   likely to be an increase in allocations in new documents, there is a
   need for some other method to report per-hop attributes.

7.2. **RRO Attributes Subobject**

   The RRO Attributes Subobject may be carried in the RECORD_ROUTE
   object if it is present. The subobject uses the standard format of
   an RRO subobject.

   The length is variable as for the Attributes Flags TLV. The content
   is the same as the Attribute Flags TLV -- that is, it is a series of
   bit flags.

   There is a one-to-one correspondence between bits in the Attributes
   Flags TLV and the RRO Attributes Subobject. If a bit is only
   required in one of the two places, it is reserved in the other place.
   See the procedures sections, below, for more information.

```
 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           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
//                      Attribute Flags                        //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

      Type

         0x05

Length

   The Length contains the total length of the subobject in bytes,
   including the Type and Length fields. This length must be a
   multiple of 4 and must be at least 8.

Attribute Flags

   The attribute flags recorded for the specific hop.

7.3. Procedures

7.3.1. Subobject Presence Rules

As will be clear from [RFC3209], the RECORD_ROUTE object is managed
as a "stack" with each LSR adding subobjects to the start of the
object. The Attributes subobject is pushed onto the RECORD_ROUTE
object immediately prior to pushing the node's IP address or link
identifier. Thus, if label recording is being used, the Attributes
subobject SHOULD be pushed onto the RECORD_ROUTE object after the
Record Label subobject(s).

A node MUST NOT push an Attributes subobject onto the RECORD_ROUTE
object without also pushing an IPv4, IPv6, or Unnumbered Interface ID
subobject.

This means that an Attributes subobject is bound to the LSR
identified by the subobject found in the RRO immediately before the
Attributes subobject.

If the new subobject causes the RRO to be too big to fit in a Path
(or Resv) message, the processing MUST be as described in Section
4.4.3 of [RFC3209].

If more than one Attributes subobject is found between a pair of
subobjects that identify LSRs, only the first one found (that is, the
nearest to the top of the stack) SHALL have any meaning within the
context of this document. All such subobjects MUST be forwarded
unmodified by transit LSRs

7.3.2. Reporting Compliance with LSP Attributes

To report compliance with an attribute requested in the Attributes
Flags TLV, an LSR MAY set the corresponding bit (see Section 8) in
the Attributes subobject. To report non-compliance, an LSR MAY clear
the corresponding bit in the Attributes subobject.

The requirement to report compliance MUST be specified in the
document that defines the usage of any bit. This will reduce to a
statement of whether hop-by-hop acknowledgement is required.

7.3.3. Reporting Per-Hop Attributes

To report a per-hop attribute, an LSR sets the appropriate bit in the
Attributes subobject.

The requirement to report a per-hop attribute MUST be specified in
the document that defines the usage of the bit.

7.3.4. Default Behavior

By default, all bits in an Attributes subobject SHOULD be set to
zero.

If a received Attribute subobject is not long enough to include a
specific numbered bit, that bit MUST be treated as though present and
as if set to zero.

If the RRO subobject is not present for a hop in the LSP, all bits
MUST be assumed to be set to zero.

8. Summary of Attribute Bit Allocation

This document defines two uses of per-LSP attribute flag bit fields.
The bit numbering in the Attributes Flags TLV and the RRO Attributes
subobject is identical. That is, the same attribute is indicated by
the same bit in both places. This means that only a single registry
of bits is maintained.

The consequence is a degree of clarity in implementation and
registration.

Note, however, that it is not always the case that a bit will be used
in both the Attributes Flags TLV and the RRO Attributes subobject.
For example, an attribute may be requested using the Attributes Flags
TLV, but there is no requirement to report the handling of the
attribute on a hop-by-hop basis. Conversely, there may be a
requirement to report the attributes of an LSP on a hop-by-hop basis,
but there is no corresponding request attribute.

In these cases, a single bit number is still assigned for both the
Attributes Flags TLV and the RRO Attributes subobject even though the
bit may be irrelevant in either the Attributes Flags or the RRO

Attributes subobject. The document that defines the usage of the new
bit MUST state in which places it is used and MUST handle a default
setting of zero.

9. Message Formats

The LSP_ATTRIBUTES object and the LSP_REQUIRED_ATTRIBUTES object MAY
be carried in a Path message. The LSP_ATTRIBUTES object MAY be
carried in a Resv message.

The order of objects in RSVP-TE messages is recommended, but
implementations must be capable of receiving the objects in any
meaningful order.

On a Path message, the LSP_ATTRIBUTES object and
LSP_REQUIRED_ATTRIBUTES objects are RECOMMENDED to be placed
immediately after the SESSION_ATTRIBUTE object if it is present, or
otherwise immediately after the LABEL_REQUEST object.

If both the LSP_ATTRIBUTES object and the LSP_REQUIRED_ATTRIBUTES
object are present, the LSP_REQUIRED_ATTRIBUTES object is RECOMMENDED
to be placed first.

LSRs MUST be prepared to receive these objects in any order in any
position within a Path message. Subsequent instances of these
objects within a Path message SHOULD be ignored and those objects
MUST be forwarded unchanged.

On a Resv message, the LSP_ATTRIBUTES object is placed in the flow
descriptor and is associated with the FILTER_SPEC object that
precedes it. It is RECOMMENDED that the LSP_ATTRIBUTES object be
placed immediately after the LABEL object.

LSRs MUST be prepared to receive this object in any order in any
position within a Resv message subject to the previous note. Only
one instance of the LSP_ATTRIBUTES object is meaningful within the
context of a FILTER_SPEC object. Subsequent instances of the object
SHOULD be ignored and MUST be forwarded unchanged.

10. Guidance for Key Application Scenarios

As described in the Introduction section of this document, it may be
that requested LSP attributes need to be acted on by only the egress
LSR of the LSP, by certain key transit points (such as ABRs and
ASBRs), or by all LSRs along the LSP. This section briefly describes
how each of these scenarios is met. This section is informational
and does not define any new procedures.

10.1. Communicating to Egress LSRs

   When communicating LSP attributes that must be acted on only by the
   LSP egress LSR, the attributes should be communicated in the
   LSP_ATTRIBUTES object. Because of its C-Num, this object may be
   ignored (passed onwards, untouched) by transit LSRs that do not
   understand it. This means that the Path message will not be rejected
   by LSRs that do not understand the object. In this way, the
   requested LSP attributes are guaranteed to reach the egress LSR.

   Attributes are set within the LSP_ATTRIBUTES object according to
   which LSP attributes are required. Each attribute is defined in some
   RFC and is accompanied by a statement of what the expected behavior
   is. This behavior will include whether the attribute must be acted
   on by any LSR that recognizes it, or specifically by the egress LSR.
   Thus, any attribute that must be acted on only by an egress LSR will
   be defined in this way -- any transit LSR seeing this attribute
   either will understand the semantics of the attribute and ignore it
   (forwarding it, unchanged) or will not understand the attribute and
   ignore it (forwarding it, unchanged) according to the rules of the
   LSP_ATTRIBUTES object.

   The remaining issue is how the ingress LSR can know whether the
   egress LSR has acted correctly on the required LSP attribute.
   Another part of the definition of the attribute (in the defining RFC)
   is whether reporting is required. If reporting is required, the
   egress LSR is required to use the RRO Attributes subobject to report
   whether it has acted on the received attribute.

   If an egress LSR understands a received attribute as mandatory for an
   egress LSR, but does not wish to satisfy the request, it will reject
   the Path message. If an egress LSR understands the attribute, but
   believes it to be optional and does not wish to satisfy the request,
   it will report its non-compliance in the RRO Attributes subobject.
   If the egress LSR does not understand the received attribute, it may
   report non-compliance in the RRO Attributes subobject explicitly, or
   may omit the RRO Attributes subobject implying that it has not
   satisfied the request.

10.2. Communicating to Key Transit LSRs

   Processing for key transit LSRs (such as ABRs and ASBRs) follows
   exactly as for egress LSR. The only difference is that the
   definition of the LSP attribute in the defining RFC will state that
   the attribute must be acted on by these transit LSRs.

10.3. Communicating to All LSRs

In order to force all LSRs to examine the LSP attributes, the
LSP_REQUIRED_ATTRIBUTES object is used. The C-Num of this object is
such that any LSR that does not recognize the object must reject a
received Path message containing the object.

An LSR that recognizes the LSP_REQUIRED_ATTRIBUTES object, but does
not recognize an attribute, will reject the Path message.

An LSR that recognizes an attribute, but does not wish to support the
attribute, reacts according to the definition of the attribute in the
defining RFC. This may allow the LSR to ignore the attribute and
forward it unchanged, or may require it to fail the LSP setup. The
LSR may additionally be required to report whether it supports the
attribute using the RRO Attributes subobject.

11. IANA Considerations

The IANA allocations made for RFC 4420 now apply to this document
and are listed here for completeness.

This document makes no further requests for IANA action, but IANA
should update the registry entries created for RFC 4420 to reference
this document which is now the normative reference for those entries.

11.1. New RSVP C-Nums and C-Types

Two new RSVP C-Nums are defined in this document and have been
assigned by IANA.

o LSP_ATTRIBUTES object

  The C-Num (value 197) is of the form 11bbbbbb so that LSRs that do
  not recognize the object will ignore the object but forward it,
  unexamined and unmodified, in all messages resulting from this
  message.

  One C-Type is defined for this object and has been assigned by
  IANA.

  o LSP Attributes TLVs

    Recommended C-Type value 1.

o LSP_REQUIRED_ATTRIBUTES object

  The C-Num (value 67) is of the form 0bbbbbbb so that LSRs that do
  not recognize the object will reject the message that carries it

      with an "Unknown Object Class" error.

      One C-Type is defined for this object and has been assigned by
      IANA.

      o LSP Required Attributes TLVs

         Recommended C-Type value 1.

11.2. New TLV Space

   The two new objects referenced above are constructed from TLVs. Each
   TLV includes a 16-bit type identifier (the T-field). The same
   T-field values are applicable to both objects.

   The IANA has created a new registry and will manage TLV type
   identifiers as follows:

   - TLV Type (T-field value)
   - TLV Name
   - Whether allowed on LSP_ATTRIBUTES object
   - Whether allowed on LSP_REQUIRED_ATTRIBUTES object.

   This document defines one TLV type as follows:

   - TLV Type = 1
   - TLV Name = Attributes Flags TLV
   - allowed on LSP_ATTRIBUTES object
   - allowed on LSP_REQUIRED_ATTRIBUTES object.

   New TLV type values may be allocated only by an IETF Consensus
   action.

11.3. Attributes Flags

   This document provides new attributes bit flags for use in other
   documents that specify new RSVP-TE attributes. These flags are
   present in the Attributes Flags TLV referenced in the previous
   section.

   The IANA has created a new registry and will manage the space of
   attributes bit flags numbering them in the usual IETF notation
   starting at zero and continuing at least through 31.

   New bit numbers may be allocated only by an IETF Consensus action.

   Each bit should be tracked with the following qualities:

   - Bit number
   - Defining RFC

   - Name of bit
   - Whether there is meaning in the Attribute Flags TLV on a Path
   - Whether there is meaning in the Attribute Flags TLV on a Resv
   - Whether there is meaning in the RRO Attributes Subobject.

   Note that this means that all bits in the Attribute Flags TLV and the
   RRO Attributes Subobject use the same bit number regardless of
   whether they are used in one or both places. Thus, only one list of
   bits is required to be maintained. (It would be meaningless in the
   context of this document for a bit to have no meaning in either the
   Attribute Flags TLV or the RRO Attributes Subobject.)

11.4. New Error Codes

   This document defines the following new Error Codes and Error Values.
   Numeric values have been assigned by IANA.

   Error Code   Error Value
   29 "Unknown Attributes TLV"   Identifies the unknown TLV type code.
   30 "Unknown Attributes Bit"   Identifies the unknown Attribute Bit.

11.5. New Record Route Subobject Identifier

   A new subobject is defined for inclusion in the RECORD_ROUTE object.

   The RRO Attributes subobject is identified by a Type value of 5.

12. Security Considerations

   This document adds two new objects to the RSVP Path message as used
   in MPLS and GMPLS signaling, and a new subobject to the RECORD_ROUTE
   object carried on many RSVP messages. It does not introduce any new
   direct security issues, and the reader is referred to the security
   considerations expressed in [RFC2205], [RFC3209], and [RFC3473].

   It is of passing note that any signaling request that indicates the
   functional preferences or attributes of an MPLS LSP may provide
   anyone with unauthorized access to the contents of the message with
   information about the LSP that an administrator may wish to keep
   secret. Although this document adds new objects for signaling
   desired LSP attributes, it does not contribute to this issue, which
   can only be satisfactorily handled by encrypting the content of the
   signaling message.

   Similarly, the addition of attribute recording information to the RRO
   may reveal information about the status of the LSP and the
   capabilities of individual LSRs that operators wish to keep secret.

The same strategy that applies to other RRO subobjects also applies
here. Note, however, that there is a tension between notifying the
head end of the LSP status at transit LSRs, and hiding the existence
or identity of the transit LSRs.

13. Acknowledgements

Credit to the OSPF Working Group for inspiration from their solution
to a similar problem. Thanks to Rahul Aggarwal for his careful
review and support of this work. Thanks also to Raymond Zhang,
Kireeti Kompella, Philip Matthews, Jim Gibson, and Alan Kullberg for
their input. As so often, thanks to John Drake for useful offline
discussions. Thanks to Mike Shand for providing the Routing
Directorate review and to Joel Halpern for the General Area review --
both picked up on some unclarities.

Thanks to the OIF for noticing the discrepency in RFC 4420 that is
fixed in this document. Alfred Hoenes noted several typographical
errors.

14. Changes from RFC 4420 to RFC XXXX

[RFC Editor - Please replace "XXXX" with the RFC number assigned for
this work and remove this note.]

This document obsoletes RFC 4420. The only change is in Section 3.
Section 3 describes the semantic of the Length field of the
Attributes TLV.

Prior to the change, the Length field indicated the length of the
Value field only. After the change, as described in Section 3, the
Length field indicates the length of the whole TLV. This change means
that this document is consistent with the subobject format defined in
[RFC3209] and the TLV format defined in [RFC3471].

15. 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.

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

   [RFC3471]   Berger, L. (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.

16. Informative References

   [RFC3031]   Rosen, E., Viswanathan, A., and R. Callon, "Multiprotocol
               Label Switching Architecture", RFC 3031, January 2001.

   [RFC4090]   Pan, P., Swallow, G., and A. Atlas, "Fast Reroute
               Extensions to RSVP-TE for LSP Tunnels", RFC 4090, May
               2005.

   [RFC4206]   Kompella, K. and Y. Rekhter, "Label Switched Paths (LSP)
               Hierarchy with Generalized Multi-Protocol Label Switching
               (GMPLS) Traffic Engineering (TE)", RFC 4206, October
               2005.

17. Authors' Addresses

   Adrian Farrel
   Old Dog Consulting

   Phone: +44 (0) 1978 860944
   EMail: adrian@olddog.co.uk

   Dimitri Papadimitriou
   Alcatel
   Fr. Wellesplein 1,
   B-2018 Antwerpen, Belgium

   Phone: +32 3 240-8491
   EMail: dimitri.papadimitriou@alcatel.be

   Jean Philippe Vasseur
   Cisco Systems, Inc.
   1414 Massachusetts Avenue
   Boxborough, MA - 01719
   USA

   EMail: jpv@cisco.com

   Arthi Ayyangar
   Juniper Networks, Inc.
   1194 N.Mathilda Ave
   Sunnyvale, CA 94089
   USA

   EMail: arthi@juniper.net

18. 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.

19. Full Copyright Statement