Network Working Group
Request for Comments: 4874
Updates: 3209, 3473
Category: Standards Track

CY. Lee
A. Farrel
Old Dog Consulting
S. De Cnodder
Alcatel-Lucent

April 2007

Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)

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.

Copyright Notice

Abstract

This document specifies ways to communicate route exclusions during path setup using Resource ReserVation Protocol-Traffic Engineering (RSVP-TE).

The RSVP-TE specification, "RSVP-TE: Extensions to RSVP for LSP Tunnels" (RFC 3209) and GMPLS extensions to RSVP-TE, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions" (RFC 3473) allow abstract nodes and resources to be explicitly included in a path setup, but not to be explicitly excluded.

In some networks where precise explicit paths are not computed at the head end, it may be useful to specify and signal abstract nodes and resources that are to be explicitly excluded from routes. These exclusions may apply to the whole path, or to parts of a path between two abstract nodes specified in an explicit path. How Shared Risk Link Groups (SRLGs) can be excluded is also specified in this document.

Table of Contents

1. Introduction

The RSVP-TE specification [RFC3209] and GMPLS extensions [RFC3473] allow abstract nodes and resources to be explicitly included in a path setup, using the Explicit Route Object (ERO).

In some systems, it may be useful to specify and signal abstract nodes and resources that are to be explicitly excluded from routes. This may be because loose hops or abstract nodes need to be prevented from selecting a route through a specific resource. This is a special case of distributed path calculation in the network.

For example, route exclusion could be used in the case where two non-overlapping Label Switched Paths (LSPs) are required. In this case, one option might be to set up one path and collect its route using route recording, and then to exclude the routers on that first path from the setup for the second path. Another option might be to set up two parallel backbones, dual home the provider edge (PE) routers to both backbones, and then exclude the local router on backbone A the first time that you set up an LSP (to a particular distant PE), and exclude the local router on backbone B the second time that you set up an LSP.

Two types of exclusions are required:

1. Exclusion of certain abstract nodes or resources on the whole path. This set of abstract nodes is referred to as the Exclude Route list.

2. Exclusion of certain abstract nodes or resources between a specific pair of abstract nodes present in an ERO. Such specific exclusions are referred to as Explicit Exclusion Route.

To convey these constructs within the signaling protocol, a new RSVP object and a new ERO subobject are introduced respectively.

- A new RSVP-TE object is introduced to convey the Exclude Route list. This object is the EXCLUDE_ROUTE object (XRO).

- The second type of exclusion is achieved through a modification to the existing ERO. A new ERO subobject type the Explicit Exclusion Route Subobject (EXRS) is introduced to indicate an exclusion between a pair of included abstract nodes.

The knowledge of SRLGs, as defined in [RFC4216], may be used to compute diverse paths that can be used for protection. In systems where it is useful to signal exclusions, it may be useful to signal SRLGs to indicate groups of resources that should be excluded on the whole path or between two abstract nodes specified in an explicit path.

This document introduces a subobject to indicate an SRLG to be signaled in either of the two exclusion methods described above. This document does not assume or preclude any other usage for this subobject. This subobject might also be appropriate for use within an Explicit Route object (ERO) or Record Route object (RRO), but this is outside the scope of this document.

1.1. Requirements Notation

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].

1.2. Scope of Exclude Routes

This document does not preclude a route exclusion from listing arbitrary nodes or network elements to avoid. The intent is, however, to indicate only the minimal number of subobjects to be explicitly avoided. For instance, it may be necessary to signal only the SRLGs (or Shared Risk Link Groups) to avoid. That is, the route exclusion is not intended to define the actual route by listing all of the choices to exclude at each hop, but rather to constrain the normal route selection process where loose hops or abstract nodes are to be expanded by listing certain elements to be avoided.

It is envisaged that most of the conventional inclusion subobjects are specified in the signaled ERO only for the area where they are pertinent. The number of subobjects to be avoided, specified in the signaled XRO, may be constant throughout the whole path setup, or the subobjects to be avoided may be removed from the XRO as they become irrelevant in the subsequent hops of the path setup.

For example, consider an LSP that traverses multiple computation domains. A computation domain may be an area in the administrative or IGP sense, or may be an arbitrary division of the network for active management and path computational purposes. Let the primary path be (Ingress, A1, A2, AB1, B1, B2, BC1, C1, C2, Egress) where:

- Xn denotes a node in domain X, and

- XYn denotes a node on the border of domain X and domain Y.

Note that Ingress is a node in domain A, and Egress is a node in domain C. This is shown in Figure 1 where the domains correspond with areas.

```
           area A               area B              area C
    <-------------------> <----------------> <------------------>

   Ingress-----A1----A2----AB1----B1----B2----BC1----C1----C2----Egress
   ^  \                / | \              / | \                /
   |   \              /  |  \            /  |  \              /
   |    A3----------A4--AB2--B3--------B4--BC2--C3----------C4
   |                     ^                  ^
   |                     |                  |
   |                     |                  |
   |                     |              ERO: (C3-strict, C4-strict,
   |                     |                    Egress-strict)
   |                     |              XRO: Not needed
   |                     |
   |               ERO: (B3-strict, B4-strict, BC2-strict, Egress-loose)
   |               XRO: (BC1, C1, C2)
   |
   ERO: (A3-strict, A4-strict, AB2-strict, Egress-loose)
   XRO: (AB1, B1, B2, BC1, C1, C2, Egress)
```

Figure 1: Domains Corresponding to IGP Areas

Consider the establishment of a node-diverse protection path in the example above. The protection path must avoid all nodes on the primary path. The exclusions for area A are handled during Constrained Shortest Path First (CSPF) computation at Ingress, so the ERO and XRO signaled at Ingress could be (A3-strict, A4-strict, AB2-strict, Egress-loose) and (AB1, B1, B2, BC1, C1, C2), respectively. At AB2, the ERO and XRO could be (B3-strict, B4-strict, BC2-strict, Egress-loose) and (BC1, C1, C2), respectively. At BC2, the ERO could be (C3-strict, C4-strict, Egress-strict) and an XRO is not needed from BC2 onwards.

In general, consideration SHOULD be given (as with explicit route) to the size of signaled data and the impact on the signaling protocol.

1.3. Relationship to MPLS TE MIB

[RFC3812] defines managed objects for managing and modeling MPLS-based traffic engineering. Included in [RFC3812] is a means to configure explicit routes for use on specific LSPs. This configuration allows the exclusion of certain resources.

In systems where the full explicit path is not computed at the ingress (or at a path computation site for use at the ingress), it may be necessary to signal those exclusions. This document offers a means of doing this signaling.

2. Shared Risk Link Groups

The identifier of an SRLG is defined as a 32-bit quantity in [RFC4202]. An SRLG subobject is introduced such that it can be used in the exclusion methods as described in the following sections. This document does not assume or preclude any other usage for this subobject. This subobject might also be appropriate for use within Explicit Route object (ERO) or Record Route object (RRO), but this is outside the scope of this document.

2.1. SRLG Subobject

The new SRLG subobject is defined by this document as follows. Its format is modeled on the ERO subobjects defined in [RFC3209].

```
 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
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|    Type     |     Length    |       SRLG Id (4 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SRLG Id (continued)      |           Reserved            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L

The L bit is an attribute of the subobject. The L bit is set if the subobject represents a loose hop in the explicit route. If the bit is not set, the subobject represents a strict hop in the explicit route.

For exclusions (as used by XRO and EXRS defined in this document), the L bit SHOULD be set to zero and ignored.

Type

The type of the subobject (34)

Length

The Length contains the total length of the subobject in bytes, including the Type and Length fields. The Length is always 8.

SRLG Id

The 32-bit identifier of the SRLG.

Reserved

This field is reserved. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

3. Exclude Route List

The exclude route identifies a list of abstract nodes that should not be traversed along the path of the LSP being established. It is RECOMMENDED that the size of the exclude route list be limited to a value local to the node originating the exclude route list.

3.1. EXCLUDE_ROUTE Object (XRO)

Abstract nodes to be excluded from the path are specified via the EXCLUDE_ROUTE object (XRO).

Currently, one C_Type is defined, Type 1 EXCLUDE_ROUTE. The EXCLUDE_ROUTE object has the following format:

```
      Class = 232, 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 an EXCLUDE_ROUTE object are a series of variable-length data items called subobjects. This specification adapts ERO subobjects as defined in [RFC3209], [RFC3473], and [RFC3477] for use in route exclusions. The SRLG subobject as defined in Section 2 of this document has not been defined before. The SRLG subobject is defined here for use with route exclusions.

The following subobject types are supported.

Type	Subobject
1	IPv4 prefix
2	IPv6 prefix
4	Unnumbered Interface ID
32	Autonomous system number
34	SRLG

The defined values for Type above are specified in [RFC3209] and in this document.

The concept of loose or strict hops has no meaning in route exclusion. The L bit, defined for ERO subobjects in [RFC3209], is reused here to indicate that an abstract node MUST be excluded (value

0) or SHOULD be avoided (value 1). The distinction is that the path of an LSP must not traverse an abstract node listed in the XRO with the L bit clear, but may traverse one with the L bit set. A node responsible for routing an LSP (for example, for expanding a loose hop) should attempt to minimize the number of abstract nodes listed in the XRO with the L bit set that are traversed by the LSP according to local policy. A node generating XRO subobjects with the L bit set must be prepared to accept an LSP that traverses one, some, or all of the corresponding abstract nodes.

Subobjects 1, 2, and 4 refer to an interface or a set of interfaces. An Attribute octet is introduced in these subobjects to indicate the attribute (e.g., interface, node, SRLG) associated with the interfaces that should be excluded from the path. For instance, the attribute node allows a whole node to be excluded from the path by specifying an interface of that node in the XRO subobject, in contrast to the attribute interface, which allows a specific interface (or multiple interfaces) to be excluded from the path without excluding the whole node. The attribute SRLG allows all SRLGs associated with an interface to be excluded from the path.

3.1.1. IPv4 Prefix Subobject

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    | IPv4 address (4 bytes)        |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv4 address (continued)      | Prefix Length |   Attribute   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L
: 0 indicates that the attribute specified MUST be excluded.
  1 indicates that the attribute specified SHOULD be avoided.

Attribute

Interface attribute values
0 indicates that the interface or set of interfaces associated with the IPv4 prefix should be excluded or avoided.

Node attribute value
1 indicates that the node or set of nodes associated with the IPv4 prefix should be excluded or avoided.

SRLG attribute values
2 indicates that all the SRLGs associated with the IPv4 prefix should be excluded or avoided.

The rest of the fields are as defined in [RFC3209].

3.1.2. IPv6 Prefix Subobject

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    | IPv6 address (16 bytes)       |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv6 address (continued)                                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv6 address (continued)                                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv6 address (continued)                                      |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   | IPv6 address (continued)      | Prefix Length |   Attribute   |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L
0 indicates that the attribute specified MUST be excluded.
1 indicates that the attribute specified SHOULD be avoided.

Attribute

Interface attribute value
0 indicates that the interface or set of interfaces associated with the IPv6 prefix should be excluded or avoided.

Node attribute value
1 indicates that the node or set of nodes associated with the IPv6 prefix should be excluded or avoided.

SRLG attribute value
2 indicates that all the SRLGs associated with the IPv6 prefix should be excluded or avoided.

The rest of the fields are as defined in [RFC3209].

3.1.3. Unnumbered Interface ID Subobject

```
    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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    |    Reserved   |  Attribute    |
   | |             |               |(must be zero) |               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                        TE Router ID                           |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                     Interface ID (32 bits)                    |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L
0 indicates that the attribute specified MUST be excluded.
1 indicates that the attribute specified SHOULD be avoided.

Attribute

Interface attribute value
0 indicates that the Interface ID specified should be excluded or avoided.

Node attribute value
1 indicates that the node with the Router ID should be excluded or avoided (this can be achieved using an IPv4/v6 subobject as well, but is included here because it may be convenient to use information from subobjects of an RRO, as defined in [RFC3477], in specifying the exclusions).

SRLG attribute value
2 indicates that all the SRLGs associated with the interface should be excluded or avoided.

Reserved
This field is reserved. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

The rest of the fields are as defined in [RFC3477].

3.1.4. Autonomous System Number Subobject

The meaning of the L bit is as follows:
0 indicates that the abstract node specified MUST be excluded.
1 indicates that the abstract node specified SHOULD be avoided.

The rest of the fields are as defined in [RFC3209]. There is no Attribute octet defined.

3.1.5. SRLG Subobject

The meaning of the L bit is as follows:
0 indicates that the SRLG specified MUST be excluded
1 indicates that the SRLG specified SHOULD be avoided

The Attribute octet is not present. The rest of the fields are as defined in the "SRLG Subobject" section of this document.

3.2. Processing Rules for the EXCLUDE_ROUTE Object (XRO)

The exclude route list is encoded as a series of subobjects contained in an EXCLUDE_ROUTE object. Each subobject identifies an abstract node in the exclude route list.

Each abstract node may be a precisely specified IP address belonging to a node, or an IP address with prefix identifying interfaces of a group of nodes, an Autonomous System, or an SRLG.

The Explicit Route and routing processing is unchanged from the description in [RFC3209] with the following additions:

1. When a Path message is received at a node, the node MUST check that it is not a member of any of the abstract nodes in the XRO if it is present in the Path message. If the node is a member of any of the abstract nodes in the XRO with the L-flag set to "exclude", it SHOULD return a PathErr with the error code "Routing Problem" and error value of "Local node in Exclude Route". If there are SRLGs in the XRO, the node SHOULD check that the resources the node uses are not part of any SRLG with the L-flag set to "exclude" that is specified in the XRO. If it is, it SHOULD return a PathErr with error code "Routing Problem" and error value of "Local node in Exclude Route".

2. Each subobject MUST be consistent. If a subobject is not consistent then the node SHOULD return a PathErr with error code "Routing Problem" and error value "Inconsistent Subobject". An example of an inconsistent subobject is an IPv4 Prefix subobject containing the IP address of a node and the attribute field is set to "interface" or "SRLG".

3. The subobjects in the ERO and XRO SHOULD NOT contradict each other. If a Path message is received that contains contradicting ERO and XRO subobjects, then:

   - Subobjects in the XRO with the L flag not set (zero) MUST take precedence over the subobjects in the ERO -- that is, a mandatory exclusion expressed in the XRO MUST be honored and an implementation MUST reject such a Path message. This means that a PathErr with error code "Routing Problem" and error value of "Route blocked by Exclude Route" is returned.

- Subobjects in the XRO with the L flag set do not take precedence over ERO subobjects -- that is, an implementation MAY choose to reject a Path message because of such a contradiction, but MAY continue and set up the LSP (ignoring the XRO subobjects that contradict the ERO subobjects).

4. When choosing a next hop or expanding an explicit route to include additional subobjects, a node:

   a. MUST NOT introduce an explicit node or an abstract node that equals or is a member of any abstract node that is specified in the EXCLUDE_ROUTE object with the L-flag set to "exclude". The number of introduced explicit nodes or abstract nodes with the L flag set to "avoid", which indicates that it is not mandatory to be excluded but that it is less preferred, SHOULD be minimized in the computed path.

   b. MUST NOT introduce links, nodes, or resources identified by the SRLG Id specified in the SRLG subobjects(s). The number of introduced SRLGs with the L flag set to "avoid" SHOULD be minimized.

   If these rules preclude further forwarding of the Path message, the node SHOULD return a PathErr with the error code "Routing Problem" and error value of "Route blocked by Exclude Route".

   Note that the subobjects in the XRO is an unordered list of subobjects.

A node receiving a Path message carrying an XRO MAY reject the message if the XRO is too large or complicated for the local implementation or the rules of local policy. In this case, the node MUST send a PathErr message with the error code "Routing Error" and error value "XRO Too Complex". An ingress LSR receiving this error code/value combination MAY reduce the complexity of the XRO or route around the node that rejected the XRO.

The XRO Class-Num is of the form 11bbbbbb so that nodes that do not support the XRO forward it uninspected and do not apply the extensions to ERO processing described above. This approach is chosen to allow route exclusion to traverse parts of the network that are not capable of parsing or handling the new function. Note that

Record Route may be used to allow computing nodes to observe violations of route exclusion and attempt to re-route the LSP accordingly.

If a node supports the XRO, but not a particular subobject or part of that subobject, then that particular subobject is ignored. Examples of a part of a subobject that can be supported are: (1) only prefix 32 of the IPv4 prefix subobject could be supported, or (2) a particular subobject is supported but not the particular attribute field.

When a node forwards a Path message, it can do the following three operations related to XRO besides the processing rules mentioned above:

1. If no XRO was present, an XRO may be included.

2. If an XRO was present, it may remove the XRO if it is sure that the next nodes do not need this information anymore. An example is where a node can expand the ERO to a full strict path towards the destination. See Figure 1 where BC2 is removing the XRO from the Path message.

3. If an XRO was present, the content of the XRO can be modified. Subobjects can be added or removed. See Figure 1 for an example where AB2 is stripping off some subobjects.

In any case, a node MUST NOT introduce any explicit or abstract node in the XRO (irrespective of the value of the L flag) that it also has introduced in the ERO.

4. Explicit Exclusion Route

The Explicit Exclusion Route defines abstract nodes or resources (such as links, unnumbered interfaces, or labels) that must not or should not be used on the path between two inclusive abstract nodes or resources in the explicit route.

4.1. Explicit Exclusion Route Subobject (EXRS)

A new ERO subobject type is defined. The Explicit Exclusion Route Subobject (EXRS) has type 33. Although the EXRS is an ERO subobject and the XRO is reusing the ERO subobject, an EXRS MUST NOT be present in an XRO. An EXRS is an ERO subobject that contains one or more subobjects of its own, called EXRS subobjects.

The format of the EXRS is 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
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |L|    Type     |     Length    |           Reserved            |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   |                                                               |
   //                one or more EXRS subobjects                  //
   |                                                               |
   +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

L

It MUST be set to zero on transmission and MUST be ignored on receipt. (Note: The L bit in an EXRS subobject is as defined for the XRO subobjects.)

Type

The type of the subobject (33).

Reserved

This field is reserved. It SHOULD be set to zero on transmission and MUST be ignored on receipt.

EXRS subobjects

An EXRS subobject indicates the abstract node or resource to be excluded. The format of an EXRS subobject is exactly the same as the format of a subobject in the XRO. An EXRS may include all subobjects defined in this document for the XRO.

Thus, an EXRS for an IP hop may look 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
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |L|    Type     |     Length    |           Reserved            |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    |L|    Type     |     Length    | IPv4 address (4 bytes)        |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
    | IPv4 address (continued)      | Prefix Length |   Attribute   |
    +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

4.2. Processing Rules for the Explicit Exclusion Route Subobject (EXRS)

Each EXRS may carry multiple exclusions. The exclusion is encoded exactly as for XRO subobjects and prefixed by an additional Type and Length.

The scope of the exclusion is the step between the previous ERO subobject that identifies an abstract node, and the subsequent ERO subobject that identifies an abstract node. The processing rules of the EXRS are the same as the processing rule of the XRO within this scope. Multiple exclusions may be present between any pair of abstract nodes.

Exclusions may indicate explicit nodes, abstract nodes, or Autonomous Systems that must not be traversed on the path to the next abstract node indicated in the ERO.

Exclusions may also indicate resources (such as unnumbered interfaces, link ids, and labels) that must not be used on the path to the next abstract node indicated in the ERO.

SRLGs may also be indicated for exclusion from the path to the next abstract node in the ERO by the inclusion of an EXRS containing an SRLG subobject. If the L bit in the SRLG subobject is zero, the resources (nodes, links, etc.) identified by the SRLG MUST NOT be used on the path to the next abstract node indicated in the ERO. If the L bit is set, the resources identified by the SRLG SHOULD be avoided.

If a node is called upon to process an EXRS and does not support handling of exclusions it will behave as described in [RFC3209] when an unrecognized ERO subobject is encountered. This means that this node will return a PathErr with error code "Routing Error" and error value "Bad EXPLICIT_ROUTE object" with the EXPLICIT_ROUTE object included, truncated (on the left) to the offending EXRS.

If the presence of EXRS precludes further forwarding of the Path message, the node SHOULD return a PathErr with the error code "Routing Problem" and error value "Route Blocked by Exclude Route".

A node MAY reject a Path message if the EXRS is too large or complicated for the local implementation or as governed by local policy. In this case, the node MUST send a PathErr message with the error code "Routing Error" and error value "EXRS Too Complex". An ingress LSR receiving this error code/value combination MAY reduce the complexity of the EXRS or route around the node that rejected the EXRS.

5. Processing of XRO together with EXRS

When an LSR performs ERO expansion and finds both the XRO in the Path message and EXRS in the ERO, it MUST exclude all the SRLGs, nodes, links, and resources listed in both places. Where some elements appear in both lists, it MUST be handled according to the stricter exclusion request. That is, if one list says that an SRLG, node, link, or resource must be excluded, and the other says only that it should be avoided, then the element MUST be excluded.

6. Minimum Compliance

An implementation MUST be at least compliant with the following:

1. The XRO MUST be supported with the following restrictions:

   - The IPv4 Prefix subobject MUST be supported with a prefix length of 32, and an attribute value of "interface" and "node". Other prefix values and attribute values MAY be supported.

   - The IPv6 Prefix subobject MUST be supported with a prefix length of 128, and an attribute value of "interface" and "node". Other prefix values and attribute values MAY be supported.

2. The EXRS MAY be supported. If supported, the same restrictions as for the XRO apply. If not supported, an EXRS encountered during normal ERO processing MUST be rejected as an unknown ERO subobject as described in Section 4.2. Note that a node SHOULD NOT parse ahead into an ERO, and if it does, it MUST NOT reject the ERO if it discovers an EXRS that applies to another node.

3. If XRO or EXRS are supported, the implementation MUST be compliant with the processing rules of the supported, not supported, or partially supported subobjects as specified within this document.

7. Security Considerations

Security considerations for MPLS-TE and GMPLS signaling are covered in [RFC3209] and [RFC3473]. This document does not introduce any new messages or any substantive new processing, and so those security considerations continue to apply.

Note that any security concerns that exist with explicit routes should be considered with regard to route exclusions. For example, some administrative boundaries may consider explicit routes to be security violations and may strip EROs from the Path messages that they process. In this case, the XRO should also be considered for removal from the Path message.

It is possible that an arbitrarily complex XRO or EXRS sequence could be introduced as a form of denial-of-service attack since its presence will potentially cause additional processing at each node on the path of the LSP. It should be noted that such an attack assumes that an otherwise trusted LSR (i.e., one that has been authenticated by its neighbors) is misbehaving. A node that receives an XRO or EXRS sequence that it considers too complex according to its local policy may respond with a PathErr message carrying the error code "Routing Error" and error value "XRO Too Complex" or "EXRS Too Complex".

8. IANA Considerations

It might be considered that an alternative approach would be to assign one of the bits of the ERO subobject type field (perhaps the top bit) to identify that a subobject is intended for inclusion rather than exclusion. However, [RFC3209] states that the type field (seven bits) should be assigned as 0 - 63 through IETF consensus action, 64 - 95 as first come first served, and 96 - 127 are reserved for private use. It would not be acceptable to disrupt existing implementations, so the only option would be to split the IETF consensus range leaving only 32 subobject types. It is felt that 32 would be an unacceptably small number for future expansion of the protocol.

8.1. New ERO Subobject Type

IANA registry: RSVP PARAMETERS
Subsection: Class Names, Class Numbers, and Class Types

A new subobject has been added to the existing entry for:

20 EXPLICIT_ROUTE

The registry reads:

33 Explicit Exclusion Route subobject (EXRS)

The Explicit Exclusion Route subobject (EXRS) is defined in Section 4.1, "Explicit Exclusion Route Subobject (EXRS)". This subobject may be present in the Explicit Route Object, but not in the Route Record Object or in the new EXCLUDE_ROUTE object, and it should not be listed among the subobjects for those objects.

8.2. New RSVP-TE Class Numbers

IANA registry: RSVP PARAMETERS
Subsection: Class Names, Class Numbers, and Class Types

A new class number has been added for EXCLUDE_ROUTE object (XRO) as defined in Section 3.1, "EXCLUDE_ROUTE Object (XRO)".

EXCLUDE_ROUTE
Class-Num of type 11bbbbbb
Value: 232
Defined CType: 1 (EXCLUDE_ROUTE)

Subobjects 1, 2, 4, and 32 are as defined for Explicit Route Object. An additional subobject has been registered as requested in Section 8.1, "New ERO Subobject Type". The text should appear as:

Sub-object type

1	IPv4 address	[RFC3209]
2	IPv6 address	[RFC3209]
4	Unnumbered Interface ID	[RFC3477]
32	Autonomous system number	[RFC3209]
33	Explicit Exclusion Route subobject (EXRS)	[RFC4874]
34	SRLG	[RFC4874]

The SRLG subobject is defined in Section 3.1.5, "SRLG Subobject". The value 34 has been assigned.

8.3. New Error Codes

IANA registry: RSVP PARAMETERS
Subsection: Error Codes and Globally-Defined Error Value Sub-Codes

New Error Values sub-codes have been registered for the Error Code 'Routing Problem' (24).

64 = Unsupported Exclude Route Subobject Type
65 = Inconsistent Subobject
66 = Local Node in Exclude Route
67 = Route Blocked by Exclude Route
68 = XRO Too Complex
69 = EXRS Too Complex

9. Acknowledgments

This document reuses text from [RFC3209] for the description of EXCLUDE_ROUTE.

The authors would like to express their thanks to Lou Berger, Steffen Brockmann, Igor Bryskin, Dimitri Papadimitriou, Cristel Pelsser, and Richard Rabbat for their considered opinions on this document. Also thanks to Yakov Rekhter for reminding us about SRLGs!

Thanks to Eric Gray for providing GenArt review and to Ross Callon for his comments.

10. References

10.1. Normative References

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

[RFC3473] Berger, L., "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (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.

[RFC4202] Kompella, K. and Y. Rekhter, "Routing Extensions in Support of Generalized Multi-Protocol Label Switching (GMPLS)", RFC 4202, October 2005.

10.2. Informative References

[CRANKBACK] Farrel, A., Satyanarayana, A., Iwata, A., Ash, G., and S. Marshall-Unitt, "Crankback Signaling Extensions for MPLS Signaling", Work in Progress, January 2007.

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", RFC 3630, September 2003.

[RFC3784] Smit, H. and T. Li, "Intermediate System to Intermediate System (IS-IS) Extensions for Traffic Engineering (TE)", RFC 3784, June 2004.

[RFC3812] Srinivasan, C., Viswanathan, A., and T. Nadeau, "Multiprotocol Label Switching (MPLS) Traffic Engineering (TE) Management Information Base (MIB)", RFC 3812, June 2004.

[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.

[RFC4216] Zhang, R. and JP. Vasseur, "MPLS Inter-Autonomous System (AS) Traffic Engineering (TE) Requirements", RFC 4216, November 2005.

Appendix A. Applications

This section describes some applications that can make use of the XRO. The intention is to show that the XRO is not an application-specific object, but that it can be used for multiple purposes. In a few examples, other solutions might be possible for that particular case, but the intention is to show that a single object can be used for all the examples, hence making the XRO a rather generic object without having to define a solution and new objects for each new application.

A.1. Inter-Area LSP Protection

One method to establish an inter-area LSP is where the ingress router selects an ABR, and then the ingress router computes a path towards this selected ABR such that the configured constraints of the LSP are fulfilled. In the example of Figure A.1, an LSP has to be established from node A in area 1 to node C in area 2. If no loose hops are configured, then the computed ERO at A could look as follows: (A1-strict, A2-strict, ABR1-strict, C-loose). When the Path message arrives at ABR1, then the ERO is (ABR1-strict, C-loose), and it can be expanded by ABR1 to (B1-strict, ABR3-strict, C-loose). Similarly, at ABR3 the received ERO is (ABR3-strict, C-loose), and it can be expanded to (C1-strict, C2-strict, C-strict). If a backup LSP also has to be established, then A takes another ABR (ABR2 in this case) and computes a path towards this ABR that fulfills the constraints of the LSP and that is disjoint from the path of the primary LSP. The ERO generated by A looks as follows for this example: (A3-strict, A4-strict, ABR2-strict, C-loose).

In order to let ABR2 expand the ERO, it also needs to know the path of the primary LSP so that the ERO expansion is disjoint from the path of the primary LSP. Therefore, A also includes an XRO that at least contains (ABR1, B1, ABR3, C1, C2). Based on these constraints, ABR2 can expand the ERO such that it is disjoint from the primary LSP. In this example, the ERO computed by ABR2 would be (B2-strict, ABR4-strict, C-loose), and the XRO generated by B contains at least (ABR3, C1, C2). The latter information is needed for ABR4 to expand the ERO so that the path is disjoint from the primary LSP in area 2.

```
            Area 1           Area 0          Area 2
       <---------------><--------------><--------------->

       +---A1---A2----ABR1-----B1-----ABR3----C1---C2---+
       |        |              |              |         |
       |        |              |              |         |
       A        |              |              |         C
       |        |              |              |         |
       |        |              |              |         |
       +---A3---A4----ABR2-----B2-----ABR4----C3---C4---+
```

Figure A.1: Inter-area LSPs

In this example, a node performing the path computation first selects an ABR and then computes a strict path towards this ABR. For the backup LSP, all nodes of the primary LSP in the next areas have to be put in the XRO (with the exception of the destination node if node protection and no link protection is required). When an ABR computes the next path segment, i.e., the path over the next area, it may remove the nodes from the XRO that are located in that area with the exception of the ABR where the primary LSP is exiting the area. The latter information is still required because when the selected ABR (ABR4 in this example) further expands the ERO, it has to exclude the ABR on which the primary LSP is entering that area (ABR3 in this example). This means that when ABR2 generates an XRO, it may remove the nodes in area 0 from the XRO but not ABR3. Note that not doing this would not cause harm in this example because there is no path from ABR4 to C via ABR3 in area 2. If there is a link between ABR4-ABR3 and ABR3-C, then it is required to have ABR3 in the XRO generated by ABR2.

Discussion on the length of the XRO: When link or node protection is requested, the length of the XRO is bounded by the length of the RRO of the primary LSP. It can be made shorter by removing nodes by the ingress node and the ABRs. In the example above, the RRO of the primary LSP contains 8 subobjects, while the maximum XRO length can be bounded by 6 subobjects (nodes A1 and A2 do not have to be in the XRO). For SRLG protection, the XRO has to list all SRLGs that are crossed by the primary LSP.

A.2. Inter-AS LSP Protection

When an inter-AS LSP (which has to be protected by a backup LSP to provide link or node protection) is established, the same method as for the inter-area LSP case can be used. The difference is when the backup LSP is not following the same AS-path as the primary LSP because then the XRO should always contain the full path of the primary LSP. In case the backup LSP is following the same AS-path (but with different ASBRs -- at least in case of node protection), it is similar to the inter-area case: ASBRs expanding the ERO over the next AS may remove the XRO subobjects located in that AS. Note that this can only be done by an ingress ASBR (the ASBR where the LSP is entering the AS).

Discussion on the length of the XRO: the XRO is bounded by the length of the RRO of the primary LSP.

Suppose that SRLG protection is required, and the ASs crossed by the main LSP use a consistent way of allocating SRLG-ids to the links (i.e., the ASs use a single SRLG space). In this case, the SRLG-ids of each link used by the main LSP can be recorded by means of the RRO; the SRLG-ids are then used by the XRO. If the SRLG-ids are only meaningful when local to the AS, putting SRLG-ids in the XRO crossing many ASs makes no sense. To provide SRLG protection for inter-AS LSPs the link IP address of the inter-AS link used by the primary LSP can be put into the XRO of the Path message of the detour LSP or bypass tunnel. The ASBR where the detour LSP or bypass tunnel is entering the AS can translate this into the list of SRLG-ids known to the local AS.

Discussion on the length of the XRO: the XRO only contains 1 subobject, which contains the IP address of the inter-AS link traversed by the primary LSP (assuming that the primary LSP and detour LSP or bypass tunnel are leaving the AS in the same area, and that they are also entering the next AS in the same area).

A.3. Protection in the GMPLS Overlay Model

When an edge-node wants to establish an LSP towards another edge-node over an optical core network as described in [RFC4208] (see Figure A.2), the XRO can be used for multiple purposes.

```
     Overlay                                                  Overlay
     Network        +--------------------------------+        Network
   +----------+     |                                |     +----------+
   |   +----+ |     |  +-----+   +-----+   +-----+   |     | +----+   |
   |   |    | |     |  |     |   |     |   |     |   |     | |    |   |
   | --+ EN1+-+-----+--+ CN1 +---+ CN2 +---+ CN3 +---+-----+-+ EN3+-- |
   |   |    | |  +--+--+     |   |     |   |     +---+--+  | |    |   |
   |   +----+ |  |  |  +--+--+   +--+--+   +--+--+   |  |  | +----+   |
   |          |  |  |     |         |         |      |  |  |          |
   +----------+  |  |     |         |         |      |  |  +----------+
                 |  |     |         |         |      |  |
   +----------+  |  |     |         |         |      |  |  +----------+
   |          |  |  |  +--+--+      |      +--+--+   |  |  |          |
   |   +----+ |  |  |  |     |      +------+     |   |  |  | +----+   |
   |   |    +-+--+  |  | CN4 +-------------+ CN5 |   |  +--+-+    |   |
   | --+ EN2+-+-----+--+     |             |     +---+-----+-+ EN4+-- |
   |   |    | |     |  +-----+             +-----+   |     | |    |   |
   |   +----+ |     |                                |     | +----+   |
   |          |     +--------------------------------+     |          |
   +----------+                 Core Network               +----------+

        Overlay                                                 Overlay
        Network                                                 Network

    Legend:
         EN - Edge-Node
         CN - Core-Node
```

Figure A.2

A first application is where an edge-node wants to establish multiple LSPs towards the same destination edge-node, and these LSPs need to have few or no SRLGs in common. In this case EN1 could establish an LSP towards EN3, and then it can establish a second LSP listing all links used by the first LSP with the indication to avoid the SRLGs of these links. This information can be used by CN1 to compute a path for the second LSP. If the core network consists of multiple areas, then the SRLG-ids have to be listed in the XRO. The same example applies to nodes and links.

Another application is where the edge-node wants to set up a backup LSP that is also protecting the links between the edge-nodes and core-nodes. For instance, when EN2 establishes an LSP to EN4, it sends a Path message to CN4, which computes a path towards EN4 over (for instance) CN5. When EN2 gets back the RRO of that LSP, it can signal a new LSP to CN1 with EN4 as the destination and the XRO computed based on the RRO of the first LSP. Based on this information, CN1 can compute a path that has the requested diversity properties (e.g., a path going over CN2 and CN3, and then to EN4).

It is clear that in these examples, the core-node may not alter the RRO in a Resv message to make its only contents be the subobjects from the egress core-node through the egress edge-node.

A.4. LSP Protection inside a Single Area

The XRO can also be used inside a single area. Take for instance a network where the TE extensions of the IGPs as described in [RFC3630] and [RFC3784] are not used. Hence, each node has to select a next-hop and possibly crankback [CRANKBACK] has to be used when there is no viable next-hop. In this case, when signaling a backup LSP, the XRO can be put in the Path message to exclude the links, nodes, or SRLGs of the primary LSP. An alternative way to provide this functionality would be to indicate the following in the Path message of the backup LSP: the primary LSP and which type of protection is required. This latter solution would work for link and node protection, but not for SRLG protection.

When link or node protection is requested, the XRO is of the same length as the RRO of the primary LSP. For SRLG protection, the XRO has to list all SRLGs that are crossed by the primary LSP. Note that for SRLG protection, the link IP address to reference the SRLGs of that link cannot be used since the TE extensions of the IGPs are not used in this example. Hence, a node cannot translate any link IP address located in that area to its SRLGs.

Authors' Addresses

Cheng-Yin Lee
EMail: c.yin.lee@gmail.com

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

Stefaan De Cnodder
Alcatel-Lucent
Copernicuslaan 50
B-2018 Antwerp
Belgium
Phone: +32 3 240 85 15
EMail: stefaan.de_cnodder@alcatel-lucent.be

Full Copyright Statement

Intellectual Property

Acknowledgement

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