Evaluation of IP Fast Reroute Proposals

Minas Gjoka
University of California, Irvine
Irvine, CA 92697
Email: mgjoka@uci.edu

Vinayak Ram
University of California, Irvine
Irvine, CA 92697
Email: vram@uci.edu

Xiaowei Yang
University of California, Irvine
Irvine, CA 92697
Email: xwy@ics.uci.edu

***Abstract*— With the increasing demand for low-latency applications in the Internet, the slow convergence of the existing routing protocols is a growing concern. A number of IP fast reroute mechanisms have been developed by the IETF to address the issue. The goal of the IPFRR mechanisms is to activate alternate routing paths which avoid micro loops under node or link failures. In this paper we present a comprehensive analysis of these proposals by evaluating their coverage for a variety of inferred and synthetic ISP topologies.**

I. Introduction

One of the main issues that motivates the need for IP Fast Reroute mechanisms is the slow convergence time in today's networks. These convergence times can be of the order of 100's of milliseconds or even 10's of seconds in the BGP networks. IP-based networks have evolved significantly over the past years from carrying just text data to carrying all sorts of multimedia traffic today. The best effort Internet model was sufficient for typical applications such as email and file transfer. However, when we consider recent applications such as VoIP, Video on Demand etc, a *better than best effort* model and a much faster convergence time is required for the network in case of a failure. Considering their low latency requirements, achieving a convergence time which is of the order of 10s of milliseconds is the key.

In the current networks, when there is a link or node failure, the routers affected need to recalculate their routing tables and propagate the updates to all the routers concerned with this failure. That results in a period of disruption of traffic until convergence is achieved. In the interim period, while the network is converging, micro-loops may be created. The impact of this was not significantly noticeable so far, as most of the applications were acceptable with the delay and packet loss(if any) involved. However, this assumption is no longer valid as we make the move to multimedia applications over the Internet.

A number of factors affect the time required for the network to recover from a failure. The time taken to detect a failure and the recalculation of the SPF at the local router takes roughly around $70ms$. Further, the time taken for the generation and propagation of the updates and the recomputation of the forwarding tables at all the nodes is of the order of a few ms (depending on the network size). Finally, the FIB updates take several 100's of milliseconds.

Thus, a mechanism that helps the network converge faster is the need of the hour. There are some limitations that we must consider here, for example achieving a sub 50ms convergence time though desirable is very improbabilistic considering the distributed nature of the network and the propagation time of the various updates. The aim is therefore to get as close to it as possible. IPFRR mechanisms strive to maintain network connectivity while the generation of the updates and the resultant FIB updates occur in the background. This significantly helps in reducing the network convergence time.

We believe that the IP Fast Reroute mechanisms have a strong potential in helping us reach this goal. There has been considerable interest amongst the networking community on these mechanisms in the recent past and few proposals have been submitted to the IETF RTWG. In this paper, we present a thorough analysis of these proposals by simulating them for a variety of networks and inferred topologies and identifying their strengths and weaknesses.

The rest of the paper is organized as follows : Section II contains an overview of the related work in IP Fast Re-Routing (IPFRR) techniques. In section III we describe the various IPFRR mechanisms as defined in work-in-progress Internet Drafts[3] - [8]. In section IV we describe our methodology for our simulations. In section V we summarize our experimental results and present a general analysis of our findings.

II. Related Work

In [11] a classification of failures is presented after an extensive analysis of routing updates in Sprint's IP backbone network. The results show that 80% of all failures are unplanned. Of those, 70% affect a single link at a time while 30% involve shared link risk groups. These numbers make the need for IP fast recovery techniques more imperative. Planned link failures which constitute the remaining 20% can be handled by techniques such as graceful shutdown, ordered updates etc.

There has been some prior work in Fast Rerouting Mechanisms related to the MPLS domain. MPLS-FRR requires that $O(nk)$ repair paths be set up for a link failure where *n* is the number of nodes and *k* is the number of links in the network. In the case of a node failure , the number of paths that is required is $O(nk^2)$. We further observe that MPLS-FRR needs a source routed path around each failure and moreover the deployment of such a solution needs an MPLS infrastructure to be present. Currently, pure IP based networks are much more prevalent than MPLS networks thus limiting the scope of the above solution.

IPFRR mechanisms are meant to provide alternative paths for a temporary time period until the network converges to a stable state with normal forwarding tables. In the interim period, microloops may occur and must be prevented. A solution which prevents microloops was described in [3] using path locking via safe neighbours. Their in-progress work provides a fixed convergence time regardless of the network size but does not guarantee full coverage for all microloops. An alternate solution was presented in [2], in which the authors describe a proposal for ordered updates for FIBs. The basic idea in their solution is that for a link-down event the router closest to the link/node failure will update its FIB only after all its upstream routers have done so. This ensures there are no microloops as a result of delayed updates of the FIB on some routers. This proposal is able to provide 100% coverage for all microloops at the cost of larger convergence time. In [12] the authors present an analysis of the number of failure scenarios that can be covered using IPFRR.

III. IP Fast ReRoute Mechanisms

A. Equal Cost Multiple Paths

Equal Cost Multiple Paths (ECMP) exist when a router can reach the destination by multiple paths of the same cost but traversing different links. In such cases these alternate paths may be precomputed and used to maintain connectivity when a link failure occurs. The basic criteria for the selection of these paths is the trivial condition:

$$cost(Ni) = cost(Nj), Ni \neq Nj \tag{1}$$

where Ni is the current primary next hop neighbor and Nj is alternate next hop neighbor which provides an ECMP path to destination D. The cost(Ni) is the shortest path cost from node Ni to the destination D. While selecting the recovery node we must ensure that the path to the destination must not contain the failed link/node. It must be noted that loops are not possible if we assume a non-zero value for links.

Consider the graph shown in figure 1(a). Node S has three ECMP paths via the next hop neighbours N1, N3, N5. If we assume the primary next hop neighbor to be N1 and the link S-N1 fails then any of the remaining two paths may serve as an alternate ECMP path. However if we consider the source destination pair as N3-D there are no ECMP paths. Thus no protection is offered for the link failure N3-N4.

B. Loop Free Alternates

The basic idea behind Loop Free Alternates [5] is to use a pre-computed alternate next hop in the event of a link failure so that traffic can be routed through this alternate next hop when a failure is detected. Thus, the traffic flow is not disrupted and the network can converge in the background. Once the network has converged and the routing tables are updated, the traffic then flows along the newly calculated primary next hop. There is a hold down timer associated which ensures that the nodes wait for the time duration which is enough for the network to converge before they switch to the newly updated tables.

It is essential to ensure that while choosing the loop free alternates, micro loops don't arise either during the link down or link up phase. The criteria for a neighbour to provide a loop-free alternate is given by the following two relations :

Loop-Free Criterion:

$$cost(Ni) < link(Ni, S) + cost(S) \tag{2}$$

Downstream Path Criterion:

$$cost(Ni) < cost(S) \tag{3}$$

where S is the source , D is the destination , Ni is the LFA for S

1) Failure Scenarios : Link Failures: Consider the sample scenario as shown in figure 1(b). S is sending traffic to D via its shortest path (S-E-D). If link S-E fails, S uses its pre-calculated LFA ie. N to route traffic to D. Note that N satisfies the selection criteria as described in (2). Thus, the traffic is re-routed along the path (S-N-E-D) until the network converges again.

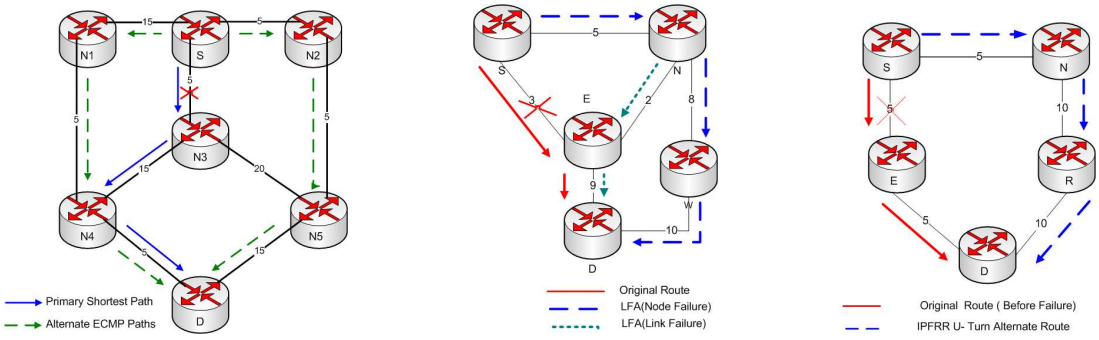
2) Failure Scenarios : Node Failures: Consider the scenario where a node failure occurs and Node E is down. Now, in the absence of the downstream path criteria (3), the LFAs for nodes S and N are $\{N\}$ and $\{S,W\}$ respectively. Thus, Nodes S and N would have realized the failure of Node E and forwarded packets to each other resulting in a micro loop. However, the condition imposed by inequality (3) allows S to use N as its LFA but not vice-versa. Consequently, N forwards the packets to its LFA (Node W) and thus a microloop is avoided. Thus, we observe that there is a tradeoff involved in the prevention of microloops and the number of alternate nodes available for recovery.

C. U-Turn Alternates

The existence of a loop-free alternate as described in the previous section is topology dependent. There can be scenarios such as the one described in figure 1(c) where in a given topology, there exist no LFAs but it is possible for U-turn Alternates [6] to exist.

A U-turn alternate uses a neighbor, whose primary next hop to the destination is the router S and which itself has a loop free node protecting alternate, which does not go through S. In figure 1(c), S does not have any LFAs to reach D as N which is its primary next hop neighbour fails to meet both the criteria as defined by (2) and (3). We observe that for node N, the primary next hop to reach the destination is S. Further, N has a LFA node R to reach the destination. Thus, if N is capable of identifying this traffic, it could avoid the loop between N and S and thus it could be used as a U-Turn Alternate for node S. This U-Turn traffic can either be explicitly marked or implicitly detected.

Thus, if N receives U-Turn traffic from its primary next-hop neighbour S, it recognizes it and instead of forwarding the traffic back to S, it forwards the traffic to its LFA R. Thus, this mechanism allows S to redirect its traffic to join the SPF at its neighbour's neighbour. The main drawback of this approach is that it requires the marking of the U-Turn packets and an increased computational complexity as compared to LFA.



(a) Equal cost multiple paths scenario

(b) Loop Free Alternate scenario

(c) U-Turn Alternate Paths

Fig. 1.

The criteria for a node to be selected as a U-Turn Alternate are as follows:

- Node Selection Criteria

$$cost(Ni) \geq linkcost(Ni, S) + cost(S) \qquad (4)$$

- S must always be the primary next hop neighbour on all shortest paths from N to D that traverse S.
- N has a loop free alternate.

Identifying U-Turn Traffic

U-turn traffic can be identified by either Implicit or Explicit packet identification. Implicit packets identification requires no modification of the packet sent to the U-Turn Alternate. When a U-Turn alternate receives a packet for a destination D from a primary next hop neighbour to which it would normally forward the packet, it identifies the packet as a U-Turn packet and forwards it to its LFA node.

The Explicit packet identification method requires the packet to be marked using either MPLS labels or by setting specific bits in the layer 2 header. Alternately, techniques such as those described in [13] where the authors use a combination of tags, which overload the bits in the identification field and TTL field of the IP header, could be used.

D. Tunnels

IPFRR Tunnels, described in [7], comprise an even more generic mechanism. When a router S detects an adjacent failure it uses a precomputed set of repair paths to bypass the failed neighboring node E or the failed link to E. A tunnel is used to carry traffic to a router, called the tunnel endpoint, where loop free alternate paths to the destination router exist, using normal forwarding. All packets are encapsulated by S and routed towards the tunnel endpoint. The tunnel endpoint decapsulates the packets and either forwards them according to its shortest path table, depending on the final destination, or uses the release point as the next hop for the packet if indicated by S.

Since the goal is to route traffic around a failure, repair paths must be created for all neighbors of the neighboring node E plus for the node E itself to provision for link failures. In figure 2 one repair path is precomputed for each of E's neighbors and a third repair path for packets which have E as their destination. It should be mentioned here that it is possible that not all these neighbors may need a repair path, for example when no traffic is initially forwarded through them.

1) Properties required to avoid looping: The routers in the tunneled repair path may not be aware of the failure yet. It is necessary for node S, called the repairing node, to ensure that the packets sent through the tunnel will not return back to S. That may not be possible if the failed component (node or link) is on the default shortest path towards the tunnel endpoint in any of the routers in the tunnelled repair path. Therefore a careful selection of the tunnel endpoint is necessary so that it is reachable from any of the intermediate routers in the tunneled repair path without passing through the failed component. For example, in figure 2 for the repair path S-S1 only routers X, Y, Z, V are suitable for tunnel endpoints, assuming a failed link S-E. The reason for that is that only those nodes have a default shortest path which avoids entirely the failed link S-E. Furthermore we must guarantee that the decapsulated packets at the tunnel endpoint will reach the destination router using normal forwarding. A viable solution requires both of the above properties to be satisfied.

Directed forwarding is a method used at the tunnel endpoint to avoid the formation of microloops. The problem mentioned may occur when decapsulated packets at the tunnel endpoint have a shortest path towards the destination router which uses the failed component. Directed forwarding allows the repairing node to specify the release point at the tunnel endpoint. For example if the cost for link X-Y in figure 2 increased to 5 the repairing router S could choose node X as the tunnel endpoint and at the same time designate router Y as the release point. This allows decapsulated packets to avoid entirely the failed link and thus reach the destination.

The possible tunnel endpoints, called F-space [7], include the set of routers reachable from the repairing node without any path passing through the failed link while the possible release points, called G-space [7], include the set of routers from which the repair path target can be reached without any path transiting the failed link.

The set produced by the intersection of the above defined sets

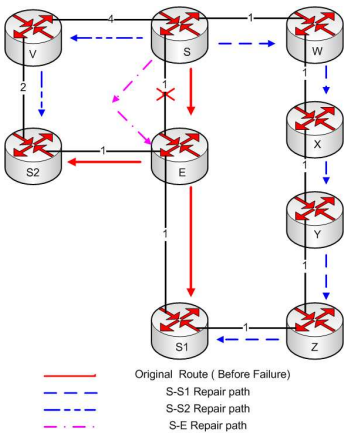


Fig. 2. Repair paths in anticipation of a router failure for Tunnels

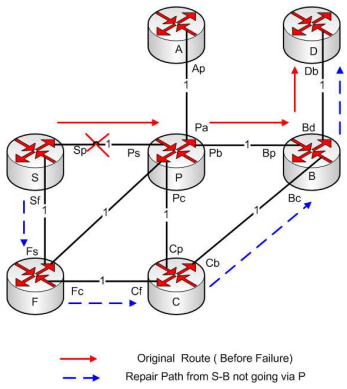


Fig. 3. Set of Not-Via Addresses for each protected interface

provides us with the release points which don't need directed forwarding. In case of an empty set, a pair of routers must be identified each of which belongs to one of the two sets and have a direct link between each other. These two routers can serve as the tunnel endpoint and the release endpoint in directed forwarding mode.

It is possible that no repair paths for a repair target are available even when directed forwarding is enabled. Some reasons listed in [7] for this observed state are :

- the neighboring router E is a single point of failure
- severely asymmetric link costs
- interference between different repair paths

E. Not-via Addresses

Not-via addresses [8] are special addresses assigned to each protected interface. The semantics of a not-via address is that "a packet addressed to a not-via address must be delivered to the router advertising that address, not via the protected component with which that address is associated". For each protected interface two addresses are required, the normal IP address and the Not-via address. Once a failure is detected, the repairing router tunnels traffic towards the Not-via address of the protected component. In the scenario depicted in figure 3, S routes traffic to D through B. When S suspects that P has failed, it tunnels the related traffic to the Not-via address Bp, which is interpreted as the shortest path from S to B not going via P. After reaching B, the involved packets will be forwarded to D without any further problems. This mechanism requires the participation of the intermediate routers on the repair path since they must be able to tell the link which they must avoid traversing from the semantics of the Not-via address.

Every router in the repair path routes the tunnelled traffic using the shortest path obtained by running SPF on a graph where the protected link is excluded. Once traffic reaches the decapsulating router it is then forwarded to its final destination. It should be noticed that backtracking may be possible if the repair path contains the final destination. Even if that is the case the packets will eventually be delivered without any more loops.

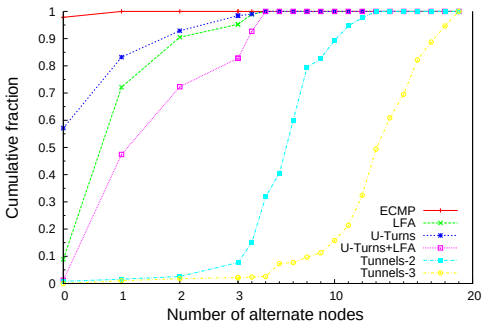
	Router Only	Bottom up
Grouping Model	-	Random pick
Model	GLP	GLP
Node placement	Random	Random
Growth Type	Incremental	Incremental
Preferential Connectivity	On	On
BW Distribution	Uniform	Uniform
Minimum BW	100	100
Maximum BW	1024	1024
m	1-2	1-3

TABLE I
BRITE TOPOLOGY GENERATOR PARAMETERS

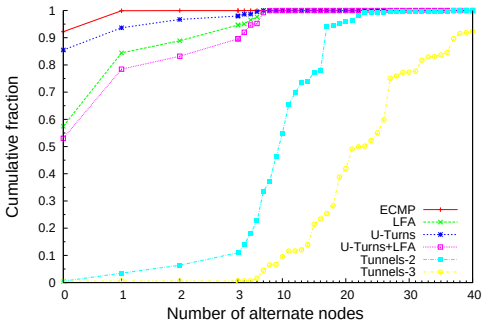
Not-via addresses constitute a mechanism able to provide full coverage but accompanied with an increased amount of complexity. The fact that each router must have precomputed backup routes combined with the possibility of it being in any shortest path of an initiated repair would require n-1 SPFs for each of the n routers of the topology. Fortunately a clever implementation of these computations using incremental SPF with early termination, mentioned in [9], decreases the computational effort required. Furthermore it is common to deploy Not-Via addresses so as to be used as the last resort when simpler mechanisms, such as Loop free alternates, fail to provide repair paths. For example, if a Loop free alternate is found by the repairing router S for a specific protected interface the routers are signaled not to proceed into the computation of the SPF for the protected interface.

IV. METHODOLOGY

In our simulations we tested the above mechanisms on a variety of randomly generated topologies, inferred ISP topologies (from Rocketfuel project[10]) and real networks such as ABILENE, GEANT and BELNET. The network size of these topologies ranges from a handful of nodes up to large graphs of hundreds of nodes. One of the reasons for using such diverse network topologies was to study the behavior of the above mechanisms under a wide range of scenarios. Our results are based on the assumption that the topologies are static in terms of link costs which are considered to be symmetric.

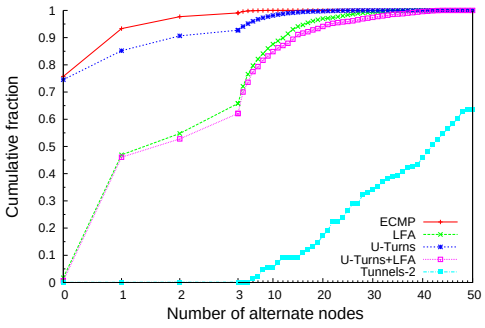


(a) Geant

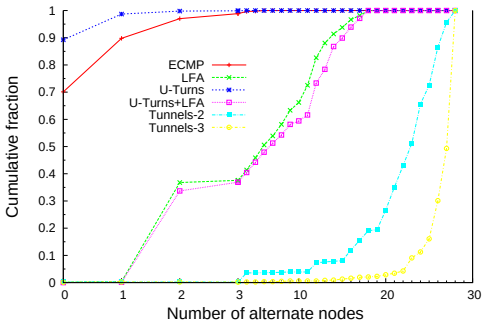


(b) Telstra

Fig. 4. Link Failures - All mechanisms

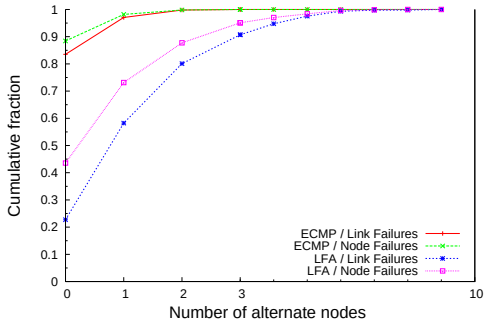


(a) AT&T

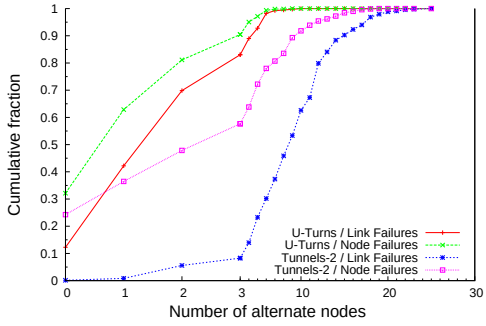


(b) DFN

Fig. 5. Link Failures - All mechanisms



(a) ECMP/LFA Link/Node Failures



(b) U-Turns/Tunnels Link/Node Failures

Fig. 6. Exodus

We chose to use BRITE topology generator [14] on Router only and Bottom-up modes. We relied on experiments performed in [15] to select parameters which produce realistic results. Table I contains the parameters used to generate random topologies which simulate real topologies. The link cost in the converted graphs for our simulator was computed as a function of the inverse bandwidth. A uniform bandwidth distribution which ranges from 100-1024 was selected. Incremental growth and preferential connectivity are selected as intuitive mechanisms for topology generation. Parameter m sets the number of links added per new node which directly affects the average node degree of the generated graph.

In [15] the authors present in detail the parameters used to generate synthetic topologies which approximate AT&T and DFN real topologies in Bottom-up mode. We have included these two topologies as-is in our simulations. Furthermore we have also included topologies with network size ranging from 50 to 200 nodes for Router only and Bottom-up modes to examine the effect of varying connectivity and network size in the coverage of the IPFRR mechanisms.

We measure the coverage provided by each of the mechanisms under scenarios of link and node failures. The algorithm used to count the number of successful cases of recovery is as follows:

For every source S in a source-destination pair we iteratively choose each of its immediate neighbors Ni. We then mark the link between S and Ni or the node Ni itself (for node failures) as failed. Cases where the node/link which is marked as failed is not a part of the normal shortest path forwarding table are

dropped. As a result of that, the total number of cases where recovery must be initiated is n*(n-1) for link failures where n is the number of nodes since for each source-destination pair there is only one shortest path in the normal forwarding table. For node failures that number may be somewhat larger since more than one neighbors may be used in the shortest path to the destination. The set of neighbors of the source node, which does not include the failed neighbor, is examined on whether it satisfies the necessary condition to be considered eligible for an alternate routing path. This set of nodes will vary depending on the IPFRR mechanism to be examined. If any of the neighbors are found to provide recovery for failure then the case is considered successful. For mechanisms that do not use any kind of signaling, since the eligibility condition may choose looping neighbors we further check whether the qualified neighbors will reroute back to the source node using their normal forwarding table. That is possible since these nodes may have not been notified of the failure yet. For link failures we check whether the next hop of the recovery neighbor is the source itself while for node failures whether the failed node lies in the path from the recovery neighbor to the destination. The latter check is more strict since we want to avoid entirely the failed node.

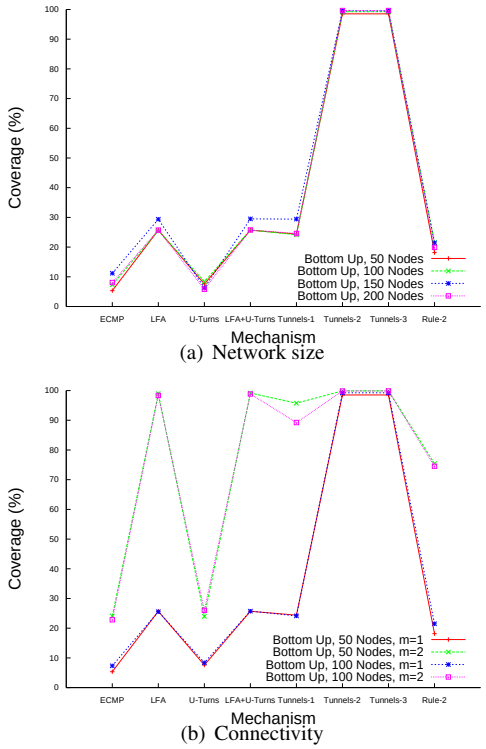
The implementation of tunnels is more complex since it involves the usage of nodes other than the immediate neighbors. Using breadth first search we build a table indexed by the number of hops a node is reachable from the source. In the rest of the paper when we refer to Tunnels-n we imply tunneling mechanisms where the alternate recovery nodes are the nodes reachable within "n" hops.

Decisions on the existence of alternate loop-free paths are made based on what a source node considers as the present state of the network. In reality, in a failure scenario the information will be propagated to the rest of the network and the rest of the nodes in the path towards the destination may react differently than expected to the announced failure. The possibility of microloops exists but solutions such as the ordered updates of routing tables [2] have been proposed by the community to avoid them. IPFRR mechanisms are based on the assumption that adjacent nodes to the failing link/node are able to detect the incurred failure within 50ms and take corrective action. Accordingly in our simulations we compute coverage for failures adjacent to the node under consideration.

V. Experimental Results & Analysis

From the data gathered from our custom simulator we present an analysis of the number of alternate nodes available per source-destination pair and the respective total protection coverage offered against link and node failures. Figures 4-6 contain the cumulative distribution of the number of alternate nodes under different mechanisms. The x-axis is zoomed in the interval from [0,3] to gain more relevant insight. In figures 4 and 5 we compare the protection for link failures between the various mechanisms.

In table II we present a detailed and comprehensive analysis of various inferred ISP topologies and open networks with a



(a) Network size

(b) Connectivity

Fig. 7. Bottom up Topologies - Link Failures

varying degree of connectivity and network size. The results as shown reaffirm our belief that ECMP is generally the weakest as compared to other schemes in both node and link failure recovery. U-Turns effectiveness ranges between that of ECMP and LFAs. However the deployment of U-Turns depends on other signalling mechanism to identify the traffic which adds to the deployment complexity. Strangely for DFN topology ECMP is found to be superior to U-Turns. We attribute this result to the fact that DFN is a synthetic topology. The coverage offered by tunnels approaches 100%. Based on our results we observed that in most cases two hop tunneling mechanisms achieve close to the absolute coverage and any increase in the number of hops is not worth the computational complexity vs increased coverage tradeoff. Another interesting result to be noted is that Telstra, in spite of its relatively large size, has negligible ECMP coverage. GEANT has no more than 1 alternate neighbor for ECMP for the very small number of succesful recovery cases observed. Sprint on the other hand provided a good number of alternate neighbors for ECMP.

In figure 6(a) and 6(b) a comparison is made showing the number of recovery nodes achieved for link and node failures under different mechanisms for Exodus ISP topology. We chose to compare ECMP/LFA and U-Turns+LFA/Tunnels-2 as the two sets are close to each other in terms of coverage. For the same reason mentioned above we did not include more than two hop tunneling. The results confirm that node failures are more difficult to handle as compared to link failures. The recovery nodes for two hop tunneling in case of node failures was significantly lower as compared to link failures.

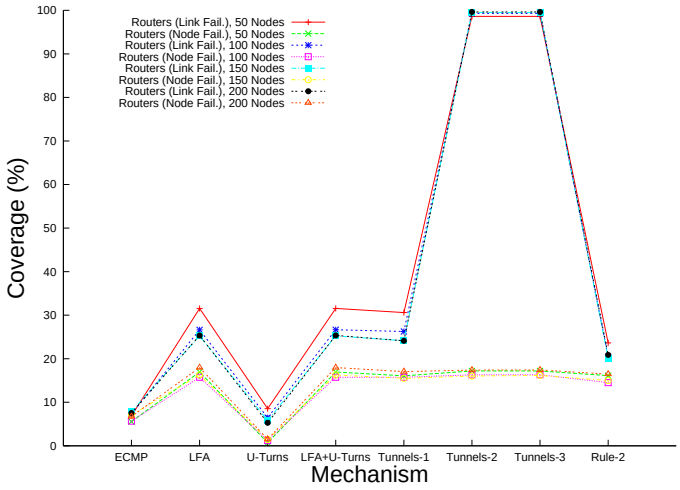


Fig. 8. Router Level - Node/Link Failures

The coverage obtained for Bottom-up synthetic topologies of varying network size is presented figure 7(a). The results seem to be consistent for networks of different sizes. In figure 7(b) we examine how increased network connectivity affects the achieved coverage. Synthetic topologies with average node degree up to twice higher clearly perform better. For Tunnels-2 and Tunnels-3 a similar coverage is achieved in all graphs regardless of the average node degree. Rule 2 from [13] is shown to perform on par with LFA, LFA+U-Turns and Tunnels-1.

Figure 8 shows the coverage achieved for node and link failures for different network size router-level topologies under all mechanisms. We must note here that average node degree does not differ by a significant percentage in each of these topologies. The results show that the obtained coverage under node failures is approximately the same for different mechanisms. The same is observed for link failures with the exception of the topology with 50 nodes which has a higher coverage for LFA and U-Turns. We should also note here the unexpectedly low coverage provided for node failures in all router level topologies which must be related to the topology generator model

A short comparison is presented below on how each of the above mechanisms performs in regard to the following:

1) Encapsulation overhead: ECMP and LFA do not use tunneling and therefore no overhead is incurred. U-turn alternates need to use 1 U-turn label while IP tunnels require more than one IP header (IP-to-IP encapsulation) plus 1 directed forwarding label if needed.

2) Coverage: Based on our observations, ECMP, LFA and U-Turns cannot provide full coverage. Only large topologies with a high degree of connectivity could reach high numbers above 95%. Tunnels seem to reach almost full coverage using no more than two to three hops but they too fall short of providing 100% coverage. Not-Via addresses have the potential to provide coverage to all failures. These estimations assume that connectivity allows for possible repairs.

3) Complexity: ECMP involves no further complexity. LFA and U-turns and Rule-2 have low to moderate complexity and involve the computation of SPFs of the order of $O(k)$ where k is the number of neighbors. Tunneling may require up to $O(k^2)$ computation of SPFs.

VI. Conclusion and Future Work

In this paper, we have implemented the various IPFRR mechanisms and Rule 2 from [13]. Based on our findings, we have presented the results as documented in table II. We have compared and analyzed the coverage as well as the diversity of recovery choices provided by these techniques. We believe that such an analysis will be useful to the research community and will be helpful for evaluating the pros and cons of the IPFRR techniques. As a part of our future work we plan to implement the other IPFRR techniques, such as Not-via and directed forwarding in Tunnels. We also plan to examine how the presence of asymmetric link costs affects coverage. In order to test the performance of the IPFRR mechanisms in a real environment we are currently working on implementing them in Zebra [16]. The results show that IPFRR is a promising framework for handling unexpected failures. Selecting the maximum protection needed is a matter of tradeoff between complexity and cost. Given the encouraging simulation results we believe that the adoption of the presented IPFRR mechanisms by ISPs would lead to a significant increase in network availability during recovery phase.

VII. Acknowledgments

We would like to thank the anonymous reviewers for their insightful feedback and comments.

References

[1] P. Francois, C. Filsfils, J. Evans, and O. Bonaventure, "Achieving sub-second IGP convergence in large IP networks", ACM SIGCOMM 2005.
[2] P. Francois and O. Bonaventure, "Avoiding transient loops during IGP convergence in IP networks", IEEE INFOCOM 2005
[3] S. Bryant, M. Shand, "A Framework for Loop-free Convergence", IETF Internet Draft 2006, draft-bryant-shand-lf-conv-frmwk-02.txt
[4] M. Shand, S. Bryant, "IP Fast Reroute Framework", IETF Internet Draft, March 2006, draft-ietf-rtgwg-ipfrr-framework-05.txt.
[5] A. Atlas and A. Zinin, "Basic specification for IP fast reroute: Loop-free alternates", IETF Internet Draft, 2005, draft-ietf-rtgwg-ipfrr-spec-base-04.txt.
[6] A. Atlas et al, "U-turn Alternates for IP/LDP Fast-Reroute", IETF Internet Draft, 2006, draft-atlas-ip-local-protect-uturn-03.txt
[7] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, "IP fast reroute using tunnels", IETF Internet Draft, Apr. 2005, draft-bryant-ipfrr-tunnels-02.txt.
[8] S. Bryant, M. Shand, and S. Previdi, "IP fast reroute using not-via addresses", IETF Internet Draft, 2005, draft-bryant-shand-IPFRR-notvia-addresses-01.txt.
[9] Alia K. Atlas, Gagan Choudhury, David Ward, "IP Fast Reroute Overview and Things we are struggling to solve", NANOG 2005
[10] "Rocketfuel: An ISP Topology Mapping Engine", http://www.cs.washington.edu/research/networking/rocketfuel/
[11] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Christophe Dio, "Characterization of Failures in an IP Backbone", INFOCOMM 2004
[12] Audun Fosselie Hansen , Tarik Cicic, Stein Gjessing, "Alternative Schemes for Proactive IP Recovery", NGI 2006
[13] Xiaowei Yang, David Wetherall, "Source Selectable Path Diversity via Routing Deflections", SIGCOMM 2006
[14] BRITE Topology Generator, http://www.cs.bu.edu/brite/
[15] Oliver Heckmann et al "How to use topology generators to create realistic topologies", Technical Report, Dec 2002
[16] GNU Zebra - http://www.zebra.org

TABLE II

EXPERIMENTAL RESULTS

Network	Mechanism	Link Failures				Node Failures			
		Alternate Nodes		Valid Paths	Coverage	Alternate Nodes		Valid Paths	Coverage
		Average	> 1			Average	> 1		
Abilene	ECMP	0	0%	0	0%	0	0%	0	0%
Nodes: 11	LFA	0.8	22%	72	65.45%	0.47	8.33%	48	43.64%
Degree: 2.54	LFA + U-Turns	1.24	37.38%	99	90%	0.71	18.18%	66	60.00%
Src-Dst Pairs: 110	Tunnels-1	0.8	22.22%	72	65.45%	0.47	8.33%	48	43.64%
	Tunnels-2	3.25	9.36%	110	100%	1.14	63.63%	66	60.00%
	Tunnels-3	5	99.09%	110	100%	1.609	80.00%	70	63.64%
	Rule 2			48	43.18%			35	31.82%
Belnet	ECMP	0.74	0%	156	74.28%	0.74	0%	156	73.58%
Nodes: 15	LFA	1.10	0.96%	208	99.04%	0.86	1.26%	158	74.53%
Degree: 3.6	LFA + U-Turns	1.86	13.33%	210	100%	0.86	1.26%	158	74.53%
Src-Dst Pairs: 210	Tunnels-1	1.71	13.46%	208	99.04%	0.85	1.26%	158	74.53%
	Tunnels-2	12.24	100%	210	100%	0.85	1.26%	158	74.53%
	Tunnels-3	12.24	100%	210	100%	0.85	1.26%	158	74.53%
	Rule 2			185	88.05%			158	74.53%
Geant	ECMP	0.02	0%	11	2.17%	0.02	0%	11	2.17%
Nodes: 23	LFA	1.34	30.58%	461	91.11%	1.02	30.77%	351	69.37%
Degree: 3.22	LFA + U-Turns	2.04	53.2%	500	98.81%	1.33	46.59%	367	72.53%
Src-Dst Pairs: 506	Tunnels-1	1.36	30.70%	469	92.69%	1.02	30.77%	351	69.37%
	Tunnels-2	6.96	99.20%	502	99.21%	2.73	80.52%	385	76.08% s
	Tunnels-3	13.39	99.01%	506	100%	4.12	82.83%	396	78.26%
	Rule 2			354	70.00%			293	57.86%
ATT	ECMP	0.35	27.53%	5719	24.27%	0.33	26.74%	5534	22.93%
Nodes: 154	LFA	4.44	54.02%	23156	98.27%	3.16	50.35%	21372	88.55%
Degree: 6.75	LFA + U-Turns	5.34	54.31%	23407	99.34%	3.62	50.98%	21679	89.82%
Src-Dst Pairs:	Tunnels-1	3.96	54.41%	21755	92.33%	2.78	51.33%	19922	82.54%
23562	Tunnels-2	46.31	99.98%	23548	99.94%	21.39	96.51%	21189	87.79%
	Tunnels-3	120.332	99.99%	23552	99.96%	45.96	97.21%	21284	88.18%
	Rule 2			18259	77.49%			17314	71.74%
DFN	ECMP	0.45	34.23%	260	29.88%	0.33	28.05%	221	23.76%
Nodes: 30	LFA	7.22	99.77%	869	99.88%	3.26	90.98%	621	66.77%
Degree: 9.4	LFA + U-Turns	7.92	99.88%	870	100%	3.39	91.68%	625	67.20%
Src-Dst Pairs: 870	Tunnels-1	6.21	89.50%	838	96.32%	2.66	77.78%	585	62.90%
	Tunnels-2	21.96	100%	868	99.77%	8.33	95.05%	606	65.16%
	Tunnels-3	26.58	99.88%	869	99.88%	9.59	95.04%	606	65.16%
	Rule 2			703	80.85%			562	60.47%
Telstra(1221)	ECMP	0.07	0.96%	833	7.77%	0.05	1.41%	565	5.27%
Nodes: 108	LFA	0.85	36.62%	4557	42.52%	0.43	22.55%	3143	29.33%
Degree: 2.90	LFA + U-Turns	1.14	45.88%	5034	46.97%	0.57	33.03%	3257	30.39%
Src-Dst Pairs:	Tunnels-1	0.93	39.32%	4664	43.52%	0.46	25.29%	3170	29.58%
11556	Tunnels-2	10.46	97.03%	10661	99.48%	1.57	80.06%	4173	38.94%
	Tunnels-3	24.08	99.93%	10661	99.48%	3.99	96.28%	4251	39.66%
	Rule 2	-	-	2876	26.83%	-	-	2217.08	20.68%
Sprintlink(1239)	ECMP	0.39	23.94%	26987	27.28%	0.38	23.91%	26471	26.65%
Nodes: 315	LFA	3.79	77.11%	85905	86.85%	2.12	58.58%	72069	72.58%
Degree: 6.17	LFA + U-Turns	4.52	78.28%	88084	89.05%	2.63	60.45%	74184	74.71%
Src-Dst Pairs:	Tunnels-1	4.12	76.98%	86040	86.98%	2.30	58.77%	72138	72.65%
98910	Tunnels-2	33.85	99.97%	98876	99.96%	14.63	94.36%	84815	85.41%
	Tunnels-3	-	-	-	-	-	-	-	-
	Rule 2			67149	67.88%			59323	59.74%
EBONE(1755)	ECMP	0.19	10.78%	1307	17.46%	0.15	13.31%	991	13.15%
Nodes: 87	LFA	1.54	58.67%	5384	71.95%	1.02	43.20%	4226	56.07%
Degree: 3.70	LFA + U-Turns	2.11	70.76%	6184	82.65%	1.37	53.13%	5023	66.65%
Src-Dst Pairs: 7482	Tunnels-1	1.69	62.97%	5632	75.27%	1.05	44.62%	4258	56.50%
	Tunnels-2	8.34	97.49%	7470	99.83%	3.38	86.12%	5319	70.58%
	Tunnels-3	18.93	99.87%	7471	99.85%	6.61	92.66%	5550	73.64%
	Rule 2			3723	49.76%			3016	40.02%
Tiscali(3257)	ECMP	0.16	18.54%	3257	13.69%	0.12	23.81%	2448	9.28%
Nodes: 161	LFA	1.92	59.08%	16403	63.67%	1.44	55.06%	14298	54.24%
Degree: 4.07	LFA + U-Turns	2.47	67.73%	17671	68.59%	1.81	62.88%	15886	60.27%
Src-Dst Pairs:	Tunnels-1	1.93	61.42%	16501	64.05%	1.42	55.32%	14142	53.65%
25760	Tunnels-2	14.62	97.38%	25712	99.81%	5.73	91.97%	16090	61.04%
	Tunnels-3	36.22	99.96%	25714	99.98%	12.00	95.98%	16751	63.55%
	Rule 2			12148	47.15%			11054	41.94%
Exodus(3967)	ECMP	0.19	17.81%	1016	16.48%	0.13	15.69%	739	11.57%
Nodes: 79	LFA	1.57	54.05%	4762	77.28%	1.05	47.57%	3605	56.44%
Degree: 3.72	LFA + U-Turns	2.14	65.85%	5407	87.74%	1.42	54.73%	4334	67.85%
Src-Dst Pairs: 6162	Tunnels-1	1.58	53.61%	4818	78.18%	1.01	44.41%	3530	55.26%
	Tunnels-2	9.30	99.28	6154	99.87%	3.94	83.89%	4837	75.73%
	Tunnels-3	20.55	100.00%	6155	99.88%	8.47	94.44%	5199	81.39%
	Rule 2			3319	53.85%			2641	41.35%
Abovenet(6461)	ECMP	0.23	25.05%	3370	17.81%	0.18	22.75%	2755	14.35%
Nodes: 141	LFA	3.19	81.02%	17278	91.36%	2.19	72.42%	14593	76.04%
Degree: 5.39	LFA + U-Turns	3.95	85.27%	17590	93.00%	2.68	76.89%	14958	77.94%
Src-Dst Pairs:	Tunnels-1	3.23	84.51%	17000	89.89%	2.17	74.74%	14304	74.53%
19740	Tunnels-2	18.10	99.96%	18898	99.92%	7.99	95.96%	16088	83.83%
	Tunnels-3	43.19	99.97%	18898	99.92%	17.04	97.53%	16358	85.24%
	Rule 2			13764	72.77%			12475	65.00%