Telecommun Syst (2015) 60:485–502
DOI 10.1007/s11235-015-0006-9

On providing fast protection with remote loop-free alternates

Analyzing and Optimizing Unit Cost Networks

Levente Csikor · Gábor Rétvári

Published online: 19 March 2015

Abstract Up to not so long ago, loop-free alternates (LFA) was the only viable option for providing fast protection in pure IP and MultiProtocol Label Switching–Label Distribution Protocol networks. Unfortunately, LFA cannot provide protection for all possible failure cases in general. Recently, the Internet Engineering Task Force has initiated the remote loop-free alternates (rLFA) technique as a simple extension to LFA, to boost the fraction of failure cases covered by fast protection. Before further standardization and deployment, however, it is crucial to determine to what extent rLFA can improve the level of protection against single link or node failures in a general IP network, as well as to find optimization methods to tweak a network for 100 % rLFA coverage. In this paper, we take the first steps towards this goal by solving these problems in the special, but practically relevant, case when each network link is of unit cost. We also provide preliminary numerical evaluations conducted on real IP network topologies, which suggest that rLFA significantly improves the level of protection, and most networks need only 2–3 new links to be added to attain 100 % failure case coverage irrespectively of whether link or node protection is considered.

Keywords IP Fast ReRoute · Remote loop-free alternates · Link protection · Node protection · Heuristics · Unit link costs

L. Csikor (✉) · G. Rétvári
MTA-BME Future Internet Research Group, High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, Budapest University of Technology and Economics, Magyar tudósok körútja 2., Budapest 1117, Hungary
e-mail: csikor@tmit.bme.hu

G. Rétvári
e-mail: retvari@tmit.bme.hu

1 Introduction

In recent years, high availability has become an important factor in operational networks, not just due to the requirements of the increasing number of real-time applications (VoIP, IPTV, online-gaming, etc.) but also for the standard Internet applications used day by day. Low latency is very important even if one only waits for a single web page to download, but network outages might cause intolerably long time intervals of service disruptions [33], and hence, increased latency. Installing added redundancy in the network topology, as well as a sophisticated failure mitigation scheme at the routers, has the potential to reduce the latency caused by component failures, and consequently, increase network revenue [46]. In order to reduce latency and increase the availability in service provider networks, therefore, it is essential for operators to deploy network functionality to recognize the failure in a timely fashion and reroute the affected packets instantly around the failed component.

Formerly, the intra-domain routing protocols (Open Shortest Path First [37] or Intermediate System To Intermediate System [23]) used to handle failures. The failure information was distributed throughout the network in order to notify each router to recalculate shortest paths with the failed component removed from the topology. This process can take between 150 ms and a couple of seconds, depending on network size and routers' shortest path calculation efficiency [24,29]. Clearly, this recovery time is beyond what real-time applications require.

Therefore, the IETF defined a framework, called IP Fast ReRoute (IPFRR [43]), for native IP protection, in order to reduce failure reaction time to tens of milliseconds in an intra-domain, unicast setting. In order to achieve this goal, the IPFRR techniques are based on *local rerouting* and *precomputed detours* [43]. This allows instant reaction to the

failure and enables the routing protocol to converge in the background.

In the past few years, many IPFRR proposals have appeared to solve this problem. Unfortunately, the majority of them requires additional management burden, complexity, and non-standard IP forwarding functionality [2,8,16,17,31,47] to existing routing protocols, evading the possibility to be eventually applied in commercial routers.

Yet, there is an IPFRR method, called loop-free alternates (LFA) [5], which has already made its way into commercial routers [10,26]. LFA is simple, standardized and already implemented. However, it has a significant drawback: it does not guarantee protection for all possible failure cases, due to strong dependency on actual topology and link costs. Extensive simulations and numerical studies have shown that LFA can only protect 75–85 % of the link failures and 50–75 % of the node failures, respectively.

To improve the level of fast protection provided by LFA, the IETF has published a generalization of LFA, called the Remote LFA (rLFA) IP Fast ReRoute framework [7]. This method is an extension to the basic LFA that provides additional backup connectivity when none can be provided by the basic mechanisms. But even if it provides higher failure coverage, there still exist networks that are not sufficiently protected by rLFA. Unfortunately, as of now there is no information available about how it performs in different network topologies, what the fundamental lower and upper bounds on failure case coverage are, or how this can be improved [41].

In the present paper, we make the first steps in this direction. As a first approach, we shall limit our attention to the special case when link costs are uniform. Our earlier studies on LFA [41] showed that the protection capabilities of LFA crucially depend on both the graph topology and the link costs of the underlying network. Unfortunately, it turned out extremely difficult to consider both at the same time, due to the complexity of the related graph theoretical questions. Therefore, it has proven beneficial to study graph topological concerns separately from the effects of link costs. In the present paper, we follow the same course: first, we initiate the analysis for remote LFA in graphs with unit costs, and in a subsequent study we shall attempt to generalize our results to arbitrary weighted graphs. Considering unweighted graphs is fruitful for a number of further reasons. The unit cost case is highly relevant in real-world networks and, as shall be shown, results for LFA can only be generalized to rLFA under the unit cost assumption. Finally, we also found this problem particularly appealing from a theoretical point of view.

This paper is essentially a crystallization of the ideas in our preliminary study on rLFA [14] and an extension of the rLFA specification [7], as well as our analysis, from the model of single link outages to the crucial case of single node failures. In the first part, we provide the first ever basic graph theoretical toolset for analyzing rLFA failure case coverage in the case when link costs are uniform, and we establish a sufficient and necessary condition for a network to have 100 % rLFA failure coverage. We also study the "bad cases" for rLFA, in which failure coverage is particularly poor. Building on [7], we distinguish between *plain* and *extended remote LFA* and we quantify the benefits that come from the usage of extended rLFA.

Our analysis shows that many practically important graph topologies do not admit 100 % rLFA failure coverage, especially with plain rLFA. Recently, LFA network optimization methods were proposed [13,15,38,40,41] to optimize certain aspects of the network to obtain maximal failure coverage. The second part of the paper is devoted to generalize these methods to rLFA. In particular, we study the problem of optimizing a network topology for better rLFA protection and we introduce a set of algorithms for modifying the network, by adding the smallest number of new links, to improve coverage to 100 %.

The main contributions in this paper are as follows:

- We develop a set of elemental graph theoretical rLFA tools, which facilitates for analyzing rLFA failure coverage in general networks. We also extend the rLFA specification [7], originally defined for single link failures only, to the relevant case of single node failures, and we generalize our toolset to this very case as well. Furthermore, we reveal the deep relations between LFA and rLFA and we show the conclusions that can be drawn if information about one of them exists.
- Using this toolset, we provide a comprehensive analysis of rLFA failure case coverage under the assumption that network links are of uniform cost. We give sufficient and necessary conditions for full rLFA failure coverage in the case of single link as well as node outages. An attempt is also made to find lower bounds on failure case coverage. In particular, we find that in 2-node-connected graphs rLFA protection coverage for single link failures can go down to 50 %, or to 33 % for 2-edge-connected networks, and for node failures rLFA coverage can totally zero out in certain cases.
- To help inherently poorly protected networks, we study the rLFA *graph extension problem* in detail. This problem asks to augment the network with new, unit cost links to attain complete rLFA protection. In particular, we propose a complete family of heuristics in order to facilitate for picking the best approximation algorithm for the particular network under consideration.
- We provide an extensive numerical evaluation of rLFA failure case coverage and rLFA graph extension methods on a wide range of real-world network topologies. Crucially, we find that some networks have full rLFA protection without any modifications. For the rest, the

proposed heuristics turn out very effective in improving rLFA failure protection.

The rest of the paper is organized as follows. Section 2 gives a summary on the related works, Sect. 3 gives an introduction the rLFA, and then Sect. 4 presents the essential formal definitions. Section 5 gives a useful mathematical model and Sects. 6 and 7 are devoted to a graph theoretical remote LFA failure coverage analysis of many important classes of graph topologies. Section 8 discusses the remote LFA graph extension problem and describes numerical results on many real-world network topologies. Finally, in Sect. 9 we conclude our work and sketch future research directions.

2 Related works

Protection against network failures has become one of the most compelling problems of today's internet. It turned out that more than 85 % of unplanned failures affect only links and almost the half of these failures are transient [33], i.e., 50 % of all failures last less than a minute [22]. Unfortunately, such transient failures are very difficult to handle with current intra-domain routing protocols, like OSPF [37] or IS-IS [23]. For instance, just a single flapping interface can keep all other routers in the network busy, since it can cause link state flooding and significant computational overhead due to the constant need for shortest paths recalculations. This drawback comes directly from the fundamental design philosophy of the protocol, since, in case of a failure, it tries to make the network topology up-to-date in order to not to cut off packet forwarding. Accordingly, after a failure, an adjacent router recognizes it and notifies every other router throughout the network about the failure in order to induce the recalculation of the shortest paths with the failed component removed. During this re-convergence process packets are dropped due to invalid routes.

To overcome these issues, IP Fast ReRoute Framework (IPFRR, [43]) was defined by the IETF (Internet Engineering Task Force). IPFRR techniques are based on two major principles: *local rerouting* and *precomputed detours*. Local rerouting means that instead of notifying every other router about the failure, the adjacent router to the failure tries to locally solve the problem, i.e., reroute the packet to another node, this way bypassing the failed component. Precomputed means that the mechanism is proactive and alternate routes are installed long before any failure occurs. Thus, the IPFRR techniques convert the restoration scheme, standard in IP networks today to handle outages, into a faster proactive protection mechanism [45].

Lately, the IETF defined a basic specification for IPFRR, called loop-free alternates (LFA) [5]. In LFA, when the connectivity to a next-hop[1] is lost all the traffic is rerouted to an alternate next-hop, called a Loop-free Alternate, that still has a path to the destination, which is unaffected by the failure. These alternate next-hops are selected in a way as to guarantee that the packet will not be passed back, since that would lead to an IPFRR loop. However, such alternate next-hops do not always exist, depending on the actual topology and link costs. Therefore, in most network topologies not all next-hops can be protected with LFA, leaving the network vulnerable to certain failure scenarios.

In the past few years, many IPFRR proposals have appeared to guarantee 100 % failure case coverage in every network topology, however, the majority of them requires additional complexity, non-standard IP forwarding functionality, explicit signaling, etc.

The *Failure-carrying Packets* (FCP [30]) framework does not just deal with single node or link failures, but it can also guarantee delivery if simultaneous failures are present in the network. Instead of having an extremely high number of precomputed paths, in FCP all routers have a consistent view of operational links, called a Network Map. Because of its consistency, all that is required to be carried by the packets is information about which of these links have failed.

In the case of *O2 routing* [42], each router has alternate paths through at least two distinct next-hops to each destination, in order to facilitate local failure reaction and loop-free forwarding. Unfortunately, the network must meet a necessary condition, which states that each node has to form at least one triangle.

As another approach, Kvalbein et al. proposed *Multiple Routing Configurations* (MRC, [2]), wherein a small set of backup network configurations is used. Thus, in case of a failure a nearby router detects it and marks the packet with a backup configuration identifier designating an overlay topology that does not contain the failed component. They also proved that on average the number of such backup configurations is usually below four. A similar approach is [36], wherein the protection and restoration is provided by distributed multipath routing. The main idea is that if multiple paths exist in the network due to load balancing, then they can be used as backup routes as well.

The so called *Protection routing* scheme, proposed in [28], is based on a centralized control over the routing tables. A central server pre-computes forwarding decisions for common failure scenarios and download these into the routers. Thus, if a failure occurs, the appropriate new forwarding state is already available locally.

Another fast resilience scheme, called Failure Insensitive Routing (FIR [31]), uses interface specific forwarding. FIR handles only link failures, while a subsequent scheme of the

[1] In IP routing, the next router along the shortest path to a destination is called *next-hop*.

same authors, FIFR [47], deals with node failures as well. The main idea of these concepts is that if a node receives a packet through an unusual interface, it can infer implicitly that, due to a failure, the packet has not traveled along its default shortest path. Unfortunately, interface specific forwarding is generally not available in IP routers today.

In the method called *Not-via addresses* [8], when a failure occurs then the packets are forwarded on an explicitly defined detour, which definitely avoids the failed component, i.e., if an arbitrary node s wants to send a packet to a destination node d, and the link to the next-hop n or the next-hop n itself fails, then s has to pass the packet towards d not-via n. Thus, this mechanism requires additional (not-via) addresses for which there is no standardized protocol, moreover it brings extra complexity into routing if the additional IP header does not fit into the MTU (Maximum Transfer Unit),[2] which can cause packet fragmentation and time-consuming reassembly at the tunnel endpoint. To break down the management burden and computational complexity, a *lightweight version of Not-via* [17] was later proposed, which is based on the concept of redundant trees [34].

DisPath [4] can protect every single link or node failure in networks and, similarly to LFA, it has low complexity and it does not modify the IP packets. Unfortunately, the computation of backup paths relies on a reverse shortest path algorithm, crucially limiting its applicability as currently OSPF and IS-IS implements Dijkstra's standard shortest path algorithm only.

A different approach is to use explicit signaling to notify routers about the failures [12,21]. The advantage of this is that it avoids the need of the modification to standard IP forwarding, but in order to make it work, it requires separate signaling mechanism only for IPFRR.

Note that telecommunication networks are usually multilayered, therefore the physical failure of a link may cause failures in a set of virtual links in the overlay topologies. This case is termed as Shared Risk Link Group (SRLG), which also has to be covered. Therefore, in order to provide resiliency in such cases, [19] use SRLG-disjoint path pairs in optical networks to avoid the failures.

Due to the complexity of the aforementioned techniques, it is no wonder that so far only LFA has made its way into commercial IP routers. As mentioned above, however, LFA cannot protect each next-hop in all networks. As a workaround, the IETF suggested to use LFA and Not-via side-by-side in the cases when the former does not deliver sufficient levels of protection [5,8]. Nevertheless, the authors in [35] proved that in real networks, where the sheer size of the IP forwarding tables and traffic engineering also play an important role, this combined method does not provide any significant advantages over pure Not-via.

As a consequence of the above considerations, there have been proposals lately to attempt to reach full failure coverage using solely loop-free alternates. The main idea is, instead of extending the capabilities of LFA, modify the underlying topology instead. Some research works studied the question of how to augment the network with the smallest number of new links to improve the failure coverage [38,41], while others [13,15,40] attempted to optimize IGP link costs in order to generate new loop-free alternates through altering default shortest paths. In the former approaches, it was proved that an exiguous number of additional links can significantly improve failure coverages in most real network topologies. Therefore, for those operators whose budget can afford adding new physical links to the topology, these may provide good solutions. In those networks, however, where reconfiguring link costs is not an option due to load balancing and traffic engineering issues, cost optimization may not be a good approach.

Recently, the IETF has published a generalization of LFA, called the Remote LFA [7] in order to improve the failure coverage provided by simple LFA. Since it is based on LFA, it is already available in today's routers [11]. The main idea is that, in case of a failure, not only direct neighbors can be used as a potential loop-free alternate but further remote nodes as well. These remote LFA staging points are reached trough IP tunnels, but these tunnels are restricted to shortest paths as well. Note that in an MPLS/LDP (MultiProtocol Label Switching–Label Distribution Protocol) enabled network these tunnels are freely accessible via a simple label stack. Yet, even if remote LFA can produce higher failure case coverage than pure LFA, the level of this protection still depends heavily on the underlying topology and link costs. The main objective of this paper is, consequently, to quantify this dependence using a thorough graph-theoretical analysis and propose new network optimization techniques to tweak a network topology towards better remote LFA protection.

Note that IPFRR is not the only option for fast protection in IP networks, since for MPLS different fast protection schemes have been proposed [1,20,25] and already standardized [39]. These methods, however, are only available in networks with the Resource Reservation Protocol–Traffic Engineering (RSVP-TE) extension deployed. Many operators, on the other hand, rely on MPLS/LDP exclusively, which uses the IP control plane for routing information and hence depend crucially on pure IP protection schemes.

3 Remote loop-free alternates

In loop-free alternates, the backup routes are precomputed and installed in the router as the backup for the primary

[2] In computer networking, the maximum transmission unit (MTU) is the size of the largest protocol data unit that the layer can pass onwards.

routes. Once a router detects a link or adjacent node failure, it switches to the backup route to avoid traffic loss. Whole LFA considers only physically adjacent routers for backup routes, remote LFA allows the backup next-hop to be more than one hop away. After a failure, an adjacent router recognizes it and tries to find a (remote) node whose shortest path to the destination is not affected by the failed component. If such a router is found, then packets will be forwarded to it. Remote LFA relies on tunnels to provide additional logical links towards backup next-hops. After the remote node receives the package it sends it towards the primary destination. Note that the tunnelled traffic is restricted to shortest paths just like "normal" traffic, hence the tunnel must avoid the failure as well. Perhaps the easiest way to understand remote LFA, and how it differs from basic LFA, is through an example. Consider the network depicted in Fig. 1a and suppose that router s wishes to send a packet to destination d.

The next-hop of s along the shortest path towards d is a. If, however, the link (s, a) fails, then node s has to find an alternative neighbor to pass on the packet to. It cannot send the packet to, say b, as b has an ECMP (Equal Cost Multiple Path) to destination d and, as it does not know about the failure, it can send the packet back to s causing a loop. Therefore, s has no neighbor that would not pass the packet back to it if chosen as a bypass, so in this case the given source–destination pair cannot be protected via standard LFA. However, if a tunnel is created between s and e (marked by black dashed line in Fig. 1a), then e, now being an indirect neighbor of s, would become an LFA for d, thereby protecting the link (s, a).

Consequently, when a link cannot be entirely protected with local LFA neighbors, the protecting router seeks the help of a remote LFA staging point. Note that this tunnel is only used as a detour, so it does not affect the normal flow of traffic in any ways. There are numerous tunnelling mechanisms which fulfill the requirements of this design. In an MPLS/LDP (Multiprotocol Label Switching-Label Distribution Protocol [3]) enabled network, for instance, a simple label stack can be used to provide the required tunnel without any additional modification to the IP header of the packets.

Next, consider Fig. 1b where node s remains the source but node d' becomes the destination and link (s, b) fails. Then (s, b) cannot be protected for a lack of a suitable tunnel, since all nodes whose shortest path does not go through (s, b) can only be reached from s through (s, b) itself. For a formal definition, see the next section. This suggests that while the use of rLFA definitely can provide higher protection level against link failures than pure LFA, it still does not facilitate full protection for all failure cases in a general topology.

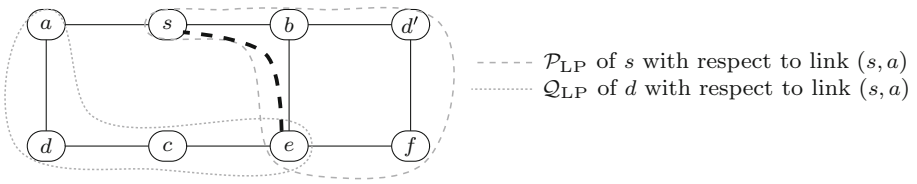
Next, examine how all the previously mentioned properties are changed when node protection is also taken into account. Consider the network depicted in Fig. 2.

Suppose that node s wants to send a packet to destination node d. The next-hop of s to d is node e. One can easily check that if link (s, e) goes down then node n and m are suitable repair tunnel endpoints, since the shortest paths from them to node d avoid the failed component. However, if not only the link (s, e) fails but the node e itself, then node m can be the one and only remote loop-free alternate, since node n has an ECMP shortest path to node d through the failed node e. It should also be noted that in case of node protection we have to deal with the so called *last-hop problem*. This says that if the destination node itself goes down, then it obviously cannot be protected. Therefore, node protection between two neighboring nodes is undefined.

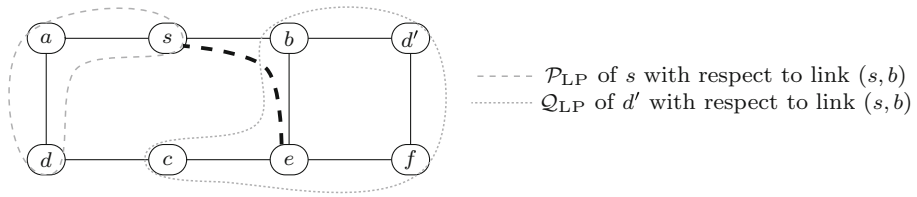
4 Model formulation

Our mathematical model for studying rLFA is as follows. We model the network topology by a *simple, undirected graph* $G(V, E)$, with V being the set of nodes and E the set of edges. Let $n = |V|$ and $m = |E|$, and denote the complement edge set with $\overline{E}$. We assume that links are bidirectional and point-to-point. As mentioned earlier, we further assume that each link in G is of the same unit cost, as this assumption allows us to study the purely graph theoretical aspects of rLFA separately from the effect of link weights. In a subsequent paper, we plan to relax this assumption. Furthermore, we presume that each node has a well-defined next-hop towards each destination even if more than one equal cost shortest paths exist. Since an arbitrary link can only be protected if the graph of the network is *2-edge-connected*, we assume this minimum topological requirement for link-protecting case. For the case of node protection, we also assume the graph to be *2-node-connected*. We use the notation $\mathrm{dist}(u, v)$ for any $u, v \in V$ to describe the length of the shortest path from u to v. Let $\mathrm{neigh}(s)$ denote the set of nodes which are the neighbors of an arbitrary node s. Furthermore, $\mathrm{LFA}(x, y)$ denotes the set of nodes protecting the (x, y) source–destination pair.

During a failure, the repair tunnel endpoint needs to be a node in the network reachable from the source without traversing the failed component. In addition, the repair tunnel endpoint needs to be a node from which packets will normally flow towards their destinations without being attracted back to the failed component. Correspondingly, in the case of link failure the set of routers which can be reached from a source without traversing the failed link is termed the *P-space* [6] of the source with respect to the failed link (hereafter $\mathcal{P}_{\mathrm{LP}}$, where LP refers to link-protecting case). Since the source router will only use a repair path when it has detected the failure of the link, the initial hop of the repair path needs *not* be subject to the source's normal forwarding decision process. Therefore, the term *extended P-space* (hereafter $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$) was also defined, which is the union of the $\mathcal{P}_{\mathrm{LP}}$s of each of the



(a) Higher protection can be attained with rLFA



(b) A basic situation that cannot be protected with rLFA either

Fig. 1 Sample network topologies with uniform link costs. *Solid lines* mark the IP network topology, while *thick dashed lines* mark a tunnel

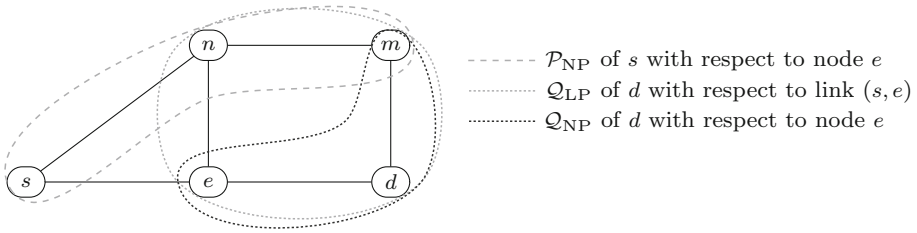


Fig. 2 A sample network topology for illustrating how node protection differs from link protection

source's neighbors. The usage of $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ may enable the source router to reach potential repair tunnel endpoints that were otherwise unreachable. Furthermore, the set of routers from which the destination can be reached without traversing the failed link is termed the *Q-space* (hereafter $\mathcal{Q}_{\mathrm{LP}}$) of the destination with respect to the failed link. The intersection of the source's $\mathcal{P}_{\mathrm{LP}}$ and the destination's $\mathcal{Q}_{\mathrm{LP}}$ with respect to the failed link defines the viable repair tunnel endpoints, known as $\mathrm{PQ}_{\mathrm{LP}}$-nodes, which are practically the remote LFAs. As can be seen, for the case of the example network depicted in Fig. 1 there is only one node (e) that protects the link (s, a), assuming that node s wants to send a packet to node d as destination. However, considering d' as the destination the $\mathcal{P}_{\mathrm{LP}}$ and $\mathcal{Q}_{\mathrm{LP}}$ turn out different. Now, there is no intersection of s' $\mathcal{P}_{\mathrm{LP}}$ and $\mathcal{Q}_{\mathrm{LP}}$ of d', thus viable $\mathrm{PQ}_{\mathrm{LP}}$-nodes do only exist if $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ is used, since if s can pass the packet to c, then node c will not pass the packet back and the packet transmission will avoid the failed (s, b).

Next, we extend these definitions to the case of node failures. Note that the rLFA specification [7] does not consider this case, so ours is the first such extension. As it turns out, the case of node protection hardly differs from the case of link protection. When the next-hop fails, the possible repair tunnel endpoint needs to be a (remote) node, which is reached from the source without traversing that failed next-hop itself (instead of only the link to it, as before). Hence, the set of such routers is termed the $\mathcal{P}_{\mathrm{NP}}$ (where the subscript NP refers to node-protecting case) of the source with respect to the failed node. As it was in the case of link protection, the term *extended P-space* (hereafter $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$) can also be defined as the union of $\mathcal{P}_{\mathrm{NP}}$s of each of the source's neighbors. The set of routers whose shortest path to destination avoid the failed node is termed the $\mathcal{Q}_{\mathrm{NP}}$ of the destination with respect to the failed node. Here again the intersection of the source's $\mathcal{P}_{\mathrm{NP}}$ and the destination's $\mathcal{Q}_{\mathrm{NP}}$ with respect to the failed node defines the viable repair tunnel endpoints, known as $\mathrm{PQ}_{\mathrm{NP}}$-nodes. For the sake of easy comprehension, see Fig. 2 and consider node s as source and node d as destination.

In this work, we slightly diverge from the terminology of the specification [7] and we say that a node is remote LFA if it is in the intersection of the "simple" $\mathcal{P}_{\mathrm{LP}}$ ($\mathcal{P}_{\mathrm{NP}}$) and $\mathcal{Q}_{\mathrm{LP}}$ ($\mathcal{Q}_{\mathrm{NP}}$, respectively) and we shall use the term "*Extended remote* LFA" henceforth when $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ ($\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$) is also to be considered for defining the rLFA nodes, i.e., $\mathrm{PQ}_{\mathrm{LP}}$-nodes ($\mathrm{PQ}_{\mathrm{NP}}$-nodes). In the rest of the paper, $\mathrm{rLFA}_{\mathrm{LP}}(x, y)$ denotes the set of nodes that protect source x and destination y with remote LFA if the link to the next-hop fails. Similarly, $\mathrm{rLFA}_{\mathrm{NP}}(x, y)$ marks the set of nodes protecting source x and destination y with remote LFA if the next-hop itself fails.

Most of our analysis will be given for "plain" rLFA, as this technique can be easily implemented and deployed since it does not require profound modifications to the forwarding

plane. Extended rLFA, on the other hand, requires sophisticated functionality. Thus, we expect, implementations to provide only the plain rLFA initially and so we mostly treat this case, and only highlight some important aspects of "Extended remote LFAs".

From the above discussion, it is clear that in general not all nodes have LFA or even remote LFA protection to every other node. To measure link and node-protecting rLFA coverages in a graph G, we adopt and redefined the simple metric from [5]:

$$\mu_{LP}(G) = \frac{\#\text{rLFA}_{\text{LP}} \text{ protected } (s,d) \text{ pairs}}{\#\text{all } (s,d) \text{ pairs}} \quad (1)$$

$$\mu_{NP}(G) = \frac{\#\text{rLFA}_{\text{NP}} \text{ protected } (s,d) \text{ pairs}}{\#\text{all non-adjacent } (s,d) \text{ pairs}} \quad (2)$$

For LFA, the coverage $\eta_{\text{LP}}(G)$ and $\eta_{\text{NP}}(G)$ can be defined in a similar way.

5 A mathematical toolset for remote LFA

Below, we give some basic machinery to handle remote LFAs somewhat more plausibly than what is provided by the mere definitions of P-spaces and Q-spaces. We shall separate the discussion into two parts. First, in line with the specification [7], we consider only single link failures. Then, in the second part we extend our techniques to single node failures as well.

5.1 Link-protecting case

An arbitrary failed link along the shortest path between a source and a destination can only be protected if the intersection of $\mathcal{P}_{\text{LP}}$ of the source and the $\mathcal{Q}_{\text{LP}}$ of the destination is not empty. First, we show an alternative characterization for $rLFA_{\text{LP}}$ that, as shall be seen, is more amenable to theoretical analysis. Consider the below reformulation of this requirement in terms of the shortest path distance function $dist$.

Observation 1 *For a source node s and next-hop e, some $n \in V$ is in $\mathcal{P}_{\text{LP}}(s,e)$ if and only if*

$$dist(s,n) < dist(s,e) + dist(e,n)\,, \quad (3)$$

and some $n \in V$ is in $\mathcal{Q}_{\text{LP}}(s,d)$ if and only if

$$dist(n,d) < dist(n,s) + dist(s,d)\,. \quad (4)$$

One can easily see, that (4) is the basic loop-free criterion of link-protecting LFAs [5], while (3) means that the repair tunnel cannot traverse the failed link. The notion of $\mathcal{P}^{\text{e}}_{\text{LP}}$ could also be expressed with distance functions:

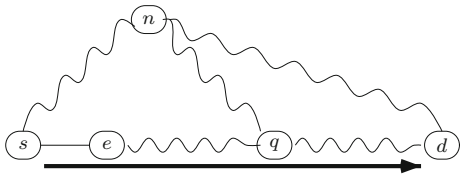


Fig. 3 Illustration for the proof of Lemma 1

Observation 2 *For a source s and next-hop e, some $n \in V$ is in the extended $\mathcal{P}^{\text{e}}_{\text{LP}}(s,e)$ if and only if $\exists v \in neigh(s)$: $dist(v,n) < dist(v,s) + dist(s,e) + dist(e,n)$.*

It should be noted that the conditions above hold for arbitrary weighted graphs as well.

Next, we formulate an important corollary of the previous observations. In particular, we show that if an arbitrary node on the shortest path between a source and a destination is rLFA_{LP} protected, then every further node along that shortest path is rLFA_{LP} protected as well.

Lemma 1 *Let (s,d) be a source–destination pair and let q be a node along the default shortest path from s to d. If $rLFA_{LP}(s,q) \neq \emptyset$, then $rLFA_{LP}(s,d) \neq \emptyset$.*

Proof Consider Fig. 3 and suppose node e is the next-hop from s to d. The wavy lines denote the existence of a path between the given nodes. The thick line indicates the shortest path from s to d. For n to be in $\text{rLFA}_{\text{LP}}(s,d)$, it has to fulfill the conditions stated in Observations 1. First, it satisfies (3) for (s,d) since $\mathcal{P}_{\text{LP}}$ does not depend on the destination node. Additionally, in case of link protection it only needs to satisfy (4), notably $\text{dist}(n,d) < \text{dist}(n,s) + \text{dist}(s,d)$. We know that $\text{dist}(n,q) < \text{dist}(n,s) + dist(s,q)$ and due to the triangle inequality[3] $\text{dist}(n,d) \leq \text{dist}(n,q) + \text{dist}(q,d)$. Therefore, $\text{dist}(n,d) < \text{dist}(n,s) + \text{dist}(s,q) + \text{dist}(q,d) \Rightarrow \text{dist}(n,d) < \text{dist}(n,s) + \text{dist}(s,d)$. □

An important consequence of Lemma 1 is the simple observation that a graph has full rLFA_{LP} protection, if and only if each node has an rLFA_{LP} to each of its next-hops.

Corollary 1 *Let G be a graph with unit link costs. Then, $\mu(G) = 1$, if and only if for each $(u,v) \in E$, u has an $rLFA_{LP}$ to v and v has an $rLFA_{LP}$ to u.*

Next, we show that there is a deep connection between basic link-protecting LFA (LFA_{LP}) and rLFA_{LP} in unit cost networks.

Theorem 1 *Let $G(V,E)$ be a graph with unit link costs, let (s,d) be a source–destination pair, let e be the default*

[3] The triangle inequality states that for any triangle, the sum of the lengths of any two sides must be greater than or equal to the length of the remaining side. It is one of the defining properties of the distance function, which is used in shortest path routing.

next-hop of s to d, and let u be an arbitrary node with $u \in neigh(s), u \neq e$. Then, $u \in rLFA_{LP}(s,d)$ if and only if $u \in LFA_{LP}(s,d)$.

Proof First, we verify the forward direction. Easily, $u \in \mathrm{rLFA}_{\mathrm{LP}}(s,d)$ implies u is in $\mathcal{Q}_{\mathrm{LP}}$, which precisely coincides with the condition for u to be a link-protecting LFA. Second, we check the reverse direction. If $u \in \mathrm{LFA}_{\mathrm{LP}}(s,d)$, then u, by definition, fulfills (4). In addition, it also satisfies (3) due to the assumption $u \in \mathrm{neigh}(s)$, because in a uniform cost network the default shortest path between adjacent nodes is through the direct link, and hence the $s \to u$ shortest path always avoids the (s,e) link. □

5.2 Node protection

In this subsection, we extend the previous statements to node protection. Now, suppose that not the link between an arbitrary source s and it's next-hop e fails but the next-hop e itself. It is easy to see from the above discussion that the condition for $\mathcal{P}_{\mathrm{NP}}$ remains the same as for $\mathcal{P}_{\mathrm{LP}}$, thus only $\mathcal{Q}_{\mathrm{NP}}$ has to be re-defined.

Observation 3 *For a source s, next-hop e and destination d, some $n \in V$ is in $\mathcal{Q}_{\mathrm{NP}}(s,d)$ if and only if*

$$dist(n,d) < dist(n,e) + dist(e,d)\ . \tag{5}$$

Similarly, it is easy to observe that (5) is the basic loop-free criterion of node-protecting LFAs [5]. The concept of $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$ could also be expressed as follows:

Observation 4 *For a source s and next-hop e, some $n \in V$ is in $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}(s,e)$ if and only if $\exists v \in neigh(s) : dist(v,n) < dist(v,e) + dist(e,n)$.*

Again, note that these conditions also hold for arbitrary weighted graphs.

Next, we reformulate Lemma 1 and show that if an arbitrary node on the shortest path between a source and a destination is $\mathrm{rLFA}_{\mathrm{NP}}$ protected, then every further node along that shortest path is $\mathrm{rLFA}_{\mathrm{NP}}$ protected as well.

Lemma 2 *Let (s,d) be a source–destination pair and let q be a node along the default shortest path from s to d. If $rLFA_{NP}(s,q) \neq \emptyset$, then $rLFA_{NP}(s,d) \neq \emptyset$.*

Proof Consider example network depicted in Fig. 3 again and suppose again node e is the next-hop from s to d. For n to be $n \in rLFA_{\mathrm{NP}}(s,d)$, it has to fulfill (3) and (5). As it was in the link-protecting case, we do not have to deal with $\mathcal{P}_{\mathrm{NP}}$ since it does not depend on the destination node. We only have to verify the condition of $\mathcal{Q}_{\mathrm{NP}}$: $\mathrm{dist}(n,d) < \mathrm{dist}(n,e) + \mathrm{dist}(e,d)$. Since $n \in \mathrm{rLFA}_{\mathrm{NP}}(s,q)$, then $\mathrm{dist}(n,q) < \mathrm{dist}(n,e) + \mathrm{dist}(e,q)$, and due to triangle inequality $\mathrm{dist}(n,d) \leq \mathrm{dist}(n,q) + \mathrm{dist}(q,d) \Rightarrow \mathrm{dist}(n,d) < \mathrm{dist}(n,e) + \mathrm{dist}(e,q) + \mathrm{dist}(q,d) \Rightarrow \mathrm{dist}(n,d) < \mathrm{dist}(n,e) + \mathrm{dist}(e,d)$, which completes the proof. □

Next, we show that, analogously to the link protecting case, node-protecting LFAs and rLFAs are deeply related to each other in unit cost networks.

Theorem 2 *Let $G(V,E)$ be a graph with unit link costs, let (s,d) be a source–destination pair, let e be the default next-hop of s to d, and let u be an arbitrary node with $u \in neigh(s), u \neq e$. Then, $u \in rLFA_{NP}(s,d)$ if and only if $u \in LFA_{NP}(s,d)$.*

The proof of the theorem goes along similar lines as the proof of Theorem 1 and so we do not present it herein.

6 Analysis of extended remote LFA

Next, we digress a little to show that extended rLFAs are a powerful tool for link-protection. In particular, first we show that in case of link failures extended $\mathrm{rLFA}_{\mathrm{LP}}$ ensures 100 % failure coverage in every network.

Theorem 3 *Let G be an arbitrary 2-edge-connected graph with uniform link costs and suppose that remote LFA can use the $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ option. Then, in case of link failures $\mu(G) = 1$.*

Proof We show that for each edge $(u,v) \in E$, u has a remote LFA to v (and vice versa). This will mean that every node has an $\mathrm{rLFA}_{\mathrm{LP}}$ to each of its next-hops, which guarantees $\mu(G) = 1$ by Corollary 1. Since G is 2-edge-connected, we know that (u,v) is contained in at least one chordless cycle. Let the length of this cycle be k. If k is odd, then the single node at distance $\frac{k-1}{2}$ from v along the cycle is a remote LFA to u. If, on the other hand, k is even, then the $\mathcal{P}_{\mathrm{LP}}(u,(u,v)) \cap \mathcal{Q}_{\mathrm{LP}}(u,v)$ is empty. Observe, however, that the single node of distance $\frac{k}{2}$ from u is contained both in $\mathcal{Q}_{\mathrm{LP}}(u,v)$ and the extended $\mathcal{P}_{\mathrm{LP}}(w,(u,v))$, where w is the neighbor of u other than v along the cycle, and so it is a remote LFA in terms of the $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ option. This completes the proof. □

Consequently, in general it can be stated that if remote LFA implementations support extended P-space then unit cost networks have full protection against single link failures. This may be an important factor to consider by an operator willing to deploy rLFA and to an IP device vendor to implement extended rLFA in its router products.

However, when node failures are also taken into account, then extended P-space with respect to the failed node is not always enough to guarantee 100 % failure coverage in every network. As a proof, consider the simple network depicted

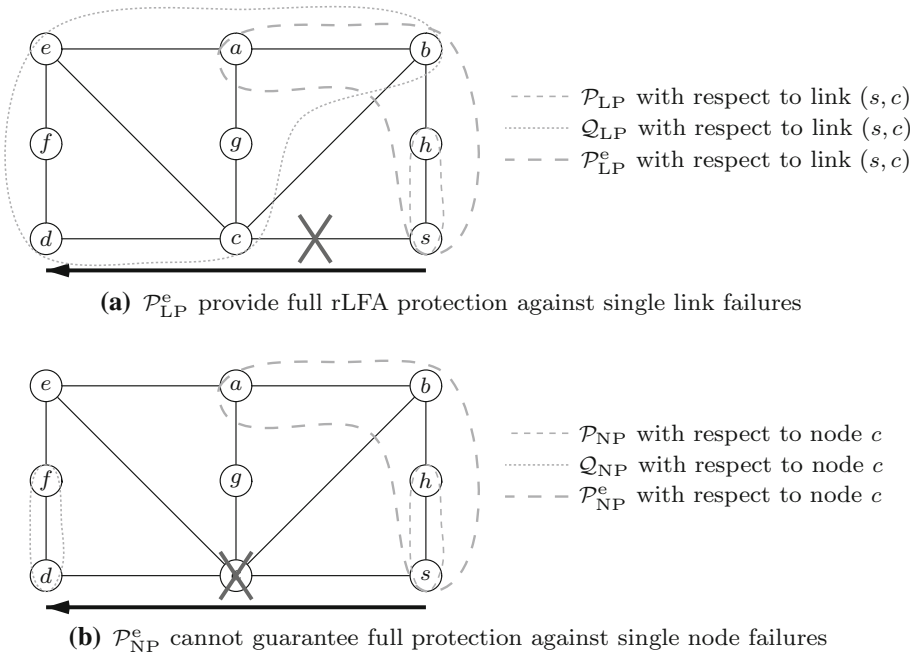


Fig. 4 Illustration of a network with different protection scenarios

(a) $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ provide full rLFA protection against single link failures

(b) $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$ cannot guarantee full protection against single node failures

in Fig. 4. Assume that node s wants to send a packet to node d. The default shortest path goes through node c.

Under the assumption that only the link (s, c) fails (see Fig. 4a), then it can be protected since $\mathcal{P}_{\mathrm{LP}} \cap \mathcal{Q}_{\mathrm{LP}} \neq \emptyset$ ($b \in \mathrm{PQ}_{\mathrm{LP}}$-nodes). Next, consider the case of Fig. 4b where the next-hop c went down. The potential repair tunnel endpoints are in $\mathrm{PQ}_{\mathrm{NP}}$-nodes, which is again the intersection of $\mathcal{P}_{\mathrm{NP}}$ of s and $\mathcal{Q}_{\mathrm{NP}}$ of d. Unfortunately, this remains $\emptyset$ even if using the extended P-space $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ would be an option. This means that there are networks that cannot be 100 % protected against node failures by nor "plain" neither extended remote LFA.

From the above discussion, one can easily see that the requirements of node protection are stricter than those for link protection. We can summarize our observations as follows.

Lemma 3 *$\mathcal{P}_{\mathrm{LP}} = \mathcal{P}_{\mathrm{NP}}$ but $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}} \subseteq \mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$, and $\mathcal{Q}_{\mathrm{NP}} \subseteq \mathcal{Q}_{\mathrm{LP}}$.*

Proof First, note that it was already concluded that the protection scheme does not affect P-spaces. Second, to prove the connection between $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$ and $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$ we use Observations 2 and 4. In the case of $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$, there is a node n: $\mathrm{dist}(v, n) < \mathrm{dist}(v, e) + \mathrm{dist}(e, n)$, where $v \in \mathrm{neigh}(s)$ and e is the default next-hop. Due to triangle inequality $\mathrm{dist}(v, e) \leq \mathrm{dist}(v, s) + \mathrm{dist}(s, e)$, and using this in our formal definition of $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}}$ results that $\mathrm{dist}(v, n) < \mathrm{dist}(v, e) + \mathrm{dist}(e, n) \leq \mathrm{dist}(v, s) + \mathrm{dist}(s, e) + \mathrm{dist}(e, n)$, which corresponds to the formal definition of $\mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$. Therefore, $\mathcal{P}^{\mathrm{e}}_{\mathrm{NP}} \subseteq \mathcal{P}^{\mathrm{e}}_{\mathrm{LP}}$. Third, since $\mathcal{Q}_{\mathrm{LP}}$ and $\mathcal{Q}_{\mathrm{NP}}$ are actually the loop-free criteria of link and node-protecting LFAs, respectively, the property $\mathcal{Q}_{\mathrm{NP}} \subseteq \mathcal{Q}_{\mathrm{LP}}$ is inherited from pure LFA. □

Hereafter, the terms $\mathcal{P}_{\mathrm{LP}}$ and $\mathcal{P}_{\mathrm{NP}}$ will be used without subscript to highlight that these sets do not differ under link protection and node protection.

7 Analysis of "plain" remote LFAs

Next, we return to the case of plain remote LFAs as this is the option that is expected to be supported first by commercial routers. Hence, in the rest of the paper we consider only the standard definitions for $\mathcal{P}$s, $\mathcal{Q}_{\mathrm{LP}}$ and $\mathcal{Q}_{\mathrm{NP}}$.

We give a graph-theoretical characterization of rLFA coverage, as measured by $\mu_{\mathrm{LP}}(G)$ and $\mu_{\mathrm{NP}}(G)$. Our main aim is to identify the attainable *lower and upper bounds* of plain rLFA failure coverage against both link and node failures. We describe some methods to easily calculate failure coverages in different families of graph topologies notable in building resilient networks. In the course of the analysis, our aim is to generalize previous propositions stated for LFAs in [13,40,41] to rLFA. First, we deal with single link failures, then in the second subsection we focus on node protection as well.

7.1 Link protection

7.1.1 Graphs with good coverage

Network operators facing with the challenge of deploying remote LFA need to ask the question, whether their current network topology is amenable to rLFA or not. Therefore, it is crucial to separate graph topologies that are "good" for

rLFA$_{\text{LP}}$ (i.e., the ones with $\mu_{\text{LP}}(G) = 1$) away from those that attain a particularly low coverage. First, we characterize the good cases for rLFA$_{\text{LP}}$.

Theorem 4 *Let G be an undirected, simple graph with uniform link costs. Now, $\mu(G) = 1$, if and only if for each $(i, j) \in E : \exists n \neq i, j$ so that $dist(i, n) = dist(j, n)$.*

Proof The result comes from applying (3) and (4) directly to (i, j). Therefore, $\mathcal{P}_{\text{LP}}$ can be defined as $\text{dist}(i, n) < \text{dist}(i, j) + \text{dist}(j, n)$, while $\mathcal{Q}_{\text{LP}}$ as $\text{dist}(j, n) < \text{dist}(n, i) + \text{dist}(i, j)$. Since link costs are unit cost, then $\text{dist}(i, j) = 1$, accordingly $\text{dist}(i, n) < 1 + \text{dist}(j, n)$ and $\text{dist}(j, n) < \text{dist}(n, i) + 1 \rightarrow \text{dist}(j, n) + 1 < \text{dist}(i, n) < \text{dist}(j, n) + 1 \Rightarrow \text{dist}(i, n) = \text{dist}(j, n)$. The backward direction of the proof comes from Corollary 1. □

Notable graph topologies with 100 % failure coverage include chordal graphs [18] (see Fig. 5d), infinite grids (see Fig. 5b) and "Möbius ladder" topologies (see Fig. 5c).

7.1.2 Worst-case graphs with rLFA$_{LP}$

In the following, we turn to discuss *lower bounds* for rLFA$_{\text{LP}}$, that is, we seek worst-case graphs, whose coverage against single link failures is particularly poor.

It has been observed previously that quintessential worst-case graphs for IPFRR are rings, i.e., cycle graphs in which all nodes are of degree two [9,16]. Consequently, we consider odd rings first, and then we shall treat even rings. Before that, we repeat a previous proposition from [41], which proved the lower bounds on the failure case coverage of link protecting LFA, denoted therein by $\eta_{\text{LP}}(G)$:

Proposition 1 *For an even ring on n nodes $\eta_{LP}(G) = \frac{1}{n-1}$, and for an odd ring on n nodes $\eta_{LP}(G) = \frac{2}{n-1}$.*

Next, we generalize these results to rLFA$_{\text{LP}}$. In fact, we shall do a bit more, as our analysis will account for the length of the repair tunnel, which is an important factor in provisioning remote LFA.[4]

Theorem 5 *Let C_n be an odd ring on n nodes with $n \geq 3$, and let $1 \leq k \leq \frac{n-1}{2}$ denote an upper bound on the length of the tunnel from the source node to its rLFA. Then, $\mu(C_n) = \frac{2k}{n-1}$.*

Proof Consider a ring topology on n nodes, n odd, let $(s, d) \in E$ be a *neighboring* source–destination and suppose that the link between them went down. In this case s needs to find a possible remote loop-free alternate since it cannot use its other neighbor because it will pass back the packet. Thus, the possible repair tunnel endpoints are situated on the other side of the ring with respect to the failed link, i.e., if an arbitrary node $u \in \text{rLFA}_{\text{LP}}(s, d)$, then $\text{dist}(s, u) \leq \frac{n-1}{2}$ which is tight if $d \in \text{neigh}(s)$. One can observe that if maximal tunnel length is permitted, i.e., $k = \frac{n-1}{2}$, then such kind of repairing node always exists ($\mu_{\text{LP}}(C_n) = \frac{n(n-1)}{n(n-1)} = 1$). However, if the tunnels need to be shorter than an arbitrary node u can only be an rLFA$_{\text{LP}}$ is $\text{dist}(s, u) \leq \frac{n-1}{2} - l$, where l is the tunnel shortening coefficient, i.e., the greater the l, the shorter the tunnel. Trivially, shortening the tunnel with l dissolves the protection among $\forall(s, d)$ pairs, where $dist(s, d) = l$. Therefore, rLFA$_{\text{LP}}$ failure coverage can be modified as follows: $\mu_{\text{LP}}(C_n) = \frac{n(n-1-2l)}{n(n-1)}$. Now, consider $\text{dist}(s, u) \leq k$, where k represents the length of the tunnel. In this manner $l = \frac{n-1}{2} - k$ meaning that $\mu_{\text{LP}}(C_n) = \frac{n-1-n+1+2k}{n-1} = \frac{2k}{n-1}$. □

Note that $k = 1$ means that only neighboring nodes can be used as repair tunnel endpoints, which essentially corresponds to simple loop-free alternates. In this case, Theorem 5 yields the same result as Proposition 1 stated for LFA$_{\text{LP}}$ for odd rings.

Theorem 6 *Leg C_n be an even ring on n nodes with $n \geq 4$, and let $1 \leq k \leq \frac{n-2}{2}$ denote an upper bound on the length of the tunnel from the source node to its rLFA$_{LP}$. Then, $\mu_{LP}(G) = \frac{2k-1}{n-1}$.*

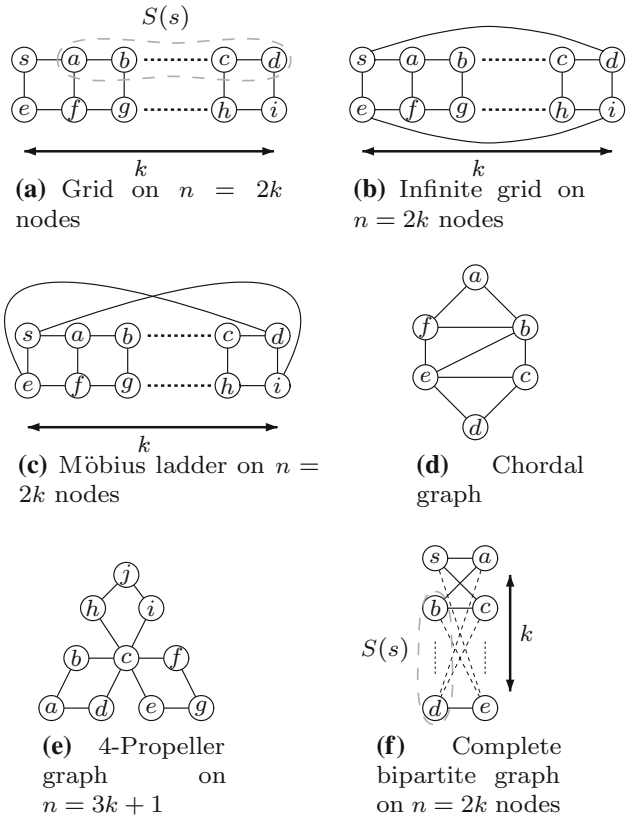
Proof Consider a ring on n nodes, n even, and suppose that link between an arbitrary *neighboring* (s, d) source–destination pair went down. According to the case of odd ring, s need to pass the packet to the other side of the ring, however, the possible repair tunnel endpoints cannot be reached without traversing the failed component. Thus, for $\forall(s, d)$ pairs, where $d \in neigh(s)$: the link (s, d) cannot be protected. One can observe, if $\text{dist}(s, d) \geq 2$, then tunnels, avoiding the failed link exist. Therefore, for an arbitrary source s has remote LFAs to $\forall d$ destination excluding its neighbors ($\mu_{\text{LP}}(C_n) = \frac{n(n-3)}{n(n-1)}$). However, assuming shorter tunnels results that for possible $u \in \text{rLFA}_{\text{LP}}(s, d)$: $\text{dist}(s, u) \leq \frac{n}{2} - l$, where l is a shortening coefficient as it was in the case of odd rings. Now, $l = \frac{n}{2} - k$ meaning that $\mu_{\text{LP}}(C_n) = \frac{n-1-n+2k}{n-1} = \frac{2k-1}{n-1}$. □

As before, supposing $k = 1$ results the corresponding statement in Proposition 1 for LFA$_{\text{LP}}$ for even rings. In this regard, rLFA$_{\text{LP}}$ can be seen as a natural generalization of LFA$_{\text{LP}}$.

7.1.3 Worst-case scenarios for rLFA$_{LP}$: 2-node-connected graphs

Below, we continue our analysis towards finding 2-node-connected graphs with low rLFA$_{\text{LP}}$ failure coverage. In what

[4] See the `remote-lfa maximum-cost` option on [11].



(a) Grid on $n = 2k$ nodes

(b) Infinite grid on $n = 2k$ nodes

(c) Möbius ladder on $n = 2k$ nodes

(d) Chordal graph

(e) 4-Propeller graph on $n = 3k + 1$

(f) Complete bipartite graph on $n = 2k$ nodes

Fig. 5 Illustration topologies

follows, we suppose that there is no constraint on the length of the tunnel.

Since the simplest 2-node-connected network with low failure coverage is a 4-cycle ($\mu_{\mathrm{LP}}(C_4) = \frac{1}{3}$), we examined graphs that contain a large number of 4-cycles as subgraphs. We considered the networks depicted in Fig. 5a where k denotes the number of 4-cycles, and Fig. 5f where k marks the number of node pairs. The following theorem concludes the results:

Theorem 7 *For any $k > 2$ there is a 2-node-connected graph G on $n = 2k$ nodes with $\mu_{LP}(G) = \frac{k-1}{2k-1}$.*

As a proof, we show that grids (G_k) and complete bipartite graphs ($K_{k,k}$) attain this limit. In grids, $\forall(s, d)$ pairs: $d \in \mathrm{neigh}(s)$ or $d \in \mathrm{S}(s)$ cannot be protected, where $\mathrm{S}(s)$ denotes the set of nodes situated on the same side. It is easy to see that every node is in a 4-cycle wherein neighbors as destinations are not protectable and the shortest paths to every node on the same side traverse one of the neighbors. Thus, such nodes are unprotected according to Lemma 1.

Similar is the case for $K_{k,k}$ as well. Each $d \in \mathrm{S}(s)$ are protected while $\forall d' \notin \mathrm{S}(s)$ are neighbors of s and, due to the property of bipartite graphs that every cycle is even, neighbors cannot be protected either.

7.1.4 Worst-case scenarios for rLFA$_{LP}$: 2-edge-connected graphs

So far, we have seen that in 2-node-connected graphs as many as 50 % of the node-pairs can go unprotected by rLFA against single link failures. Below, we show that in slightly less dense 2-edge-connected graphs the situation can be even worse.

Theorem 8 *For any $k \geq 1$ there is a 2-edge-connected graph G on $n = 3k + 1$ nodes with $\mu_{LP}(G) = \frac{1}{3}$.*

As a proof, we show that the so called "4-propeller graph" (P_k) attains this limit. Thus, consider (P_k) depicted in Fig. 5e where k denotes the number of blades. One can see that the nodes on the pitch of the propeller blades have remote LFAs to every destination except the neighbors, since they are on an even cycle. Nodes on the side of the blades considered as sources can only protect adjacent link failures if the nodes in the face of them are considered as destinations. Finally, the node in the middle has remote LFAs only for destination nodes situated on the pitch of the blades. Thus,

$$\mu_{\mathrm{LP}}(G) = \frac{k(3k-2)+2k+k}{3k(3k+1)} = \frac{3k^2+k}{3k(3k+1)} = \frac{k(3k+1)}{3k(3k+1)} = \frac{1}{3}.$$

7.2 Node protection

Next, we turn to find the graphs with the lowest and highest rLFA$_{\mathrm{NP}}$ coverage, as measured by μ_{NP}. First, we characterize the good cases and show that, as it was in the link-protecting case, there exist graphs that can be fully protected against single node failures. Then, we show that even $\mu_{\mathrm{NP}}(G) = 0$ is possible, and this can be attained even in a not so complicated network topology.

7.2.1 Graph with good coverage

Since node protection is undefined between two arbitrary neighboring nodes, we need to analyze only those (s, d) pairs, where $\mathrm{dist}(s, d) > 1$. The following theorem concludes the results:

Theorem 9 *Let G be an undirected, simple graph with uniform link costs, and let $\mathcal{S}_2$ be a set of 2-neighbors in G: $(u, v) : dist(u, v) = 2$. Now, $\mu_{NP} = 1$, if and only if for each $(s, d) \in \mathcal{S}_2$ there exists n for which*

$$dist(s, n) = dist(n, d) \text{ or } dist(s, n) + 1 = dist(n, d) .$$

Proof Consider the (s, d) pair in $\mathcal{S}_2$ depicted in Fig. 6, where $\mathrm{dist}(s, d) = 2$ and the wavy lines denote the existence of paths among the nodes. In this case, for n to be rLFA$_{\mathrm{NP}}(s, d)$ it has to fulfill (3) and (5), namely $\mathrm{dist}(s, n) < 1 + \mathrm{dist}(e, n)$ and $\mathrm{dist}(n, d) < \mathrm{dist}(e, n) + 1 \Rightarrow \mathrm{dist}(s, n) = \mathrm{dist}(n, d)$. On the other hand, consider now that $\mathrm{dist}(e, n) = k$. Then,

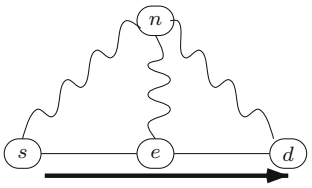


Fig. 6 Illustration for Theorem 9

due to triangle inequality, $\text{dist}(s, n)$ and $\text{dist}(n, d)$ can only be $1 \leq x \leq k$. However, if $x = 1$, then $\text{dist}(n, d)$ can only be $\text{dist}(s, n) + 1$, since if it does not, then the next-hop of s to destination d would not be node e. The backward direction of the proof comes from Lemma 2, as if a next-hop is rLFA_{NP} protected, then every further node, including the next-next-hop, is protected as well. Therefore, if it is true for each non-neighboring node pair, then $\mu_{\text{NP}}(G) = 1$. □

There is a bunch of networks for which the statement of Theorem 9 applies. For instance, odd and even rings, infinite grids, and "Möbius ladder" topologies all qualify.

7.2.2 Worst-case graphs for rLFA$_{NP}$

Next, we turn to discuss which networks are the most inconvenient for rLFA_{NP}. Note that there are certain graphs for which studying μ_{NP} does not make sense, as it happens to be undefined. Such is the case, for instance, of complete graphs with unit link costs: here, every node-pair is adjacent and hence rLFA_{NP} is not defined due to the last-hop problem. In our analysis, therefore, we only considered graphs in which at least one non-adjacent node pair exists (i.e., non-complete graphs). Even in these graphs the question is only interesting when single node failures, at least theoretically, can be repaired, so we shall focus only on 2-node-connected graphs.

Theorem 10 *For any $n > 4$, there is a 2-node-connected graph G on n nodes with $\mu_{NP}(G) = \frac{2(n-3)}{n^2-5n+6}$.*

Again, as a proof we show a particular graph on n nodes, hereafter denoted by L_n, that attains this limit. An example for L_n for the case when $n = 6$ is depicted in Fig. 7. The

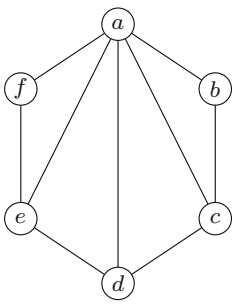


Fig. 7 Worst-case graph for rLFA_{NP} on $n = 6$ nodes

main topological characteristic of L_n is that there is one node on the top with degree of $n - 1$, there are two nodes with degree of 2, while the remaining $n - 3$ nodes have a degree of 3. Correspondingly, the number of non-adjacent source–destination pairs is $2(n-3)+(n-3)(n-4) = n^2-5n+6$. For each non-neighboring node pair (s, d) : $\text{dist}(s, d) = 2$ via the node on the top. One easily sees, in addition, that only those node pairs can be protected that have ECMPs to each other, that is, which are in opposite in the 4-cycles. The number of such node pairs equals twice the number of 4-cycles in the graph (i.e., $n-3$), and therefore there are $2(n-3)$ protected node pairs. Consequently, we have $\mu_{\text{NP}}(\text{L}_n) = \frac{2(n-3)}{n^2-5n+6}$. Observe that, in the limit, this bound tends to zero, meaning that in very large L_n graphs the fraction of rLFA node-protected source–destination pairs diminishes.

So far, we have sought a tight characterization for the lower bound on μ_{LP} and μ_{NP} for any unweighted graph G. At the moment, we do not have clear answers to this intriguing but hard graph-theoretical problem. What we could prove, however, is that in certain 2-node-connected unweighted graphs $\mu_{\text{LP}}(G)$ can be as low as $\frac{1}{2}$, and in 2-edge-connected graphs an even lower threshold of $\frac{1}{3}$ is also realizable. So far, we have not been able to identify any 2-node-connected or 2-edge-connected graph with smaller rLFA_{LP} coverage. Thus, we conjecture that $\frac{k-1}{2k-1}$ is an actual lower bound on $\mu_{\text{LP}}(G)$ for 2-node-connected graphs, while $\frac{1}{3}$ is a lower bound on $\mu_{\text{LP}}(G)$ for 2-edge-connected graphs. In the case of node protection, we have stronger results: we could show that there exist certain large graphs with $\mu_{\text{NP}} \to 0$, which, evidently, is a lower bound, at least in the limit. Regarding graphs of practical size, however, we do not have better lower bound at the moment than the one for L_n graphs.

7.2.3 Computational study

It turned out that finding a universal lower bound on rLFA_{LP} or rLFA_{NP} coverage is a hard problem. Clearly, a computational approach might be instructive to support or refute our conjectures. Hence, we generated all non-isomorphic networks on n nodes where $n \in \{1, 2, \ldots, 9\}$. Note that the generation is very time consuming even if only non-isomorphic graphs are created. Table 1 summarizes the lower bounds with the following notations: n denotes the number of nodes, μ^{2e}_{LP} and μ^{2n}_{LP} notes the failure coverage against single link failures in case of 2-edge-connected and 2-node-connected networks, while μ_{NP} denotes the failure coverage against single node failures in non-complete 2-node-connected networks. The columns marked by B_l denote the conjectural lower bounds.

In the case of link protection, it can be seen that until $n \leq 4$ results are the same, and if $n \geq 5$ coverages start to increase. One can observe that in the case of $n = 7$ the

Table 1 Lower bounds measured by μ_{LP} and μ_{NP} in worst-case graphs on n nodes

n	μ_{LP}^{2e}	B_l	μ_{LP}^{2n}	B_l	μ_{NP}	B_l
3	1	1	1	1	Undefined	Undefined
4	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$	1	1
5	$\frac{2}{5}$		$\frac{2}{5}$		$\frac{2}{3}$	$\frac{2}{3}$
6	$\frac{2}{5}$		$\frac{2}{5}$	$\frac{2}{5}$	$\frac{1}{2}$	$\frac{1}{2}$
7	$\frac{1}{3}$	$\frac{1}{3}$	$\frac{3}{7}$		$\frac{2}{5}$	$\frac{2}{5}$
8	$\frac{19}{56}$		$\frac{3}{7}$	$\frac{3}{7}$	$\frac{1}{3}$	$\frac{1}{3}$
9	$\frac{1}{3}$		$\frac{31}{72}$		$\frac{2}{7}$	$\frac{2}{7}$

given failure coverage equals to the coverage attained by 4-propeller graphs mentioned above. It also shows that lower bounds of 2-edge-connected networks are the lowest.

The case of node protection turns out different. If $n = 4$ the coverage is 1, while if $n > 4$ the coverages start to decrease. The most important observation is that the conjectured lower bounds are tight for every 2-node-connected network until $n = 9$ nodes suggesting that the result of Theorem 10 hold true as the attainable lower bound on μ_{NP} in 2-node-connected networks.

8 Remote LFA graph extension

As observed, there exist a lot of graphs with small failure coverage, measured in terms of μ_{LP} and μ_{NP}. Hence, in this section we ask to what extent we need to intervene at the graph topology to improve coverage to 100 % in both link and node-protecting cases. This problem is important since (i) this would answer how "far" are poorly protected networks from perfect rLFA failure coverage and (ii) would provide an easy way for operators to boost the protection in their networks. We adapt the formal description of the LFA graph extension problem from [41] to link-protecting rLFA as follows:

Definition 1 Link-protecting rLFA graph extension problem: Given a graph $G(V, E)$, find the smallest subset F of the complement edge set $\overline{E}$ of G such that $\mu_{\mathrm{LP}}(G(V, E \cup F) = 1$.

Similarly, in the case of node protection this definition can be formalized as follows:

Definition 2 Node-protecting rLFA graph extension problem: Given a graph $G(V, E)$, find the smallest subset F of the complement edge set $\overline{E}$ of G such that $\mu_{\mathrm{NP}}(G(V, E \cup F) = 1$.

At the moment, we do not know the complexity of this problem but, based on our former experience with similar network optimization problems for LFA, it seems highly probable that it is also NP-complete. To actually solve the problem, we adopted the *greedy graph extension algorithm* from [41], which, at least for LFA, performed almost the same as the optimal algorithm, but it is much faster and simpler. Here, we extend this algorithm to the case of rLFA, both for the link-protecting and the node-protecting cases. Moreover, we also developed a *simulated annealing-based heuristics* as another approach to complement our studies in increasing the rLFA failure coverage in different kinds of networks.

First, we show the greedy graph extension method. This algorithm adds the best edge from the complement edge set that improves the coverage at most. Formally, the algorithm is defined as follows:

Algorithm 1 Greedy rLFA graph extension for graph $G(V, E)$

```
1: while μ(G(V, E)) < 1
2:     (u, v) ← argmax_{(i,j)∈Ē} μ(G(V, E ∪ {(i, j)}))
3:     E ← E ∪ {(i, j)}
4: end while
```

Note that the pseudo-code works the same for the link-protecting and the node-protecting case. The variant for the link-protecting case is called the *greedy link-protecting rLFA graph extension algorithm*, while the one optimizing for node-protection is called the *greedy node-protecting rLFA graph extension algorithm*. The following theorems characterize the terminating conditions of these algorithms.

Theorem 11 *Let $G(V, E)$ be a graph with unit link costs. Then, the greedy link-protecting rLFA graph extension algorithm terminates with full link-protecting rLFA coverage regardless of the input graph.*

Proof Algorithm 1 surely terminates when all complement links are added, but at this point $\mu_{\mathrm{LP}}(G) = 1$ as complete graphs have full link-protecting rLFA coverage. □

Theorem 12 *Let $G(V, E)$ be a graph with unit link costs and suppose that $G(V, E)$ is not a complete graph. Then, the greedy node-protecting rLFA graph extension algorithm terminates with full node-protecting rLFA coverage regardless of the input graph.*

Proof We cannot use the same approach directly as above, because with all the complement edges added we again reach a complete graph but for this graph μ_{NP} is not defined (recall the discussion in Sect. 7.2). We observe, however, that if we add all the complement edges except one, then we get an almost complete graph in which node protection is defined between one and only one node pair. As this node pair is trivially protected against a single node failure (since the nodes in the pair are not neighbors and they are situated in a 4-cycle), therefore $\mu_{\mathrm{NP}} = 1$ for this graph. As the algorithm is guaranteed to converge to this graph, unless the input is a

complete graph or the algorithm terminates previously, the proof is complete. □

Next, we turn to the other algorithm. We chose the simulated annealing probabilistic metaheuristic as the main framework, and within this framework we obtained different heuristics by different objective functions. Basically, the algorithm works as follows: given an input graph $G(V, E)$, we try to augment the graph with a randomly chosen edge (i, j) from the complement edge set $\overline{E}$. If the failure coverage was improved, then we unconditionally accept this edge. Otherwise, if the coverage is worse, then the edge could still be accepted with a certain probability, depending on the actual objective function and a system parameter called the temperature, which was initially set to relatively high value and is decreased in every iteration. This ensures the system to escape easily from local optima in the beginning, and eventually get stuck in a good quality optimum. The process terminates at that time when temperature is dropped to 0 or failure coverage reached 1. We also used *tabu lists* to preclude the iteration from oscillating between two or more already tested new edge.

Algorithm 2 Simulated Annealing based rLFA graph extension for graph $G(V, E)$

```
T ← T_0
while μ(G(V, E)) < 1
    choose_random_edge((i, j) ∈ Ē)
    if accept_edge(Δμ, T) then
        E ← E ∪ {(i, j)}
    end if
    T ← T − 1
end while
```

The pseudo-code for the simulated annealing based heuristic is given in Algorithm 2. Note again, that it works similarly for the link-protecting and node-protecting case. The pseudo-code uses two procedures, specified as follows:

- choose_random_edge(i, j) selects randomly an edge $(i, j) \in \overline{E}$ to be added to the network.
- accept_edge$(\Delta\mu, T)$: after adding a randomly selected new edge to the network, the new failure coverage $\mu(G(V, E \cup (i, j)))$ is examined. If it was improved, then the new edge is added irrevocably to the network. Otherwise, two different objective functions (ΔD) are evaluated in order to check how bad the new solution is. One of them only checks how the failure coverage declined, formally $\Delta D = \mu(G(V, E)) - \mu(G(V, E \cup (i, j)))$. Besides, the other objective function takes into account the number of newly added links as well in order to try to keep it low, formally: $\Delta D = \Delta NP - 2m$, where NP is the number of protected source–destination pairs and m is the number of all the edges including the newly added one as well. These objective functions are tested via the so called metropolis test. Metropolis test is used in simulated annealing heuristics to accept "bad" solutions if it suits for a criterion, namely $e^{(\frac{-\Delta D}{T})} > R(0, 1)$, where ΔD is the change of the objective function, T is the actual temperature of the system, and $R(0, 1)$ is a random number in the interval $[0, 1]$. According to the output of this test, the newly added edge is left in the network permanently or promptly removed. One can observe that the two different objective functions result two different kind of heuristics. Therefore, let SA_{co} be the simulated annealing with the former objective function, and let SA_{cne} be the simulated annealing with the latter objective function.

8.1 Numerical evaluations

In this section, we examine how many links one must add in realistic graphs to achieve full rLFA coverage, both against single link- and node-failures. We chose existing real-world topologies inferred from the Rocketfuel [32] data set, the SNDLib [44] graph library, and the Topology Zoo project [27]. In all topologies, we set link costs uniformly to 1. Note that there are networks in the data set where inferred link costs were exactly unit costs.

We executed the greedy algorithm as well as the simulated annealing based heuristics. From the latter we executed 20,000 rounds, with initial temperature $T_0 = 150$ and tabu list size of 20. The detailed results of the link-protecting case are in Table 2 with the following notations: n is the number of nodes and m is the number of links in $G(V, E)$; η_{LP} is the initial link-protecting LFA coverage; μ_{LP} is the initial link-protecting rLFA coverage; Gr_{η} denotes the number of new links added by the LFA greedy graph extension algorithm to reach 100 % link-protecting LFA coverage, while Gr_{μ} gives the same result for remote LFA. $SA_{\Delta\mu}$ denotes the number of new links added by SA_{co}, and last but not least, SA_{γ} marks the number of new links added by SA_{cne}.

The first observation is that there were five networks that were fully protected with rLFA right away, even without the need of any graph extension. Second, the number of links that have to be added to reach full coverage with rLFA is much less than when only simple link-protecting LFA capable routers are present, irrespectively of which graph extension method was used. Nevertheless, on average the number of links added by the different simulated annealing based heuristics is greater than in the case of the greedy algorithm. This suggests that for the graph extension problem the greedy approach is the most plausible solution and, if we can draw conclusions from the case of pure LFA in [41], it probably performs very close to the optimal solution too. The largest

Table 2 Remote LFA graph extension results for link protection

Topology	n	m	η_{LP}	Gr_η	μ_{LP}	Gr_μ	$SA_{\Delta\mu}$	SA_γ
AS1221	7	9	0.833	1	0.833	1	1	1
AS1239	30	69	0.898	6	1	0	0	0
AS1755	18	33	0.889	4	1	0	0	0
AS3257	27	64	0.946	3	0.954	1	1	1
AS3967	21	36	0.864	7	0.969	1	1	1
AS6461	17	37	0.919	2	1	0	0	0
Abilene	12	15	0.56	6	0.833	1	1	1
Arnes	41	57	0.595	18	0.731	6	9	12
AT&T	22	38	0.823	6	0.8875	2	2	2
Deltacom	113	161	0.542	79	0.885	4	7	11
Gambia	28	28	0.037	16	0.111	8	12	13
Geant	37	55	0.646	20	0.827	4	5	5
Germ_50	50	88	0.801	22	1	0	0	0
Germany	27	32	0.695	1	0.882	1	1	1
InternetMCI	19	33	0.877	3	0.888	2	2	2
Italy	33	56	0.784	12	0.951	2	2	2
NSF	26	43	0.86	9	1	0	0	0

Table 3 Remote LFA graph extension results for node protection

Topology	n	m	η_{NP}	Gr	μ_{NP}	Gr	$SA_{\Delta\mu}$	SA_γ	μ^e_{NP}	Gr^e_μ	$SA^e_{\Delta\mu}$	SA^e_γ
AS1221	7	9	0.083	3	0.083	1	1	1	0.083	1	1	1
AS1239	30	69	0.658	16	0.843	1	1	1	0.928	1	1	1
AS1755	18	33	0.704	7	0.912	1	1	1	1	0	0	0
AS3257	27	64	0.521	20	0.702	5	8	8	0.866	3	3	3
AS3967	21	36	0.715	10	0.896	2	2	2	0.994	1	1	1
AS6461	17	37	0.505	8	0.596	3	3	3	0.747	2	2	2
Abilene	12	15	0.608	3	0.725	2	2	2	0.872	1	1	1
Arnes	41	57	0.331	35	0.426	12	24	20	0.45	12	16	15
AT&T	22	38	0.565	12	0.684	4	4	4	0.849	2	2	2
Deltacom	113	161	0.436	113	0.818	9	25	27	0.868	9	22	23
Gambia	28	28	0.04	23	0.12	14	17	22	0.12	13	19	18
Geant	37	55	0.411	30	0.676	5	11	11	0.74	5	8	8
Germ_50	50	88	0.676	37	0.977	1	1	1	0.998	1	1	1
Germany	27	32	0.599	8	0.77	2	2	2	0.955	2	2	2
InternetMCI	19	33	0.558	9	0.837	3	2	2	0.916	1	1	1
Italy	33	56	0.574	24	0.839	3	3	3	0.926	2	2	2
NSF	26	43	0.634	16	0.963	1	1	1	1	0	0	0

improvement in rLFA coverage, compared to simple LFA, is seen in networks where initially $\eta(G) < 0.9$ (see, e.g., in the Geant topology). In the Deltacom topology, the installation of 79 new links was necessary to achieve full LFA coverage, while with only 4 additional links full rLFA coverage is attainable. The results indicate that (i) more than 50 % of the networks lend themselves to rLFA extension since the maximum number of links needed is less than 2; (ii) on average 3.6 new links are necessary to attain 100 % rLFA coverage while in case of simple LFA this number is 14.5.

In the second run, we examined how the proposed algorithms could improve the failure coverage against single node failures. Since the extended rLFA variant can play an important role in the case of node protection, even if the link costs are uniform, we evaluated that possibility as well. Table 3 contains the results, where again n is the number of nodes

and m is the number of links in $G(V, E)$; η_{NP} is the initial node-protecting LFA coverage; μ_{NP} is the initial node-protecting rLFA coverage, while μ^e_{NP} is the initial node-protecting rLFA coverage with the extended rLFA option. Gr_η denotes the number of new links added by the LFA greedy graph extension algorithm. Gr_μ marks the number of new links added by the rLFA greedy graph extension algorithm, while in the case of Gr^e_μ the extended P-space option was also considered. The results in column $SA_{\Delta\mu}$ and SA_γ are similar to the link-protecting case, while columns $SA^e_{\Delta\mu}$ and SA^e_γ indicates the number of links added by the two simulated annealing based heuristics, under the assumptions that routers were able to use their extended P-space.

The first observation is that, if simple rLFA is considered then there is no network, which is initially fully rLFA protected against node failures. However, if the routers are able to use their extended P-space, then there were 3 networks with full protection out of the box. As it was in the link-protecting case, much less additional edges are needed for 100 % node-protecting rLFA failure coverage than when only simple node-protecting LFAs are only available. For instance, in Deltacom topology, 113 new edges were necessary to protect all source–destination pairs with pure LFA against single node failures, while this number is only 9 when remote LFAs can be used as well. One also can observe that the greedy approach yielded the best solutions, i.e., it needed the fewest additional edges in order to provide full protection. Namely, in the case of simple rLFA, on average it installs 4.05 new links to the network, while simulated annealing based algorithms could not reach full protection with less than 6.35 new links. Nevertheless, if extended P-space is an option, then greedy algorithm needed on average 3.3 new links, whilst the other two heuristics resulted more than 4.75 new links.

Overall, the results suggest that network operators might hugely benefit from deploying rLFA in their routers, since it can definitely protect much more source–destination pairs than pure LFA ever could do. Moreover, the provisioning of a very few number of additional new links can boost the protection provided by rLFA up to 100 %. In particular, we found that more than 50 % of the networks needed less than 4 additional links for perfect rLFA failure case coverage.

9 Conclusions

Currently, loop-free alternates is the best choice for providing fast protection in pure IP and MPLS/LDP networks, as it is readily implemented in basically all commercial IP router offerings. It is a well-known fact that LFA cannot protect every single failure. In our previous works, we showed that improvements can be made by altering the existing network topology. If modifying the network is not an option, remote LFAs may be a better approach.

As in the case of LFA, the number of failure cases protected by rLFA crucially depends on both the graph topology and the link costs. As it seems difficult to consider both at the same time, we studied graph topological concerns separately from the effect of link costs in this paper. This restriction is plausible as a first approach, and we definitely plan to generalize our results to weighted graphs in a subsequent work.

For the first time in the literature, we analyzed rLFA failure coverage in general networks by a new set of elemental graph theoretical rLFA tools. Moreover, we extended the basic specification of rLFA [7], originally defined for single link failures only, to the relevant case of single node failures, and we also made a deep analysis to this case with our toolset.

We showed that, under the unit-cost assumption, "extended P-space" results full rLFA failure coverage in every network against single link failures. This can be an important pointer for operators, currently in the position to deploy rLFA, on how to actually choose link costs. Unfortunately, it turned out that in the case of node protection this option is not enough to protect all source–destination pairs.

We gave sufficient and necessary conditions for a unit cost graph to be 100 % protectable with rLFA against both link and node failures. Then, we studied general lower and upper bounds for rLFA coverage. For upper bounds, we showed that in both link- and node-proteting cases, full rLFA coverage can be attained. For lower bounds, in the case of link protection, we found that for 2-node-connected graphs on $2k$ nodes the value $\frac{k-1}{2k-1}$ is realizable by grids and complete bipartite graphs and we confirmed computationally that this is a valid lower bound as long as the number of nodes n is smaller than 10. We also found that for 2-edge-connected graphs, this "conjectured" lower bound is $\frac{1}{3}$. We also found that for node failures rLFA coverage can, somewhat unexpectedly, straight out drop to zero in certain cases.

We defined the rLFA graph extension problem as the task to augment an unweighted graph with the fewest new links to obtain 100 % failure case coverage. Along a simple greedy algorithm we also developed a family of simulated annealing based heuristics to solve this problem approximately. We found that, as it was in the case for pure LFA [41], the greedy method is the most plausible algorithm. It turned out that, even in very big real-world ISP topologies, adding only 2–3 new links is enough to attain 100 % failure coverage against link failures, whilst the number of new links needed for full protection against node failures is only slightly more, 3–4.

In the future, we plan to study further remote LFA related network optimization questions. For instance, in the unweighted case improving rLFA coverage is possible with modifying link costs as well, which looks another intriguing, and practically relevant, network optimization problem.

Acknowledgments The authors thank the support of High Speed Networks Laboratory at BME. This project was supported by TÁMOP 4.2.2.B-10/1-2010-0009 and OTKA-PD 104939 grants. Levente Csikor was supported by the hungarian Sándor Csibi Research Grant.

References

1. Ahn, G., Jang, J., & Chun, W. (2002). An efficient rerouting scheme for mpls-based recovery and its performance evaluation. *Telecommunication Systems*, *19*(3–4), 481–495. doi:10.1023/A:1013806925464.
2. Amund, K., Fosselie, H. A., Čičic, Tarik, Stein, G., & Olav, L. (2009). Multiple routing configurations for fast IP network recovery. *IEEE/ACM Transactions on Netwworking*, *17*(2), 473–486. doi:10.1109/TNET.2008.926507.
3. Andersson, L., Minei, I., & Thomas, B. (2007). Ldp specifiaction. RFC 5036.
4. Antonakopoulos, S., Bejerano, Y., & Koppol, P. (2012). A simple ip fast reroute scheme for full coverage. In *2012 IEEE 13th International Conference on, High Performance Switching and Routing (HPSR)* (pp. 15–22). doi:10.1109/HPSR.2012.6260822.
5. Atlas, A., & Zinin, A. (2008). Basic specification for IP fast reroute: Loop-Free Alternates. RFC 5286.
6. Bryant, S., Filfils, C., Previdi, S., & Shand, M. (2007). IP fast reroute using tunnels. IETF DRAFT.
7. Bryant, S., Filfils, C., Shand, M., & So, N. (2012). Remote LFA FRR. IETF DRAFT.
8. Bryant, S., Shand, M., & Previdi, S. (2010). IP fast reroute using Not-via addresses. Internet Draft.
9. Čičic, T. (2006). An upper bound on the state requirements of link-fault tolerant multi-topology routing. *IEEE ICC*, *3*, 1026–1031.
10. Cisco Systems: Cisco ios release 12.0 and network protocols configuration guide (2011).
11. Cisco Systems: Ip routing: Ospf configuration guide, cisco ios release 15.2s—ospf ipv4 remote loop-free alternate ip fast reroute (downloaded: Apr. 2012).
12. Császár, A., Enyedi, G., & Kini, S. (March 2011). Ip fast re-route with fast notification. Internet Draft.
13. Csikor, L., Nagy, M., & Rétvári, G. (2011). Network optimization techniques for improving fast ip-level resilience with loop-free alternates. *Infocommunications Journal*, *3*(4), 2–10.
14. Csikor, L., & Rétvári, G. (2012). IP fast reroute with remote loop-free alternates: The unit link cost case. In *Proceedings of the RNDM* (pp. 16–22).
15. Csikor, L., Rétvári, G., & Tapolcai, J. (2012). Optimizing igp link costs for improving ip-level resilience with loop-free alternates. *Computer Communications*. doi:10.1016/j.comcom.2012.09.004.
16. Enyedi, G., Rétvári, G., & Cinkler, T. (2007). A novel loop-free IP fast reroute algorithm. In *EUNICE*.
17. Enyedi, G., Szilágyi, P., Rétvári, G., & Császár, A. (2009). IP Fast ReRoute: Lightweight Not-Via without additional addresses. In *INFOCOM Mini-conf.*
18. Golumbic, M. C. (2004). *Algorithmic Graph Theory and Perfect Graphs* (2nd ed.). Amsterdam: Elsevier Science.
19. Gomes, T., oes, C.S., & Fernandes, L. (2011). Resilient routing in optical networks using srlg-disjoint path pairs of min-sum cost. *Springer Telecommunication Systems Journal*.
20. Hock, D., Hartmann, M., Menth, M., Pioro, M., Tomaszewski, A., & Zukowski, C. (2011). Comparison of ip-based and explicit paths for one-to-one fastreroute in mpls networks. *Springer Telecommunication Systems Journal*, 1–12. doi:10.1007/s11235-011-9603-4.
21. Hokelek, I., Fecko, M., Gurung, P., Samtani, S., Cevher, S., & Sucec, J. (Feb 2008). Loop-free ip fast reroute using local and remote lfaps. Internet Draft.
22. Iannaccone, G., Chuah, C.N., Mortier, R., Bhattacharyya, S., & Diot, C. (2002) Analysis of link failures in an ip backbone. In *ACM SIGCOMM Internet Measurement Workshop* (pp. 237–242).
23. ISO: Intermediate ststem-to-intermediate system (is-is) routing protocol. ISO/IEC 10589 (2002).
24. Iyer, S., Bhattacharyya, S., Taft, N., & Diot, C. (2003). An approach to alleviate link overload as observed on an IP backbone. In *INFOCOM*.
25. Jarry, A. (2013). Fast reroute paths algorithms. *Telecommunication Systems*, *52*(2), 881–888. doi:10.1007/s11235-011-9582-5.
26. Juniper Networks: Junos 9.6 routing protocols configuration guide (2009).
27. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., & Roughan, M. The internet topology zoo. http://www.topology-zoo.org (downloaded: Apr. 2012)
28. Kwong, K.W., Gao, L., Guerin, R., & Zhang, Z.L. (2010). On the feasibility and efficacy of protection routing in ip networks. In *INFOCOM, long version is available in Tech. Rep. 2009*. University of Pennsylvania.
29. Labovitz, C., Malan, G. R., & Jahanian, F. (1998). Internet routing instability. *IEEE/ACM Transactions on Networking*, *6*(5), 515–528.
30. Lakshminarayanan, K., Caesar, M., Rangan, M., Anderson, T., Shenker, S., & Stoica, I. (2007). Achieving convergence-free routing using failure-carrying packets. In *Proceedings of the SIGCOMM*.
31. Lee, S., Yu, Y., Nelakuditi, S., Zhang, Z.L., & Chuah, C.N. (2004). Proactive vs reactive approaches to failure resilient routing. In *INFOCOM*.
32. Mahajan, R., Spring, N., Wetherall, D., & Anderson, T. (2002). Inferring link weights using end-to-end measurements. In *ACM IMC* (pp. 231–236).
33. Markopoulou, A., Iannacone, G., Bhattacharyya, S., Chuah, C.N., & Diot, C. (Mar. 2004). Characterization of failures in an ip backbone. In *Proceedings of the IEEE Infocom*.
34. Médard, M., Barry, R. A., Finn, S. G., & Galler, R. G. (1999). Redundant trees for preplanned recovery in arbitrary vertex-redundant or edge-redundant graphs. *IEEE/ACM Transactions on Networking*, *7*(5), 641–652.
35. Menth, M., Hartmann, M., Martin, R., Čičic, T., & Kvalbein, A. (2010). Loop-free alternates and not-via addresses: A proper combination for ip fast reroute? *Computer Networks*, *54*(8), 41300–41315. doi:10.1016/j.comnet.2009.10.020.
36. Merindol, P., Pansiot, J.J., & Cateloin, S. (2008). Providing protection and restoration with distributed multipath routing. In *International symposium on, performance evaluation of computer and telecommunication systems, 2008. SPECTS 2008* (pp. 456–463).
37. Moy, J. (1998). Ospf version 2. RFC 2328.
38. Nagy, M., Tapolcai, J., & Rétvári, G. (2012). Optimization methods for improving ip-level fast protection for local shared risk groups with loop-free alternates. *Springer Telecommunication Systems Journal*.
39. Pan, P., Swallow, G., & Atlas, A. (2005). Fast reroute extensions to RSVP-TE for LSP tunnels. RFC 4090.
40. Rétvári, G., Csikor, L., Tapolcai, J., Enyedi, G., & Császár, A. (Oct. 2011). Optimizing igp link costs for improving IP-level resilience. In *Proceedings of the DRCN* (pp. 62–69).
41. Rétvári, G., Tapolcai, J., Enyedi, G., & Császár, A. (2011). IP fast ReRoute: Loop free alternates revisited. In *INFOCOM* (pp. 2948–2956).
42. Schollmeier, G., Charzinski, J., Kirstädter, A., Reichert, C., Schrodi, K., Glickman, Y., & Winkler, C. (2003). Improving the resilience in ip networks. In *Proceedings of the HPSR*.
43. Shand, M., & Bryant, S. (2010). IP Fast Reroute framework. RFC 5714.

44. SNDLib: Survivable fixed telecommunication network design library. http://sndlib.zib.de (downloaded: Apr. 2012).
45. Sterbenz, J., Cetinkaya, E.K., Hameed, M.A., Jabbar, A., Qian, S., & Rohrer, J.P. (2011). Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation and experimentation. *Springer Telecommunication Systems Journal*, 1–32. doi:10.1007/s11235-011-9573-6.
46. Vulimiri, A., Michel, O., Godfrey, P.B., & Shenker, S. (2012). More is less: reducing latency via redundancy. In *Hotnets*.
47. Zhong, Z., Nelakuditi, S., Yu, Y., Lee, S., Wang, J., & Chuah, C.N. (2005). Failure inferencing based fast rerouting for handling transient link and node failures. In *INFOCOM*.

Levente Csikor received the M.Sc. degree in technical informatics from the Budapest University of Technology and Economics (BME) in 2010. Now, he is a third-year PhD student at the High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, BME. He is also the member of MTA-BME Future Internet Research Group. He has experience in C/C++/LEMON, Java/JUNG, Linux and web development. His research interests include fast IP level resilience with (remote) loop-free alternates in order to provide high availability in the future Internet. Currently, he is also involved in researching MultiPath TCP using real Openflow enabled testbeds. Furthermore, he is interested in smart grid opportunities and research.

Gábor Rétvári received the M.Sc. and Ph.D. degrees in electrical engineering from the Budapest University of Technology and Economics (BME), Budapest, Hungary, in 1999 and 2007, respectively. He is now a Research Fellow at the High Speed Networks Laboratory, Department of Telecommunications and Media Informatics, BME. His research interests include QoS routing, Traffic Engineering and the networking applications of computational geometry and the mathematical theory of network flows. He is a Perl expert, maintaining numerous open source scientific tools written in Perl, C and Haskell.