CCAMP Working Group
Internet Draft
Intended status: Standard Track
Expires: January 15, 2013

Zafar Ali
George Swallow
Clarence Filsfils
Ori Gerstel
Cisco Systems
Kenji Kumaki
KDDI Corporation
Ruediger Kunze
Deutsche Telekom AG
July 16, 2012

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

draft-ali-ccamp-rsvp-te-include-route-02.txt

Status of this Memo

Copyright Notice

Abstract

There are scenarios that require two or more LSPs or segments of LSPs to follow same route in the network. This document specifies methods to communicate route inclusions along the loose hops during path setup using the Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) protocol.

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 RFC 2119 [RFC2119].

Table of Contents

1. Introduction

There are scenarios that require two or more LSPs to follow same route in the network. E.g., many deployments require member LSPs of a bundle/ aggregated link (or Forwarding Adjacency (FA)) follow the same route. Possible reasons for two or more LSPs to follow the same end-to-end or partial route include, but are not limited to:

- Fate sharing: an application may require that two or more LSP fail together. In the example of bundle link this would mean that if one component goes down, the entire bundle goes down.

- Homogeneous Attributes: it is often required that two or more LSPs have the same TE metrics like latency, delay variation, etc. In the example of a bundle/ aggregated link this would meet the requirement that all component links (FAs) of a bundle should have same latency and delay variation. As noted in [OSPF-TE-METRIC] and [ISIS-TE-METRIC], in certain networks, such as financial information networks, network performance (e.g. latency and latency variation) is becoming critical and hence having bundles with component links (FAs) with homogeneous delay and delay variation is important.

The RSVP-TE specification, "RSVP-TE: Extensions to RSVP for LSP Tunnels" [RFC3209] and GMPLS extensions to RSVP-TE, "Generalized Multi-Protocol Label Switching (GMPLS) Signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Extensions" [RFC3473] allow abstract nodes and resources to be explicitly included in a path setup. However, such inclusion may not be possible when a loose hop is expanded. It is obviously possible to divide the loose hop into multiple loose hops and construct an inclusion in that fashion. However, there are scenarios where division of a loose hop into multiple explicit loose hops is not possible. Included (but not limited to) are the following:

. When the destination is in another area, AS, or across a UNI, the ingress node may not have full visibility of the topology. In cases where the ingress node lacks sufficient topological knowledge around the loose hop, it is not able to divide a loose hop into a proper sequence of strict or a sequence of finer-grained loose hops.

. The ingress node requires two Label Switched Paths (LSPs) to follow the same route but has no knowledge of how a loose hop of a reference LSP was expanded. The ingress node requires certain SLRGs to be explicitly "included" when the loose hop is expanded. This document defines inclusion use of the SRLG subobject defined in [RFC4874].

When the entire route of LSPs that need to follow the same path is computed by the ingress node, the aforementioned requirements can be met by a local decision at the ingress node. However, there are scenarios where a full route computation is not performed at the ingress but instead is performed by remote nodes. This case creates a need for relevant affinity requirements to be communicated to the route expanding nodes. These include (but are not limited to):

- LSPs with loose hops in the Explicit Route Object (ERO), e.g. inter-domain LSPs.

- Generalized Multi-Protocol Label Switching (GMPLS) User-Network Interface (UNI) where route computation may be performed by the UNI-Network (server) node;

This document addresses these requirements and defines procedures that may be used to signal LSPs such that the entire LSP or LSP segments follow the same route.

2. RSVP-TE signaling extensions

A new ERO subobject type the Explicit Inclusion Route Subobject (EIRS) is introduced to indicate an inclusion between a pair of included nodes or abstract nodes. The ERO subobject encoding and processing rules are similar to Explicit Exclusion Route Subobject (EXRS) subobject of ERO defined in [RFC4874], with the exception of include vs. exclude usage.

2.1. Explicit Inclusion Route Subobject (EIRS)

The Explicit Inclusion Route Subobject (EIRS) defines abstract nodes or resources (such as links, SRLG, Circuit IDs (see [DRAFT-LSP-XRO]), unnumbered interfaces, or labels, etc.) that must or should be used on the path between two inclusive abstract nodes or resources in the explicit route. An EIRS is an ERO subobject that contains one or more subobjects of its own, called EIRS subobjects. Each EIRS may carry multiple inclusions. The inclusion is encoded exactly as for XRO subobjects and prefixed by an additional Type and Length.

The format of the EIRS 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 EIRS subobjects                  //
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```


An example of EIRS for SLRG inclusion (SRLG Id 1 and SRLG Id 2) is provided in the following. This example is referenced in the following description.

```
 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    |       SRLG Id 1 (4 bytes)     |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SRLG Id 1 (continued)      |           Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|L|    Type     |     Length    |      SRLG Id 2(4 bytes)       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|      SRLG Id 2 (continued)      |           Reserved          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 1: Example of EIRS with SRLG subobjects

Please note that there are two or more "L bits" in an EIRS. The following convention is used to reference the individual "L bits".

EIRS.L: The L bit of the header of the EIRS subobject. E.g., EIRS.L refers to the first L bit in EIRS example in Figure 1.

EIRS.SubobjectN.L: The L bit of the nth subobject of EIRS. E.g., EIRS.Subobject2.L refers to the third L bit in EIRS example in Figure 1 (i.e., the L bit to define the expected treatment of SRLG ID2 value).

The fields of the EIRS subobject are defined as follows:

EIRS.L bit: The L bit is an attribute of the EIRS subobject. The L bit SHOULD be set, so that the subobject represents a loose hop in the explicit route.

EIRS.Type: The type of the subobject is to be defined by IANA (Suggested Value: 68).

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

EIRS subobjects: An EIRS subobject indicates the abstract node or resource to be included in the path. The format of an EIRS subobject is exactly the same as the format of a subobject in the eXclude Route Object (XRO) (See [RFC4874] and [DRAFT-LSP-XRO-SUB]). This is with the exception of the interpretation of the "EIRS.SubobjectN.L bit" of the subobjects, as detailed in the following.

EIRS.SubobjectN.L bit: For all supported subobjects of EIRS, the EIRS.SubobjectN.L bit has the following interpretation.

- EIRS.SubobjectN.L = 0 indicates that the attribute specified MUST be included.

- EIRS.SubobjectN.L = 1 indicates that the attribute specified SHOULD be included.

An EIRS may include all subobjects defined in this document for the XRO (See [RFC4874] and [DRAFT-LSP-XRO-SUB]). Specifically, an EIRS may include the following subobjects:

EIRS.SubobjectN.Type = 1: IPv4 address [RFC3209].

EIRS.SubobjectN.Type = 2: IPv6 address [RFC3209].

EIRS.SubobjectN.Type = 3: Label [RFC6001].

EIRS.SubobjectN.Type = 4: Unnumbered Interface ID [RFC3477].

EIRS.SubobjectN.Type = 32: Autonomous system number [RFC3209].

EIRS.SubobjectN.Type = 34: SRLG [RFC4874].

EIRS.SubobjectN.Type = 35: Switching Capability (SC) [RFC6001].

EIRS.SubobjectN.Type = TBD (suggested value 37): LSP [DRAFT-LSP-XRO-SUB].

Please note that EIRS.SubobjectN.Type = 33: Explicit Exclusion Route subobject (EXRS) [RFC4874] is not supported.

2.2. EIRS Subobject Processing Rule

The scope of the inclusion is the previous ERO subobject that identifies a node or an abstract node, and the subsequent ERO subobject that identifies a node or an abstract node. The processing rules of the EIRS are the same as the processing rule of the EXRS, with the exception that EIRS subobjects request resource inclusion, whereas EXRS subobjects request resource exclusion.

Multiple inclusions may be present between any pair of nodes or abstract nodes. An EIRS may be present when an EXRS is also present in the ERO and/ or an XRO is also present in the path message. Section 2.3 discusses details of processing of the EIRS with the XRO object and the EXRS subobject of ERO.

If the processing node does not understand the EIRS subobject, it behaves 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 EIRS subobject.

If the EIRS.L bit is not set, the processing node SHOULD generate a Path Error with error code "Routing Problem" and error subcode "Bad EXPLICIT_ROUTE object".

If the processing node understands the EIRS subobject and all the subobjects contained in the EIRS, it takes the following steps:

- For all subobjects contained in the EIRS such that EIRS.SubobjectN.L = 0, the processing node finds a path that MUST include the resource attribute identified by the EIRS.SubobjectN.
- For all subobjects contained in the EIRS such that EIRS.SubobjectN.L = 1, the processing node finds a path that MUST include the resource attribute identified by the EIRS.SubobjectN.
- If the processing node fails to find a route such that the all resources identified in the EIRS.SubobjectN for all N can be included in the route (depending on EIRS.SubobjectN.L bit setting), the node SHOULD return a PathErr with the error code "Routing Problem" and error value "Route Blocked by Include

  Route". The error subcode "Route Blocked by Include Route" for Path Error code "Routing Problem" is to be assigned by IANA (Suggested Value: 110).

If the processing node understands the EIRS subobject but does not understand or support a subobject contained in the EIRS (say EIRS. SubobjectN), it SHOULD 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 EIRS subobject containing the unsupported EIRS.subobjectN.

A node MAY reject a Path message if the EIRS is too large or complicated for the local implementation or as governed by local policy. In this case, the node SHOULD send a PathErr message with the error code "Routing Error" and error subcode "EIRS Too Complex". An ingress node receiving this error code/subcode combination MAY reduce the complexity of the EIRS. The error subcode "EIRS Too Complex" for Path Error code "Routing Problem" is to be assigned by IANA (Suggested Value: 111).

2.3. Processing of EIRS with XRO and EXRS

A node performing ERO expansion MAY find an XRO in the Path message and both EIRS and EXRS subobjects in ERO. In this case, the processing node MUST include all resources identified in the EIRS and exclude all resources identified in the EXRS and XRO.

If the constraints identified by the EIRS, EXRS and XRO conflict each other, the processing node SHOULD send a PathErr message with the   error code "Routing Error" and error subcode "inconsistent include/ exclude constraints". The error subcode "inconsistent include/ exclude constraints" for Path Error code "Routing Problem" is to be assigned by IANA (Suggested Value: 112).

3. Security Considerations

This document does not introduce any additional security issues above those identified in [RFC5920], [RFC2205], [RFC3209], and [RFC3473] and [RFC4874].

4. IANA Considerations

This document adds the following new subobject of the existing entry for ERO (20, EXPLICIT_ROUTE):

Value	Description
TBA (suggest value: 68)	Explicit Inclusion Route Subobject (EIRS)

These subobject may be present in the Explicit Route Object, but not in the Route Record Object.

5. Acknowledgments

Authors would like to thank Matt Hartley, Gabriele Maria Galimberti, Luyuan Fang and Walid Wakim for their review comments.

6. References

6.1. Normative References

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

[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.

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

[RFC4874] Lee, CY., Farrel, A., and S. De Cnodder, "Exclude Routes - Extension to Resource ReserVation Protocol-Traffic Engineering (RSVP-TE)", RFC 4874, April 2007.

[RFC6001] Papadimitriou, D., Vigoureux, M., Shiomoto, K., Brungard, D., and JL. Le Roux, "Generalized MPLS (GMPLS) Protocol Extensions for Multi-Layer and Multi-Region Networks (MLN/MRN)", RFC 6001, October 2010.

[RFC3477] Kompella, K. and Y. Rekhter, "Signalling Unnumbered Links in Resource ReSerVation Protocol - Traffic

Engineering (RSVP-TE)", RFC 3477, January 2003.

[DRAFT-LSP-XRO-SUB] Ali, Z., Swallow, G., Filsfils, C., et al, "Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) Route Diversity using Exclude Routes", draft-ali-ccamp-xro-lsp-subobject, work in progress.

6.2. Informative References

[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.

[RFC2209] Braden, R. and L. Zhang, "Resource ReSerVation Protocol (RSVP) -- Version 1 Message Processing Rules", RFC 2209, September 1997.

[RFC5920] Fang, L., Ed., "Security Framework for MPLS and GMPLS Networks", RFC 5920, July 2010.

Authors' Addresses

Zafar Ali
Cisco Systems, Inc.
Email: zali@cisco.com

George Swallow
Cisco Systems, Inc.
swallow@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
cfilsfil@cisco.com

Ori Gerstel
Cisco Systems
ogerstel@cisco.com

Kenji Kumaki
KDDI Corporation
Email: ke-kumaki@kddi.com

Rudiger Kunze
Deutsche Telekom AG
Ruediger.Kunze@telekom.de