

Optimizing IGP Link Costs for Improving IP-level Resilience

Gábor Rétvári, Levente Csikor, János Tapolcai
Dept. of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Email: {retvari, csikor, tapolcai}@tmit.bme.hu

Gábor Enyedi, András Császár
TrafficLab,
Ericsson Research
Email: {gabor.sandor.enyedi, andras.csaszar}@ericsson.com

Abstract—Recently, major vendors have introduced new router platforms to the market that support fast IP-level failure protection out of the box. The implementations are based on the IP Fast ReRoute–Loop Free Alternates (LFA) standard. LFA is simple, unobtrusive, and easily deployable. This simplicity, however, comes at a severe price, in that LFA usually cannot protect all possible failure scenarios. In this paper, we give new graph theoretical tools for analyzing LFA failure case coverage and we seek ways for improvement. In particular, we investigate how to optimize IGP link costs to maximize the number of protected failure scenarios, we show that this problem is NP-complete even in a very restricted formulation, and we give exact and approximate algorithms to solve it. Our simulation studies show that a deliberate selection of IGP costs can bring many networks close to complete LFA-based protection.

I. INTRODUCTION

The IP protocol suite has come a long way to become a viable bearing platform for commercial telecom services. However, there still exist missing components that make it difficult to sustain the transmission quality required by multimedia applications, like VoIP, IPTV, online gaming, etc., in a pure IP or MPLS/LDP environment. Perhaps the most prominent issue is the slow reaction to device and link failures. Interior Gateway Protocols (IGPs), like OSPF or IS-IS, adopt a restoration-based resilience approach, using a global flooding of failure information and a lengthy network-wide re-convergence process. In order to achieve a sub-50 ms convergence time essential for most multimedia applications, one needs to go beyond conventional restoration and invoke a protection-based, proactive, local recovery method, called IP Fast ReRoute (IPFRR, [1]). In IPFRR, routers precompute alternate next-hops and traffic is instantly switched to these secondary next-hops should the primary next-hop become unavailable. This ensures that traffic flows without interruption until the IGP converges in the background.

Unfortunately, combining IP’s destination-based forwarding with protection is difficult. Therefore, many IPFRR proposals require alterations to destination-based forwarding itself [2], or introduce some forms of in-band or out-of-band signaling mechanism for failure notification [3]–[5], or use tunnels to route around the failed component [6]–[8]. Deploying these

IPFRR mechanisms, therefore, would either demand non-trivial modifications to the essential IP infrastructure or impose considerable management burden on network operations [9] (or both), making network device vendors reluctant to implement them and discouraging operators from deploying IPFRR.

To our days, only a single IPFRR specification has found its way into commercial IP routers: Loop Free Alternates (LFA, [10]). LFA is as simple as it can get: traffic impacted by a failure is passed on to an alternate next-hop (called a Loop Free Alternate) that still has an intact path to the destination. LFA can be implemented with straightforward software upgrades, and so it can be deployed incrementally. Simplicity and deployability, however, comes at a significant price: depending on the network topology and IGP link costs, very often not all routers have LFAs to all destinations, making it impossible to repair certain failure scenarios rapidly.

Consequently, many operators are hesitating to enable LFA, trying to measure the expected benefits against the additional costs. In this paper, we seek ways to assist in making this important decision. In the first part, we give new graph theoretical tools for analyzing LFA failure case coverage in operational networks. Similar protectability analyses are already available for some non-standardized IPFRR mechanisms: [11] considers the O2 method and [12] discusses a centralized destination-based routing scheme. For LFA, only simulation-based reports have been available this far [13]–[16], but a mathematical apparatus for LFA coverage analysis is still missing. We took the initial steps towards this goal in [17], and in this paper we bring that work further.

Initial deployments confirmed that in many operational networks LFA indeed does not guarantee protection for all failure scenarios. There are various ways to overcome this. One is to alter the network topology (which problem we treated in [17]) and the other we focus on in this paper is altering link costs. In particular, we ask how a network operator can adjust IGP link costs in order to maximize LFA-based resilience. While improving IP resilience is a recurring theme in the literature (see [18] for deflection routing, [11] for O2, or [12] for a review), for the specific case of LFA only the joint optimization of network performance and resilience has been investigated previously [19], [20]. Thus, at the moment very little understanding is available as to how much LFA-based IP Fast ReRoute is suitable to protect an IP network

and to what extent this can be improved by optimizing link costs.

After reviewing the related literature in Section II and introducing the notations and the model in Section III, we first discuss LFA failure coverage analysis (see Section IV) and then, in Section V, we turn to discuss the LFA cost optimization problem. We show that even a very minimalistic formulation of the problem is already NP-complete, and we give exact and heuristic algorithms to solve it. In Section VI, we evaluate the proposed algorithms numerically and finally we conclude our work with Section VII.

II. RELATED WORKS

The IP Fast ReRoute framework was initiated by the Internet Engineering Task Force in [1], and the Loop Free Alternates standard, as the basic specification for IPFRR, was subsequently documented in [10]. IPFRR is not only targeted into pure IP networks, but forwarding mechanisms that also rely on the IP control plane for routing information could also benefit from it. Most notable amongst these is MultiProtocol Label Switching using the Label Distribution Protocol for label management.

It was from the very beginning made clear by the IETF that LFA does not guarantee fast protection for all possible failure scenarios in all network topologies. This was later confirmed by extensive simulation studies, which indicated that, depending on the topology and link cost settings, LFA can usually protect only about 50-80% of the possible link failure scenarios, and the level of node protection is even worse [13]–[15], [21]. These LFA coverage analyses are all *quantitative* studies, based on calculating the LFA coverage for various real-life network topologies. Perhaps the most detailed amongst these is [16], which inspects the applicability of LFA in common access network topologies. So far, no *qualitative* analyses have been available in the literature, which would help uncover the graph theoretical ingredients needed for good LFA coverage. We initiated the work in that direction in [17], and in this paper we refine our earlier results significantly. Possibly the closest to ours is the study in [12], where the authors perform a qualitative protectability analysis for a fast resilience scheme they call IP protection routing. Protection routing is appealing for such an analysis as it is theoretically much easier to approach than LFA, however, in practice it is somewhat less attractive as implementing it requires centralized control over the routing tables.

Since the appearance of the original LFA draft, many IPFRR proposals have surfaced. Implicit in these proposals is the recognition that in order to protect all failure scenarios one either needs to go beyond standard IP forwarding and/or apply some forms of explicit failure notification mechanism. The reason for this is that a router must give special treatment to packets traveling on a detour around a failure, or otherwise forwarding loops will arise in certain failure scenarios.

Most IPFRR proposals choose the former option and intervene at the level of IP packet forwarding. Failure Insensitive Routing [2], [22], [23] differentiates packets based

on the incoming interface they arrive through, letting the router to guess the failure's location from the direction of the received packets and exploit this information in the course of packet forwarding. Multiple Routing Configurations [5] call to achieve the same goal with explicit packet marking, while other proposals, like Not-via Addresses, use tunnels to this end [6]–[9]. Unfortunately, the former solution would allocate invaluable bits in the IP header, while the latter might cause painful packet fragmentation and time-consuming reassembly at the tunnel endpoint if the additional IP header does not fit into the MTU. Deflection routing for fast rerouting purposes is proposed in [18], while O2 routing, a resilient multi-path data forwarding method, is specified in [24]. Both require non-standard IP forwarding functionality, unavailable in commercial routers at the moment.

A different approach is to use explicit signaling to notify routers about failures [3], [25]. This avoids having to modify standard IP forwarding at the price of a establishing a separate signaling mechanism just for IPFRR. Proposals also exist to combine different IPFRR mechanisms to achieve full protection [21]. Good overviews on IPFRR are [13] and [15].

So far, only one IPFRR method has found its way into commercial routers, and hence into operational IP networks: LFA. Due to its appealing simplicity, operators can deploy LFA incrementally without any particular staff training, and no major alterations to installed IP hardware and software. Therefore, at least two major vendors are already providing LFA out of the box [26], [27], and other vendors are expected to follow suit.

Finding methods to design or optimize networks in an attempt to improve fast resiliency has been an actively researched topic lately. In the recent literature, [18] seems to be the first reference that, besides motivating the need for fast IP resilience with detailed failure case analysis in an operational backbone, proposes a method to improve the robustness of the network against such failures. Theory and algorithms for topology optimization for O2 are presented in [11], and a generic approach for protection routing is given in [12]. Apart from our study in [17], the only attempts at LFA-oriented network optimization seem to be [19] and (partly) [20].

A common theme shared by most approaches is that (with the exception of [11] and [17]) each one addresses the joint optimization of network resilience and routing performance simultaneously. The former aims at better protection against failures, while the latter is called to minimize congestion and distribute load evenly in the network with respect to some known, measured or predicted, traffic matrix [28]. A good example of this approach is [19], where the authors formulate the joint LFA cost optimization and traffic engineering problem as a constraint-programming task and feed it into a generic solver. We believe that this approach has several drawbacks. First, good traffic matrices are difficult to come by, and this is even more so today as traffic is becoming extremely dynamic and unpredictable. Many modern traffic engineering methods, therefore, completely eliminate the dependence on traffic matrices [29]. Second, there may be operational goals

more important than mere load balancing [30], [31], and most existing proposals leave these out of consideration. But most importantly, solving the joint problem leaves the particularities of the individual subproblems, their computational complexity and algorithmic aspects, in obscurity. For instance, the authors in [19] claim that the joint problem is NP-complete because OSPF traffic engineering in itself is already NP-complete [32], without ever getting to know anything about the computational complexity of LFA cost optimization alone.

We think that our approach, a rigorous separation of performance maximization and LFA cost optimization, allows a deeper understanding of the problem. It lets us to treat LFA cost optimization as a standalone optimization problem, determine its complexity and give efficient algorithms. This then leads to good insight into the inherent limitations of LFA-based IP Fast ReRoute and the extent to which optimizing costs just for the purpose of IPFRR can improve the resilience in IP networks. We believe that only after understanding the fundamental trade-offs involved in LFA-based IPFRR should we take the next step and address operational issues, like traffic engineering, in network optimization.

III. MODEL AND PROBLEM FORMULATION

We model the network with a connected, undirected graph $G(V, E)$, the set of nodes is denoted by V ($|V| = n$) and set of edges by E ($|E| = m$). Let N_i denote the set of neighbors of some node $i \in V$. For simplicity, we assume that the network consists of point-to-point links only and contains no broadcast LANs and Shared Risk Link Groups (SRLGs). IGP link costs are represented by an edge cost function $c : E \mapsto \mathbb{Z}^+$. The cost of an edge (i, j) is denoted by $c(i, j)$. The model assumes that costs are symmetric. We presume that $G(V, E)$ and the cost function c are readily available to the network nodes through the IGP, using which all routers can compute the shortest path distance between any two routers in the network. Denote the distance from node i to node j with $\text{dist}(i, j)$.

Fig. 1 shows a sample network, with costs indicated near the edges and shortest paths towards node f marked by arrows. For instance, node b 's next-hop along the shortest path to node f is node e . Should the link from b to its next-hop e become unavailable, b can safely switch to an alternate next-hop, in this case node d , even without explicitly notifying it about the failure, as d will never send packets destined to f through b so no loop can arise. We say that for some source s and destination d , a neighbor n of s that is not the next-hop of s towards d is a *link-protecting LFA* if [10]:

$$\text{dist}(n, d) < \text{dist}(n, s) + \text{dist}(s, d) . \quad (1)$$

That is, any neighbor that is not an upstream in the shortest path tree is a link-protecting LFA. Besides node b , e also has an LFA to f (the same d as that of b), and so has d and c (e and d , respectively). What is more, the LFAs of b and c are node-protecting as well, as they protect against both the failure of the link to the next-hop and the next-hop itself. Moreover, d is also called a per-link LFA for b , as it protects all nodes reachable from b through the link (b, e) . For a full taxonomy, see [10],

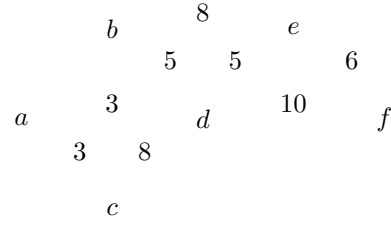


Figure 1: Sample network, edge costs and shortest paths to node f .

[16]. As single link failures account for the majority (about 70%) of unplanned outages in a generic network [33], we shall treat only this type of failures in the sequel. Consequently, the term LFA will refer to link-protecting LFAs exclusively. Other LFA types are for further study.

We observe that, in the present network topology with the given link costs, node a does not have an LFA to f . This is because it has only two neighbors, one is the next-hop d towards f whose failure we want to protect, and the other is an upstream node, which cannot provide an LFA by (1). Given a graph $G(V, E)$ and a cost function c , let $I_{s,d}(G, c)$ be an indicator variable whose value is 1 if node s has an LFA to node d , and zero otherwise. Then, given a set of source-destination pairs $\mathcal{S} = \{(s_k, d_k) : k \in 1, \dots, K, s_k \neq d_k\}$ the LFA coverage with respect to \mathcal{S} is defined as (inspired by [10]):

$$\eta_{\mathcal{S}}(G, c) = \frac{1}{|\mathcal{S}|} \sum_{(s,d) \in \mathcal{S}} I_{s,d}(G, c) .$$

We shall often confine ourselves to the special cases when \mathcal{S} is the set of all node pairs whose destination is a given terminal node d : $\mathcal{S}_d = \{(s, d) : s \in V \setminus \{d\}\}$, or when \mathcal{S} contains all distinct node pairs in $V \times V$. In the latter case, we shall neglect to indicate \mathcal{S} in the LFA coverage metric and simply write $\eta(G, c)$.

As our example shows, usually not all nodes have LFA to all destinations. There are basically two ways to remedy this: by adding new edges to the graph or by altering the edge costs. Taken the example of Fig. 1, adding the new edge (a, b) to E and setting its cost to, say, 10, will let b to become an LFA of a (and vice versa). The *LFA graph extension* problem asks, how to achieve maximal LFA protection by adding the minimum number of new edges. We address this problem in a separate paper [17]. The other way is to change edge costs: if we, for instance, reduce the cost of edge (c, d) from 8 to 5, then c 's shortest path to f will bypass a and so a and c will become LFAs for each other. This paper is devoted to investigate this very problem, called the *LFA cost optimization* problem:

Definition 1: $\text{LFACostOpt}(G, \mathcal{S})$: Given a graph $G(V, E)$ and a set of source-destination pairs \mathcal{S} , is there a cost function c so that $\eta_{\mathcal{S}}(G, c) = 1$?

We shall in many cases treat the optimization version of $\text{LFACostOpt}(G, \mathcal{S})$, that is, we shall seek the costs that maximize network-wide LFA coverage.

IV. LFA FAILURE COVERAGE ANALYSIS

Before turning to discuss how to solve the LFA cost optimization problem, first we show some simple theoretical limits on LFA coverage, intended to serve as a guideline for network operators to quickly assess the LFA-protectability of their network. In particular, we give simple graph theoretical lower and upper bounds on the LFA coverage achievable in a given graph under *any* selection of link costs. In what follows, we shall assume that $S = (V \times V) \setminus \{(v, v) : v \in V\}$.

Some preliminaries. Let Δ denote the average node degree in G and let Δ_{\max} be the maximum degree. Easily, $\Delta \geq \frac{2(n-1)}{n}$ for any connected graph, since the sparsest connected graphs are trees for which $\Delta = \frac{2(n-1)}{n}$. A Δ -regular graph is a graph in which all nodes are of constant degree Δ . An even (odd) ring is a cycle graph with an even (odd) number of nodes. Rings are the smallest-degree 2-edge-connected regular graphs (in particular, $\Delta = 2$).

In [17], we identified the following fundamental lower and upper LFA coverage bounds.

Proposition 1: The LFA coverage in a 2-edge-connected graph $G(V, E)$ on n nodes ($n \geq 3$) is bounded by $\frac{1}{n-1} \leq \eta(G, c) \leq 1$, and the lower bound is tight for even rings and uniform edge costs. For odd rings, $\eta(G, c) = \frac{2}{n-1}$ with c uniform.

In the rest of this section, we discuss how to sharpen the above bounds. The idea is that the shortest path tree to some destination d can contain only $n - 1$ edges, and all further edges provide at least 1, and at most 2, nodes with LFAs towards d . Consider the following lemma.

Lemma 1: For any connected simple graph G with $n > 2$, $\eta(G, c) \leq \frac{n}{n-1}(\Delta - 2) + \frac{2}{n-1}$.

Proof: An edge not contained in the shortest path tree rooted at some d provides at most 2 LFAs towards d . This occurs when the edge lies between two branches of the tree. Since the number of such out-of-tree edges is exactly $m - (n - 1)$, at most $2(m - n + 1) = n\Delta - 2n + 2 = n(\Delta - 2) + 2$ nodes can have LFA to d . Taken the sum over all nodes and dividing by the number of source-destination pairs gives $\eta(G) \leq \frac{n(n(\Delta-2)+2)}{n(n-1)} = \frac{n}{n-1}(\Delta - 2) + \frac{2}{n-1}$. ■

The Lemma is non-trivial for $\frac{2(n-1)}{n} \leq \Delta < 3$. For trees, in particular, we obtain $\eta(G, c) \leq 0$, which implies that the Lemma is tight for trees over arbitrary link costs. It is tight for uniform cost odd rings as well, for which we obtain $\eta(G, c) \leq \frac{2}{n-1}$ (c.f., Proposition 1).

Lemma 2: For any connected simple graph G with $n > 2$, $\eta(G, c) \geq \frac{n}{n-1} \frac{\Delta-1}{\Delta_{\max}-1} + \frac{1}{(n-1)(\Delta_{\max}-1)}$.

Proof: Again, exactly $n - 1$ nodes are contained in the shortest path tree of d , and an out-of-tree edge (of which we have $m - n + 1$) can provide at least one LFA towards d (if the edge is inside a single branch of the shortest path tree, then it provides LFA from the upstream to the downstream). So there are $m - n + 1$ out-of-tree edges that are incident to at least $\frac{m-n+1}{\Delta_{\max}-1} = \frac{n(\frac{\Delta}{2}-1)+1}{\Delta_{\max}-1}$ nodes providing LFA to them towards d ($\Delta_{\max} - 1$ because every node has at least one in-tree edge, so only the rest count as out-of-tree edges). Taking the sum

over all nodes and dividing by $n(n - 1)$ gives the required result. ■

Corollary 1: For a Δ -regular graph R_Δ on n nodes, $\eta(R_\Delta, c) \geq \frac{1}{2} - \frac{1}{2} \frac{n-\Delta-1}{(n-1)(\Delta-1)}$.

This gives $\eta(R_2, c) \geq \frac{1}{n-1}$ and $\eta(R_3, c) \geq \frac{1}{4} + \frac{3}{4} \frac{1}{n-1} > \frac{1}{4}$. From this, we conclude that the lower bound of Lemma 2 is tight for even rings (again, by Proposition 1). One easily sees that it is tight for trees as well, for which we get $\eta(G, c) \geq 0$.

The above analysis helps us identify an interesting extreme case for LFA coverage. In particular, we find that the 2-connected graph with the smallest possible average degree that can be fully protected using LFA is the 3-ring C_3 . Every other 2-connected graph with complete LFA coverage has average degree higher than 2. From Proposition 1, we have $\eta(C_3, c) = 1$, which is attained when c is uniform, and one easily sees that $\eta(C_3, c)$ is the only 2-connected graph of average degree $\Delta = 2$ with this property. Graphs with $\Delta < 2$ cannot have full protection because such graphs contain at least one node with degree 1 whose single outgoing link can never be protected. On the other hand, larger 2-connected graphs with $\Delta = 2$ are all ring topologies, and rings can only have full LFA coverage if $n = 3$ (again, by Proposition 1).

V. LFA COST OPTIMIZATION

Next, we turn to the LFA cost optimization problem. This problem asks for an IGP link cost setting that maximizes the LFA coverage, given the inherent limitations of the network topology under consideration. First, we characterize the extent to which such an optimization can improve LFA coverage, then we discuss the complexity and the algorithmic aspects of the problem.

A. The potential of LFA cost optimization

The question immediately arises as to whether it is worth optimizing costs for LFA at all. Easily, readjusting costs in most of the cases alters, possibly in a negative way, default shortest paths, which might have been previously tweaked with great accuracy to match the needs of the network in terms of load balancing, traffic engineering, etc. [28], [30], [31]. On the other hand, as shall be shown through an example below, the wins achievable with optimizing link costs for LFA can be substantial (more than 50%), and such a huge improvement in fast resiliency might compensate for the losses in forwarding efficiency in certain cases.

Consider the so called ‘‘Möbius ladder’’ topologies depicted in Fig. 2. These graphs consist of an even ring with all the main diagonals added. In Fig. 2a, the cost of diagonals is chosen so that the path between any two nodes is shorter around the ring than through it via a diagonal. This way, as one easily checks, the graph has complete LFA coverage. The graph construction can be generalized to arbitrary even n , and one can always choose the above cost setting strategy to achieve complete LFA protection. Fig. 2b also depicts a Möbius ladder (for $n = 10$), just with setting costs uniformly at all edges and drawn in a slightly awkward layout. The layout was chosen so that one can easily check the validity of the following claim for any

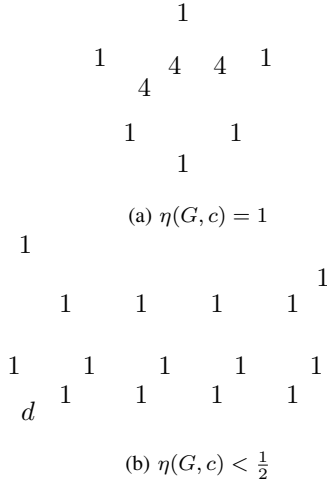


Figure 2: Möbius ladder topologies.

Möbius ladder with $\frac{n}{2}$ odd, $n > 2$ and c uniform: for every $d \in V$, exactly $\frac{n}{2} - 1$ nodes have LFA. Considering the node d we marked in Fig. 2b, there is exactly one node in each “column” that has an LFA to d , except for the column of d in which there is no protected node. This gives $\eta(G, c) = \frac{1}{2} - \frac{1}{2} \frac{1}{n-1} < \frac{1}{2}$. For instance, in our example $\eta(G, c) = \frac{4}{9}$.

This example shows that different selections of edge costs can produce dramatical differences in LFA failure case coverage. Simulation studies presented later also seem to support this claim. The other lesson is that resilience and forwarding efficiency are usually contradicting requirements in routing: in our example in the latter case all traffic flows along min-hop paths but resilience is poor, while in the former case we have full protection but long forwarding paths going around the ring instead of taking the shortcuts through it. Such “joker” links that do not carry traffic seem a general requirement for protectability [11].

B. Complexity

Next, we turn to discuss how to solve the LFA cost optimization problem as of Definition 1. First, we characterize the computational complexity of the problem.

Theorem 1: The LFA cost optimization problem $\text{LFACostOpt}(G, \mathcal{S})$ is NP-complete.

This result is not particularly unexpected, as we found basically all other LFA-related network optimization problems NP-complete [17]. Taking a closer look, we find that there are two reasons due to which the problem is difficult. First, there is an inherent coupling between the LFAs to different destinations through the link costs, which makes it difficult to make independent decisions. In particular, assigning a neighbor as an LFA towards some destination necessitates adjusting edge costs accordingly, but this may destroy LFAs to other destinations. Second, even assigning LFAs to *just a single destination* seems difficult enough. Consider the following theorem.

Theorem 2: Given a graph $G(V, E)$ and a node $d \in V$,

$\text{LFACostOpt}(G, \mathcal{S}_d)$ with $\mathcal{S}_d = \{(s, d) : s \in V \setminus \{d\}\}$ is NP-complete.

For a complete proof, see the Appendix.

Obviously, Theorem 2 proves Theorem 1 stated for the general case $\text{LFACostOpt}(G, \mathcal{S})$ as well, of which $\text{LFACostOpt}(G, \mathcal{S}_d)$ is a special case. Additionally, we also observe that the optimization version, which asks for a cost maximizing LFA coverage, is also intractable.

C. Algorithms

LFA cost optimization is difficult, yet solving it would be extremely useful for improving the resilience in operational IP networks. Next, we give an Integer Linear Program (ILP) suitable for obtaining optimal solutions only in small networks, then we discuss a heuristics better suited to large networks. For simplicity, we assume that \mathcal{S} contains all distinct node-pairs (even though the algorithms are easy to generalize to arbitrary \mathcal{S}).

The ILP is formulated in the dual space: to every node i we assign a node potential π_i^d that signifies the shortest distance from i to some d over the costs c , and then we require that the potentials and the costs together fulfill the Shortest Path Optimality Criteria [34] while also maximizing LFA coverage.

$$\max \sum_{(s,d) \in \mathcal{S}} \alpha_s^d \quad (2)$$

$$\pi_j^d + s_{ij}^d = \pi_i^d + c_{ij}, \quad 0 \leq s_{ij}^d \leq C y_{ij}^d \quad \forall (s, d) \in \mathcal{S}, \forall (i, j) \in E \quad (3)$$

$$\sum_{v \in N_s} y_{sv}^d \leq |N_s| - 1 \quad \forall (s, d) \in \mathcal{S} \quad (4)$$

$$y_{sv}^d \in \{0, 1\} \quad \forall (s, d) \in \mathcal{S}, \forall v \in N_s \quad (5)$$

$$\pi_v^s - \pi_s^s + \pi_s^d - \pi_v^d + z_{sv}^d \leq 0, \quad 0 \leq z_{sv}^d \leq 1 \quad \forall (s, d) \in \mathcal{S}, \forall v \in N_s \quad (6)$$

$$\sum_{v \in N_s} z_{sv}^d \geq \alpha_s^d, \quad 0 \leq \alpha_s^d \leq 1 \quad \forall (s, d) \in \mathcal{S} \quad (7)$$

$$c_{ij} = c_{ji}, \quad c_{ij} \in \{1, \dots, C_{\max}\} \quad \forall (i, j) \in E \quad (8)$$

In the ILP, (3)–(5) enforce the Shortest Path Optimality Criteria: for each edge (i, j) , $\pi_j^d \leq \pi_i^d + c_{ij}$ and the inequality is satisfied with strict equality for at least one neighbor. This is to ensure that the node potentials π_i^d correctly encode the shortest path distances with respect to the destination node d over the cost setting c . Furthermore, (6)–(7) represent the LFA condition as of (1): by (6) z_{sv}^d is an indicator variable whose value is positive if and only if v is an LFA from s to d , and (7) ensures that α_s^d only becomes positive if at least one neighbor of s provides LFA towards d . The requirements (8) guarantee that costs are symmetric and are selected from the interval $\{1, \dots, C_{\max}\}$. Finally, the objective function (2) maximizes the number of LFA protected node pairs. There are two problem parameters to the ILP: C_{\max} is the maximum permitted cost, while $C \geq nC_{\max}$ is the maximum allowed potential difference between two neighboring nodes.

The ILP has $O(n^3)$ integer variables, which makes it intractable in anything but the smallest topologies. Therefore,

Algorithm 1 Heuristic LFA cost optimization algorithm. Input is graph G .

```

1:  $c \leftarrow \text{random\_cost}(C_{\max}), T \leftarrow T_0$ 
2: while  $T > 0$  and  $\eta(G, c) < 1$ 
3:    $c' \leftarrow \text{argmax}_{q \in \text{neigh}(c)} \eta(G, q)$ 
4:   if  $\eta(G, c') > \eta(G, c)$  or  $T > \text{random}(T_0)$  then
5:      $c \leftarrow c'$ 
6:   end if
7:    $T \leftarrow T - 1$ 
8: end while

```

we also present an approximate algorithm roughly modeled after the Simulated Annealing probabilistic metaheuristic. The idea is to, starting from a randomly chosen cost c , search for the best c' “nearby” c and accept c' if either c' provides larger LFA coverage than c (greedy step) or c' is worse than c but the temperature T of the system is sufficiently large (escape from a local minimum). As the algorithm progresses we gradually reduce T , thus the system will increasingly tend to get stuck in a good quality local minimum.

The pseudo-code for the approximate algorithm is given in Alg. 1. The subroutine `random_cost(C)` returns a random initial cost in the range $\{1, \dots, C_{\max}\}$ for each link. The routine `neigh(c)` returns a positive, integral cost setting obtained by increasing or decreasing (if possible) the cost c at exactly one edge by 1. Line 3 searches for the best such neighbor. We unconditionally accept this cost if it is better than the previous one. Additionally, we also accept it if a random number generated in the range $[1, T_0]$ by the subroutine `random(T0)` is below T . Thus, the algorithm easily escapes from local minima initially, to eventually settle in a good local minimum by only letting greedy steps when T is low. The input to the heuristic is the graph $G(V, E)$, initial temperature T_0 and maximum allowed cost C_{\max} , and the output is the final cost c . The complexity of the algorithm is $O(T_0 mn^3)$, dominated by the need to evaluate $\eta(G, q)$ (needing $O(n^3)$ steps) in each iteration for each $2m$ neighbor q of the current cost c .

VI. NUMERICAL EVALUATIONS

In the course of our numerical studies, first we were curious as to how close the approximate LFA cost optimization algorithm can get to the optimum. Therefore, we implemented the ILP (3)–(8) and the heuristics as described in Alg. 1. We found that about the largest non-trivial graphs for which the ILP can be solved are of 8 nodes. Unfortunately, very few real topologies of this small size are available in the literature. Thus, the first round of the evaluations were run on Erdős-Rényi random graphs ($n = 8$, expected node degree 3). Out of the 20 random graphs generated, 17 was 2-connected. For each topology, the simulated annealing was executed 500 times ($T_0 = 150$, $C_{\max} = 20$) and the cost c^* that attained the highest LFA coverage was selected. A tabu list of size 20 was also applied in order to preclude the heuristics from

oscillating. Table I gives some characteristics of the graphs (number of nodes n , and number of links m); the theoretical lower and upper bounds on LFA coverage (as of Lemma 1 and Lemma 2); and the actual LFA coverage $\eta(G, c_{\text{opt}})$ for the costs c_{opt} obtained by the ILP and the heuristics ($\eta(G, c^*)$). We observe that from the 17 experiments only in 2 cases the approximation did not find the optimum (these experiments are marked by an asterisk in Table I), and the difference is at most 2-3% in LFA coverage. This indicates that in small networks the simulated-annealing-based heuristics performs quite efficiently. Additionally, we found that the theoretical bounds provide a solid estimate on the LFA coverage. Especially the upper bound seems to be of practical relevance.

In the second round, we examined the performance of the approximate LFA cost optimization algorithm in larger real network topologies where the ILP could not be solved to optimality. We used inferred ISP data maps from the Rocketfuel dataset [35] (AS1221, AS1239, AS1755, AS3257, AS3967 and AS6461). We obtained approximate POP-level maps by collapsing the topologies so that nodes correspond to cities and we eliminated leaf-nodes (this preprocessing method was suggested in [29]). These networks come with inferred link costs (these costs are needed to compute the “default” LFA coverage $\eta(G, c)$ of the network). We also chose some network topologies from [36], namely, the Abilene, Italy, Germany, NSF and AT&T networks and the 50 node extended German backbone (Germ_50). Unfortunately, except for the last network no valid link costs were available, so we set each cost to 1. We also chose some representative ISP topologies from [37], in particular, the Arnes, Deltacom, Geant, and the InternetMCI topologies. Link costs were set inversely proportional to the link capacities (this setting is recommended by Cisco, see documentation on `ospf auto-cost` in [38]). Additionally, we also ran the evaluations on some artificial topologies with uniform costs. In particular, M_n are the Möbius ladder graphs of n nodes as discussed in Section V.

Table II shows, in the order of the appearance: the characteristics of the topologies (name, number of nodes n and edges m , and the average node degree Δ); the LFA coverage obtained by the original link cost setting for the graphs; and the LFA coverage $\eta(G, c^*)$ for the best cost function c^* obtained by the approximate algorithm. There was only one topology on which we could solve the ILP to optimality: AS1221. For this particular network, the approximate solution matches the ILP optimum ($\eta(G, c_{\text{opt}}) = \eta(G, c^*) = 0.833$).

Our observations are as follows. First, we found that the LFA coverage $\eta(G, c^*)$ produced by the approximate algorithm is usually significantly higher than the LFA coverage produced by the network’s original cost setting. The improvement almost always exceeds 5%, but in many cases it attains about 15-20% (e.g., AS1239, AS3967, or the Italian backbone). This suggests that optimizing costs specifically for LFA usually attains significant improvement in network resilience. The improvement is especially significant for the artificial networks. Second, for large Möbius ladder graphs the approximation could not get closer than 10% to the optimum

Table I: LFA cost optimization in random topologies.

Num	n	m	Lower/Upper	$\eta(G, c_{\text{opt}})$	$\eta(G, c^*)$
1*	7	11	0.278/1	1	0.976
2	8	9	0.095/0.571	0.536	0.536
3*	8	13	0.214/1	1	0.982
4	7	11	0.278/1	1	1
6	8	9	0.143/0.571	0.571	0.571
9	7	11	0.208/1	0.952	0.952
10	8	11	0.114/1	0.857	0.857
11	8	10	0.143/0.857	0.75	0.75
12	8	9	0.095/0.571	0.429	0.429
13	8	11	0.143/1	0.911	0.911
14	8	11	0.19/1	0.821	0.821
15	8	11	0.19/1	0.946	0.946
16	7	8	0.111/0.667	0.5	0.5
17	8	14	0.2/1	1	1
18	8	11	0.114/1	0.714	0.714
19	8	9	0.143/0.571	0.482	0.482
20	8	10	0.143/0.857	0.679	0.679

Table II: LFA cost optimization in real and artificial topologies.

Name	n	m	Δ	$\eta(G, c)$	$\eta(G, c^*)$
AS1221	7	9	2.57	0.809	0.833
AS1239	30	69	4.60	0.873	0.957
AS1755	18	33	3.66	0.872	0.98
AS3257	27	64	4.74	0.923	0.997
AS3967	21	36	3.42	0.785	0.967
AS6461	17	37	4.35	0.933	0.996
Abilene	12	15	2.5	0.56	0.701
Italy	33	56	3.39	0.784	0.919
Germany	17	25	2.94	0.695	0.889
NSF	26	43	3.3	0.86	0.95
AT&T	22	38	3.45	0.822	0.984
Germ_50	50	88	3.52	0.9	0.934
Arnes	41	57	2.78	0.623	0.702
Deltacom	113	161	2.85	0.577	0.662
Geant	37	55	2.97	0.69	0.74
InternetMCI	19	33	3.47	0.904	0.932
M_6	6	9	3	0.4	1
M_{10}	10	15	3	0.444	0.933
M_{18}	18	27	3	0.470	0.879
M_{30}	30	45	3	0.482	0.89

(which we know is $\eta(G, c_{\text{opt}}) = 1$ in this case). This indicates that in larger topologies the efficiency of the heuristics we identified in small networks might not be present. Last but not least, we observe that the final LFA coverage $\eta(G, c^*)$ in many real network topologies is more than 95%. The denser the network, the higher the LFA coverage. It seems that networks with an average node degree exceeding about 3.5 lend themselves especially well to LFA cost optimization (AS1239, AS1755, AS3257, AS6461, AT&T, Germ_50): in these networks even the default cost settings yield a higher than 80% LFA coverage and our cost optimization tool can bring these networks well beyond 95% and close to 100% in many cases. Networks of average degree 3 are still amenable to

LFA, but when the degree falls below 3 the chances of getting a high LFA coverage rapidly vanish. For sparser networks (like the Abilene topology), the final LFA coverage $\eta(G, c^*)$ is a mere 70%. These observations are in line with our theoretical analysis in Section IV. Note, however, that node degree alone is not sufficient to assess the extent to which LFA can protect a network, as there are topologies (the Möbius ladder graphs) that have small average degree of 3 but still complete LFA protection over some appropriately chosen costs. It seems that LFA cost optimization is most difficult when the degree is about 3.

Our results suggest that most real network topologies, which are usually richly connected and highly redundant, lend themselves readily to LFA cost optimization and almost perfect LFA coverage can be achieved in most of the cases. There were, however, some exceptional topologies where LFA cost optimization was less appealing. For such networks, LFA is not an acceptable option and operators need to look after more efficient alternatives [39].

VII. CONCLUSIONS

In this paper, we have assessed the possibilities of improving fast resilience in operational IP networks using the Loop Free Alternates method. The motivation for choosing LFA over its alternatives is its simplicity, easy deployability, and availability in IP routers. We presented new tools to quickly estimate LFA failure case coverage and we sought ways to improve it by carefully adjusting IGP link costs. We showed that this problem is NP-complete and we proposed a simplistic simulated-annealing-based approximation, using which we could achieve close to perfect LFA coverage in many real-world network topologies. Considering that LFA is just a router-configuration command away in many modern IP networks, we believe that this result has huge practical relevance. Nevertheless, we also found that some topologies are less amenable to LFA cost optimization. Future work involves combining the LFA network optimization tools we gave in [17] and the algorithms presented herein to improve IP-level fast resilience in such notorious network topologies.

REFERENCES

- [1] M. Shand and S. Bryant, "IP Fast Reroute framework," RFC 5714, Jan 2010.
- [2] S. Lee, Y. Yu, S. Nelakuditi, Z.-L. Zhang, and C.-N. Chuah, "Proactive vs reactive approaches to failure resilient routing," in *INFOCOM*, 2004.
- [3] I. Hokelek, M. Fecko, P. Gurung, S. Samtani, S. Cevher, and J. Sucec, "Loop-free IP Fast Reroute using local and remote LFAPs," Internet Draft, Feb 2008.
- [4] A. Li, X. Yang, and D. Wetherall, "SafeGuard: safe forwarding during route changes," in *ACM CoNEXT*, 2009, pp. 301–312.
- [5] A. Kvalbein, A. F. Hansen, T. Čičić, S. Gjessing, and O. Lysne, "Multiple routing configurations for fast IP network recovery," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 473–486, 2009.
- [6] S. Bryant, C. Filsfils, S. Previdi, and M. Shand, "IP Fast Reroute using tunnels," Internet Draft, Nov 2007.
- [7] S. Bryant, M. Shand, and S. Previdi, "IP fast reroute using Not-via addresses," Internet Draft, March 2010.
- [8] G. Enyedi, P. Szilágyi, G. Rétvári, and A. Császár, "IP Fast ReRoute: lightweight Not-Via without additional addresses," in *INFOCOM Mini-conf*, 2009.

- [9] A. Li, P. Francois, and X. Yang, "On improving the efficiency and manageability of NotVia," in *ACM CoNEXT*, 2007.
- [10] A. Atlas and A. Zinin, "Basic specification for IP fast reroute: Loop-Free Alternates," RFC 5286, 2008.
- [11] C. Reichert and T. Magedanz, "Topology requirements for resilient IP networks," in *MMB*, 2004, pp. 379–388.
- [12] K.-W. Kwong, L. Gao, R. Guerin, and Z.-L. Zhang, "On the feasibility and efficacy of protection routing in IP networks," in *INFOCOM 2010, long version appears as Tech. Rep. 2009, University of Pennsylvania*, 2010.
- [13] P. Francois and O. Bonaventure, "An evaluation of IP-based fast reroute techniques," in *ACM CoNEXT*, 2005, pp. 244–245.
- [14] S. Previdi, "IP Fast ReRoute technologies," APRICOT, 2006.
- [15] M. Gjoka, V. Ram, and X. Yang, "Evaluation of IP fast reroute proposals," in *IEEE Comsware*, 2007.
- [16] C. Filsfils *et al.*, "LFA applicability in SP networks," Internet Draft, March 2010.
- [17] G. Rétvári, J. Tapolcai, G. Enyedi, and A. Császár, "IP Fast ReRoute: Loop Free Alternates revisited," in *INFOCOM*, 2011, pp. 2948–2956.
- [18] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot, "An approach to alleviate link overload as observed on an IP backbone," in *INFOCOM*, 2003.
- [19] H. T. Viet, P. Francois, Y. Deville, and O. Bonaventure, "Implementation of a traffic engineering technique that preserves IP Fast Reroute in COMET," in *Rencontres Francophones sur les Aspects Algorithmiques des Telecommunications, Algotel (2009)*, 2009.
- [20] M. Menth, M. Hartmann, and D. Hock, "Routing optimization with IP Fast Reroute," Internet Draft, July 2010.
- [21] M. Menth, M. Hartmann, R. Martin, T. Čičić, and A. Kvalbein, "Loop-free alternates and not-via addresses: A proper combination for IP fast reroute?" *Comput. Netw.*, vol. 54, no. 8, pp. 1300–1315, 2010.
- [22] Z. Zhong, S. Nelakuditi, Y. Yu, S. Lee, J. Wang, and C.-N. Chuah, "Failure inferencing based fast rerouting for handling transient link and node failures," in *INFOCOM*, 2005.
- [23] G. Enyedi, G. Rétvári, and T. Cinkler, "A novel loop-free IP fast reroute algorithm," in *EUNICE*, 2007.
- [24] G. Schollmeier, J. Charzinski, A. Kirstadter, C. Reichert, K. Schrodli, Y. Glickman, and C. Winkler, "Improving the resilience in IP networks," in *High Performance Switching and Routing, 2003, HPSR. Workshop on*, 2003, pp. 91–96.
- [25] A. Csaszar, G. Enyedi, and S. Kini, "IP Fast Re-Route with Fast Notification," Internet Draft, March 2011.
- [26] Cisco Systems, "Cisco IOS XR Routing Configuration Guide, Release 3.7," 2008.
- [27] Juniper Networks, "JUNOS 9.6 Routing protocols configuration guide," 2009.
- [28] B. Fortz, J. Rexford, and M. Thorup, "Traffic engineering with traditional IP routing protocols," *IEEE Comm. Mag.*, vol. 40, no. 10, pp. 118–124, Oct 2002.
- [29] D. Applegate and E. Cohen, "Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs," in *ACM SIGCOMM*, 2003, pp. 313–324.
- [30] G. Swallow, S. Bryant, and L. Andersson, "Avoiding equal cost multipath treatment in MPLS networks," RFC 4928, June 2007.
- [31] M. Thorup and M. Roughan, "Avoiding ties in shortest path first routing," 2001, aT&T, Shannon Laboratory, Florham Park, NJ, Technical Report, http://www.research.att.com/~mthorup/PAPERS/ties_ospf.ps.
- [32] G. Rétvári, R. Szabó, and J. J. Bíró, "On the representability of arbitrary path sets as shortest paths: Theory, algorithms, and complexity," in *Lecture Notes in Computer Science: Proceedings of the Third International IFIP-TC6 Networking Conference*, Athens, Greece, May 2004, pp. 1180–1191.
- [33] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, Y. Ganjali, and C. Diot, "Characterization of failures in an operational IP backbone network," *IEEE/ACM Trans. Netw.*, vol. 16, no. 4, pp. 749–762, 2008.
- [34] R. Ahuja, T. Magnanti, and J. Orlin, *Network Flows: Theory, Algorithms and Applications*. Prentice-Hall, New Jersey, 1993.
- [35] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, "Inferring link weights using end-to-end measurements," in *ACM IMC*, 2002, pp. 231–236.
- [36] SNDlib, "Survivable fixed telecommunication network design library," <http://sndlib.zib.de>.
- [37] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The Internet Topology Zoo," <http://www.topology-zoo.org>.
- [38] Cisco Systems, "Cisco IOS Release 12.0, Network Protocols Configuration Guide," 2011.
- [39] P. Pan, G. Swallow, and A. Atlas, "Fast reroute extensions to RSVP-TE for LSP tunnels," RFC 4090, 2005.

APPENDIX

Proof of Theorem 2: Easily, $\text{LFACostOpt}(G, \mathcal{S}_d)$ is in NP. To prove NP-hardness, we show that it is essentially equivalent to the protection routing problem, proved to be NP-complete in [12].

Definition 2: $\text{PR}(G, d)$: given a graph $G(V, E)$ and some $d \in V$, is there a directed spanning DAG $R_d(V, E_d) : E_d \subseteq E$ rooted at d , so that for any single node or link failure f every node $s \in V \setminus \{d\}$ has a neighbor $k : (s, k) \notin E_d$ for which it holds that (i) k is not upstream of s in R_d^f , and (ii) there is a $k \rightarrow d$ path in R_d^f , where R_d^f is obtained from R_d by removing the failed component f .

The basic differences are that (a) $\text{LFACostOpt}(G, \mathcal{S}_d)$ is defined in terms of costs, while $\text{PR}(G, d)$ in terms of a routing DAG R_d , (b) $\text{PR}(G, d)$ is for both node and link failures, while $\text{LFACostOpt}(G, \mathcal{S}_d)$ is only for link failures, and (c) item (ii) in the above definition. To show equivalence, we need to handle all these differences.

First, we show that a cost function c uniquely determines R_d and vice versa, in that we can show a mapping from c to R_d so that a path is shortest path over c if and only if it is contained in R_d (this will handle (a)). Easily, the shortest paths over c are always in a DAG. The reverse direction, that is, taking R_d and creating a cost c of it, is equally easy: take a topological ordering $o(v) : v \in V$ of R_d (this always exists) and for each $(i, j) \in E$ set $c(i, j) = o(j) - o(i)$ if $(i, j) \in E_d$ and $c(i, j) = n$ otherwise.

Second, taking a close look on the NP-completeness proof of $\text{PR}(G, d)$ in [12], we observe that the proof remains valid if we treat link failures only and disregard node failures. Thus, we can state stronger claim: $\text{PR}(G, d)$ is NP-complete, even if we only allow link failures. This handles (b).

Finally, (c) means that in $\text{PR}(G, d)$ we only take a node for protected, if after a failure f all its downstream neighbors' path in R_d avoids f . However, when we only consider single link failures, item (i) guarantees this. ■