IP Fast Reroute Applicability

Pierre Francois
Institute IMDEA Networks
Pierre.Francois@imdea.org

Agenda

- IGP (Fast) Convergence
- IGP Fast Reroute
- (Hitless maintenance operations)

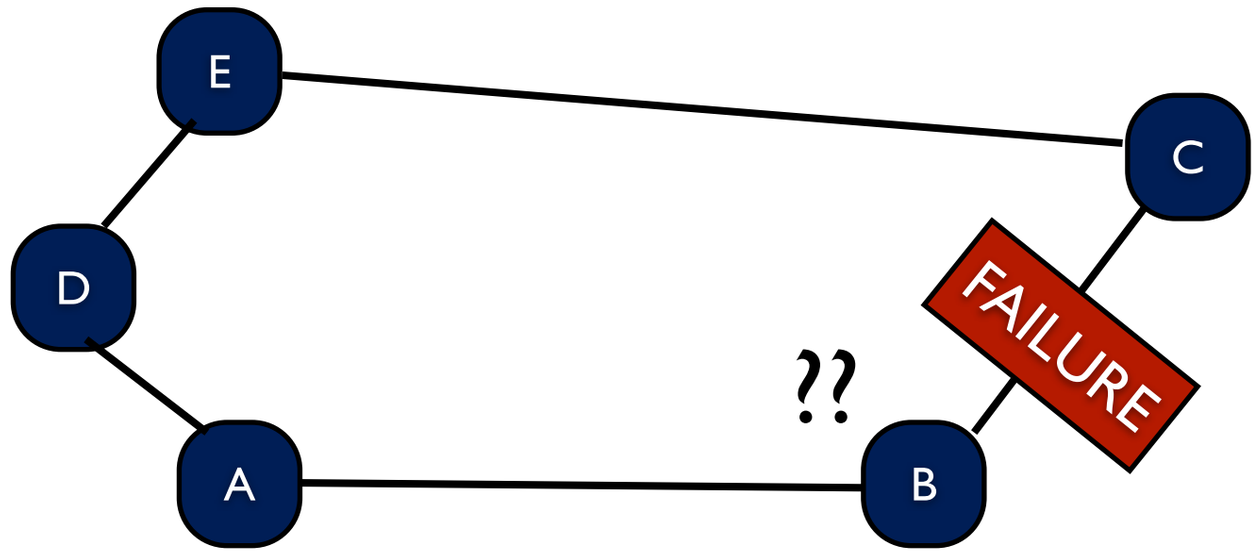
IGP Fast convergence

- *"Pushing the IGP to the limits"*
  - Implementation
  - Configuration
- Check with your vendor if your requirements for convergence time are really pushy
  - Depends on router performance
  - Depends on network topology

Components of the IGP convergence

- Failure detection
- Link-State Packet generation / propagation
- LSDB update, SPF run, RIB update
- FIB updates

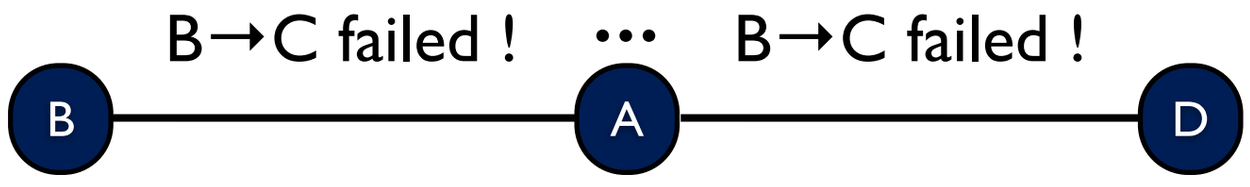
Failure detection



BFD, L2 alarms,..

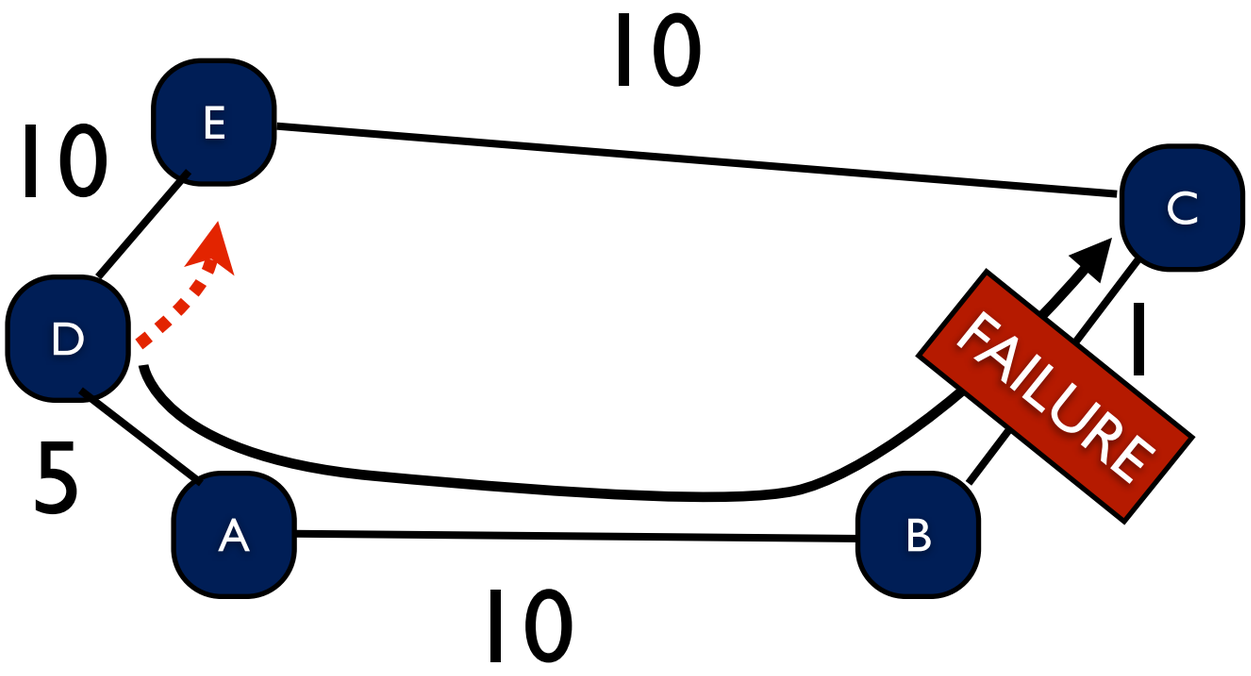
Fast detection, no more control plane stress

Link-State generation/propagation



- Generation
  - No more fixed “wait” time 5s, 5s, 5s, ...
  - Exponential back off 0ms, 500ms, 1s
- Propagation
  - “Fast-flood” vs. fixed Pacing timer
- No more artificial delaying under normal operation

Link-State update propagation



- Distance to the **rerouting** nodes
  - msec before updating routers know about the failure
  - Topology dependent
- Repair vs. Re-optimization

Paths recomputation time

- iSPF
- Full SPF takes a few msec now anyway

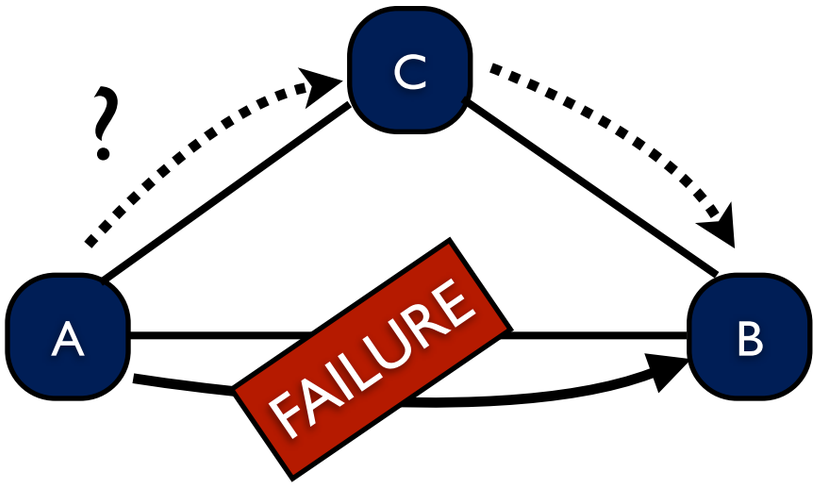
FIB Updates

- Bottleneck component of the convergence time
- x prefixes at y µsec / prefix...
- Prioritized prefixes

IGP Fast Convergence

- convergence time much below 1s now...
- but convergence time scaling factors exist
  - number of prefixes in the IGP
- **recovery** mechanisms can help

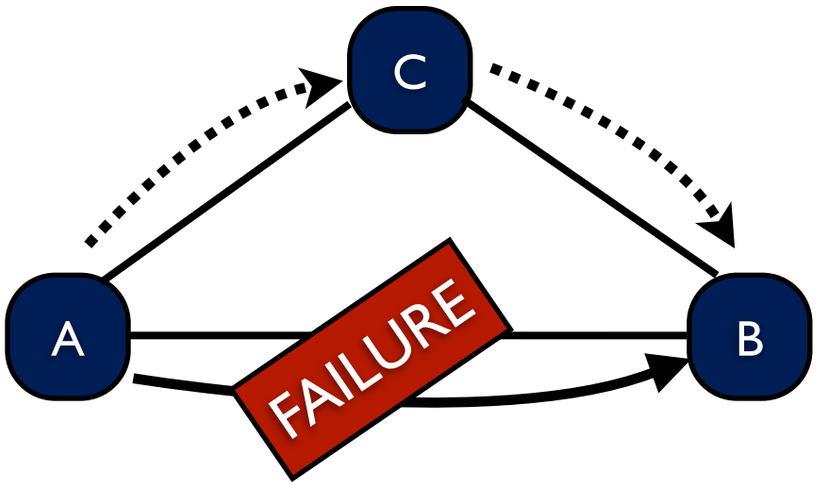
Loop-free Alternates



C is a loop free alternate
of A, for the failure of A→B,
if C would not forward the traffic
sent along A→B back to A,
when A deviates it to C

FIB design to allow direct deviation of traffic
when A→B is flagged down

LFA FIB support
per-link

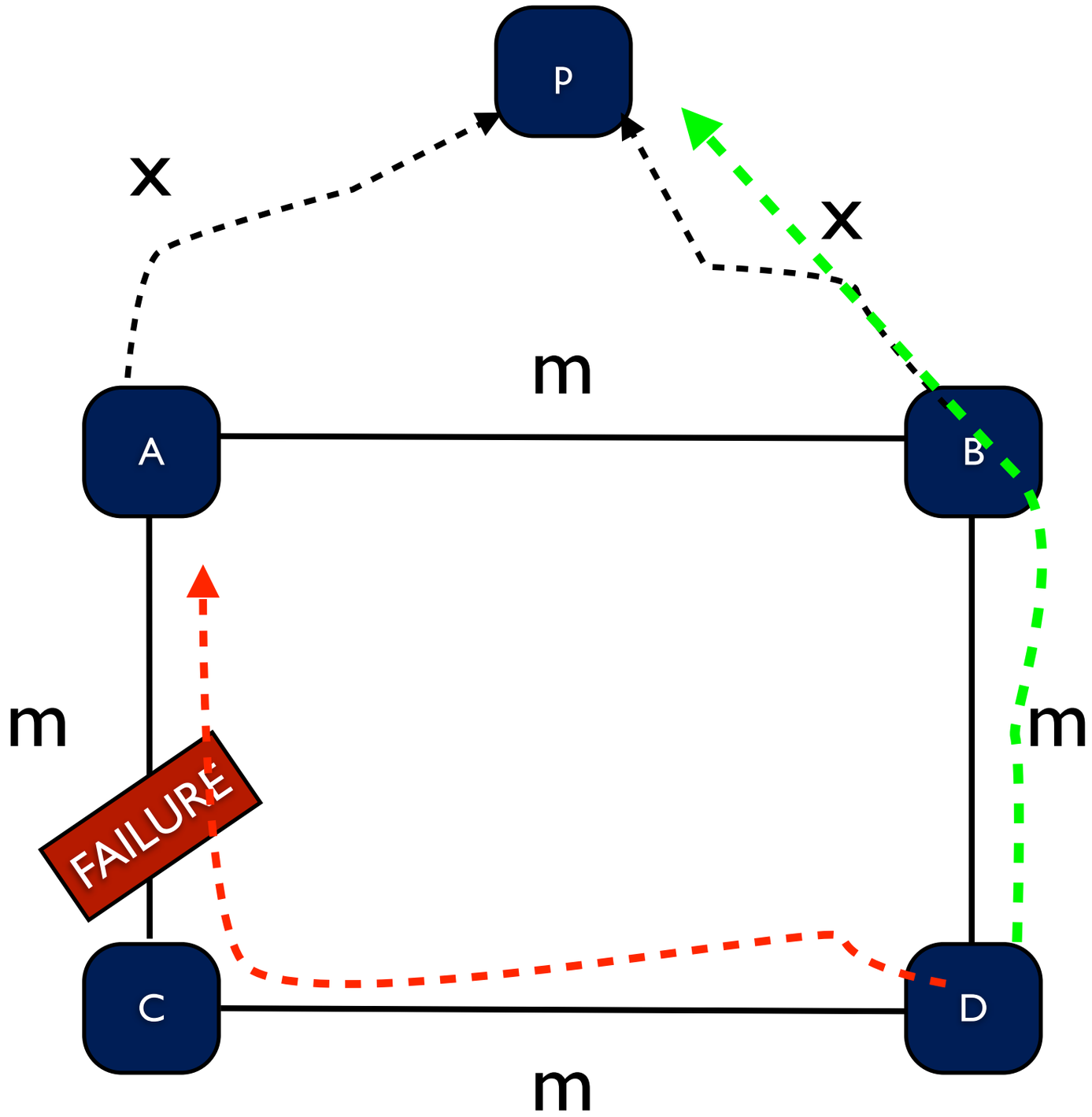


prefix	oif
p/P	--B

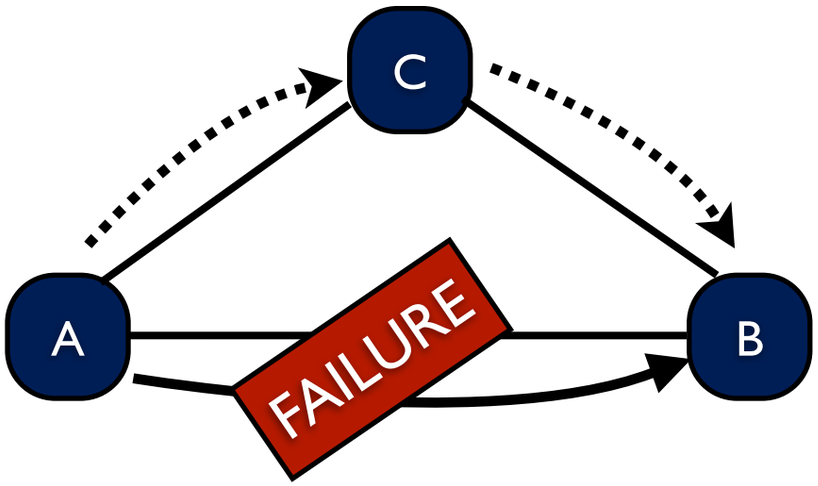
FIXED recovery time

--B	**DOWN**	**--C**
--C	UP	--B

per-link is constraining



LFA FIB support
per-prefix



prefix	oif	back. oif
p/P	--B	--C

FIXED recovery time

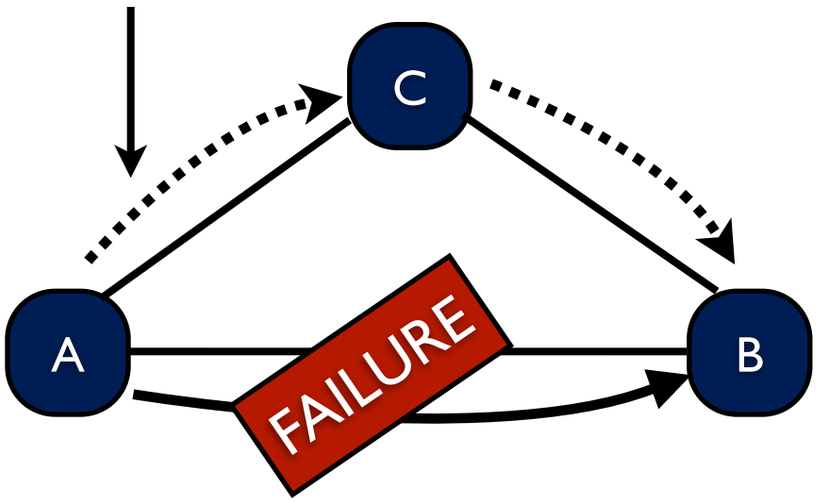
--B	**DOWN**
--C	UP

per-prefix LFA

- My favorite
  - Increases coverage
  - Allows for LB'ing LFA'ed traffic
  - Allows to pick the optimal LFA for each prefix
  - Favors post-convergence paths
- More complex
  - Not really to compute, but to manage in the FIB
  - LFAs for LDP-established LSP make it necessary

LFA FIB support
per-link

Label for p/P
over --C ?

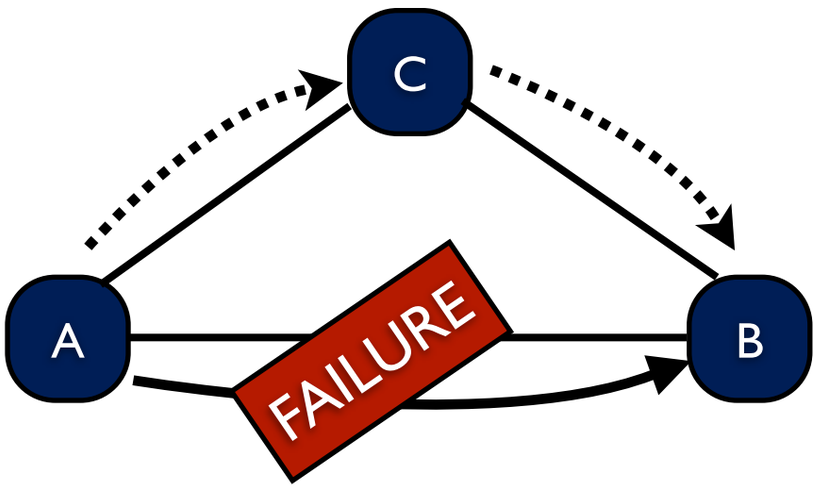


prefix	oif
p/P	--B:**L I**

FIXED recovery time

--B	**DOWN**	**--C**
--C	UP	--B

LFA FIB support
per-link



prefix	oif	back. oif
p/P	--B:**L1**	--C:**L2**
z/Z	--B:**L2**	--C:**L3**

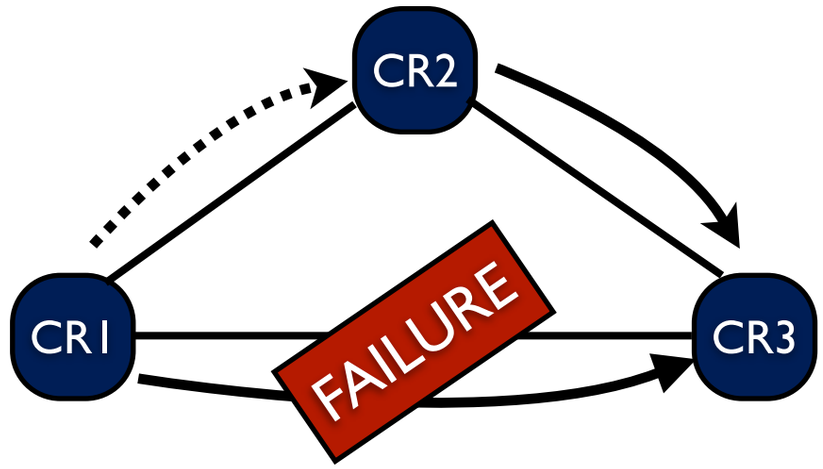
--B	**DOWN**
--C	UP

FIXED recovery time

C must have LDP downstream unsolicited turned on
or FEC-label must be requested to LFAs

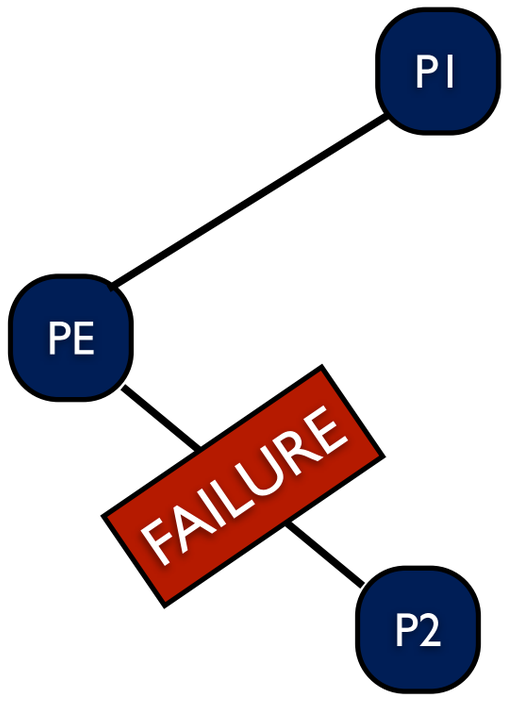
LFAs : where does it apply ?

meshed part of cores



LFAs : where does it apply

PE→P links

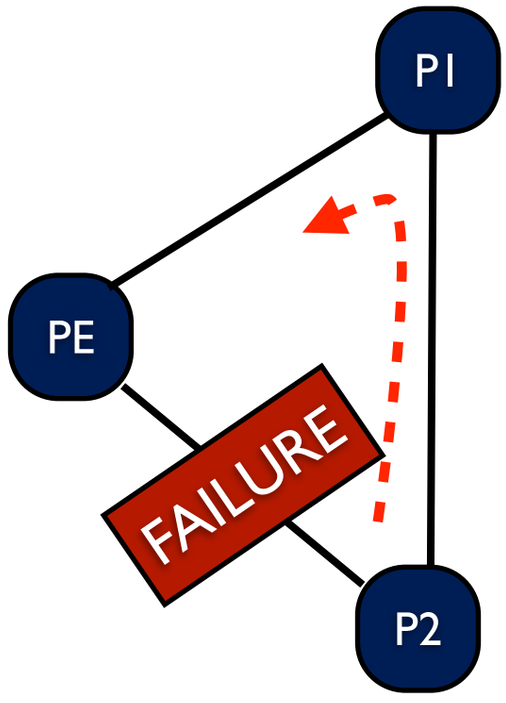


All the links of a non transit node can protect each other

These links track many prefixes

LFAs : where does it apply

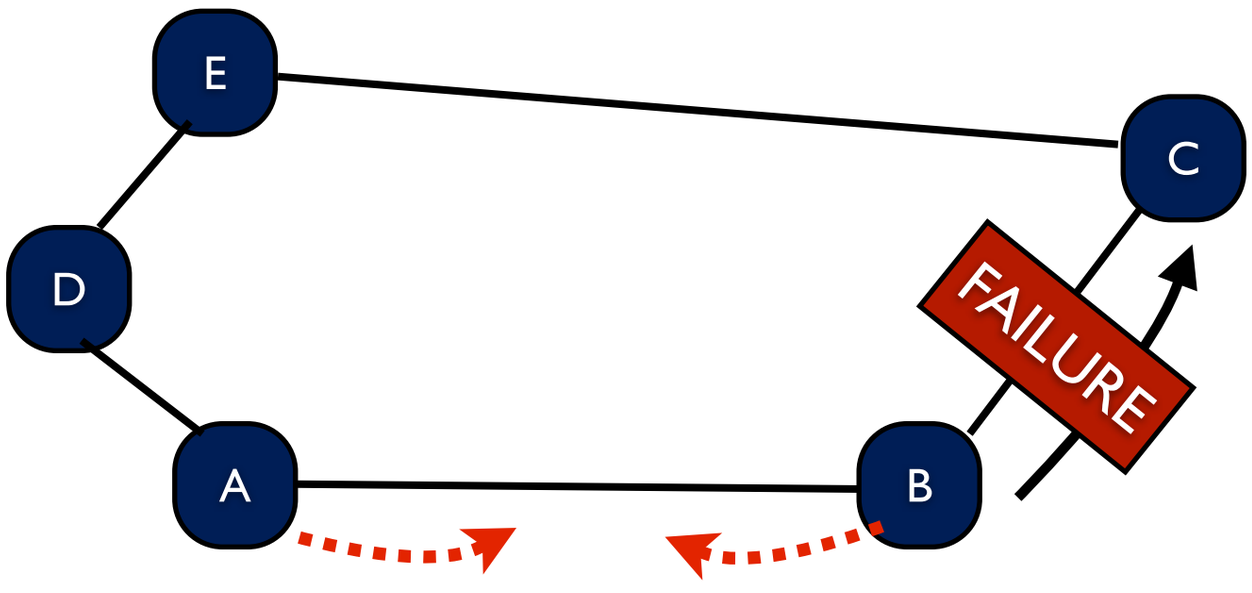
P→PE links



Less prefix tracked on these links

LFAs : where it does not apply

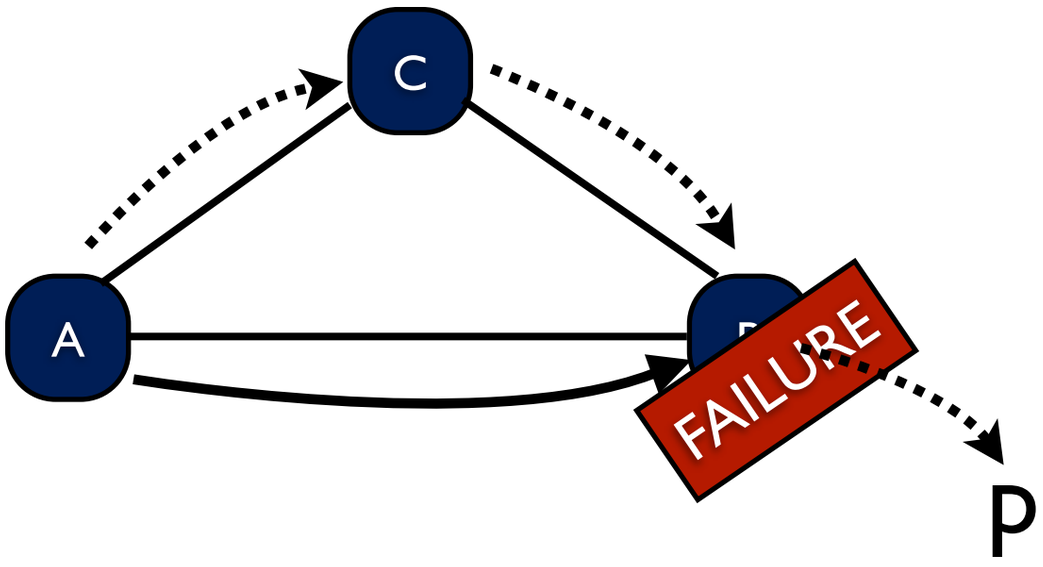
"Ring-ish" parts of topologies

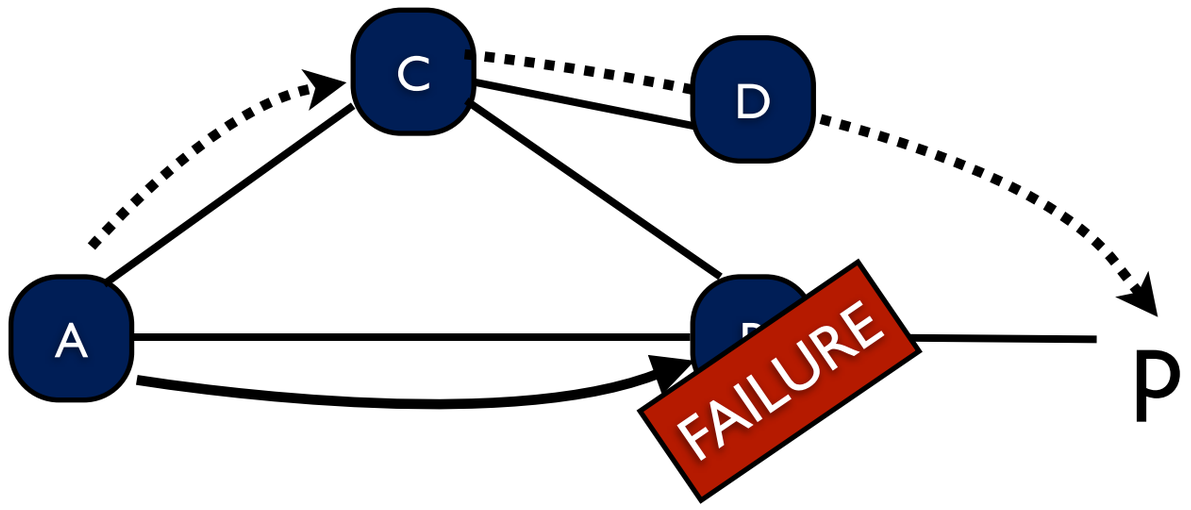


Node protecting LFAs

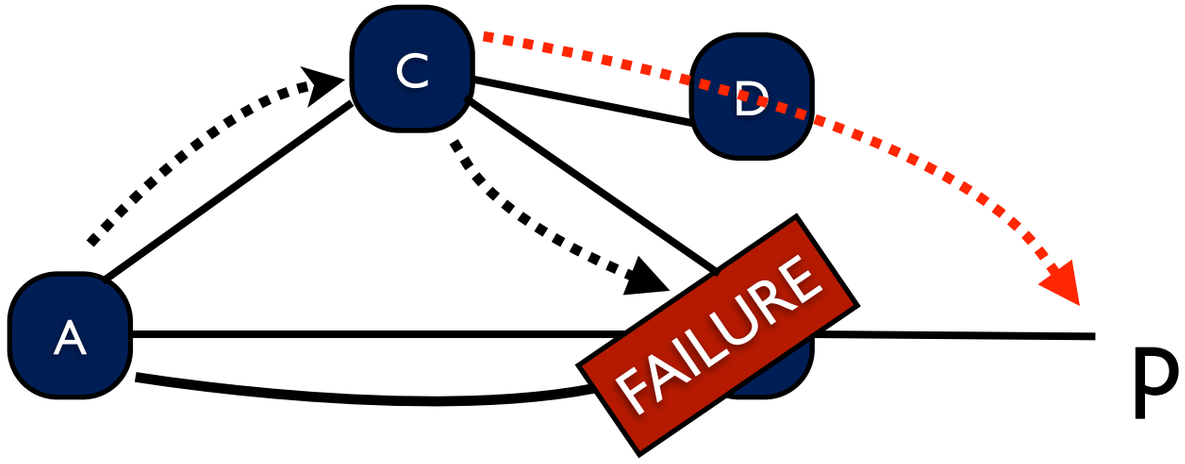
- An LFA which survives the failure of the entire neighboring node

Node protecting LFAs





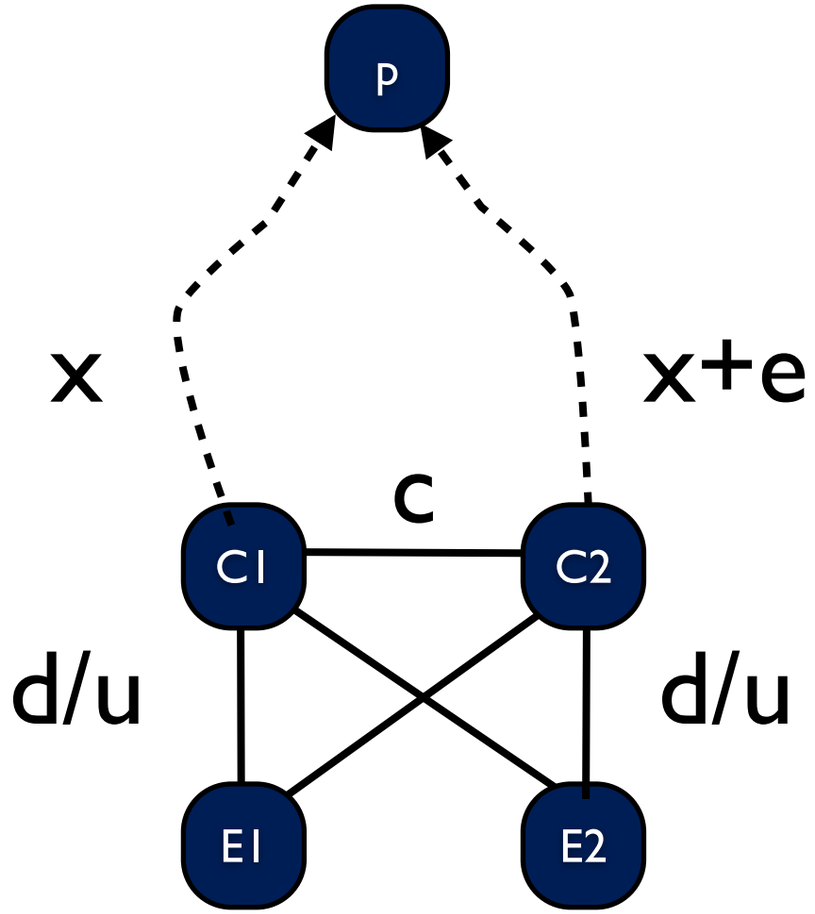
De facto node protecting LFA



LFA applicability

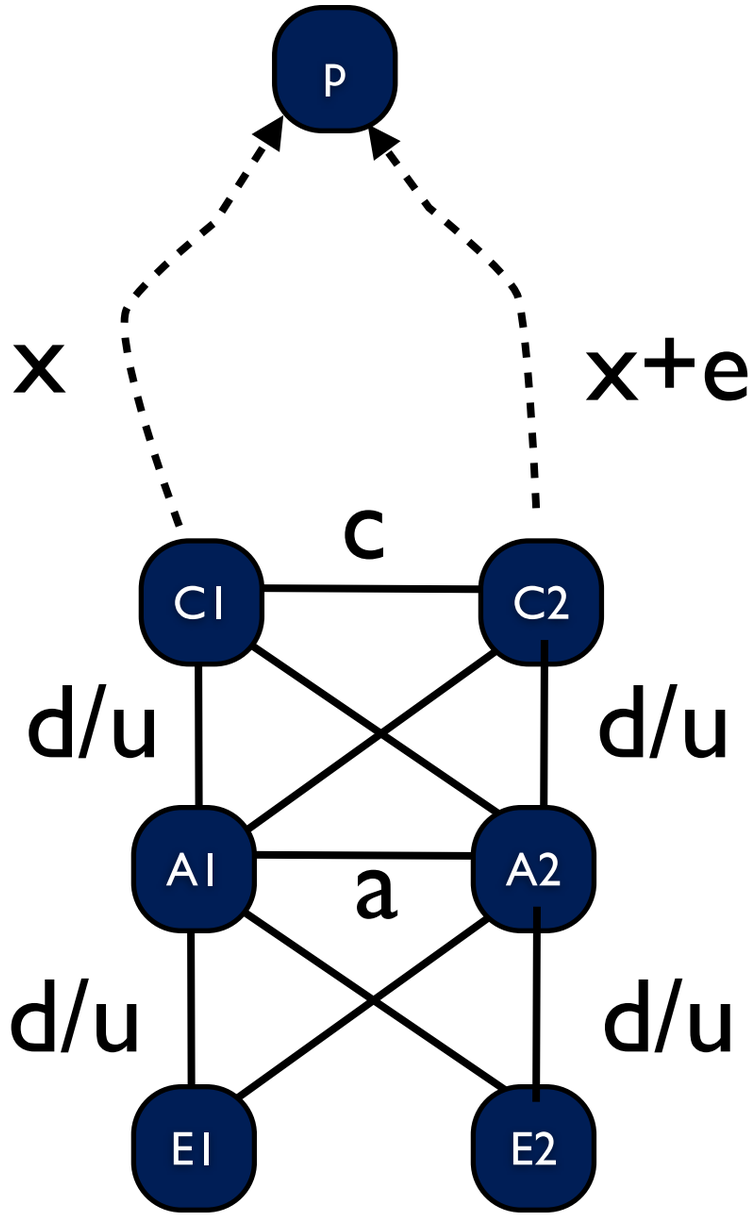
- draft-ietf-rtgwg-lfa-applicability-03
  cisco, IMDEA, FT, ATT, DT
  Under IESG review
- Focus on LFA applicability for PoP Designs

Triangle



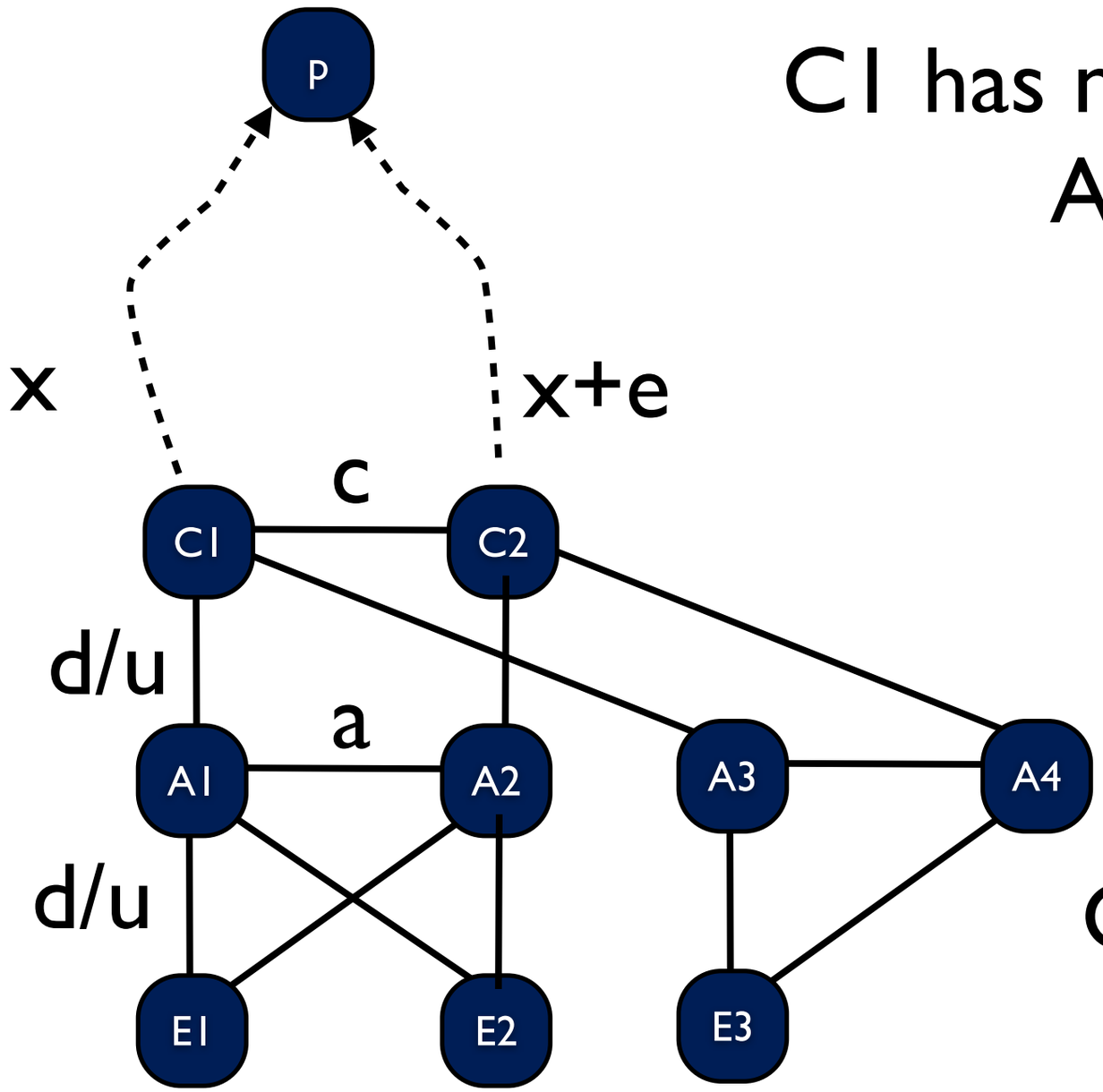
Fully protected against link and node failures (intra and inter)

Full-mesh



Fully protected against link and node failures (intra and inter)

Square



C1 has no LFA for destination A1 when c <= a

Not a big deal if per-prefix LFAs are used

Convergence happens on a small number of prefixes

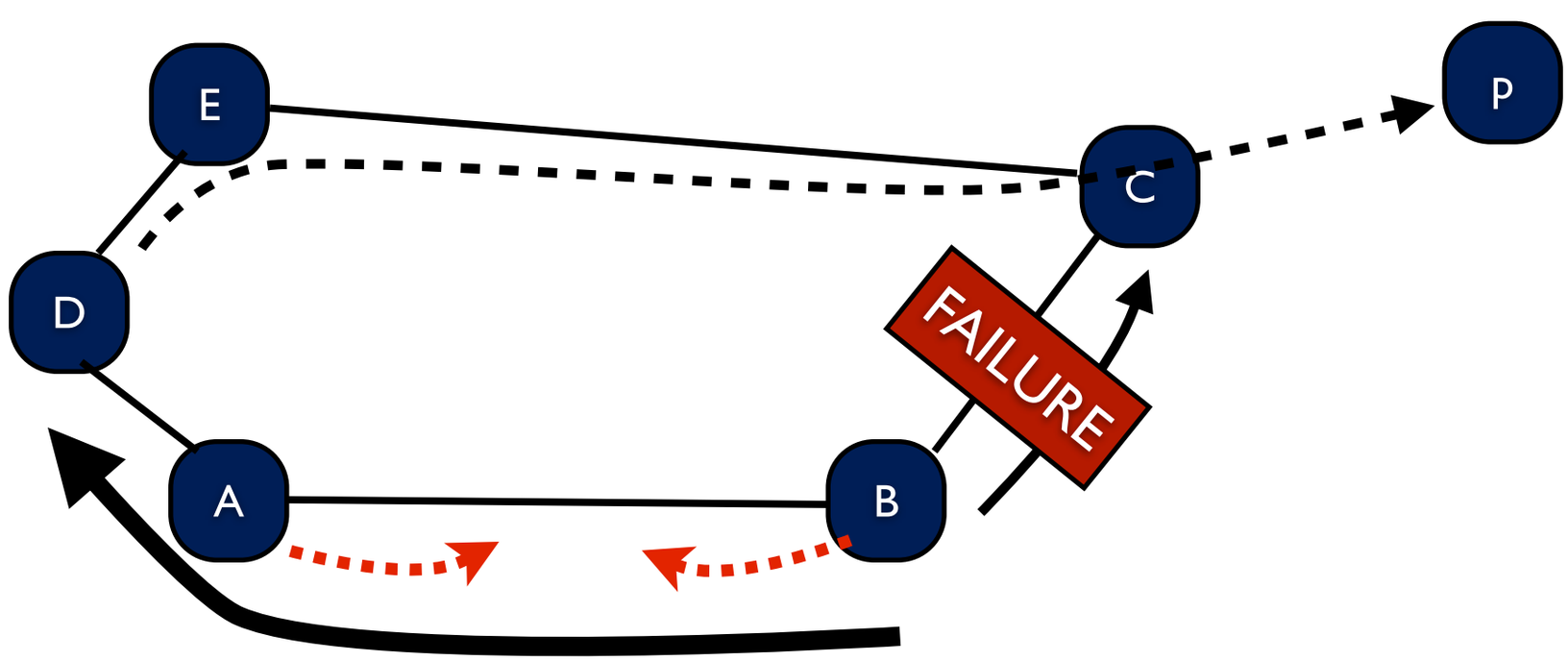
Summary

- Most typical PoP designs can be covered by LFAs
- Node protection is usually obtained, sometimes de facto
- Square has one link not fully covered

How to reach 100% coverage ?

- Do you need to ?
- Using more complex techniques
  - U-Turns, NotVia...
  - Tunnels
  - MPLS-FRR

Tunnels



Tunnels

- In theory, you can get crazy end-to-end paths
- Within a PoP, less of a concern

LFAs in the core ?

- Less obvious
- Coverage is highly topology dependent
- Topology design driven by many other factors
- Should not be a primary design objective

Profile I

- Designer relies on fast IGP convergence
- LFAs come as a bonus when applicable (50msec vs 500msec)

Profile 2

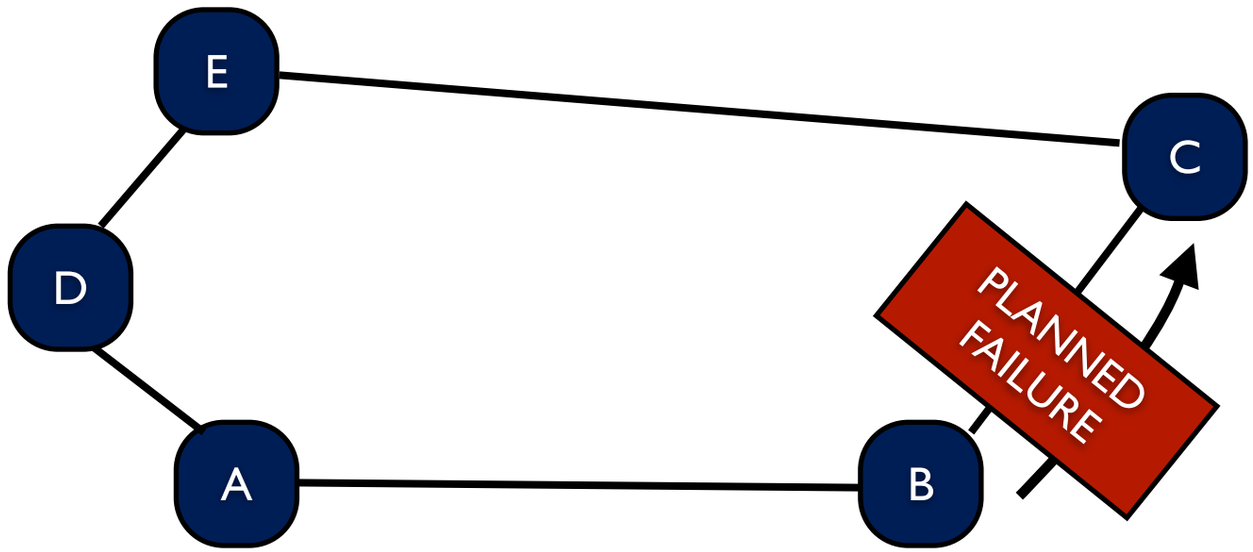
- Designer seeks for high FRR coverage and cannot (re-)engineer his backbone
- Forget about LFAs
- MPLS TE FRR

Profile 3

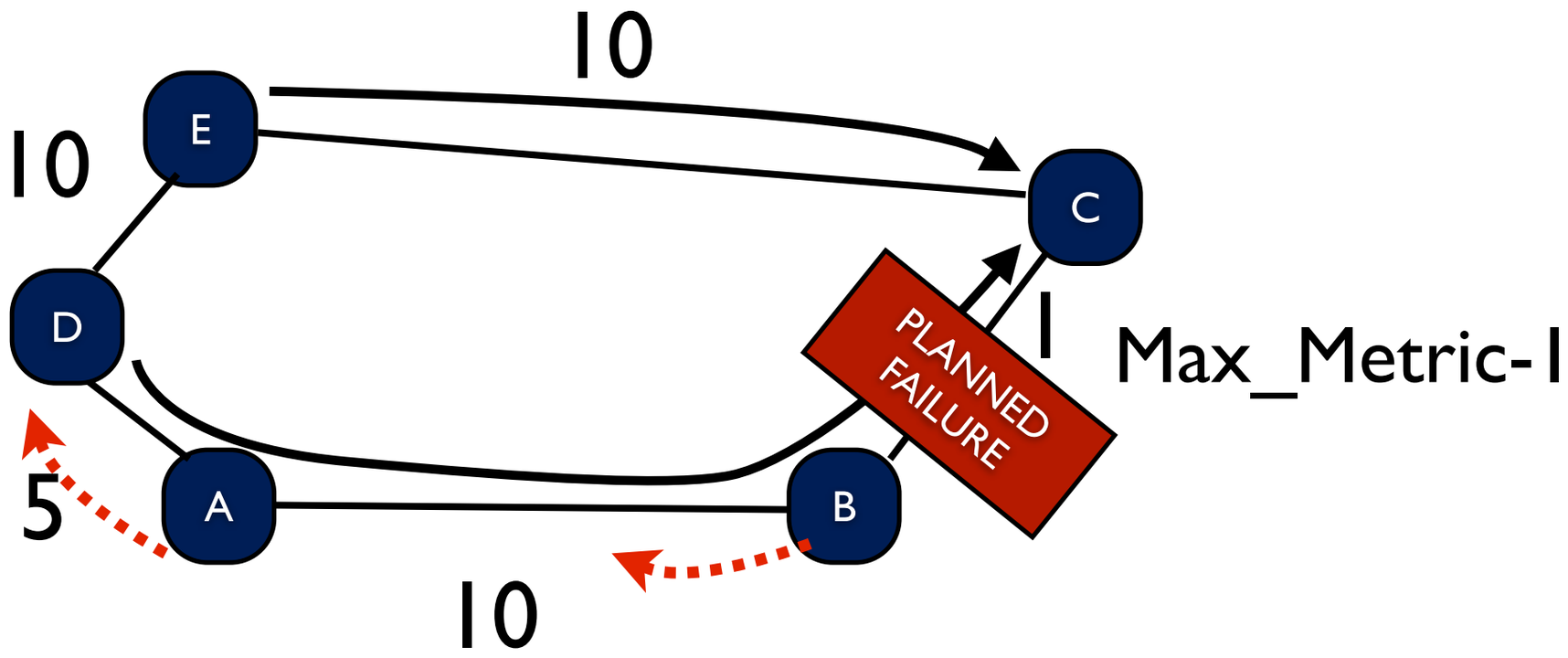
- Designer looks for high LFA coverage and can (re-)engineer the topology
  - e.g. for some specific demands in some parts of the networks
  - Tools to perform capacity planning accounting for LFAs

Tool

IGP Hitless maintenance operations

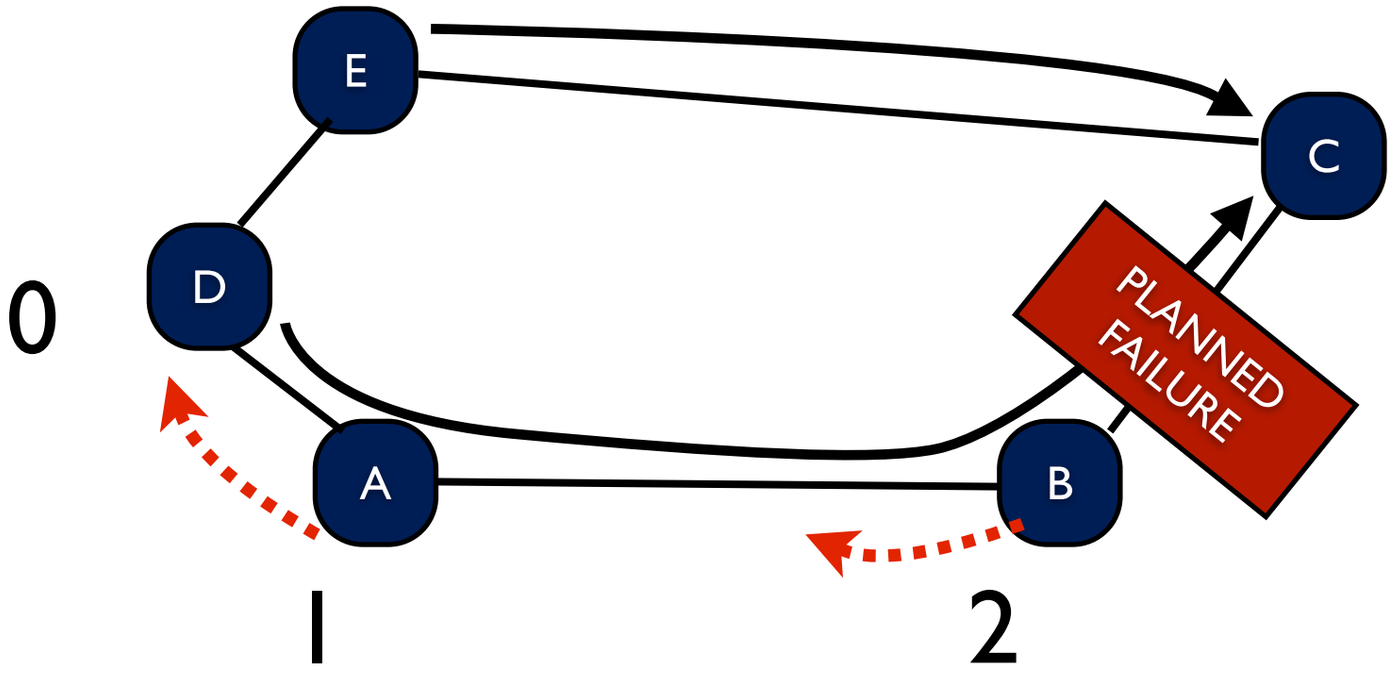


IGP Hitless maintenance operations



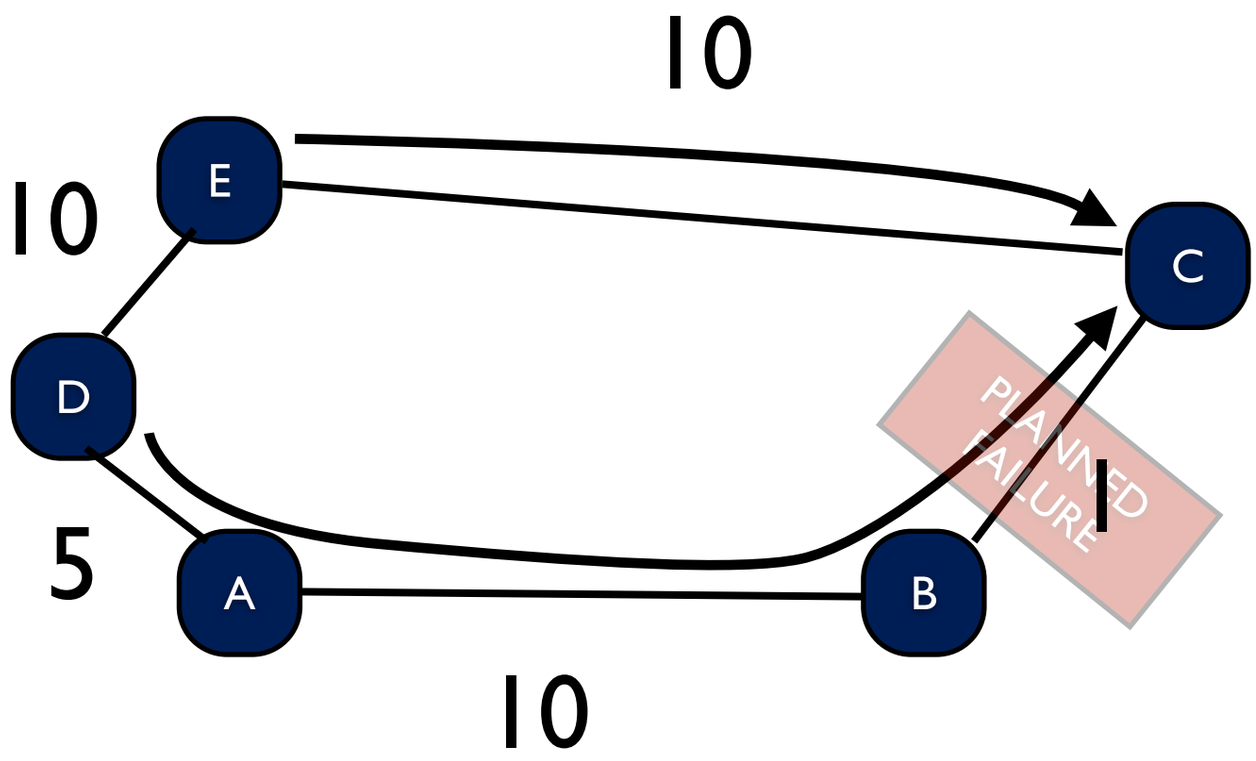
ordered FIB updates

Modifications to OSPF / IS-IS to enforce
a loop-free ordering of the FIB updates
upon planned maintenance

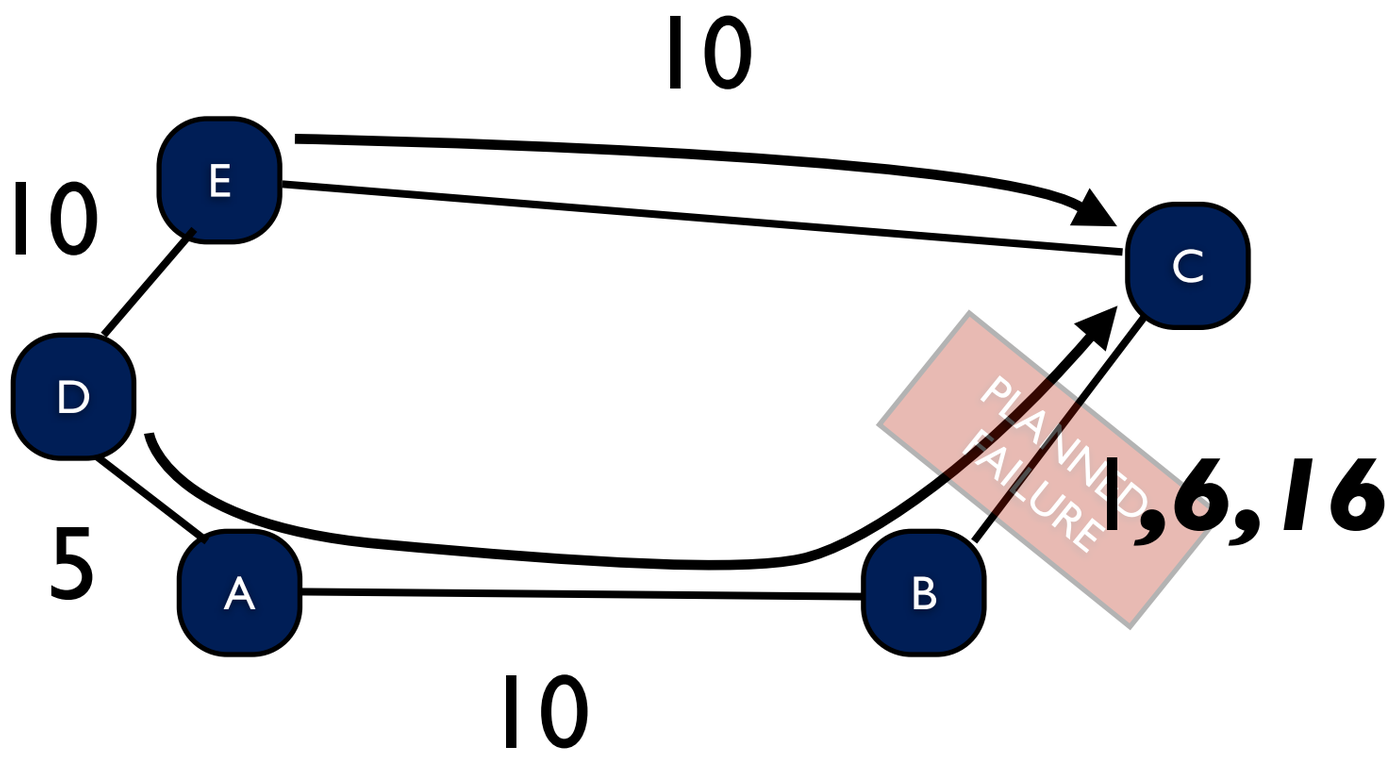


Metric-increments

Reconfiguration of OSPF / IS-IS to enforce a loop-free transition to post-convergence state



Metric-increments



Usually, one intermediate metric is sufficient

Conclusion

- Sub-second convergence is conservative now
- Local FRR mechanisms make it close to Failure Detection time
  - Easy, simple in the PoP
  - Use with care in the Core, consider as bonus
- (Graceful shutdown is achievable)