Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: February 11, 2016

Pierre Francois
Clarence Filsfils
Ahmed Bashandy
Cisco Systems, Inc.
Bruno Decraene
Stephane Litkowski
Orange

Aug 10, 2015

Topology Independent Fast Reroute using Segment Routing
draft-francois-spring-segment-routing-ti-lfa-02

Abstract

This document presents Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at providing link and node protection of node and adjacency segments within the Segment Routing (SR) framework. This Fast Re-route (FRR) behavior builds on proven IP-FRR concepts being LFAs, remote LFAs (RLFA), and remote LFAs with directed forwarding (DLFA). It extends these concepts to provide guaranteed coverage in any IGP network. We accommodate the FRR discovery and selection approaches in order to establish protection over post-convergence paths from the point of local repair, dramatically reducing the operational need to control the tie-breaks among various FRR options.

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

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."

This Internet-Draft will expire on February 11, 2016.

Copyright Notice

Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (http://trustee.ietf.org/license-info) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1. Introduction 3
2. Terminology 4
3. Intersecting P-Space and Q-Space with post-convergence paths 5
3.1. P-Space property computation for a resource X 5
3.2. Q-Space property computation for a link S-F, over post-convergence paths 5
3.3. Q-Space property computation for a node F, over post-convergence paths 6
4. TI-LFA Repair Tunnel 6
4.1. The repair node is a direct neighbor 6
4.2. The repair node is a PQ node 6
4.3. The repair is a Q node, neighbor of the last P node 7
4.4. Connecting distant P and Q nodes along post-convergence paths 7
5. Protecting segments 7
5.1. The active segment is a node segment 7
5.2. The active segment is an adjacency segment 7
5.2.1. Protecting [Adjacency, Adjacency] segment lists 8
5.2.2. Protecting [Adjacency, Node] segment lists 8
6. References 8
Authors' Addresses 9

1. Introduction

Segment Routing aims at supporting services with tight SLA guarantees [1]. This document provides local repair mechanisms using SR, capable of restoring end-to-end connectivity in the case of a sudden failure of a link or a node, with guaranteed coverage properties.

Using segment routing, there is no need to establish TLDP sessions with remote nodes in order to take advantage of the applicability of remote LFAs (RLFA) or remote LFAs with directed forwarding (DLFA) [2]. As a result, preferring LFAs over RLFAs or DLFAs, as well as minimizing the number of RLFA or DLFA repair nodes is not required. Using SR, there is no need to create state in the network in order to enforce an explicit FRR path. As a result, we can use optimized detour paths for each specific destination and for each possible failure in the network without creating additional forwarding state.

Building on such an easier forwarding environment, the FRR behavior suggested in this document tailors the repair paths over the post-convergence path from the PLR to the protected destination.

As the capacity of the post-convergence path is typically planned by the operator to support the post-convergence routing of the traffic for any expected failure, there is much less need for the operator to tune the decision among which protection path to choose. The protection path will automatically follow the natural backup path that would be used after local convergence. This also helps to reduce the amount of path changes and hence service transients: one transition (pre-convergence to post-convergence) instead of two (pre-convergence to FRR and then post-convergence).

We provide the TI-LFA approach that achieves guaranteed coverage against link or node failure, in any IGP network, relying on the flexibility of SR.

```
              L         ____
         S----------F--{____}--D
        _|_       ___________ /
       {___}--Q--{___________}
```

Figure 1: TI-LFA Protection

We use Figure 1 to illustrate the TI-LFA approach.

The Point of Local Repair (PLR), S, needs to find a node Q (a repair node) that is capable of safely forwarding the traffic to a destination D affected by the failure of the protected link L, or node F. The PLR also needs to find a way to reach Q without being affected by the convergence state of the nodes over the paths it wants to use to reach Q.

In Section 2 we define the main notations used in the document. They are in line with [2].

In Section 3, we suggest to compute the P-Space and Q-Space properties defined in Section 2, for the specific case of nodes lying over the post-convergence paths towards the protected destinations. The failure of a link S-F as well as the failure of a neighbor F is discussed.

Using the properties defined in Section 3, we describe how to compute protection lists that encode a loopfree post-convergence towards the destination, in Section 4.

Finally, we define the segment operations to be applied by the PLR to ensure consistency with the forwarding state of the repair node, in Section 5.

2. Terminology

We define the main notations used in this document as the following.

We refer to "old" and "new" topologies as the LSDB state before and after the considered failure.

SPT_old(R) is the Shortest Path Tree rooted at node R in the initial state of the network.

SPT_new(R, X) is the Shortest Path Tree rooted at node R in the state of the network after the resource X has failed.

Dist_old(A,B) is the distance from node A to node B in SPT_old(A).

Dist_new(A,B, X) is the distance from node A to node B in SPT_new(A, X).

Similarly to [4], we rely on the concept of P-Space and Q-Space for TI-LFA.

The P-Space P(R,X) of a node R w.r.t. a resource X (e.g. a link S-F, or a node F) is the set of nodes that are reachable from R without passing through X. It is the set of nodes that are not downstream of X in SPT_old(R).

The Extended P-Space P'(R,X) of a node R w.r.t. a resource X is the set of nodes that are reachable from R or a neighbor of R, without passing through X.

The Q-Space Q(D,X) of a destination node D w.r.t. a resource X is the set of nodes which do not use X to reach D in the initial state of the network. In other words, it is the set of nodes which have D in their P-Space w.r.t. S-F (or F).

A symmetric network is a network such that the IGP metric of each link is the same in both directions of the link.

3. Intersecting P-Space and Q-Space with post-convergence paths

In this section, we suggest to determine the P-Space and Q-Space properties of the nodes along on the post-convergence paths from the PLR to the protected destination and compute an SR-based explicit path from P to Q when they are not adjacent. Such properties will be used in Section 4 to compute the TI-LFA repair list.

3.1. P-Space property computation for a resource X

A node N is in P(R, X) if it is not downstream of X in SPT_old(R).

A node N is in P'(R,X) if it is not downstream of X in SPT_old(N), for at least one neighbor N of R.

3.2. Q-Space property computation for a link S-F, over post-convergence paths

We want to determine which nodes on the post-convergence from the PLR to the destination D are in the Q-Space of destination D w.r.t. link S-F.

This can be found by intersecting the post-convergence path to D, assuming the failure of S-F, with Q(D, S-F).

The post-convergence path to D requires to compute SPT_new(S, S-F).

A node N is in Q(D,S-F) if it is not downstream of S-F in rSPT_old(D).

3.3. Q-Space property computation for a node F, over post-convergence paths

We want to determine which nodes on the post-convergence from the PLR to the destination D are in the Q-Space of destination D w.r.t. node F.

This can be found by intersecting the post-convergence path to D, assuming the failure of F with Q(D, F).

The post-convergence path to D requires to compute SPT_new(S, F).

A node N is in Q(D,F) if it is not downstream of F in rSPT_old(D).

4. TI-LFA Repair Tunnel

The TI-LFA repair tunnel consists of an outgoing interface and a list of segments (repair list) to insert on the SR header. The repair list encodes the explicit post-convergence path to the destination, which avoids the protected resource X.

The TI-LFA repair tunnel is found by intersecting P(S,X) and Q(D,X) with the post-convergence path to D and computing the explicit SR-based path EP(P, Q) from P to Q when these nodes are not adjacent along the post convergence path. The TI-LFA repair list is expressed generally as (Node_SID(P), EP(P, Q)).

Most often, the TI-LFA repair list has a simpler form, as described in the following sections.

4.1. The repair node is a direct neighbor

When the repair node is a direct neighbor, the outgoing interface is set to that neighbor and the repair segment list is empty.

This is comparable to an LFA FRR repair.

4.2. The repair node is a PQ node

When the repair node is in P(S,X), the repair list is made of a single node segment to the repair node.

This is comparable to an RLFA repair tunnel.

4.3. The repair is a Q node, neighbor of the last P node

When the repair node is adjacent to P(S,X), the repair list is made of two segments: A node segment to the adjacent P node, and an adjacency segment from that node to the repair node.

This is comparable to a DLFA repair tunnel.

4.4. Connecting distant P and Q nodes along post-convergence paths

In some cases, there is no adjacent P and Q node along the post-convergence path. However, the PLR can perform additional computations to compute a list of segments that represent a loopfree path from P to Q.

5. Protecting segments

In this section, we explain how a protecting router S processes the active segment of a packet upon the failure of its primary outgoing interface.

The behavior depends on the type of active segment to be protected.

5.1. The active segment is a node segment

The active segment is kept on the SR header, unchanged (1). The repair list is inserted at the head of the list. The active segment becomes the first segment of the inserted repair list.

A future version of the document will describe the FRR behavior when the active segment is a node segment destined to F, and F has failed.

Note (1): If the SRGB at the repair node is different from the SRGB at the PLR, then the active segment must be updated to fit the SRGB of the repair node.

5.2. The active segment is an adjacency segment

We define hereafter the FRR behavior applied by S for any packet received with an active adjacency segment S-F for which protection was enabled. We distinguish the case where this active segment is followed by another adjacency segment from the case where it is followed by a node segment.

5.2.1. Protecting [Adjacency, Adjacency] segment lists

If the next segment in the list is an Adjacency segment, then the packet has to be conveyed to F.

To do so, S applies a "NEXT" operation on Adj(S-F) and then two consecutive "PUSH" operations: first it pushes a node segment for F, and then it pushes a protection list allowing to reach F while bypassing S-F.

Upon failure of S-F, a packet reaching S with a segment list matching [adj(S-F),adj(M),...] will thus leave S with a segment list matching [RT(F),node(F),adj(M)], where RT(F) is the repair tunnel for destination F.

5.2.2. Protecting [Adjacency, Node] segment lists

If the next segment in the stack is a node segment, say for node T, the packet segment list matches [adj(S-F),node(T),...].

A first solution would consist in steering the packet back to F while avoiding S-F, similarly to the previous case. To do so, S applies a "NEXT" operation on Adj(S-F) and then two consecutive "PUSH" operations: first it pushes a node segment for F, and then it pushes a repair list allowing to reach F while bypassing S-F.

Upon failure of S-F, a packet reaching S with a segment list matching [adj(S-F),node(T),...] will thus leave S with a segment list matching [RT(F),node(F),node(T)].

Another solution is to not steer the packet back via F but rather follow the new shortest path to T. In this case, S just needs to apply a "NEXT" operation on the Adjacency segment related to S-F, and push a repair list redirecting the traffic to a node Q, whose path to node segment T is not affected by the failure.

Upon failure of S-F, packets reaching S with a segment list matching [adj(L), node(T), ...], would leave S with a segment list matching [RT(Q),node(T), ...].

6. References

[1] Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", draft-ietf-spring-segment-routing-04 (work in progress), July 2015.

[2] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, January 2010.

[3] Filsfils, C., Francois, P., Shand, M., Decraene, B., Uttaro, J., Leymann, N., and M. Horneffer, "Loop-Free Alternate (LFA) Applicability in Service Provider (SP) Networks", RFC 6571, June 2012.

[4] Bryant, S., Filsfils, C., Previdi, S., Shand, M., and N. So, "Remote Loop-Free Alternate (LFA) Fast Reroute (FRR)", RFC 7490, DOI 10.17487/RFC7490, April 2015, <http://www.rfc-editor.org/info/rfc7490>.

Authors' Addresses

Pierre Francois
Cisco Systems, Inc.
Vimercate
IT

Email: pifranco@cisco.com

Clarence Filsfils
Cisco Systems, Inc.
Brussels
BE

Email: cfilsfil@cisco.com

Ahmed Bashandy
Cisco Systems, Inc.
San Jose
US

Email: bashandy@cisco.com

Bruno Decraene
Orange
Issy-les-Moulineaux
FR

Email: bruno.decraene@orange.com

Stephane Litkowski
Orange
FR

Email: bruno.decraene@orange.com