Router Virtualization for Improving IP-level Resilience

János Tapolcai, Gábor Rétvári

MTA-BME Future Internet Research Group
High Speed Networks Laboratory
Department of Telecommunications and Media Informatics
Budapest University of Technology and Economics
Email: {tapolcai, retvari}@tmit.bme.hu

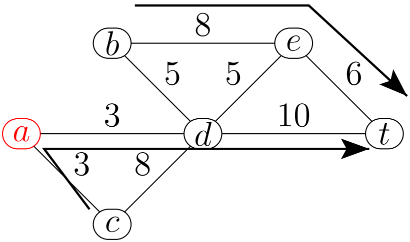
MŰEGYETEM 1782

Background

- Many operators provide commercial telecom services over a pure IP control plane
- Legacy IP failure recovery is slow (>150 ms)
- For <50 ms resilience fast IP-level protection is the way to go: **IP Fast ReRoute** (IPFRR)
- There is only one IPFRR scheme available in off the shelf routers: **Loop Free Alternates** (LFA)
- But with LFA certain failure cases are impossible to repair
- **Challenge:** tune the network for better LFA-based protection, without interfering with normal operation

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
- Enough to ensure that the alternate is not upstream
- So it will not loop the packet back



Why LFA?

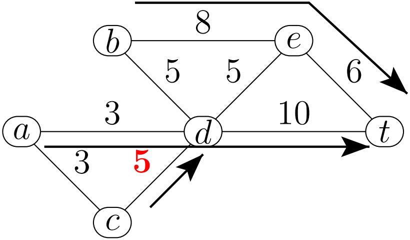
- **IPFRR is hard**, as destination-based forwarding does not play well with local rerouting
- LFA alternatives induce extra-management burden, added complexity and non-trivial infrastructure upgrade
- Standardization and deployment barrier
- LFA is unobtrusive and incrementally deployable
  - standardized and commercially available
  - Cisco IOS Release 3.7, JUNOS 9.6, HP 6600 series
  - the only IPFRR technique widely implemented
  - industrial requirement by *Seamless MPLS*
  - but it does not provide complete protection!
  - neither its successor “Remote LFA” does

What if some nodes do not have LFA?

Change the link costs

M. Menth, “Routing optimization with IP Fast Reroute,” Internet Draft, July 2010.

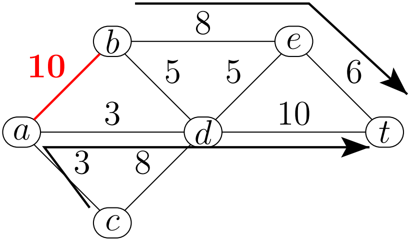
- simple and cheap
- but alters shortest paths
- often not allowed



Change the topology

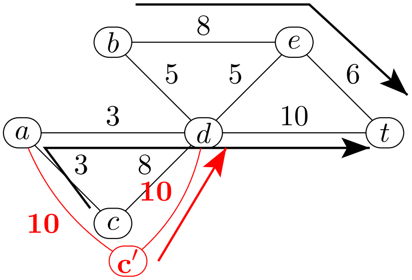
G. Rétvári et al., “IP Fast ReRoute: Loop Free Alternates revisited,” in *INFOCOM*, 2011.

- adding “joker” links
- shortest paths intact
- can be costly



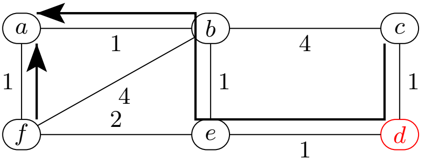
Main idea: Router virtualization

- Provision virtual routers as LFAs to unprotected routers
- Runs a separate IGP instance: eligible as LFA
- Protect the network **without touching the shortest paths and the physical topology in any ways**



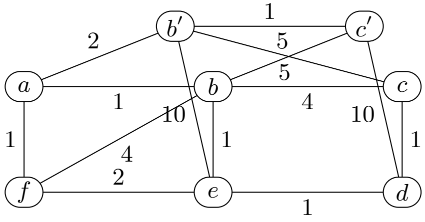
LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)



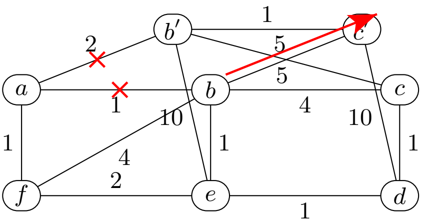
LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)



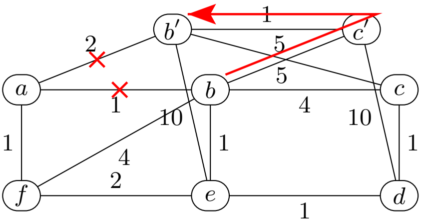
LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)



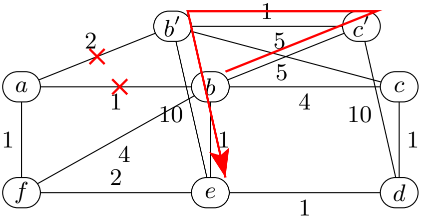
LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)



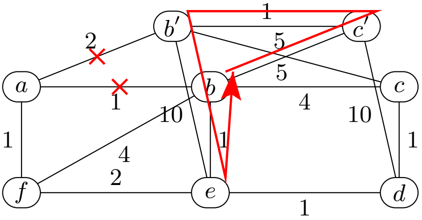
LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)



LFA Virtual Router Augmentation

- Design a **virtual overlay on top of the physical network to maximize LFA failure case coverage** against single failures
  - use the fewest possible virtual router instances
  - Layer 2 virtual links only
  - take care of Shared Risk Link Groups (SRLGs)

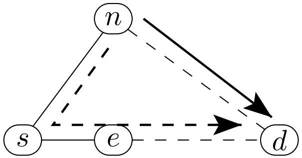


  - never decrease LFA coverage by virtualization!

LFA Virtual Router Augmentation

- A neighbor n of s is a **link-protecting LFA** for s to d if

LFA-1 n is different from the default $s \to d$ next-hop e

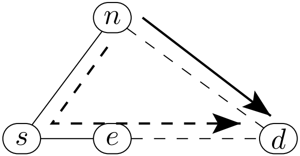


LFA Virtual Router Augmentation

- A neighbor n of s is a **link-protecting LFA** for s to d if

LFA-1 n is different from the default $s \to d$ next-hop e

LFA-2 $\mathrm{dist}(n, d) < \mathrm{dist}(n, s) + \mathrm{dist}(s, d)$



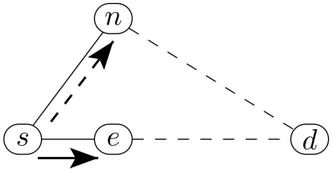
LFA Virtual Router Augmentation

- A neighbor n of s is a **link-protecting LFA** for s to d if

LFA-1 n is different from the default $s \to d$ next-hop e

LFA-2 $\mathrm{dist}(n, d) < \mathrm{dist}(n, s) + \mathrm{dist}(s, d)$

LFA-3 links (s, n) and (s, e) do not share an SRLG



LFA Virtual Router Augmentation

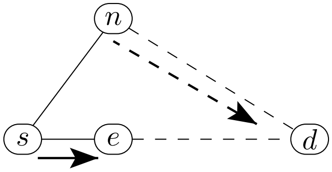
- A neighbor n of s is a **link-protecting LFA** for s to d if

LFA-1 n is different from the default $s \rightarrow d$ next-hop e

LFA-2 $\mathrm{dist}(n, d) < \mathrm{dist}(n, s) + \mathrm{dist}(s, d)$

LFA-3 links (s, n) and (s, e) do not share an SRLG

LFA-4 each $n \rightarrow d$ shortest path is SRLG-disjoint from (s, e)



LFA Virtual Router Augmentation

- A neighbor n of s is a **link-protecting LFA** for s to d if

LFA-1 n is different from the default $s \to d$ next-hop e

LFA-2 $\mathrm{dist}(n, d) < \mathrm{dist}(n, s) + \mathrm{dist}(s, d)$

LFA-3 links (s, n) and (s, e) do not share an SRLG

LFA-4 each $n \to d$ shortest path is SRLG-disjoint from (s, e)

- IGP support for SRLGs varies across implementations
  - either support **no-SRLGs** (only LFA-1 and LFA-2)
  - or only support **local-SRLGs** (LFA-1, LFA-2, LFA-3)
  - no implementation we know of has **general SRLGs**
- We support both the **no-SRLG** and **local-SRLG** models

Results: Complexity

- **LFAVirt:** a relaxed version, where the task is to add a single virtual router v' to a known node v with
  - ruling out fake LFAs and
  - maximizing LFA failure case coverage

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

- **Theorem: LFAVirt** is NP-complete under any SRLG model
- Transformation is from the minimum feedback arc set problem [GT8]

Results: Algorithms

- Greedy framework: in every iteration, add the virtual node v' to v that maximizes LFA coverage on $v \in V$
- An Integer Linear Program (ILP) to select the virtual router's next-hops, by pre-computing
  - node pairs sd that can gain an LFA from v'
  - **escape nodes** $\mathcal{E}_{sd}$ that can provide an $s \rightarrow d$ LFA
  - **trap nodes** $\mathcal{T}_{sd}$ that might create an LFA loop for some node-pair sd
- Choose a next-hop that is an escape node to the most node pairs, but never a trap node for others
- Only $O(\Delta^2 n)$ integer variables for both SRLG models

Results: Numerical evaluations

- 21 ISP topologies, inferred or real IGP link costs

Name	$\eta_E(0)$	$\eta_E(\frac{1}{3})$	$\eta_E(\frac{2}{3})$	$\eta_E(1)$	$\eta_E(2)$	time [s]
Germany	0.694	0.886	0.944	0.981	1.000	0.025
AS1755	0.872	0.983	1.000	1.000	1.000	0.027
AS3967	0.785	0.983	1.000	1.000	1.000	0.052
BellSouth	0.797	0.997	1.000	1.000	1.000	0.043
Italy	0.784	0.923	0.969	0.982	0.985	0.170
Deltacom	0.632	0.906	0.951	0.954	0.954	1.159

- Adding a virtual router to only 33% of the nodes boosts LFA coverage beyond 90%
- Almost 100% protection with 2 virtual routers per node
- Improvement is 10–30% for link failures, and 40–50% for node failures (not shown here)
- The ILP can be solved fast

Conclusions

- Huge industrial demand for IPFRR (heavy IETF activity)
- IPFRR schemes providing 100% protection are still years from standardization and deplyment
- LFA is simple, widely supported, and well-tested
- LFA network optimization: tune the topology for LFA
  - can be done without modifying the physical topology in any ways
  - LFA virtual router augmentation
  - theoretically difficult, but well-approximable
  - numerical evaluations indicate huge potential
- **This can be deployed in your network right now!**