Disruption-free network reconfiguration *

Laurent Vanbever — ICTEAM, Pôle d'ingénierie informatique, IP Networking Lab

Reconfiguring a network (digraph)
changes the way traffic is forwarded

Reconfiguring a network
can be beneficial in terms of

- manageability
- stability
- security
- services ($$)

Reconfiguring a network is often avoided as
it is operationally complex and disruptive

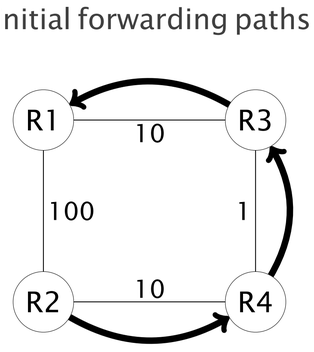
Reconfigure the network when it is *running*
since networks carry traffic 24/7

Reconfigure each node independently
node-by-node, in a coordinated manner

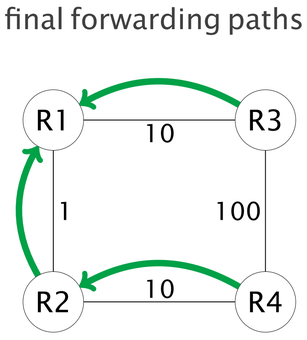
Face potential (services affecting) traffic losses
as non-reconfigured and reconfigured nodes interact

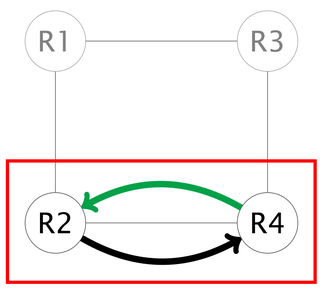
Problem: Transform an initial digraph into a final one, node-by-node, without creating any loop

initial forwarding paths



final forwarding paths

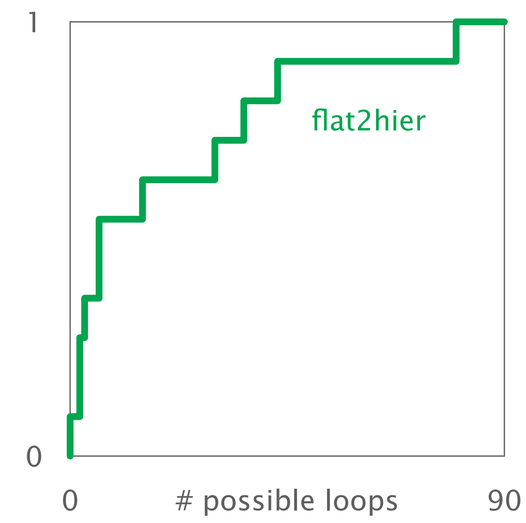




forwarding loop

The loop appears iff R2
is reconfigured before R4.

Tested networks (cumul. frequency)

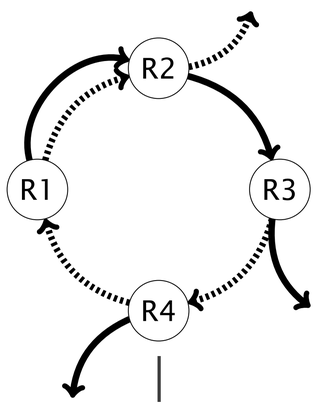


Up to 90 loops can arise during the reconfiguration.
Each loop can lead to *significative* losses of traffic.

Although finding a reconfiguration ordering is computationally hard (NP-complete),
finding one is doable *in practice*, even in large networks (150+ nodes)

→ initial path
⇢ final path

The Enumeration Algorithm [correct & complete]



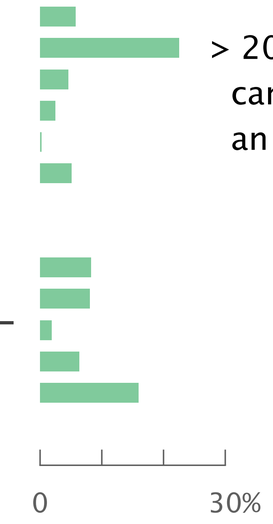
1. Merge the initial and the final forwarding paths
2. For each migration loop in the merged graph,
   Output ordering constraints such that
   at least one router in the initial state
   is migrated before at least one in the final
3. Solve the system by using Linear Programming

Migrate R2 before R3 or R4 avoids the loop

Tested networks

> 20% of the nodes can be involved in an ordering

An ordering was found in all the tested networks



Routers involved in ordering

(*) Laurent Vanbever, Stefano Vissicchio, Cristel Pelsser, Pierre Francois and Olivier Bonaventure.
Seamless Network-Wide IGP Migrations. In *Proceedings of the 2011 ACM SIGCOMM Conference*, Toronto, Canada, Aug. 2011. ACM.