

Exclude Routes - Extension to RSVP-TE

[draft-lee-ccamp-rsvp-te-exclude-route-00.txt](#)

1. Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC2026](#).

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

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

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

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

2. Abstract

The current RSVP-TE specification [[RSVP-TE](#)] and GMPLS extensions [[GMPLS-RSVP-TE](#)] allow abstract nodes and resources to be explicitly included in a path setup, but not to be explicitly excluded.

In some systems 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 of a path, or to parts of a path between two abstract nodes specified in an explicit route.

Shared Risk Link Groups (SRLGs) allow the definition of resources or groups of resources that share the same risk of failure. The knowledge of SRLGs 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 of a path or between two abstract nodes specified in an explicit path.

This draft specifies ways to communicate route exclusions during path setup using RSVP-TE.

These approaches are equally applicable to other MPLS TE signaling protocols such as CR-LDP.

3. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [[RFC2119](#)].

4. Overview

The current RSVP-TE specification [[RSVP-TE](#)] and GMPLS extensions [[GMPLS-RSVP-TE](#)] 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.

Two types of exclusions are required:

- i) Do not include any of the abstract nodes in a given set anywhere on the path. This set of abstract nodes to exclude is referred to as the Exclude Route list.
- ii) Do not include certain abstract nodes or resources between a specific pair of abstract nodes present in an ERO. Such specific exclusions are referred to as Explicit Route Exclusions.

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 subobject type (the Exclude Route Subobject) is introduced to indicate an exclusion between a pair of included abstract nodes.

At the same time, it is recognized that SRLGs are a useful means of indicating resources that share the same risk of failure. When establishing protection LSPs they are often required to be node and link diverse from the LSPs that they protect. Further, where SRLGs are known, the protection LSPs are required to not utilize resources in the SRLGs traversed by the protected LSPs.

This draft introduces an ERO subobject to indicate an SRLG to be signaled in either of the two exclusion methods described above. This subobject might also be appropriate for use within Explicit Routes, but that discussion is outside the scope of this draft.

5. Shared Risk Link Groups

The identifier of a SRLG is defined as a 32 bit quantity in [GMPLS-OSPF]. These 32 bits are divided into an 8 bit type field and a 24 bit identifier in [IPO-SRLG].

5.1 SRLG ERO Subobject

The format of the ERO and its subobjects are defined in [RSVP-TE].

The SRLG subobject is defined 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   |   Tolerance   |   Reserved   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|   SRLG Id   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

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, the L bit SHOULD be set to zero and ignored.

Type

The type of the subobject [TBD].

Length

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

Tolerance

The level to which it is permissible for this SRLG to be included in the path when more than one SRLG is specified. A value of zero indicates that this SRLG MUST be avoided. A tolerance value of $n < m$ indicates that the SRLG MUST be avoided in preference to an SRLG with tolerance value m .

If only one SRLG is present, then a value other than zero indicates the SRLG SHOULD be avoided.

SRLG Id

The 32 bit identifier of the SRLG.

5.2 Exclusion Tolerance Semantics

The Tolerance field in the SRLG subobject indicates the degree to which the SRLG must be avoided. (The degree to which it is permissible to include it.)

If the Tolerance field has the value zero (0), the LSP MUST NOT traverse or use any resource that is a member of the SRLG.

If the value is non-zero, all path computation elements SHOULD attempt to select routes that avoid all resources that are members of the SRLG.

Where more than one SRLG with non-zero Tolerance value is specified for exclusion and no route can be found that avoids both SRLGs, a route SHOULD be chosen that avoids the SRLG with the lower Tolerance value.

6. Exclude Route List

The exclude route identifies a list of abstract nodes that MUST NOT be traversed along the path.

6.1 Exclude Route Object

Abstract nodes to be excluded from the path are specified via the EXCLUDE_ROUTE object (XR0). The Exclude Route Class value is [TBD].

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

Class = TBD, 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)              //
|
+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+--+

```

Subobjects

The contents of an EXCLUDE_ROUTE object are a series of variable-length data items called subobjects. The subobjects are identical to those defined in [[RSVP-TE](#)] and [[GMPLS-RSVP-TE](#)] for use in EROs.

The following subobject types are supported.

- | | |
|-----|--------------------------|
| 1 | IPv4 prefix |
| 2 | IPv6 prefix |
| 32 | Autonomous system number |
| TBD | SRLG |

The defined values for Type above are specified in [[RSVP-TE](#)] and in this document.

The L bit that denotes a loose hop when the subobject is used in the ERO has no meaning in the XRO and should be ignored.

6.2. Semantics and 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 of a single node, an IP address with prefix identifying a group of nodes, or an Autonomous System.

The Explicit Route and routing processing is unchanged from the description in [[RSVP-TE](#)] with the following additions:

- a. 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 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 it and the resources it uses are not part of any SRLG that is specified with Tolerance value of zero. If it is, it should return a PathErr with the error code "Routing Problem" and error value of "Local node in Exclude Route".
The node may be a member of an SRLG in the XRO that is specified with a non-zero Tolerance value.
- b. When choosing a next hop or expanding an explicit route to include additional subobjects, a node:
 - i) 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.
 - ii) must not (or should not, in the case of a non-zero Tolerance value) introduce links, nodes or resources identified by the SRLG ID specified in the SRLG subobjects(s).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".

The XRO Class-Num is of the form 11bbbbbb so that nodes which do not support the XRO will forward it uninspected and will not apply the extensions to ERO processing described above. This makes the XRO a 'best effort' process.

This 'best-effort' 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.

7. Explicit Route Exclusions

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

7.1. Exclude Object Subobject

A new subobject type is defined. The Exclude Object Subobject has type [TBD].

The format of the Exclude Object Subobject is as follows.

```

0                               1
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|L|   Type       |   Length   | Exclude subobject contents   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
```

L (ignored, must be 0)

Type

The type of the subobject [TBD]

Exclude subobject contents

An ERO subobject indicating the abstract node or resource to be excluded. The format of this field is exactly the format of an ERO subobject contained in an ERO and may include an SRLG subobject as described earlier in this draft.

Thus, an exclude subobject for an IP hop might 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   |R|   Type   |   Length   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| IPv4 address (4 bytes)                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
| Prefix Length |   Reserved   |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

R is reserved and must be zero

7.2. Semantics and Processing Rules for the Exclude Object Subobject

Each Exclude Object Subobject carries a single exclusion. The exclusion is encoded exactly as an inclusion in the ERO 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. 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, 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 Exclude Object Subobject containing an SRLG subobject. If the Tolerance value 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 Tolerance value is non-zero, the resources identified by the SRLG should be avoided, but may be used in preference to resources associated with another SRLG indicated for exclusion if that SRLG has a (numerically) lower Tolerance value.

If a node is called upon to process an Exclude Object Subobject and does not support handling of exclusions it will return a PathErr with a "Bad EXPLICIT_ROUTE object" error.

If the presence of Exclude Object Subobjects precludes 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".

8. Security

The new exclude route object poses no security exposures over and above [[RSVP-TE](#)] and [[GMPLS-RSVP-TE](#)]. Note that any security concerns that exist with Explicit Routes should be considered with regard to route exclusions.

9. IANA Considerations

9.1. New Class Numbers

One new class number is required.

```
EXCLUDE_ROUTE
  Class-Num = 011bbbbbb
  CType: 1
```

9.2. Explicit Route Subobject Types

Two new subobject types for the Explicit Route Object are required.

```
SRLG subobject
Exclude object subobject
```

9.3. New Error Codes

New error values are needed for the error code 'Routing Problem'.

```
Unsupported Exclude Route Subobject Type
Local node in Exclude Route
Route blocked by Exclude Route
```

10. Acknowledgments

This draft reuses text from [[RSVP-TE](#)] for the description of EXCLUDE_ROUTE.

The authors would like to express their thanks to Igor Bryskin, Lou Berger and Stefaan de Cnodder for their considered opinions on this draft. Also thanks to Yakov Rekhter for reminding us about SRLGs.

11. Authors' Information

Cheng-Yin Lee
600 March Road
Ottawa, Ontario
Canada K2K 2E6
email: Cheng-Yin.Lee@alcatel.com

Adrian Farrel
Movaz Networks, Inc.
7926 Jones Branch Drive, Suite 615
McLean VA, 22102 USA
Phone: +1-703-847-1867
Email: afarrel@movaz.com

12. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997
- [RSVP-TE] D. Awduche, et al., "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), December 2001.
- [GMPLS-RSVP-TE] P. Ashwood-Smith, et al., "Generalized MPLS Signaling - RSVP-TE Extensions", Internet Draft, [draft-ietf-mpls-generalized-rsvp-te-07.txt](#), April 2002 (work in progress).
- [GMPLS-OSPF] K. Kompela, et al., "OSPF Extensions in Support of Generalized MPLS", Internet Draft, [draft-ietf-ccamp-ospf-gmpls-extensions-07.txt](#), May 2002 (work in progress).
- [IPO-SRLG] D. Papadimitriou, et al., "Inference of Shared Risk Link Groups", Internet Draft, [draft-many-inference-srlg-02.txt](#), November 2001 (work in progress).

13. Informational References

- [MPLS-BUNDLE] Kompella, K., Rekhter, Y., and Berger, L., "Link Bundling in MPLS Traffic Engineering", Internet Draft, [draft-ietf-mpls-bundle-02.txt](#), May 2002, (work in progress).
- [MPLS-UNNUM] Kompella, K., Rekhter, Y., "Signalling Unnumbered Links in RSVP-TE", Internet Draft, [draft-ietf-mpls-rsvp-unnum-06.txt](#), May 2002, (work in progress).

[GMPLS-SIG] P. Ashwood-Smith, et al, "Generalized MPLS -
Signaling Functional Description",
[draft-ietf-mpls-generalized-signaling-08.txt](#)
April 2002, (work in progress).

14. Full Copyright Statement

Copyright (C) The Internet Society (2002). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns. This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

