IP Fast ReRoute: Loop Free Alternates Revisited

Gábor Rétvári, János Tapolcai

High Speed Networks Laboratory

Department of Telecommunications and Media Informatics

Budapest University of Technology and Economics

Email: {retvari, tapolcai}@tmit.bme.hu

Gábor Enyedi, András Császár

TrafficLab

Ericsson Telecommunications Hungary

Email: {gabor.sandor.enyedi,andras.csaszar}@ericsson.com

Backgrounds

- Many operators provide commercial telecom services over pure IP
- Legacy IP failure recovery is slow (>150 ms)
- For <50 ms resilience, IP-level protection is the way to go
- „Can we turn it on today?”
- „Well, sort of . . .”
- There *is* an IP fast-resilience scheme available in many off-the-shelf routers: Loop Free Alternates (LFA)
- But with LFA certain failure cases are impossible to repair
- Can we improve?
- **Not** by changing LFA!

IP Fast ReRoute

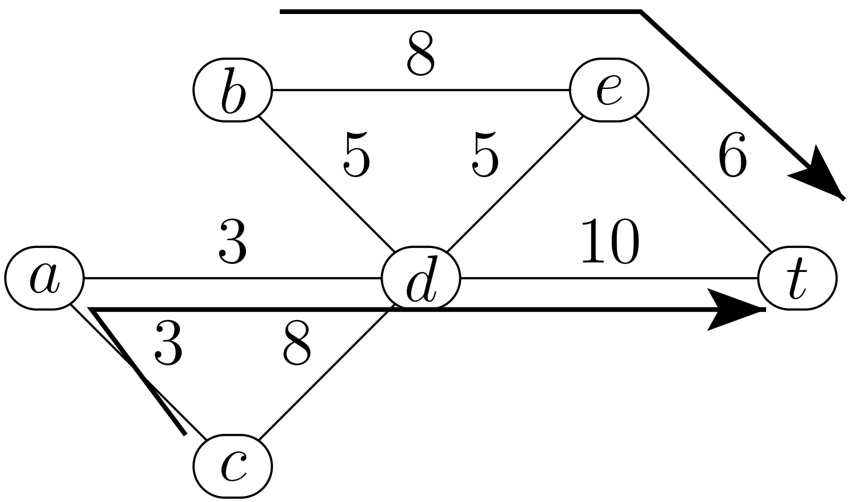
- A framework for fast protection implemented in pure IP
  - instant failure detection (e.g., BFD, layer 2)
  - switch to precomputed detours
  - locally route around the failure
  - then get packet back to shortest path
  - let the IGP converge in the background
  - recompute detours
- Benefits both pure IP and MPLS-LDP

Basic IPFRR: Loop Free Alternates

- Piggy-back IPFRR on a standard link-state IP shortest path routing protocol (OSPF, IS-IS)
- When next-hop goes away, pass packet on to a neighbor that still has an intact route to the destination
- Basically any neighbor that will not send it back
- Enough to ensure that the alternate neighbor is not upstream
- So it will not loop the packet back

Basic IPFRR: Loop Free Alternates

- In the sample network nodes are routers, destination is t
  - the default next-hop from b to t is e
  - if e goes away, b can still pass packets to d



- Nodes b, c, d and e all have an LFA to t
- Node a has no LFA: no fast protection!

Alternatives of LFA

- IPFRR is hard: destination-based forwarding does not play well with local rerouting
- For full protection, packets on detour must be distinguished from packets on default paths
- Alter destination-based forwarding (FIR & co.)

  S. Nelakuditi et al. „Fast local rerouting for handling transient link failures", INFOCOM'04.

  - consider packet's incoming interface in forwarding
  - full protection, but per-interface FIB is not supported
- Explicit failure signaling (e.g., remote LFAPs)

  I. Hokelek et al.„Loop-free IP Fast Reroute using local and remote LFAPs" Internet Draft, Feb 2008.

  - standalone signaling mechanism for IPFRR
  - operators reluctant to deploy

Alternatives of LFA

- In-band signaling (MRC, SafeGuard, IP redundant trees)

  A. Kvalbein et al. „Fast IP Network Recovery Using Multiple Routing Configurations", INFOCOM'06.

  - e.g., mark detours in the IP header
  - could never be pushed through IETF
- Tunneling (near-side/far-side tunneling, Not-via)

  S. Bryant et al. „IP fast reroute using Not-via addresses", Internet Draft, March 2007.

  - „lightweight in-band signaling": mark packets in destination address
  - wire-speed tunneling not reachable everywhere
  - MTU issues can cause debug nightmare
- Various combinations

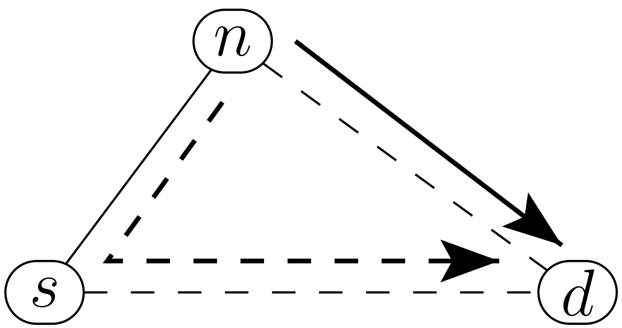
  M. Menth et al. „Loop-free alternates and not-via addresses: A proper combination for IP fast reroute?", Comput. Netw., 54/8 pp. 1300–1315, 2010.

Revisit LFA

- Alternatives are too complex
  - extra-management burden, added complexity and non-trivial infrastructure upgrade: deployment barrier
- In contrast, LFA is unobtrusive and incrementally deployable
  - standardized and commercially available
  - Cisco IOS Release 3.7, JUNOS 9.6
  - remains the only IPFRR technique widely implemented
  - but it does not provide complete protection!
- Before deployment of LFA, some questions must be answered
  1. To what extent LFA can protect real networks?
  2. Which topologies are good for LFA, and which are bad?
  3. If LFA turns out inefficient in a particular case, how can we improve?

Link-protecting LFAs: some definitions

- p2p links, no LANs, no ECMP, no SRLGs, only link failures
- Some neighbor n of s is a **link-protecting LFA** for s to d if
  (i) n is not the default (shortest-path) next-hop of s to d
  (ii) $\mathrm{dist}(n, d) < \mathrm{dist}(n, s) + \mathrm{dist}(s, d)$



- **LFA coverage metric** $\eta(G)$: characterize network topologies based on their amenability to LFA

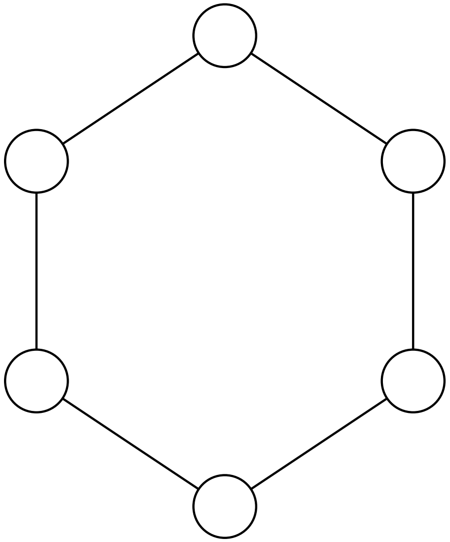
$$\eta(G) = \frac{\text{\#LFA protected } (s, d) \text{ pairs}}{\text{\#all } (s, d) \text{ pairs}}$$

Graph theoretical LFA coverage analysis

- **Theorem:** for any 2-connected graph G on n nodes

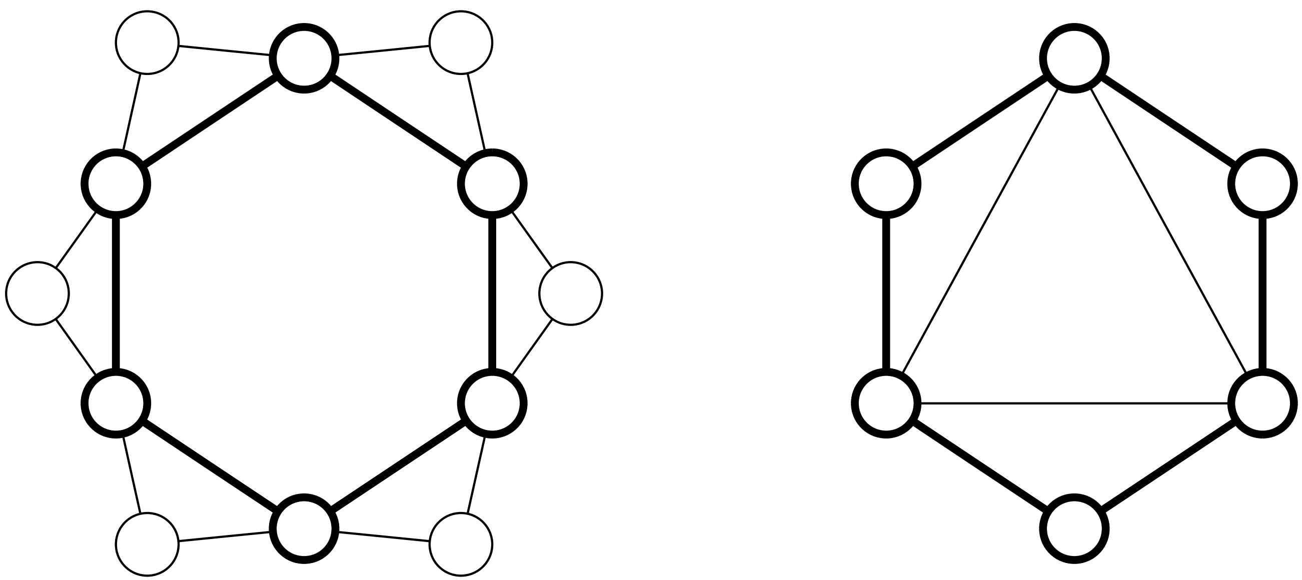
$$\frac{1}{n-1} \leq \eta(G) \leq 1$$

  - lower bound is tight for even rings/uniform costs
  - upper bound is tight for complete graphs/uniform costs
- The worst topologies for LFA are rings



Networks with full LFA protection

- Treat the uniform cost and the weighted case separately
- Generalize from the former to the latter
- **Theorem (uniform cost case):** $\eta(G) = 1$, if and only if each edge is contained in a triangle (cycle of length 3)



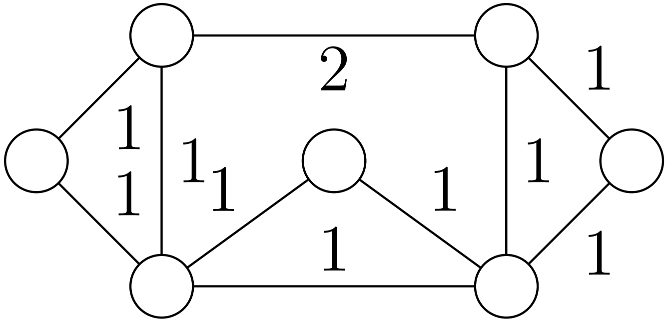
- Complete graphs, chordal graphs and maximal planar graphs have full LFA coverage

Networks with full LFA protection

- **Theorem (weighted case):** $\eta(G) = 1$, if each forwarding edge is in a triangle for which the triangle inequality holds

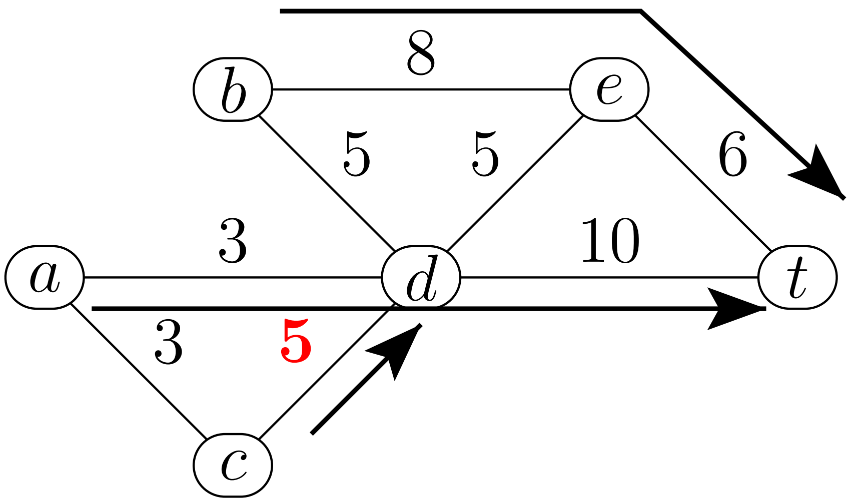
$$\mathrm{dist}(i,j) < \mathrm{dist}(i,k) + \mathrm{dist}(k,j)$$
$$\mathrm{dist}(i,k) < \mathrm{dist}(i,j) + \mathrm{dist}(j,k)$$
$$\mathrm{dist}(k,j) < \mathrm{dist}(k,i) + \mathrm{dist}(i,j)$$

- Only a sufficient condition but not necessary

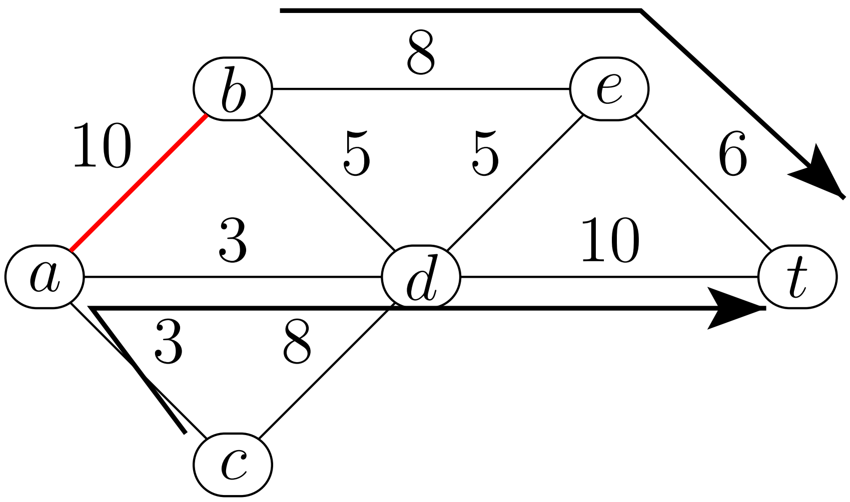


What if some nodes do not have LFA?

1.) Change link costs
- cheap but alters shortest paths
- might be too much of a price for improved LFA coverage



2.) Alter the topology by adding new links
- can be costly
- but leaves shortest paths intact
- at least, if new links are of sufficiently high cost



LFA coverage improvement

- Again, treat weighted and unweighted case separately
- LFA graph extension problem in the uniform cost case:

$$\min_{F \in \overline{E}} |F| : \eta(G(V, E \cup F)) = 1 \qquad (\text{minLFAu})$$

- We ask for the smallest complement edge set so that all edges are included in a triangle
- **Theorem:** *minLFAu* is NP-complete
- Gave an ILP and a greedy approximation
- The greedy approximation adds the link that improves the most
- **Theorem:** the greedy algorithm terminates with full LFA coverage

LFA coverage improvement: weighted case

- LFA graph extension problem, weighted case (*minLFAw*): do *minLFAu* without changing any shortest paths at all
- We must choose link costs appropriately as well
- **Theorem:** *minLFAw* is solvable, if and only if each node n has at least two upstream nodes in the shortest path tree rooted at n
- Gave a pre-processing algorithm
  - for each node violating the above requirements, adds at most one link and changes at most one cost
- **Theorem:** if solvable, *minLFAw* is NP-complete
- Again, gave an ILP and a greedy approximation
- In fact, the previous algorithm works here too with minimal modifications

Numerical results

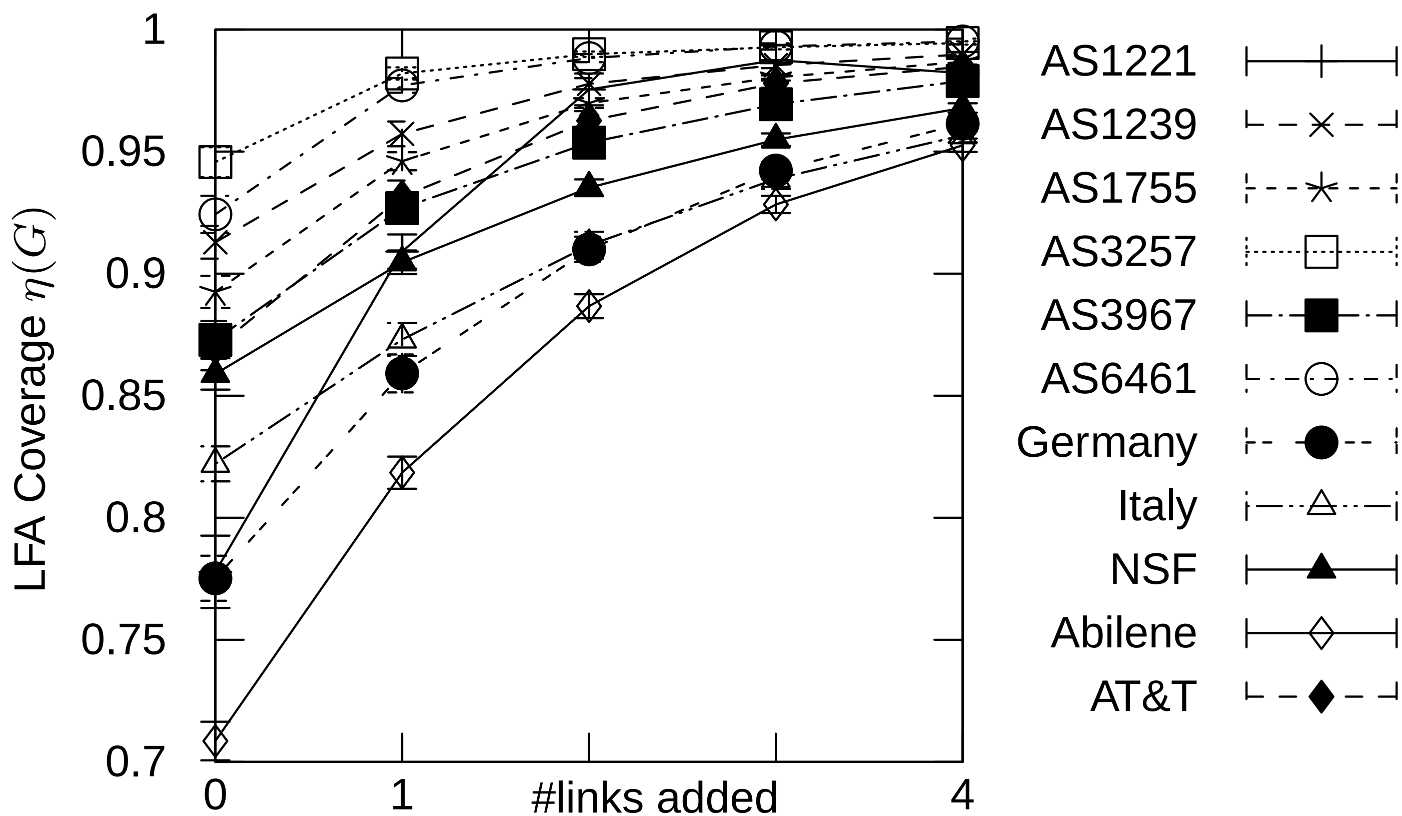
- Ran the ILP and the approximation on select ISP topologies

	Uniform cost			Weighted			
Topology	η_0	ILP	greedy	η_0	preproc.	ILP	greedy
AS1221	0.833	1	1	0.833	1/1	2	2
AS1239	0.898	6	6	0.877	0/0	6	7
AS1755	0.889	4	4	0.886	0/0	8	8
AS3257	0.946	2	3	0.903	7/7	10	11
AT&T	0.823	5	6	0.823	0/0	10	13
Germ_50	0.801	21	22	0.92	0/0	18	21

- Default coverage is usually 70-90%
- The greedy approximation is efficient
- In many cases, very few new links needed

Numerical results

- LFA coverage in the first 4 iterations of the greedy algorithm



- Only 2-4 new links is enough for >95% LFA coverage

Conclusions

- IPFRR is under wide-scale deployment
  - LFA is the only commercially implemented technique
  - simple, but no protection for all failure scenarios
- In this paper: theoretical and practical studies on how to actually deploy LFA
  - which networks are good/bad for deploying LFA
  - introduced the LFA graph extension problem
  - computationally hard, but efficiently approximable
  - just by adding a couple of links/changing a few link costs LFA coverage can be increased drastically
- We since submitted a paper on the „LFA cost optimization" version too