Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 2020

S. Litkowski
Cisco
A. Bashandy
Individual
C. Filsfils
Cisco Systems
B. Decraene
Orange
P. Francois
INSA Lyon
D. Voyer
Bell Canada
F. Clad
P. Camarillo
Cisco Systems

January 18, 2020

Topology Independent Fast Reroute using Segment Routing
draft-ietf-rtgwg-segment-routing-ti-lfa-02

Abstract

This document presents Topology Independent Loop-free Alternate Fast Re-route (TI-LFA), aimed at providing 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. A key aspect of TI-LFA is the FRR path selection approach establishing protection over the expected 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 https://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 July 18, 2020.

Table of Contents

1. Introduction

Segment Routing aims at supporting services with tight SLA guarantees [RFC8402]. By relying on SR this document provides a local repair mechanism for standard IGP shortest path capable of restoring end-to-end connectivity in the case of a sudden directly connected failure of a network component. Non-SR mechanisms for local repair are beyond the scope of this document. Non-local failures are addressed in a separate document [I-D.bashandy-rtgwg-segment-routing-uloop].

The term topology independent (TI) refers to the ability to provide a loop free backup path irrespective of the topologies used in the network. This provides a major improvement compared to LFA ([RFC5286]) and remote LFA ([RFC7490]) which cannot be applicable in some topologies ([RFC6571]).

For each destination in the network, TI-LFA pre-installs a backup forwarding entry for each protected destination ready to be activated upon detection of the failure of a link used to reach the destination. TI-LFA provides protection in the event of any one of the following: single link failure, single node failure, or single SRLG failure. In link failure mode, the destination is protected assuming the failure of the link. In node protection mode, the destination is protected assuming that the neighbor connected to the primary link has failed. In SRLG protecting mode, the destination is protected assuming that a configured set of links sharing fate with the primary link has failed (e.g. a linecard or a set of links sharing a common transmission pipe).

Protection techniques outlined in this document are limited to protecting links, nodes, and SRLGs that are within a routing domain. Protecting domain exit routers and/or links attached to another routing domains are beyond the scope of this document

Thanks to SR, TI-LFA does not require the establishment of TLDP sessions with remote nodes in order to take advantage of the applicability of remote LFAs (RLFA) [RFC7490][RFC7916] or remote LFAs with directed forwarding (DLFA)[RFC5714]. All the Segment Identifiers (SIDs) are available in the link state database (LSDB) of the IGP. As a result, preferring LFAs over RLFAs or DLFAs, as well as minimizing the number of RLFA or DLFA repair nodes is not required anymore.

Thanks to SR, there is no need to create state in the network in order to enforce an explicit FRR path. This relieves the nodes themselves from having to maintain extra state, and it relieves the operator from having to deploy an extra protocol or extra protocol sessions just to enhance the protection coverage.

[RFC7916] raised several operational considerations when using LFA or remote LFA. [RFC7916] Section 3 presents a case where a high bandwidth link between two core routers is protected through a PE router connected with low bandwidth links. In such a case, congestion may happen when the FRR backup path is activated. [RFC7916] introduces a local policy framework to let the operator tuning manually the best alternate election based on its own requirements.

From a network capacity planning point of view, it is often assumed that if a link L fails on a particular node X, the bandwidth consumed on L will be spread over some of the remaining links of X. The remaining links to be used are determined by the IGP routing considering that the link L has failed (we assume that the traffic uses the post-convergence path starting from the node X). In Figure 1, we consider a network with all metrics equal to 1 except the metrics on links used by PE1, PE2 and PE3 which are 1000. An easy network capacity planning method is to consider that if the link L (X-B) fails, the traffic actually flowing through L will be spread over the remaining links of X (X-H, X-D, X-A). Considering the IGP metrics, only X-H and X-D can only be used in reality to carry the traffic flowing through the link L. As a consequence, the bandwidth of links X-H and X-D is sized according to this rule. We should observe that this capacity planning policy works, however it is not

fully accurate.

In Figure 1, considering that the source of traffic is only from PE1 and PE4, when the link L fails, depending on the convergence speed of the nodes, X may reroute its forwarding entries to the remote PEs onto X-H or X-D; however in a similar timeframe, PE1 will also reroute a subset of its traffic (the subset destined to PE2) out of its nominal path reducing the quantity of traffic received by X. The capacity planning rule presented previously has the drawback of oversizing the network, however it allows to prevent any transient congestion (when for example X reroutes traffic before PE1 does).

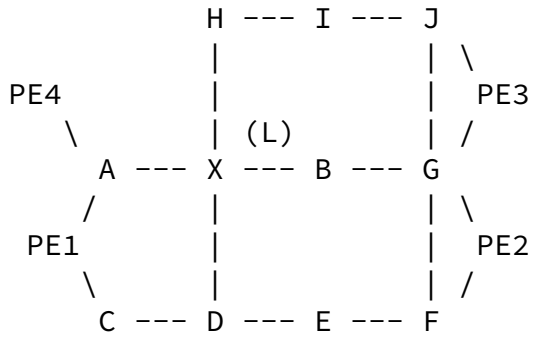

Figure 1

Based on this assumption, in order to facilitate the operation of FRR, and limit the implementation of local FRR policies, it looks interesting to steer the traffic onto the post-convergence path from the PLR point of view during the FRR phase. In our example, when link L fails, X switches the traffic destined to PE3 and PE2 on the post-convergence paths. This is perfectly inline with the capacity planning rule that was presented before and also inline with the fact X may converge before PE1 (or any other upstream router) and may spread the X-B traffic onto the post-convergence paths rooted at X.

It should be noted, that some networks may have a different capacity planning rule, leading to an allocation of less bandwidth on X-H and

X-D links. In such a case, using the post-convergence paths rooted at X during FRR may introduce some congestion on X-H and X-D links. However it is important to note, that a transient congestion may possibly happen, even without FRR activated, for instance when X converges before the upstream routers. Operators are still free to use the policy framework defined in [RFC7916] if the usage of the post-convergence paths rooted at the PLR is not suitable.

Readers should be aware that FRR protection is pre-computing a backup path to protect against a particular type of failure (link, node, SRLG). When using the post-convergence path as FRR backup path, the computed post-convergence path is the one considering the failure we are protecting against. This means that FRR is using an expected post-convergence path, and this expected post-convergence path may be actually different from the post-convergence path used if the failure that happened is different from the failure FRR was protecting against. As an example, if the operator has implemented a protection against a node failure, the expected post-convergence path used during FRR will be the one considering that the node has failed. However, even if a single link is failing or a set of links is failing (instead of the full node), the node-protecting post-convergence path will be used. The consequence is that the path used during FRR is not optimal with respect to the failure that has actually occurred.

Another consideration to take into account is: while using the expected post-convergence path for SR traffic using node segments only (for instance, PE to PE traffic using shortest path) has some advantages, these advantages reduce when SR policies ([I-D.ietf-spring-segment-routing-policy]) are involved. A segment-list used in an SR policy is computed to obey a set of path constraints defined locally at the head-end or centrally in a controller. TI-LFA cannot be aware of such path constraints and there is no reason to expect the TI-LFA backup path protecting one the segments in that segment list to obey those constraints. When SR policies are used and the operator wants to have a backup path which still follows the policy requirements, this backup path should be computed as part of the SR policy in the ingress node (or central controller) and the SR policy should not rely on local protection. Another option could be to use FlexAlgo ([I-D.ietf-lsr-flex-algo]) to express the set of constraints and use a single node segment associated with a FlexAlgo to reach the destination. When using a node segment associated with a FlexAlgo, TI-LFA keeps providing an

optimal backup by applying the appropriate set of constraints. The relationship between TI-LFA and the SR-algorithm is detailed in Section 6.

Thanks to SR and the combination of Adjacency segments and Node segments, the expression of the expected post-convergence path rooted at the PLR is facilitated and does not create any additional state on intermediate nodes. The easiest way to express the expected post-convergence path in a loop-free manner is to encode it as a list of adjacency segments. However, in an MPLS world, this may create a long stack of labels to be pushed that some hardware may not be able to push. One of the challenges of TI-LFA is to encode the expected post-convergence path by combining adjacency segments and node segments. Each implementation will be free to have its own path compression optimization algorithm. This document details the basic concepts that could be used to build the SR backup path as well as the associated dataplane procedures.

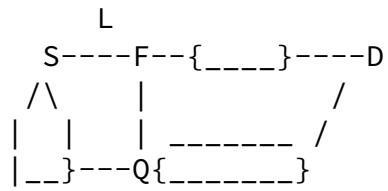

Figure 2 TI-LFA Protection

We use Figure 2 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, a set of links including L (SRLG), or the node F itself. 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: the PLR needs a loop-free path to reach Q.

Section 2 defines the main notations used in the document. They are in line with [RFC5714].

Section 3 suggests 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.

Using the properties defined in Section 3, Section 4 describes how to compute protection lists that encode a loop-free post-convergence path towards the destination.

Section 5 defines the segment operations to be applied by the PLR to ensure consistency with the forwarding state of the repair node.

By applying the algorithms specified in this document to actual service providers and large enterprise networks, we provide real life measurements for the number of SIDs used by repair paths. Section 8 summarizes these measurements.

1.1. Conventions used in this document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when and only when, they appear in all capitals, as shown here.

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.

PLR stands for "Point of Local Repair". It is the router that applies fast traffic restoration after detecting failure in a directly attached link, set of links, and/or node.

Similar to [RFC7490], we use 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, a node F, or a SRLG) 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, F, or a set of links adjacent to S).

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

One of the challenges of defining an SR path following the expected post-convergence path is to reduce the size of the segment list. In order to reduce this segment list, an implementation MAY determine the P-Space/Extended P-Space and Q-Space properties (defined in [RFC7490]) of the nodes along the expected post-convergence path 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). X can be a link, a node, or a set of links adjacent to the PLR. 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 path 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).

3.3. Q-Space property computation for a set of links adjacent to S, over post-convergence paths

We want to determine which nodes on the post-convergence path from the PLR to the destination D are in the Q-Space of destination D w.r.t. a set of links adjacent to S (S being the PLR). That is, we aim to find the set of nodes on the post-convergence path that use none of the members of the protected set of links, to reach D.

This can be found by intersecting the post-convergence path to D, assuming the failure of the set of links, with the intersection among Q(D, S->X) for all S->X belonging to the set of links.

3.4. 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).

3.5. Scaling considerations when computing Q-Space

[RFC7490] raises scaling concerns about computing a Q-Space per destination. Similar concerns may affect TI-LFA computation if an implementation tries to compute a reverse SPT for every destination in the network to determine the Q-Space. It will be up to each implementation to determine the good tradeoff between scaling and accuracy of the optimization.

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 and, at the same time, is guaranteed to be loop-free irrespective of the state of FIBs along the nodes belonging to the explicit path. Thus there is no need for any co-ordination or message exchange between the PLR and any other router in the network.

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. Section 8 provides statistics for the number of SIDs in the explicit path to protect against various failures.

4.1. FRR path using a direct neighbor

When a direct neighbor is in P(S,X) and Q(D,x) and on the post-convergence path, the outgoing interface is set to that neighbor and the repair segment list MUST be empty.

This is comparable to a post-convergence LFA FRR repair.

4.2. FRR path using a PQ node

When a remote node R is in P(S,X) and Q(D,x) and on the post-convergence path, the repair list MUST be made of a single node segment to R and the outgoing interface MUST be set to the outgoing interface used to reach R.

This is comparable to a post-convergence RLFA repair tunnel.

4.3. FRR path using a P node and Q node that are adjacent

When a node P is in P(S,X) and a node Q is in Q(D,x) and both are on the post-convergence path and both are adjacent to each other, the repair list MUST be made of two segments: A node segment to P (to be processed first), followed by an adjacency segment from P to Q.

This is comparable to a post-convergence 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 loop-free path from P to Q. How these computations are done is out of scope of this document.

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 for the packet, S-F.

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

5.1. The active segment is a node segment

The active segment MUST be kept on the SR header unchanged and the repair list MUST be inserted at the head of the list. The active segment becomes the first segment of the inserted repair list.

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on the tail-end of the active segment, and if the active segment has been signalled with penultimate hop popping, the active segment MUST be popped before pushing the repair list.

- If the SRGB at the Q node is different from the SRGB at the PLR, then the active segment (before the insertion of the repair list) MUST be updated to fit the SRGB of the Q node.

In Section 5.3, we describe the node protection behavior of PLR S, for the specific case where the active segment is a prefix segment for the neighbor F itself.

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. As protection has been enabled for the segment S-F and signalled in the IGP, any SR policy using this segment knows that it may be transiently rerouted out of S-F in case of S-F failure.

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 MUST apply 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. For details on the "NEXT" and "PUSH" operations, refer to [RFC8402].

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

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on F, and if the node segment of F has been signalled with penultimate hop popping, the implementation MUST pop Adj(S-F) and then push the repair list (the node segment of F is not pushed). The packet will leave S with a segment list matching [RT(F),adj(F-M)].

- If the SRGB at the Q node is different from the SRGB at the PLR, then MPLS label representing node(F) MUST be calculated as per the SRGB of the Q node.

In Section 5.3.2, we describe the TI-LFA behavior of PLR S when node protection is applied and the two first segments are Adjacency Segments.

5.2.2. Protecting [Adjacency, Node] segment lists

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

A first solution would consist in steering the packet back to F while avoiding S-F. To do so, S MUST apply 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)].

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on F, and if the node segment of F has been signalled with penultimate hop popping, the implementation MUST pop Adj(S-F) and then push the repair list (the node segment of F is not pushed). The packet will leave S with a segment list matching [RT(F),node(T)].

- If the SRGB at the Q node is different from the SRGB at the PLR, then MPLS label representing node(F) MUST be calculated as per the SRGB of the Q node.

Another solution is to not steer the packet back via F but rather follow the new shortest path to T. In this case, S MUST apply a "NEXT" operation on the Adjacency segment related to S-F, followed by a "PUSH" of 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(S-F), node(T), ...], would leave S with a segment list matching [RT(Q),node(T), ...]. Note that this second behavior is the one followed for node protection, as described in Section 5.3.1.

This behavior is slightly modified when SR-MPLS is used:

- If the repair list ends with an adjacency segment terminating on T (T being the Q node), and if the node segment of T has been signalled with penultimate hop popping, the implementation MUST pop Adj(S-F) and then push the repair list (the node segment of T is not pushed). The packet will leave S with a segment list matching [RT(Q=T), ...].

- If the SRGB at the Q node is different from the SRGB at the PLR, then the MPLS label representing node(T) MUST be calculated as per the SRGB of the Q node.

The first proposal which merges back the traffic at the remote end of the adjacency segment has the advantage of keeping as much as possible the traffic on the existing path. As stated in Section 1, when SR policies are involved and a strict compliance of the policy is required, an end-to-end protection should be preferred over a local repair mechanism.

5.3. Protecting SR policy midpoints against node failure

In this section, we describe the behavior of a node S configured to interpret the failure of link S->F as the node failure of F, in the specific case where the active segment of the packet received by S is a Prefix SID of F represented as "F"), or an Adjacency SID for the link S-F (represented as "S->F").

5.3.1. Protecting {F, T, D} or {S->F, T, D}

This section describes the protection behavior of S when all of the following conditions are true:

1. the active segment is a prefix SID for a neighbor F, or an adjacency segment S->F

2. the primary interface used to forward the packet failed

3. the segment following the active segment is a prefix SID (for node T)

4. node protection is active for that interface.

In such a case, the PLR MUST:

1. apply a NEXT operation; the segment F or S->F is removed

2. Confirm that the next segment is in the SRGB of F, meaning that the next segment is a prefix segment, e.g. for node T

3. Retrieve the segment ID of T (as per the SRGB of F)

4. Apply a NEXT operation followed by a PUSH operation of T's segment based on the SRGB of node S.

5. Look up T's segment (based on the updated label value) and forward accordingly.

5.3.2. Protecting {F, F->T, D} or {S->F, F->T, D}

This section describes the protection behavior of S when all of the following conditions are true:

1. the active segment is a prefix SID for a neighbor F, or an adjacency segment S->F

2. the primary interface used to forward the packet failed

3. the segment following the active segment is an adjacency SID (F->T)

4. node protection is active for that interface.

In such a case, the PLR MUST:

1. Apply a NEXT operation; the segment F or S->F is removed

2. Confirm that the next segment is an adjacency SID of F, say F->T

3. Retrieve the node segment ID associated to T (as per the set of Adjacency Segments of F)

4. Apply a NEXT operation on the next segment followed by a PUSH of T's segment based on the SRGB of the node S.

5. Look up T's segment (based on the updated label value) and forward accordingly.

It is noteworthy to mention that node "S" in the procedures described in Sections 5.3.1 and 5.3.2 can always determine whether the segment after popping the top segment is an adjacency SID or a prefix-SID of the next-hop "F" as follows:

1. In a link state environment, the node "S" knows the SRGB and the adj-SIDs of the neighboring node "F"

2. If the new segment after popping the top segment is within the SRGB or the adj-SIDs of "F", then node "S" is certain that the failure of node "F" is a midpoint failure and hence node "S" applies the procedures specified in Sections 5.3.1 or 5.3.2, respectively.

3. Otherwise the failure is not a midpoint failure and hence the node "S" may apply other protection techniques that are beyond the scope of this document or simply drop the packet and wait for normal protocol convergence.

6. TI-LFA and SR Algorithms

SR allows an operator to bind an algorithm to a prefix SID (as defined in [RFC8402]. The algorithm value dictates how the path to the prefix is computed. The SR default algorithm is known has the "Shortest Path" algorithm. The SR default algorithm allows an operator to override the IGP shortest path by using local policies. When TI-LFA uses Node-SIDs associated with the default algorithm, there is no guarantee that the path will be loop-free as a local policy may have overriden the expected IGP path. As the local policies are defined by the operator, it becomes the responsibility of this operator to ensure that the deployed policies do not affect the TI-LFA deployment. It should be noted that such situation can already happen today with existing mechanisms as remote LFA.

When a Node-SID is associated with the SR default algorithm, enforcing TI-LFA to use Node-SIDs associated with a strict SPF algorithm is a definitive solution to this problem.

[I-D.ietf-lsr-flex-algo] defines a flexible algorithm (FlexAlgo) framework to be associated with Prefix SIDs. FlexAlgo allows a user to associate a constrained path to a Prefix SID rather than using the regular IGP shortest path. An implementation MAY support TI-LFA to protect Node-SIDs associated to a FlexAlgo. In such a case, rather than computing the expected post-convergence path based on the regular SPF, an implementation SHOULD use the constrained SPF algorithm bound to the FlexAlgo instead of the regular Dijkstra in all the SPF/rSPF computations that are occurring during the TI-LFA computation. This includes the computation of the P-Space and Q-Space as well as the post-convergence path.

7. Usage of Adjacency segments in the repair list

The repair list of segments computed by TI-LFA may contain one or more adjacency segments. An adjacency segment may be protected or not protected.

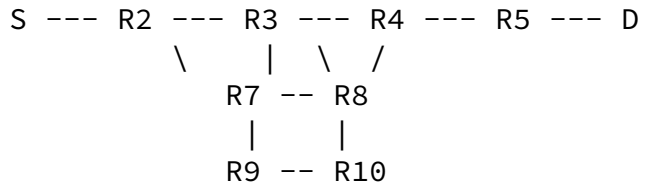

Figure 3

In Figure 3, all the metrics are equal to 1 except R2-R7,R7-R8,R8-R4,R7-R9 which have a metric of 1000. Considering R2 as a PLR to protect against the failure of node R3 for the traffic S->D, the repair list computed by R2 will be [adj(R7-R8),adj(R8-R4)] and the outgoing interface will be to R7. If R3 fails, R2 pushes the repair list onto the incoming packet to D. During the FRR, if R7-R8 fails and if TI-LFA has picked a protected adjacency segment for adj(R7-R8), R7 will push an additional repair list onto the packet following the procedures defined in Section 5.

To avoid the possibility of this double FRR, an implementation of TI-LFA MAY pick only non protected adjacency segments when building the repair list.

8. Measurements on Real Networks

This section presents measurements performed on real service provider and large enterprise networks. The objective of the measurements is to assess the number of SIDs required in an explicit path when the mechanisms described in this document are used to protect against the failure scenarios within the scope of this document. The number of segments described in this section are applicable to instantiating segment routing over the MPLS forwarding plane.

The measurements below indicate that for link and local SRLG protection, a 1 SID repair path delivers more than 99% coverage. For node protection a 2 SIDs repair path yields 99% coverage.

Table 1 below lists the characteristics of the networks used in our measurements. The measurements are carried out as follows

- For each network, the algorithms described in this document are applied to protect all prefixes against link, node, and local SRLG failure

- For each prefix, the number of SIDs used by the repair path is recored

- The percentage of number of SIDs are listed in Tables 2A/B, 3A/B, and 4A/B

The measurements listed in the tables indicate that for link and local SRLG protection, 1 SID repair paths are sufficient to protect more than 99% of the prefix in almost all cases. For node protection 2 SIDs repair paths yield 99% coverage.

Network	Nodes	Circuits	Node-to-Link Ratio	SRLG info?
T1	408	665	1 : 63	Yes
T2	587	1083	1 : 84	No
T3	93	401	4 : 31	Yes
T4	247	393	1 : 59	Yes
T5	34	96	2 : 82	Yes
T6	50	78	1 : 56	No
T7	82	293	3 : 57	No
T8	35	41	1 : 17	Yes
T9	177	1371	7 : 74	Yes

Table 1: Data Set Definition

The rest of this section presents the measurements done on the actual topologies. The convention that we use is as follows

- 0 SIDs: the calculated repair path starts with a directly connected neighbor that is also a loop free alternate, in which case there is no need to explicitly route the traffic using additional SIDs. This scenario is described in Section 4.1.

- 1 SIDs: the repair node is a PQ node, in which case only 1 SID is needed to guarantee loop-freeness. This scenario is covered in Section 4.2.

- 2 or more SIDs: The repair path consists of 2 or more SIDs as described in Sections 4.3 and 4.4. We do not cover the case for 2 SIDs (Section 4.3) separately because there was no granularity in the result. Also we treat the node-SID+adj-SID and node-SID + node-SID the same because they do not differ from the data plane point of view.

Table 2A and 2B below summarize the measurements on the number of SIDs needed for link protection

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.227%	25.256%	0.517%	0.001%
T2	81.097%	18.738%	0.165%	0.0%
T3	95.878%	4.067%	0.056%	0.0%
T4	62.547%	35.666%	1.788%	0.0%
T5	85.733%	14.267%	0.0%	0.0%
T6	81.252%	18.714%	0.033%	0.0%
T7	98,857%	1.143%	0.0%	0.0%
T8	94,118%	5.882%	0.0%	0.0%
T9	98.950%	1.050%	0.0%	0.0%

Table 2A: Link protection (repair size distribution)

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.227%	99.482%	99.999%	100.0%
T2	81.097%	99.835%	100.0%	100.0%
T3	95.878%	99.944%	100.0%	100.0%
T4	62.547%	98.212%	100.0%	100.0%
T5	85.733%	100.000%	100.0%	100.0%
T6	81.252%	99.967%	100.0%	100.0%
T7	98,857%	100.000%	100.0%	100.0%
T8	94,118%	100.000%	100.0%	100.0%
T9	98,950%	100.000%	100.0%	100.0%

Table 2B: Link protection repair size cumulative distribution

Table 3A and 3B summarize the measurements on the number of SIDs needed for local SRLG protection.

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.177%	25.306%	0.517%	0.001%
T2	No SRLG Information			
T3	93.650%	6.301%	0.049%	0.0%
T4	62,547%	35.666%	1.788%	0.0%
T5	83.139%	16.861%	0.0%	0.0%
T6	No SRLG Information			

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.177%	99.482%	99.999%	100.001%
T2	No SRLG Information			
T3	93.650%	99.951%	100.000%	0.0%
T4	62,547%	98.212%	100.000%	100.0%
T5	83.139%	100.000%	100.0%	100.0%
T6	No SRLG Information			
T7	No SRLG Information			
T8	85.185%	14.815%	0.0%	0.0%
T9	98,940%	1.060%	0.0%	0.0%

Table 3A: Local SRLG protection repair size distribution

Network	0 SIDs	1 SID	2 SIDs	3 SIDs
T1	74.177%	99.482%	99.999%	100.001%
T2	No SRLG Information			
T3	93.650%	99.951%	100.000%	0.0%
T4	62,547%	98.212%	100.000%	100.0%
T5	83.139%	100.000%	100.0%	100.0%
T6	No SRLG Information			
T7	No SRLG Information			
T8	85.185%	100,000%	100.000%	100.0%
T9	98,940%	100,000%	100.000%	100.0%

Table 3B: Local SRLG protection repair size Cumulative distribution

The remaining two tables summarize the measurements on the number of SIDs needed for node protection.

Network	0 SIDs	1 SID	2 SIDs	3 SIDs	4 SIDs
T1	49.771%	47.902%	2.156%	0.148%	0.023%
T2	36,528%	59.625%	3.628%	0.194%	0.025%
T3	73,287%	25,574%	1,128%	0.010%	0%

T4	36.112%	57.350%	6.329%	0.199%	0.010%
T5	73.185%	26.815%	0%	0%	0%
T6	78.362%	21.320%	0.318%	0%	0%
T7	66.106%	32.813%	1.082%	0%	0%
T8	59.712%	40.288%	0%	0%	0%
T9	98.950%	1.050%	0%	0%	0%

Table 4A: Node protection (repair size distribution)

Network	0 SIDs	1 SID	2 SIDs	3 SIDs	4 SIDs
T1	49.771%	97.673%	99.829%	99.977%	100%
T2	36,528%	96.153%	99.781%	99.975%	100%
T3	73,287%	98.862%	99.990%	100.0%	100%
T4	36.112%	93.461%	99.791%	99.990%	100%
T5	73.185%	100.0%	100.0%	100.0%	100%
T6	78.362%	99.682%	100.0%	100.0%	100%
T7	66.106%	98,918%	100.0%	100.0%	100%
T8	59.712%	100.0%	100.0%	100.0%	100%
T9	98.950%	100.0%	100.0%	100.0%	100%

Table 4B: Node protection (repair size cumulative distribution)

9. Security Considerations

The techniques described in this document are internal functionalities to a router that result in the ability to guarantee an upper bound on the time taken to restore traffic flow upon the failure of a directly connected link or node. As these techniques steer traffic to the post-convergence path as quickly as possible, this serves to minimize the disruption associated with a local failure which can be seen as a modest security enhancement. The protection mechanisms does not protect external destinations, but rather provides quick restoration for destination that are internal to a routing domain.

10. IANA Considerations

No requirements for IANA

11. Conclusions

This document proposes a mechanism that is able to pre-calculate a backup path for every primary path so as to be able to protect

against the failure of a directly connected link, node, or SRLG. The mechanism is able to calculate the backup path irrespective of the topology as long as the topology is sufficiently redundant.

12. Acknowledgments

We would like to thank Les Ginsberg, Stewart Bryant, Alexander Vainsthein, Chris Bowers for their valuable comments.

This document was prepared using 2-Word-v2.0.template.dot.

13. References

13.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC7916] Litkowski, S., Ed., Decraene, B., Filsfils, C., Raza, K., Horneffer, M., and P. Sarkar, "Operational Management of

Loop-Free Alternates", RFC 7916, DOI 10.17487/RFC7916, July 2016, <https://www.rfc-editor.org/info/rfc7916>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[RFC8402] Filsfils, C., Previdi, S., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402 July 2018, <http://www.rfc-editor.org/info/rfc8402>.

13.2. Informative References

[I-D.bashandy-rtgwg-segment-routing-uloop] Bashandy, A., Filsfils, C., Litkowski, S., Decraene, B., Francois, P., and Psenak, P. " Loop avoidance using Segment Routing", draft-bashandy-rtgwg-segment-routing-uloop-07, (work in progress), January 2020

[I-D.ietf-lsr-flex-algo] Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", draft-ietf-lsr-flex-algo-05 (work in progress), November 2019.

[I-D.ietf-spring-segment-routing-policy] Filsfils, C., Sivabalan, S., daniel.voyer@bell.ca, d., bogdanov@google.com, b., and P. Mattes, "Segment Routing Policy Architecture", draft-ietf-spring-segment-routing-policy-06 (work in progress), December 2019.

[RFC5286] Atlas, A., Ed. and A. Zinin, Ed., "Basic Specification for IP Fast Reroute: Loop-Free Alternates", RFC 5286, DOI 10.17487/RFC5286, September 2008, <https://www.rfc-editor.org/info/rfc5286>.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", RFC 5714, DOI 10.17487/RFC5714 January 2010, <http://www.rfc-editor.org/info/rfc5714>.

[RFC6571] 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, DOI 10.17487/RFC6571 June 2012, <http://www.rfc-editor.org/info/rfc6571>.

[RFC7490] 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

Stephane Litkowski
Cisco
Email: slitkows.ietf@gmail.com

Ahmed Bashandy
Individual
Email: abashandy.ietf@gmail.com

Clarence Filsfils
Cisco Systems

Brussels
Belgium
Email: cfilsfil@cisco.com

Bruno Decraene
Orange
Issy-les-Moulineaux
France
Email: bruno.decraene@orange.com

Pierre Francois
INSA Lyon
Email: pierre.francois@insa-lyon.fr

Daniel Voyer
Bell Canada
Canada
Email: daniel.voyer@bell.ca

Francois Clad
Cisco Systems
Email: fclad@cisco.com

Pablo Camarillo
Cisco Systems
Email: pcamaril@cisco.com