

Network Working Group  
Internet-Draft  
Intended status: Standards Track  
Expires: May 26, 2014

S. Bryant  
C. Filsfils  
S. Previdi  
Cisco Systems  
M. Shand  
Independent Contributor  
N. So  
Tata Communications  
November 22, 2013

Remote LFA FRR  
draft-ietf-rtgwg-remote-lfa-04

## Abstract

This draft describes an extension to the basic IP fast re-route mechanism described in [RFC5286](#) that provides additional backup connectivity for link failures when none can be provided by the basic mechanisms.

## Requirements Language

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](#) [[RFC2119](#)].

## 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 May 26, 2014.

## Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

Internet-Draft

Remote LFA FRR

November 2013

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

<a href="#">1.</a>	<a href="#">Terminology</a>	<a href="#">2</a>
<a href="#">2.</a>	<a href="#">Introduction</a>	<a href="#">3</a>
<a href="#">3.</a>	<a href="#">Repair Paths</a>	<a href="#">5</a>
<a href="#">3.1.</a>	<a href="#">Tunnels as Repair Paths</a>	<a href="#">5</a>
<a href="#">3.2.</a>	<a href="#">Tunnel Requirements</a>	<a href="#">5</a>
<a href="#">4.</a>	<a href="#">Construction of Repair Paths</a>	<a href="#">6</a>
<a href="#">4.1.</a>	<a href="#">Identifying Required Tunneled Repair Paths</a>	<a href="#">6</a>
<a href="#">4.2.</a>	<a href="#">Determining Tunnel End Points</a>	<a href="#">6</a>
<a href="#">4.2.1.</a>	<a href="#">Computing Repair Paths</a>	<a href="#">7</a>
<a href="#">4.2.2.</a>	<a href="#">Selecting Repair Paths</a>	<a href="#">9</a>
<a href="#">4.3.</a>	<a href="#">A Cost Based RLFA Algorithm</a>	<a href="#">10</a>
<a href="#">5.</a>	<a href="#">Example Application of Remote LFAs</a>	<a href="#">14</a>
<a href="#">6.</a>	<a href="#">Node Failures</a>	<a href="#">15</a>
<a href="#">7.</a>	<a href="#">Operation in an LDP environment</a>	<a href="#">16</a>
<a href="#">8.</a>	<a href="#">Analysis of Real World Topologies</a>	<a href="#">17</a>
<a href="#">8.1.</a>	<a href="#">Topology Details</a>	<a href="#">17</a>
<a href="#">8.2.</a>	<a href="#">LFA only</a>	<a href="#">18</a>
<a href="#">8.3.</a>	<a href="#">RLFA</a>	<a href="#">18</a>
<a href="#">8.4.</a>	<a href="#">Comparison of LFA an RLFA results</a>	<a href="#">20</a>
<a href="#">9.</a>	<a href="#">Management Considerations</a>	<a href="#">21</a>
<a href="#">10.</a>	<a href="#">Historical Note</a>	<a href="#">21</a>
<a href="#">11.</a>	<a href="#">IANA Considerations</a>	<a href="#">21</a>
<a href="#">12.</a>	<a href="#">Security Considerations</a>	<a href="#">21</a>
<a href="#">13.</a>	<a href="#">Acknowledgments</a>	<a href="#">22</a>
<a href="#">14.</a>	<a href="#">Informative References</a>	<a href="#">22</a>
	<a href="#">Authors' Addresses</a>	<a href="#">24</a>

## [1.](#) Terminology

This draft uses the terms defined in [\[RFC5714\]](#). This section defines additional terms used in this draft.

The union of the P-space of the neighbours of a specific router with respect to the protected link (see [Section 4.2.1.2](#)).

**P-space** P-space is the set of routers reachable from a specific router without any path (including equal cost path splits) transiting the protected link.

For example, the P-space of S, is the set of routers that S can reach without using the protected link S-E.

**PQ node** A node which is a member of both the (extended) P-space and the Q-space. In remote LFA this is used as the repair tunnel endpoint.

**Q-space** Q-space is the set of routers from which a specific router can be reached without any path (including equal cost path splits) transiting the protected link.

**Repair tunnel** A tunnel established for the purpose of providing a virtual neighbor which is a Loop Free Alternate.

**Remote LFA (RLFA)** The use of a PQ node rather than a neighbour of the repairing node as the next hop in an LFA repair.

In this document we use the notation X-Y to mean the path from X to Y over the link directly connecting X and Y, whilst the notation X->Y refers to the shortest path from X to Y via some set of unspecified nodes including the null set (i.e. including over a link directly connecting X and Y).

## [2.](#) Introduction

[RFC 5714](#) [[RFC5714](#)] describes a framework for IP Fast Re-route and provides a summary of various proposed IPFRR solutions. A basic mechanism using loop-free alternates (LFAs) is described in [[RFC5286](#)]

that provides good repair coverage in many topologies[RFC6571], especially those that are highly meshed. However, some topologies, notably ring based topologies are not well protected by LFAs alone. This is illustrated in Figure 1 below.

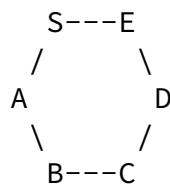


Figure 1: A simple ring topology

If all link costs are equal, the link S-E cannot be fully protected by LFAs. The destination C is an ECMP from S, and so can be protected when S-E fails, but D and E are not protectable using LFAs

This draft describes extensions to the basic repair mechanism in which tunnels are used to provide additional logical links which can then be used as loop free alternates where none exist in the original topology. For example if a tunnel is provided between S and C as shown in Figure 2 then C, now being a direct neighbor of S would become an LFA for D and E. The non-failure traffic distribution is not disrupted by the provision of such a tunnel since it is only used for repair traffic and MUST NOT be used for normal traffic.

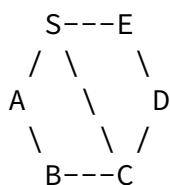


Figure 2: The addition of a tunnel

The use of this technique is not restricted to ring based topologies, but is a general mechanism which can be used to enhance the protection provided by LFAs. A study of the protection achieved using remote LFA in typical service provider core networks is provided in [Section 8](#), and a side by side comparison between LFA and remote LFA is provided in [Section 8.4](#).

Remote LFA is suitable for incremental deployment within a network, including a network that is already deploying LFA. Computation of the repair path is relatively simple, and takes place exclusively on the repairing node. In MPLS networks the targeted LDP protocol needed to learn the label binding at the repair tunnel endpoint is a well understood and widely deployed technology.

This technique described in this document is directed at providing repairs in the case of link failures. Considerations regarding node failures are discussed in [Section 6](#).

### [3.](#) Repair Paths

As with LFA FRR, when a router detects an adjacent link failure, it uses one or more repair paths in place of the failed link. Repair paths are pre-computed in anticipation of later failures so they can be promptly activated when a failure is detected.

A tunneled repair path tunnels traffic to some staging point in the network from which it is assumed that, in the absence of multiple failures, it will travel to its destination using normal forwarding without looping back. This is equivalent to providing a virtual loop-free alternate to supplement the physical loop-free alternates. Hence the name "Remote LFA FRR". In its simplest form, when a link cannot be entirely protected with local LFA neighbors, the protecting router seeks the help of a remote LFA staging point. Network manageability considerations may lead to a repair strategy that uses a remote LFA more frequently [[I-D.ietf-rtgwg-lfa-manageability](#)].]

#### [3.1.](#) Tunnels as Repair Paths

Consider an arbitrary protected link S-E. In LFA FRR, if a path to the destination from a neighbor N of S does not cause a packet to

loop back over the link S-E (i.e. N is a loop-free alternate), then S can send the packet to N and the packet will be delivered to the destination using the pre-failure forwarding information. If there is no such LFA neighbor, then S may be able to create a virtual LFA by using a tunnel to carry the packet to a point in the network which is not a direct neighbor of S from which the packet will be delivered to the destination without looping back to S. In this document such a tunnel is termed a repair tunnel. The tail-end of this tunnel (the repair tunnel endpoint) is a "PQ node" and the repair mechanism is a "remote LFA".

Note that the repair tunnel terminates at some intermediate router between S and E, and not E itself. This is clearly the case, since if it were possible to construct a tunnel from S to E then a conventional LFA would have been sufficient to effect the repair.

### 3.2. Tunnel Requirements

There are a number of IP in IP tunnel mechanisms that may be used to fulfil the requirements of this design, such as IP-in-IP [[RFC1853](#)] and GRE[RFC1701] .

In an MPLS enabled network using LDP[RFC5036], a simple label stack[RFC3032] may be used to provide the required repair tunnel. In this case the outer label is S's neighbor's label for the repair tunnel end point, and the inner label is the repair tunnel end

point's label for the packet destination. In order for S to obtain the correct inner label it is necessary to establish a directed LDP session[RFC5036] to the tunnel end point.

The selection of the specific tunnelling mechanism (and any necessary enhancements) used to provide a repair path is outside the scope of this document. The deployment in an MPLS/LDP environment is relatively simple in the data plane as an LDP LSP from S to the repair tunnel endpoint (the selected PQ node) is readily available, and hence does not require any new protocol extension or design change. This LSP is automatically established as a basic property of LDP behavior. The performance of the encapsulation and decapsulation is efficient as encapsulation is just a push of one label (like conventional MPLS TE FRR) and the decapsulation occurs naturally at the penultimate hop before the repair tunnel endpoint. In the

control plane, a targeted LDP (TLDP) session is needed between the repairing node and the repair tunnel endpoint, which will need to be established and the labels processed before the tunnel can be used. The time to establish the TLDP session and acquire labels will limit the speed at which a new tunnel can be put into service, but this will not be a problem in normal operation.

When a failure is detected, it is necessary to immediately redirect traffic to the repair path. Consequently, the repair tunnel used must be provisioned beforehand in anticipation of the failure. Since the location of the repair tunnels is dynamically determined it is necessary to automatically establish the repair tunnels. Multiple repairs may share a tunnel end point

#### [4.](#) Construction of Repair Paths

##### [4.1.](#) Identifying Required Tunneled Repair Paths

Not all links will require protection using a tunneled repair path. Referring to Figure 1, if E can already be protected via an LFA, S-E does not need to be protected using a repair tunnel, since all destinations normally reachable through E must therefore also be protectable by an LFA. Such an LFA is frequently termed a "link LFA". Tunneled repair paths are only required for links which do not have a link LFA.

It should be noted that using the Q-space of E as a proxy for the Q-space of each destination can result in failing to identify valid remote LFAs. The extent to which this reduces the effective protection coverage is topology dependent.

##### [4.2.](#) Determining Tunnel End Points

The repair tunnel endpoint needs to be a node in the network reachable from S without traversing S-E. In addition, the repair tunnel end point needs to be a node from which packets will normally flow towards their destination without being attracted back to the failed link S-E.

Note that once released from the tunnel, the packet will be forwarded, as normal, on the shortest path from the release point to

its destination. This may result in the packet traversing the router E at the far end of the protected link S-E., but this is obviously not required.

The properties that are required of repair tunnel end points are therefore:

- o The repair tunneled point MUST be reachable from the tunnel source without traversing the failed link; and
- o When released, tunneled packets MUST proceed towards their destination without being attracted back over the failed link.

Provided both these requirements are met, packets forwarded over the repair tunnel will reach their destination and will not loop.

In some topologies it will not be possible to find a repair tunnel endpoint that exhibits both the required properties. For example if the ring topology illustrated in Figure 1 had a cost of 4 for the link B-C, while the remaining links were cost 1, then it would not be possible to establish a tunnel from S to C (without resorting to some form of source routing).

#### [4.2.1.](#) Computing Repair Paths

To compute the repair path for link S-E we need to determine the set of routers which can be reached from S without traversing S-E, and match this with the set of routers from which the node E can be reached, by normal forwarding, without traversing the link S-E.

We will proceed as follows: we will describe how to compute the set of routers which can be reached from S without traversing S-E. We call this the S's P-space with respect to the failure of link S-E. We will then note that S is able to use P-Space of its neighbours since S can determine which neighbour it will use as the next hop for the repair. We call this the S's Extended P-Space with respect to the failure of link S-E. The use of extended P-Space allows greater repair coverage and is the preferred approach. Finally we will show how to compute the set of routers from which the node E can be reached, by normal forwarding, without traversing the link S-E. This

is called the Q-space of E with respect to the link S-E. The



selection of the preferred node from the set of nodes that are in both Extended P-Space and Q-Space is described in [Section 4.2.2](#).

A suitable cost based algorithm to compute the set of nodes common to both extended P-space and Q-space is provided in [Section 4.3](#).

#### [4.2.1.1](#). P-space

The set of routers which can be reached from S without traversing S-E is termed the P-space of S with respect to the link S-E. The P-space can be obtained by computing a shortest path tree (SPT) rooted at S and excising the sub-tree reached via the link S-E (including those which are members of an ECMP). In the case of Figure 1 the P-space comprises nodes A and B only. Expressed in cost terms the set of routers {P} are those for which the shortest path cost S->P is strictly less than the shortest path cost S->E->P.

#### [4.2.1.2](#). Extended P-space

The description in [Section 4.2.1.1](#) calculated router S's P-space rooted at S itself. However, since router S will only use a repair path when it has detected the failure of the link S-E, the initial hop of the repair path need not be subject to S's normal forwarding decision process. Thus we introduce the concept of extended P-space. Router S's extended P-space is the union of the P-spaces of each of S's neighbours (N). This may be calculated by computing an SPT at each of S's neighbors (excluding E) and excising the subtree reached via the path N->S->E. The use of extended P-space may allow router S to reach potential repair tunnel end points that were otherwise unreachable. In cost terms a router (P) is in extended P-space if the shortest path cost N->P is strictly less than the shortest path cost N->S->E->P. In other words, once the packet is forced to N by S, it is lower cost for it to continue on to P by any path except one that takes it back to S and then across the S->E link.

Since in the case of Figure 1 node A is a per-prefix LFA for the destination node C, the set of extended P-space nodes comprises nodes A, B and C. Since node C is also in E's Q-space, there is now a node common to both extended P-space and Q-space which can be used as a repair tunnel end-point to protect the link S-E.

#### [4.2.1.3](#). Q-space

The set of routers from which the node E can be reached, by normal forwarding, without traversing the link S-E is termed the Q-space of E with respect to the link S-E. The Q-space can be obtained by computing a reverse shortest path tree (rSPT) rooted at E, with the

sub-tree which traverses the failed link excised (including those which are members of an ECMP). The rSPT uses the cost towards the root rather than from it and yields the best paths towards the root from other nodes in the network. In the case of Figure 1 the Q-space comprises nodes C and D only. Expressed in cost terms the set of routers {Q} are those for which the shortest path cost Q->E is strictly less than the shortest path cost Q->S->E. In Figure 1 the intersection of the E's Q-space with S's P-space defines the set of viable repair tunnel end-points, known as "PQ nodes". As can be seen, for the case of Figure 1 there is no common node and hence no viable repair tunnel end-point.

Note that the Q-space calculation could be conducted for each individual destination and a per-destination repair tunnel end point determined. However this would, in the worst case, require an SPF computation per destination which is not currently considered to be scalable. We therefore use the Q-space of E as a proxy for the Q-space of each destination. This approximation is obviously correct since the repair is only used for the set of destinations which were, prior to the failure, routed through node E. This is analogous to the use of link-LFAs rather than per-prefix LFAs.

#### [4.2.2.](#) Selecting Repair Paths

The mechanisms described above will identify all the possible repair tunnel end points that can be used to protect a particular link. In a well-connected network there are likely to be multiple possible release points for each protected link. All will deliver the packets correctly so, arguably, it does not matter which is chosen. However, one repair tunnel end point may be preferred over the others on the basis of path cost or some other selection criteria.

There is no technical requirement for the selection criteria to be consistent across all routers, but such consistency may be desirable from an operational point of view. In general there are advantages in choosing the repair tunnel end point closest (shortest metric) to S. Choosing the closest maximises the opportunity for the traffic to be load balanced once it has been released from the tunnel. For consistency in behavior, is RECOMMENDED that member of the set of routers {PQ} with the lowest cost S->P be the default choice for P. In the event of a tie the router with the lowest node identifier SHOULD be selected.

It is a local matter whether the repair path selection policy used by the router favours LFA repairs over RLFA repairs. An LFA repair has the advantage of not requiring the use of tunnel, however network

manageability considerations may lead to a repair strategy that uses a remote LFA more frequently [[I-D.ietf-rtgwg-lfa-manageability](#)].

As described in [[RFC5286](#)], always selecting a PQ node that is downstream with respect to the repairing node, prevents the formation of loops when the failure is worse than expected. The use of downstream nodes reduces the repair coverage, and operators are advised to determine whether adequate coverage is achieved before enabling this selection feature.

#### [4.3.](#) A Cost Based RLFA Algorithm

The preceding text has mostly described the computation of the remote LFA repair target (PQ) in terms of the intersection of two reachability graphs computed using SPF. This section describes a method of computing the remote LFA repair target using a cost based algorithm. The pseudo-code provided in this section avoids unnecessary SPF computations, but for the sake of readability, it does not otherwise try to optimize the code. In this description  $D_{opt}(a,b)$  is the shortest distance from node a to node b as computed by the SPF.

The following notation is used:

- o  $D_{opt}(a,b)$  is the shortest distance from node a to node b as computed by the SPF.
- o `dest` is the packet destination
- o `fail_intf` is the failed interface (S-E in the example)
- o `fail_intf.remote_node` is the node reachable over interface `fail_intf` (node E in the example)
- o `intf.remote_node` is the node reachable over interface `intf`
- o `root` is the root of the SPF calculation
- o `self` is the node carrying out the computation
- o `y` is the node in the network under consideration

```
////////////////////////////////////  
//  
//    Main Function
```

```
////////////////////////////////////  
//  
// We have already computed the forward SPF from self to all nodes
```

```
// y in network and thus we know D_opt (self, y). This is needed  
// for normal forwarding.  
// However for completeness.
```

```
Compute_and_Store_Forward_SPF(self)
```

```
// To extend P-space we compute the SPF at each neighbour except  
// the neighbour that is reached via the link being protected.  
// We will also need D_opt(fail_intf.remote_node,y) so compute  
// that at the same time.
```

```
Compute_Neighbor_SPFs()
```

```
// Compute the set of nodes {P} reachable other than via the  
// failed link
```

```
Compute_Extended_P_Space(fail_intf)
```

```
// Compute the set of nodes that can reach the node on the far  
// side of the failed link without traversing the failed link.
```

```
Compute_Q_Space(fail_intf)
```

```
// Compute the set of candidate RLFA tunnel endpoints
```

```
Intersect_Extended_P_and_Q_Space()
```

```
// Make sure that we cannot get looping repairs when the  
// failure is worse than expected.
```

```
if (guarantee_no_looping_on_worse_than_protected_failure)  
    Apply_Downstream_Constraint()
```

```

//
// End of Main Function
//
/////////////////////////////////////////////////////////////////

/////////////////////////////////////////////////////////////////
//
// Procedures
//

/////////////////////////////////////////////////////////////////
//
// This computes the SPF from root, and stores the optimum
// distance from root to each node y

```

```

Compute_and_Store_Forward_SPF(root)
    Compute_Forward_SPF(root)
    foreach node y in network
        store D_opt(root,y)

/////////////////////////////////////////////////////////////////
//
// This computes the optimum distance from each neighbour (other
// than the neighbour reachable through the failed link) and
// every other node in the network

Compute_Neighbor_SPFs()
    foreach interface intf in self
        Compute_and_Store_Forward_SPF(intf.remote_node)

/////////////////////////////////////////////////////////////////
//
// The reverse SPF computes the cost from each remote node to
// root. This is achieved by running the normal SPF algorithm,
// but using the link cost in the direction from the next hop
// back towards root in place of the link cost in the direction
// away from root towards the next hop.

```

```

Compute_and_Store_Reverse_SPF(root)
  Compute_Reverse_SPF(root)
  foreach node y in network
    store D_opt(y,root)

```

```

////////////////////////////////////
//
// Calculate extended P-space
//
// Note the strictly less than operator is needed to
// avoid ECMP issues.

```

```

Compute_Extended_P_Space(fail_intf)
  foreach node y in network
    y.in_extended_P_space = false
    // Extend P-space to the P-spaces of all reachable
    // neighbours
    foreach interface intf in self
      if (intf.remote_node != fail_intf.remote_node)
        if ( D_opt(intf.remote_node, y) <
            D_opt(intf.remote_node, self) +
            D_opt(self,fail_intf.remote_node) +

```

```

        D_opt(fail_intf.remote_node,y) )
        y.in_extended_P_space = true

```

```

////////////////////////////////////
//
// Compute the nodes in Q-space
//

```

```

Compute_Q_Space(fail_intf)
  // Compute the cost from every node the network to the
  // node normally reachable across the failed link
  Compute_and_Store_Reverse_SPF(fail_intf.remote_node)

  // Compute the cost from every node the network to self
  Compute_and_Store_Reverse_SPF(self)

  foreach node y in network

```

```

        if ( D_opt(y,fail_intf.remote_node) < D_opt(y,self) +
            D_opt(self,fail_intf.remote_node) )
            y.in_Q_space = true
        else
            y.in_Q_space = false

////////////////////////////////////
//
// Compute set of nodes in both extended P-space and in Q-space

Intersect_Extended_P_and_Q_Space()
    foreach node y in network
        if ( y.in_extended_P_space && y.in_Q_space )
            y.valid_tunnel_endpoint = true
        else
            y.valid_tunnel_endpoint = false

////////////////////////////////////
//
// A downstream route is one where the next hop is strictly
// closer to the destination. By sending the packet to a
// PQ node that is downstream, we know that if the PQ node
// detects a failure, it will not loop the packet back to self.
// This is useful when there are two failures, or a node has
// failed rather than a link.

Apply_Downstream_Constraint()
    foreach node y in network

```

```

        if (y.valid_tunnel_endpoint)
            Compute_and_Store_Forward_SPF(y)
            if ((D_opt(y,dest) < D_opt(self,dest))
                y.valid_tunnel_endpoint = true
            else
                y.valid_tunnel_endpoint = false

//
////////////////////////////////////

```

## 5. Example Application of Remote LFAs

An example of a commonly deployed topology which is not fully protected by LFAs alone is shown in Figure 3. PE1 and PE2 are connected in the same site. P1 and P2 may be geographically separated (inter-site). In order to guarantee the lowest latency path from/to all other remote PEs, normally the shortest path follows the geographical distance of the site locations. Therefore, to ensure this, a lower IGP metric (5) is assigned between PE1 and PE2. A high metric (1000) is set on the P-PE links to prevent the PEs being used for transit traffic. The PEs are not individually dual-homed in order to reduce costs.

This is a common topology in SP networks.

When a failure occurs on the link between PE1 and P2, PE1 does not have an LFA for traffic reachable via P1. Similarly, by symmetry, if the link between PE2 and P1 fails, PE2 does not have an LFA for traffic reachable via P2.

Increasing the metric between PE1 and PE2 to allow the LFA would impact the normal traffic performance by potentially increasing the latency.

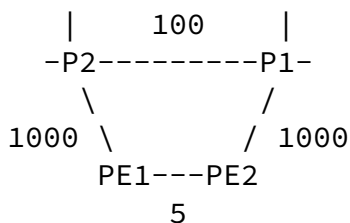


Figure 3: Example SP topology

Clearly, full protection can be provided, using the techniques described in this draft, by PE1 choosing P1 as the remote LFA repair target node, and PE2 choosing P2 as the remote LFA repair target.



## 6. Node Failures

When the failure is a node failure rather than a link failure there is a danger that the RLFA repair will loop. This is discussed in detail in [[I-D.bryant-ipfrr-tunnels](#)]. In summary problem is that two of more of E's neighbors each with E as the next hop to some destination D may attempt to repair a packet addressed to destination D via the other neighbor and then E, thus causing a loop to form. As will be noted from [[I-D.bryant-ipfrr-tunnels](#)], this can rapidly become a complex problem to address.

There are a number of ways to minimize the probability of a loop forming when a node failure occurs and there exists the possibility that two of E's neighbors may form a mutual repair.

1. Detect when a packet has arrived on some interface I that is also the interface used to reach the first hop on the RLFA path to the remote LFA repair target, and drop the packet. This is useful in the case of a ring topology.
2. Require that the path from the remote LFA repair target to destination D never passes through E (including in the ECMP case), i.e. only use node protecting paths in which the cost from the remote LFA repair target to D is strictly less than the cost from the remote LFA repair target to E plus the cost E to D.
3. Require that where the packet may pass through another neighbor of E, that node is down stream (i.e. strictly closer to D than the repairing node). This means that some neighbor of E (X) can repair via some other neighbor of E (Y), but Y cannot repair via X.

Case 1 accepts that loops may form and suppresses them by dropping packets. Dropping packets may be considered less detrimental than looping packets. This approach may also lead to dropping some legitimate packets. Cases 2 and 3 above prevent the formation of a loop, but at the expense of a reduced repair coverage and at the cost of additional complexity in the algorithm to compute the repair path.

The probability of a node failure and the consequences of node failure in any particular topology will depend on the node design, the particular topology in use, and node failure strategy (including the null strategy). It is recommended that a network operator perform an analysis of the consequences and probability of node

failure in their network, and determine whether the incidence and consequence of occurrence are acceptable.

This topic is further discussed in [\[I-D.psarkar-rtgwg-rlfa-node-protection\]](#).

## 7. Operation in an LDP environment

Where this technique is used in an MPLS network using LDP [\[RFC5036\]](#), and S is a transit node, S will need to swap the top label in the stack for the remote LFA repair target's (PQ's) label to the destination, and to then push its own label for the remote LFA repair target.

In the example [Section 2](#) S already has the first hop (A) label for the remote LFA repair target (C) as a result of the ordinary operation of LDP. To get the remote LFA repair target's label (C's label) for the destination (D), S needs to establish a targeted LDP session with C. The label stack for normal operation and RLFA operation is shown below in Figure 4.



X = Normal label stack packet arriving at S

Y = Normal label stack packet leaving S

Z = RLFA label stack to D via C as the remote LFA repair target.

Figure 4

To establish an targeted LDP session with a candidate remote LFA repair target node the repairing node (S) needs to know what IP address that the remote LFA repair target is willing to use for targeted LDP sessions. Ideally this is provided by the remote LFA repair target advertising this address in the IGP in use. Which address is used, how this is advertised in the IGP, and whether this is a special IP address or an IP address also used for some other purpose is out of scope for this document and must be specified in an IGP specific RFC.

In the absence of a protocol to learn the preferred IP address for targeted LDP, an LSR should attempt a targeted LDP session with the Router ID [[RFC2328](#)] [[RFC5305](#)] [[RFC5340](#)], unless it is configured otherwise.

No protection is available until the TLDP session has been established and a label for the destination has been learned from the remote LFA repair target. If for any reason the TLDP session cannot not be established, an implementation SHOULD advise the operator about the protection setup issue using any well known mechanism such as Syslog [[RFC5424](#)] or SNMP [[RFC3411](#)].

## [8.](#) Analysis of Real World Topologies

This section gives the results of analysing a number of real world service provider topologies collected between October 2012 and the date of this draft.

### [8.1.](#) Topology Details

The figure below characterises each topology (topo) studied in terms of :

- o The number of nodes (# nodes) excluding pseudonodes.
- o The number of bidirectional links ( # links) including parallel links and links to and from pseudonodes.
- o The number of node pairs that are connected by one or more links (# pairs).

- o The number of node pairs that are connected by more than one (i.e. parallel) link ( # para).
- o The number of links (excluding pseudonode links, which are by definition asymmetric) that have asymmetric metrics (#asym).

topo	# nodes	# links	# pairs	# para	# asym
1	315	570	560	10	3
2	158	373	312	33	0
3	655	1768	1314	275	1195
4	1281	2326	2248	70	10
5	364	811	659	80	86
6	114	318	197	101	4
7	55	237	159	67	2
8	779	1848	1441	199	437
9	263	482	413	41	12
10	86	375	145	64	22
11	162	1083	351	201	49
12	380	1174	763	231	0
13	1051	2087	2037	48	64
14	92	291	204	64	2

## 8.2. LFA only

The figure below shows the percentage of protected destinations (% prot) and percentage of guaranteed node protected destinations ( % gtd N) for the set of topologies characterized in [Section 8.1](#) achieved using only LFA repairs.

topo	% prot	% gtd N
1	78.5	36.9
2	97.3	52.4
3	99.3	58
4	83.1	63.1

	5		99		59.1	
	6		86.4		21.4	
	7		93.9		35.4	
	8		95.3		48.1	
	9		82.2		49.5	
	10		98.5		14.9	
	11		99.6		24.8	
	12		99.5		62.4	
	13		92.4		51.6	
	14		99.3		48.6	
+-----+-----+-----+						

### [8.3.](#) RLFA

The figure below shows the percentage of protected destinations (% prot) and % guaranteed node protected destinations ( % gtd N) for RLFA protection in the topologies studies. In addition, it show the percentage of destinations using an RLFA repair (% PQ) together with the total number of unidirectional RLFA targeted LDP session established (# PQ), the number of PQ sessions which would be required for complete protection, but which could not be established (no PQ). It also shows the 50 (p50), 90 (p90) and 100 (p100) percentiles for the number of individual LDP sessions terminating at an individual node (whether used for TX, RX or both).

For example, if there were LDP sessions required A->B, A->C, C->A, C->D, these would be counted as 2, 1, 2, 1 at nodes A,B,C and D respectively because:-

A has two sessions (to nodes B and C)

B has one session (to node A)

C has two sessions (to nodes A and D)

D has one session (to node D)

In this study, remote LFA is only used when necessary. i.e. when there is at least one destination which is not reparable by a per destination LFA, and a single remote LFA tunnel is used (if

available) to repair traffic to all such destinations. The remote LFA repair target points are computed using extended P space and choosing the PQ node which has the lowest metric cost from the repairing node.

topo	% prot	% gtd N	% PQ	# PQ	no PQ	p50	p90	p100
1	99.7	53.3	21.2	295	3	1	5	14
2	97.5	52.4	0.2	7	40	0	0	2
3	99.999	58.4	0.7	63	5	0	1	5
4	99	74.8	16	1424	54	1	3	23
5	99.5	59.5	0.5	151	7	0	2	7
6	100	34.9	13.6	63	0	1	2	6
7	99.999	40.6	6.1	16	2	0	2	4
8	99.5	50.2	4.3	350	39	0	2	15
9	99.5	55	17.3	428	5	1	2	67
10	99.6	14.1	1	49	7	1	2	5
11	99.9	24.9	0.3	85	1	0	2	8
12	99.999	62.8	0.5	512	4	0	0	3
13	97.5	54.6	5.1	1188	95	0	2	27

Bryant, et al.

Expires May 26, 2014

[Page 19]

Internet-Draft

Remote LFA FRR

November 2013

14	100	48.6	0.7	79	0	0	2	4
----	-----	------	-----	----	---	---	---	---

Another study[ISOCORE2010] confirms the significant coverage increase provided by Remote LFAs.

#### 8.4. Comparison of LFA an RLFA results

The table below provides a side by side comparison the LFA and the remote LFA results. This shows a significant improvement in the percentage of protected destinations and normally a modest improvement in the percentage of guaranteed node protected destinations.

topo	LFA	RLFA	LFA	RLFA
------	-----	------	-----	------

	% prot	%prot	% gtd N	% gtd N
1	78.5	99.7	36.9	53.3
2	97.3	97.5	52.4	52.4
3	99.3	99.999	58	58.4
4	83.1	99	63.1	74.8
5	99	99.5	59.1	59.5
6	86.4	100	21.4	34.9
7	93.9	99.999	35.4	40.6
8	95.3	99.5	48.1	50.2
9	82.2	99.5	49.5	55
10	98.5	99.6	14.9	14.1
11	99.6	99.9	24.8	24.9
12	99.5	99.999	62.4	62.8
13	92.4	97.5	51.6	54.6
14	99.3	100	48.6	48.6

As shown in the table, remote LFA provides close to 100% prefix protection against link failure in 11 of the 14 topologies studied, and provides a significant improvement in two of the remaining three cases. In an MPLS network, this is achieved without any scalability impact, as the tunnels to the PQ nodes are always present as a property of an LDP-based deployment. In the very few cases where P and Q spaces have an empty intersection, one could select the closest node in the Q space and signal an explicitly-routed RSVP TE LSP to that Q node. A directed LDP session is then established with the selected Q node and the rest of the solution is identical to that

described elsewhere in this document. Alternatively the segment routing technology being defined in the IETF may be used to carry the traffic between non-collocated P and Q nodes  
[\[I-D.filsfils-rtgwg-segment-routing-use-cases\]](#),  
[\[I-D.filsfils-rtgwg-segment-routing\]](#),  
[\[I-D.gredler-rtgwg-igp-label-advertisement\]](#).

## 9. Management Considerations

The management of LFA and remote LFA is the subject of ongoing work within the IETF[\[I-D.ietf-rtgwg-lfa-manageability\]](#) to which the reader is referred. Management considerations may lead to a

preference for the use of a remote LFA over an available LFA. This preference is a matter for the network operator, and not a matter of protocol correctness.

## 10. Historical Note

The basic concepts behind Remote LFA were invented in 2002 and were later included in [[I-D.bryant-ipfrr-tunnels](#)], submitted in 2004.

[[I-D.bryant-ipfrr-tunnels](#)], targeted a 100% protection coverage and hence included additional mechanisms on top of the Remote LFA concept. The addition of these mechanisms made the proposal very complex and computationally intensive and it was therefore not pursued as a working group item.

As explained in [[RFC6571](#)], the purpose of the LFA FRR technology is not to provide coverage at any cost. A solution for this already exists with MPLS TE FRR. MPLS TE FRR is a mature technology which is able to provide protection in any topology thanks to the explicit routing capability of MPLS TE.

The purpose of LFA FRR technology is to provide for a simple FRR solution when such a solution is possible. The first step along this simplicity approach was "local" LFA [[RFC5286](#)]. We propose "Remote LFA" as a natural second step. The following section motivates its benefits in terms of simplicity, incremental deployment and significant coverage increase.

## 11. IANA Considerations

There are no IANA considerations that arise from this architectural description of IPFRR. The RFC Editor may remove this section on publication.

## 12. Security Considerations

The security considerations of [RFC 5286](#) also apply.

To prevent their use as an attack vector the repair tunnel endpoints SHOULD be assigned from a set of addresses that are not reachable from outside the routing domain.



### 13. Acknowledgments

The authors wish to thank Levente Csikor and Chris Bowers for their contribution to the cost based algorithm text. We thank Stephane Litkowski for his review of this work.

### 14. Informative References

[I-D.bryant-ipfrr-tunnels]

Bryant, S., Filsfils, C., Previdi, S., and M. Shand, "IP Fast Reroute using tunnels", [draft-bryant-ipfrr-tunnels-03](#) (work in progress), November 2007.

[I-D.filsfils-rtgwg-segment-routing-use-cases]

Filsfils, C., Francois, P., Previdi, S., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., Kini, S., and E. Crabbe, "Segment Routing Use Cases", [draft-filsfils-rtgwg-segment-routing-use-cases-02](#) (work in progress), October 2013.

[I-D.filsfils-rtgwg-segment-routing]

Filsfils, C., Previdi, S., Bashandy, A., Decraene, B., Litkowski, S., Horneffer, M., Milojevic, I., Shakir, R., Ytti, S., Henderickx, W., Tantsura, J., and E. Crabbe, "Segment Routing Architecture", [draft-filsfils-rtgwg-segment-routing-01](#) (work in progress), October 2013.

[I-D.gredler-rtgwg-igp-label-advertisement]

Gredler, H., Amante, S., Scholl, T., and L. Jalil, "Advertising MPLS labels in IGPs", [draft-gredler-rtgwg-igp-label-advertisement-05](#) (work in progress), May 2013.

[I-D.ietf-rtgwg-lfa-manageability]

Litkowski, S., Decraene, B., Filsfils, C., and K. Raza, "Operational management of Loop Free Alternates", [draft-ietf-rtgwg-lfa-manageability-00](#) (work in progress), May 2013.

[I-D.psarkar-rtgwg-rlfa-node-protection]

psarkar@juniper.net, p., Gredler, H., Hegde, S., Raghuveer, H., cbowers@juniper.net, c., and S. Litkowski,

"Remote-LFA Node Protection and Manageability", [draft-psarkar-rtgwg-rlfa-node-protection-02](#) (work in progress), November 2013.

[ISOCORE2010]

So, N., Lin, T., and C. Chen, "LFA (Loop Free Alternates) Case Studies in Verizon's LDP Network", 2010.

[RFC1701] Hanks, S., Li, T., Farinacci, D., and P. Traina, "Generic Routing Encapsulation (GRE)", [RFC 1701](#), October 1994.

[RFC1853] Simpson, W., "IP in IP Tunneling", [RFC 1853](#), October 1995.

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

[RFC2328] Moy, J., "OSPF Version 2", STD 54, [RFC 2328](#), April 1998.

[RFC3032] Rosen, E., Tappan, D., Fedorkow, G., Rekhter, Y., Farinacci, D., Li, T., and A. Conta, "MPLS Label Stack Encoding", [RFC 3032](#), January 2001.

[RFC3411] Harrington, D., Presuhn, R., and B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks", STD 62, [RFC 3411](#), December 2002.

[RFC5036] Andersson, L., Minei, I., and B. Thomas, "LDP Specification", [RFC 5036](#), October 2007.

[RFC5286] Atlas, A. and A. Zinin, "Basic Specification for IP Fast Reroute: Loop-Free Alternates", [RFC 5286](#), September 2008.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), October 2008.

[RFC5340] Coltun, R., Ferguson, D., Moy, J., and A. Lindem, "OSPF for IPv6", [RFC 5340](#), July 2008.

[RFC5424] Gerhards, R., "The Syslog Protocol", [RFC 5424](#), March 2009.

[RFC5714] Shand, M. and S. Bryant, "IP Fast Reroute Framework", [RFC 5714](#), January 2010.

[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](#), June 2012.

Internet-Draft

Remote LFA FRR

November 2013

#### Authors' Addresses

Stewart Bryant  
Cisco Systems  
250, Longwater, Green Park,  
Reading RG2 6GB, UK  
UK

Email: [stbryant@cisco.com](mailto:stbryant@cisco.com)

Clarence Filsfils  
Cisco Systems  
De Kleetlaan 6a  
1831 Diegem  
Belgium

Email: [cfilsfil@cisco.com](mailto:cfilsfil@cisco.com)

Stefano Previdi  
Cisco Systems

Email: [sprevidi@cisco.com](mailto:sprevidi@cisco.com)

Mike Shand  
Independent Contributor

Email: [imc.shand@gmail.com](mailto:imc.shand@gmail.com)

Ning So  
Tata Communications  
Mobile Broadband Services

Email: [Ning.So@tatacommunications.com](mailto:Ning.So@tatacommunications.com)

Bryant, et al.

Expires May 26, 2014

[Page 24]