



## A Survey on IP Fast Rerouting Schemes using Backup Topology

Vimal S Pal  
M.TECH (CSE), CBIT,  
Hyderabad, India.

Y Rama Devi  
Professor, Department of CSE, CBIT,  
Hyderabad, India.

**Abstract:** Paper describes the different IP fast rerouting schemes using backup topology. This paper describes the scheme which comes under complete IPFRR frame work as according to IETF (Internet engineering task force). The first section gives the basic idea about the IPFRR used for the rerouting purpose. The second section deals about the different methods used under IPFRR with concepts and examples. The third section describes the basic three configuration scenario for IPFRR. The fourth section gives the conclusion on the basis of the comparative theoretical studies made in the above sections and tries to find the best for rerouting. The conclusion suggests that any of the single scheme is not capable of providing full coverage as a result it shows that to obtain a full coverage of 100% these schemes must be used in combination with the other schemes.

**Index Terms:** IP Fast Reroute, Congestion Avoidance, Special Node, Backup Topology.

### 1. Introduction

General connectionless networks contain no mechanisms to establish disjoint end to-end paths as described for connection-oriented networks. Recovery schemes in Connectionless networks presently are IP re-convergence and IP Fast reroute. The overall reason for the long time-scale of IP re-convergence is the fact that it is a reactive and global process. To obtain fast reroute a scheme must be proactive and local, which means that backup routing information must be installed in advance and that the rerouting is performed locally without any failure notifications. IPFRR provides fast reroute capabilities using pure IP (non-MPLS) protocols, such as OSPF and IS-IS. IPFRR can be enabled on one or more routers, after which it calculates one LFA backup path for every prefix. If there is a failure and a router cannot forward packets on the required outbound interface, it can switch quickly, before reconvergence to an LFA interface. The IPFRR enabled interface ensures that the packets rejoin the original route downstream from the failure. If the rerouting rejoins the original route at the remote node of the protected interface, the LFA provides circuit protection. If it rejoins further downstream than the remote node, then the LFA provides node failure protection. The LFAs on the interfaces, as well as the routes over those interfaces, are based on the network topology. The topology could result in LFAs being available to protect all routes over some interfaces, to protect only some routes, or to protect none at all. With fast rerouting, packets can be rerouted to alternative routes in a time-scale that may help on the performance of real-time applications. Another key contribution of such schemes is the ability to suppress IP re-convergence under transient failures, and hence prevent instability and potentially reduce the amount of micro-loops. Micro-loops may also occur during a transition from backup routing to original routing when the transient failure is repaired, however the frequency may be reduced.

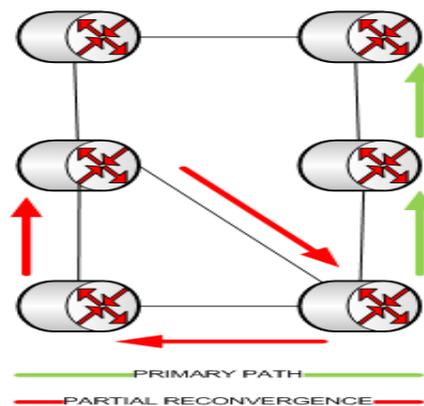


Fig 1.LFA problem

When using IPFRR, only routers adjacent to the failed link or node know anything about the failure at all. Other routers use their normal routes for packet forwarding, which may cause problems if a router passes a packet back from where it has received it, believing that there is a usable path leading thitherto. Therefore, IPFRR techniques must ensure that no forwarding loops can be formed based on such misbelieves.

The next sections in this paper describe the different algorithms used under IPFRR.

## 2 IPFRR METHODS

### 2.1 Equal Cost Multiple Path (ECMP)

One of the oldest and simplest IP Fast ReRoute techniques is Equal Cost Multiple Path (ECMP) [7], which is an extension enabled in the majority of today's networks. ECMP is usable in those cases when more than one (different) shortest paths are available towards a destination. The traffic is distributed equally among the paths by default, offering increased bandwidth. Additionally, when a failure occurs on one path, routers balance traffic among the remaining routes. ECMP is easy to implement, but it works only when multiple paths of equal cost are available between the source and destination. ECMP is easy to implement, but it works only when multiple paths of equal cost are available between the source and destination

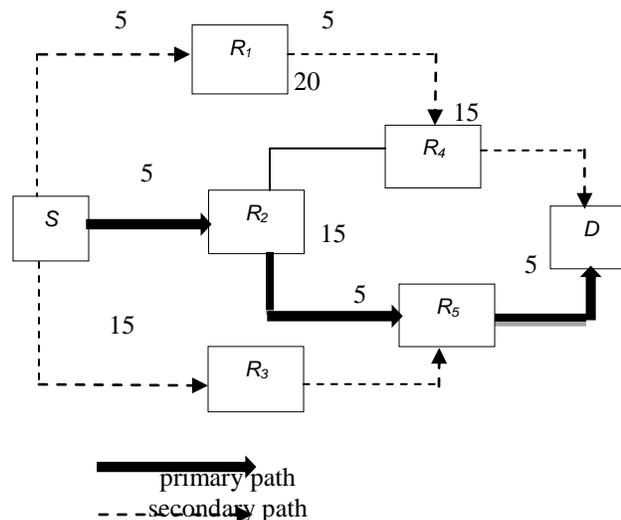


Fig 2: Equal Cost Multiple Path [2]

In Fig. 2, node S has three ECMP paths to node D: via R1, R2 and R3. Suppose that link S → R2 on the primary path is down. In this case, node S continues to use its two secondary paths when forwarding packets to node D. However, node R2 has only one shortest path to node D, thus ECMP would not be able to handle the failure of link R2 → R5.

### 2.2 Loop-free Alternates (LFA)

The principles of Loop-free Alternates (LFA [8]) are similar to that of ECMP, but LFA covers more cases. The primary path is always the shortest one, and all other paths along which the next-hop is closer to the destination than the sender are potential secondary paths. There is no global signaling, failures are not advertised throughout the whole network, but loops still cannot appear, since each router forwards packets to its neighbors so that the distance to the destination decreases in every step. A Loop-Free Alternate path [5] exists when a direct neighbor of the router adjacent to the failure has a path to the destination that can be guaranteed not to traverse the failure (loop-free neighbor condition). The average coverage on common networks (that is strongly dependent on the topology) shows variations from 60 to 90%. Indeed, when a link or a node fails, only the neighbors of the failure are initially aware that the failure has occurred and only neighboring nodes to the failure repair the failure. These repairing routers have to steer datagrams to their destinations despite the fact that most other routers in the network are unaware of the nature and the location of the failure. A common limitation in most of the base LFA mechanism is an inability to indicate the identity of the failure and to explicitly steer the repaired datagram round the failure. Consequently, the extent to which this limitation affects the repair coverage is topology dependent. An advanced LFA solution [6] consists in sequencing the FIB updates either spatially (topologically ordered FIB update from far-end to the near-end neighbor contiguous to the failure) or temporally (timely synchronized FIB updates). For instance, ordered FIB update provides 100% loop-free convergence at the expense of a FIB update time proportional to  $R \times \text{MAX\_FIB}$ , where, R is the max (hop) length among paths to edge r used to reach destination t (downstream SPF neighbor prior to the failure) and MAX\_FIB is a network-wide constant that reflects the maximum time T<sub>max</sub> required to update a FIB irrespective of the change required. Hence, it degrades proportionally to the path length i.e. FIB updates are actually committed at the near-end after reception of a completion message traveling back from the source of max (hop) length among path to edge r used to reach destination t. This solution is not considered outside network maintenance operation as it suffers from slow activation.

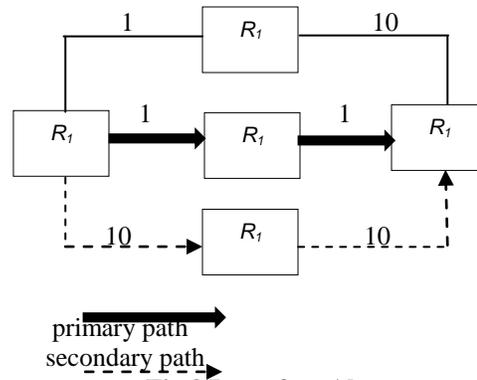


Fig.3 Loop-free Alternates

As shown in Fig. 3, the next-hop from S towards D along the shortest path is R2. When link S → R2 is down, the second shortest path from S is via R1. However, R1 does not know anything about the failure, and its shortest path to D is through S; hence, it cannot be used as a backup next-hop as it would pass packets back to S. Instead, S forwards packets towards R3 in order to avoid the loop. In case the weight of link S → R3 was 1, LFA would not be able to find an alternate path to D. These are alternative paths that are longer than the primary path, but still provide loop-free routing to the destination. Such a path exists when a direct neighbor (N) of the detecting node (S) has a path to the destination which can be guaranteed to not traverse the failure, i.e. the failed link or node is not included in the alternative path. IP Fast Reroute specifies a condition for Link-protecting alternates and a more restrictive condition for Node-protecting alternates.

1. Link-protecting alternates to guarantee loop-free alternates for link failures, the following condition must hold:

$$\text{cost}(N,D) < \text{cost}(N, S) + \text{cost}(S,D) \quad (1)$$

Figure 2.6 shows a failure scenario where this condition holds. In this scenario, node N would not route the packets back to the failure.

2. Node-protecting alternates alternate next-hops for node failures require a stronger condition than what is the case for link failures. If node E failed in figure 4 node N would choose node E as next hop towards destination D, and hence node N cannot be used as a backup next hop to protect the failure of node E. To guarantee loop-free alternates for node failures, the following condition must hold:

$$\text{cost}(N,D) < \text{cost}(N,E) + \text{cost}(E,D) \quad (2)$$

Figure 5 gives an example of a failure scenario where the condition holds for a failure of node E.

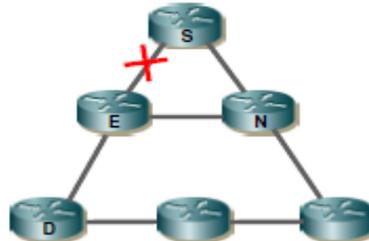


Fig 4: Illustrates a failure scenario where the condition for Link-protecting alternates is fulfilled.

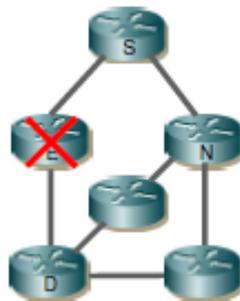


Fig 5: Illustrates a failure scenario where the condition for Node-protecting alternates is fulfilled.

### 2.3 Multi-hop repair paths

When there is no feasible loop-free alternate path it may still be possible to locate a router, which is more than one hop away from the router adjacent to the failure, from which traffic will be forwarded to the destination without traversing the failure. ECMP and loop-free alternate paths (as described in [RFC5286]) offer the simplest repair paths and would normally be used when they are available. It is anticipated that around 80% of failures (see Section 5.2.2) can be repaired using these basic methods alone. Multi-hop repair paths are more complex, both in the computations required to determine their existence, and in the mechanisms required to invoke them. They can be further classified as:

1. Mechanisms where one or more alternate FIBs are pre-computed in all routers, and the repaired packet is instructed to be forwarded using a "repair FIB" by some method of per-packet signaling such as detecting a "U-turn" [UTURN], [FIFR] or by marking the packet [SIMULA].
2. Mechanisms functionally equivalent to a loose source route that is invoked using the normal FIB. These include tunnels [TUNNELS], alternative shortest paths [ALT-SP], and label-based mechanisms.
3. Mechanisms employing special addresses or labels that are installed in the FIBs of all routers with routes pre-computed to avoid certain components of the network. For example, see [NOTVIA]. In many cases, a repair path that reaches two hops away from the router detecting the failure will suffice, and it is anticipated that around 98% of failures can be repaired by this method. However, to provide complete repair coverage, some use of longer multi-hop repair paths is generally necessary.

#### Scope of Repair Paths

A particular repair path may be valid for all destinations which require repair or may only be valid for a subset of destinations. If a repair path is valid for a node immediately downstream of the failure, then it will be valid for all destinations previously reachable by traversing the failure. However, in cases where such a repair path is difficult to achieve because it requires a high order multi-hop repair path, it may still be possible to identify lower-order repair paths (possibly even loop-free alternate paths) that allow the majority of destinations to be repaired. When IPFRR is unable to provide complete repair, it is desirable that the extent of the repair coverage can be determined and reported via network management. There is a trade-off between minimizing the number of repair paths to be computed, and minimizing the overheads incurred in using higher-order multi-hop repair paths for destinations for which they are not strictly necessary. However, the computational cost of determining repair paths on an individual destination basis can be very high. It will frequently be the case that the majority of destinations may be repaired using only the "basic" repair mechanism, leaving a smaller subset of the destinations to be repaired using one of the more complex multi-hop methods. Such a hybrid approach may go some way to resolving the conflict between completeness and complexity. The use of repair paths may result in excessive traffic passing over a link, resulting in congestion discard. This reduces the effectiveness of IPFRR. Mechanisms to influence the distribution of repaired traffic to minimize this effect are therefore desirable.

#### Tunneling

The repair strategies described in this draft operate on the basis that if a packet can somehow be sent to the other side of the failure, it will subsequently proceed towards its destination exactly as if it had traversed the failed component. See Figure 5.

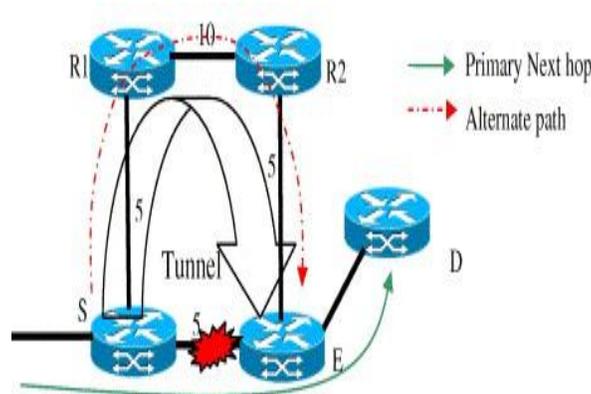


Fig 6: Simple Link Repair.

Creating a repair path from S to E may require a packet to traverse an unnatural route. If a suitable natural path starts at a neighbor (i.e. it is a loop-free alternate), then S can force the packet directly there. If this is not the case, then S may create one by using a tunnel to carry the packet to a point in the network where there is a real loop-free alternate. Note that the tunnel does not have to go from S to E. The tunnel can terminate at any router in the network, provided that S can be sure that the packet will proceed correctly to its destination from that router.

### Tunnel Requirements

There are a number of IP in IP tunnel mechanisms that may be used to fulfill the requirements of this design. Suitable candidates include IP-in-IP [RFC1853], GRE [RFC1701] and L2TPv3 [RFC3931]. The selection of the specific tunneling mechanism (and any necessary enhancements) used to provide a repair path is outside the scope of this document. However the following sections describe the requirements for the tunneling mechanism.

### Not-via Repairs

This section provides a brief overview of the not-via method of IPFRR.

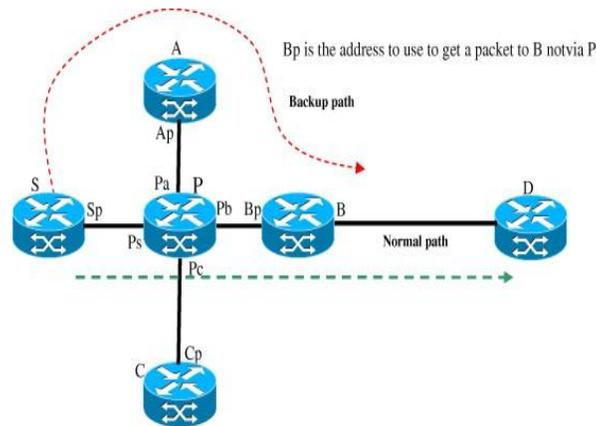


Fig 7: Tunnel to notvia address [8].

Assume that S has a packet for some destination D that it would normally send via P and B, and that S suspects that P has failed. S encapsulates the packet to Bp. The path from S to Bp is the shortest path from S to B not going via P. If the network contains a path from S to B that does not transit router P, i.e. the network is not partitioned by the failure of P, then the packet will be successfully delivered to B. When the packet addressed to Bp arrives at B, B removes the encapsulation and forwards the repaired packet towards its final destination. Note that if the path from B to the final destination includes one or more nodes that are included in the repair path, a packet may back track after the encapsulation is removed. However, because the decapsulating router is always closer to the packet destination than the encapsulating router, the packet will not loop. For complete protection, all of P's neighbors will require a not-via address that allows traffic to be directed to them without traversing P.

### 3 Ip Fast Reroute Configuration Strategies

We envision 3 different configuration scenarios of IP Fast Reroute. Link failures are the most common failure, and hence a strategy for only providing link failure coverage may be an alternative (1). If however there is a requirement for handling node failures as well, configuring for covering node failures is an alternative (2). Such an alternative will also cover link failures. In both strategy 1 and 2 we allow the use of multi-hop repair paths to obtain full coverage. Configuration strategy 3 represents a scenario where the routers or the operator does not support the use of multi-hop repair paths, e.g. due to the complexity. Then, only ECMP and other loop free alternates are allowed.

#### 1. Covering link failures

In this case, we configure IP Fast Reroute according to the condition for link protecting alternates. When no loop-free link protecting alternates exist, we configure u-turns, multi-hop tunneling or tunneling using Not-via addresses, respectively.

#### 2. Covering node failures

In this case, we configure IP Fast Reroute according to the condition for node protecting alternates. When no loop-free node protecting alternates exist, we configure u-turns, multi-hop tunneling or tunneling using Not-via addresses, respectively. Link failures will also be covered with this configuration strategy.

#### 3. Loop-free alternates only

In this case, we configure IP Fast Reroute according to the condition for node protecting alternates. If the condition for node protection alternates is not satisfied for a given destination, we try to configure according to the less restrictive condition for link protecting alternates. This strategy will use no multi-hop repair paths, and hence some failure scenarios may not be covered.

### 4 Conclusions

In theory, Not-via is the only IETF IP fast reroute scheme that can obtain full recovery from any single link or node failure. Full coverage cannot be obtained by using loop free alternates only. An alternative to obtain full coverage is to successively try to configure ECMP, other loop-free alternates, Uturns, general tunnels and Not-via tunnels (configuration strategy 1 and 2). From a management point of view this alternative provides a mix of relatively complex mechanisms to implement and

configure. From these findings, conclusion can be made that a network that is supposed to support fast reroute should support a method that guarantees full failure recovery from both single link and node failures. A feasible configuration would be to use ECMP and loop-free alternates since these are quite simple to configure and manage, and then use a full coverage method like Not-via to fulfill the guarantees of 100 % recovery from single failures.

#### **References**

- [1] AudunFossellie Hansen, "Fast Reroute in IP Networks," Doctoral Dissertation at the University of Oslo, May 2007.
- [2] PeterSzilagyi, ZoltanToth, "Design, Implementation and Evaluation of an IP Fast ReRoute Prototype," Budapest University of Technology and Economics, Faculty of Electrical Engineering and Informatics, Dept. of Telecommunications and Media Informatics, 2008.
- [3] Simon Tembo, Ken-ichiYukimatsu, Ryota Takahashi, ShoeiKamamura, Takashi Miyamura, KoheiShiomoto, "A New Backup Topology Design Method for Congestion Avoidance in IP Fast Reroute," International Journal of Networks and Communications 2012, 2(5): 123-131.
- [4] Wouter Tavernier,Dimitri Papadimitriou, Didier Colle, Mario Pickavet, Piet Demeester, " Automated Learning of Loop-Free Alternate Paths for Fast Re-Routing," Department of Information Technology (INTEC), Ghent University – IBBT, Gaston Crommenlaan 8, 9050 Gent, Belgium,2011.
- [5] S. Bryant, C. Filsfils, S.Previdi, M. Shand, "IP Fast Reroute using tunnels", internet-Draft Internet Engineering Task Force, November 16, 2007.
- [6] S. Bryant, M. Shand,"IP Fast Reroute Framework," internet-Draft , Internet Engineering Task Force, January, 2010.
- [7] S. Bryant, S. Previdi, M. Shand, "IP Fast Reroute Using Not-via Addresses", internet-Draft , Internet Engineering Task Force, December 21, 2011.