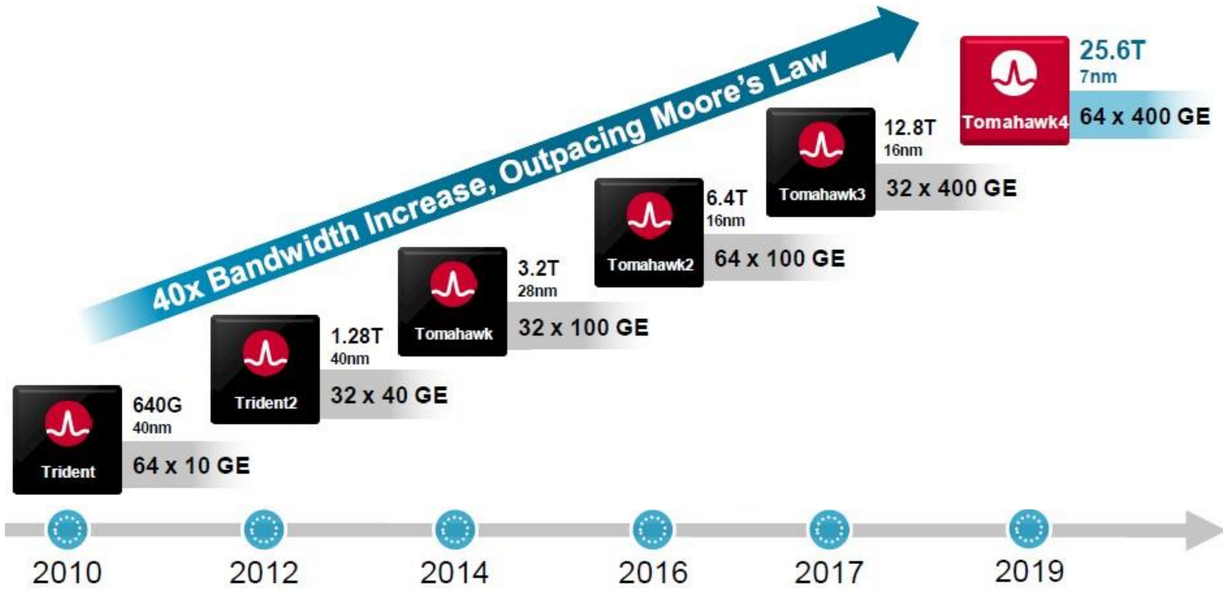
Hashing Linearity Enables Relative Path Control in Data Centers

Zhehui Zhang (UCLA)
Haiyang Zheng, Jiayao Hu, Xiangning Yu,
Chenchen Qi, Xuemei Shi, Guohui Wang (Alibaba Group)

Requirements for Data Centers

High bandwidth

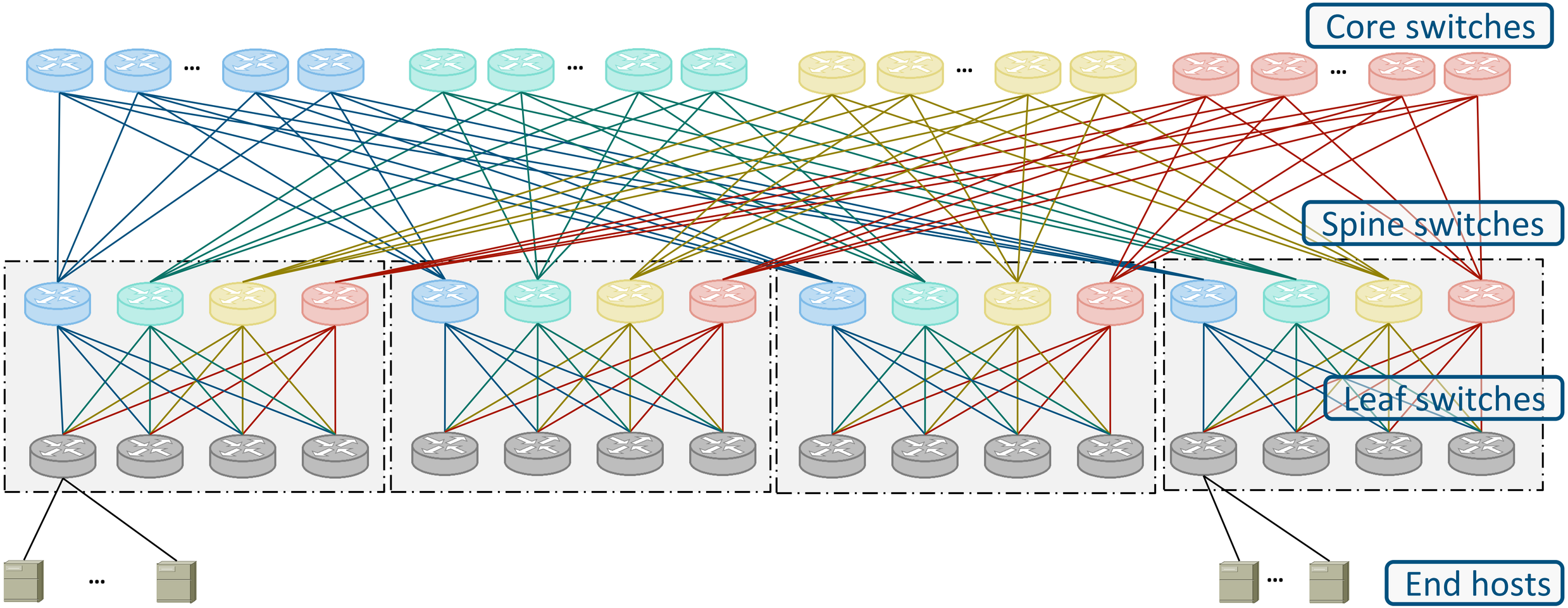


src: broadcom

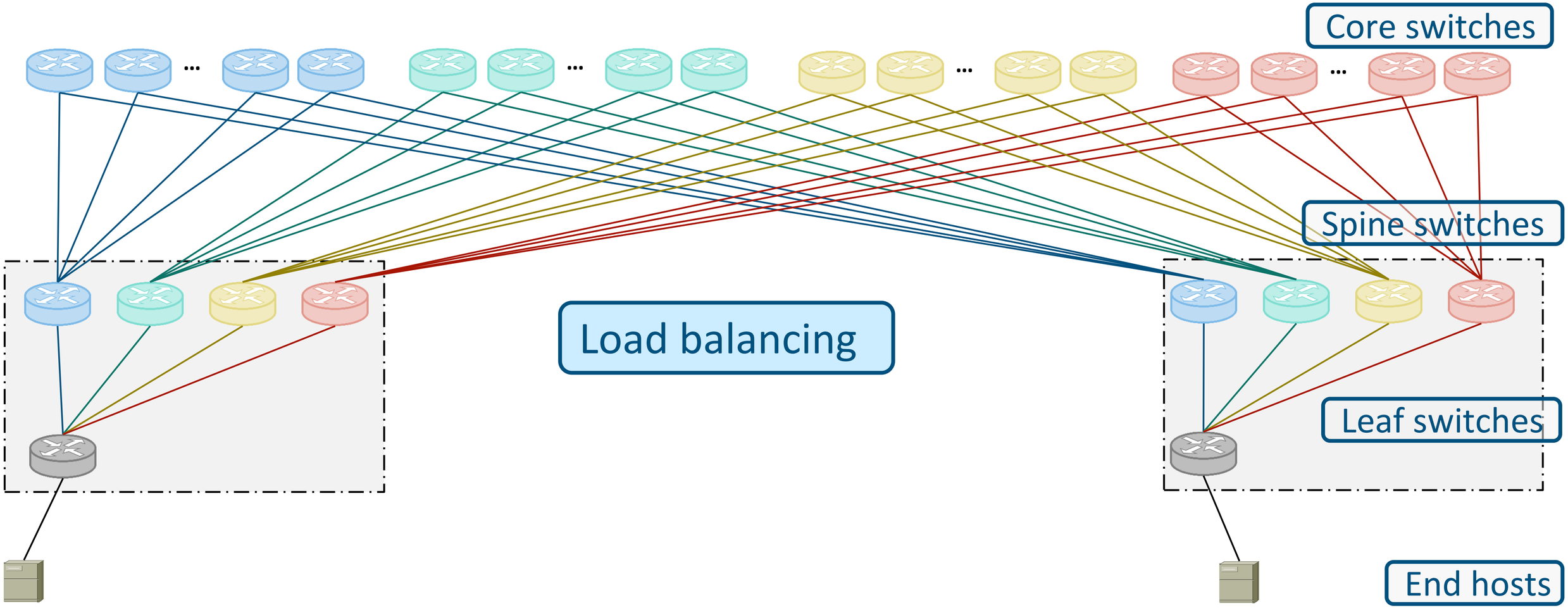
‘Five Nines’ availability

src: alibabacloud.com

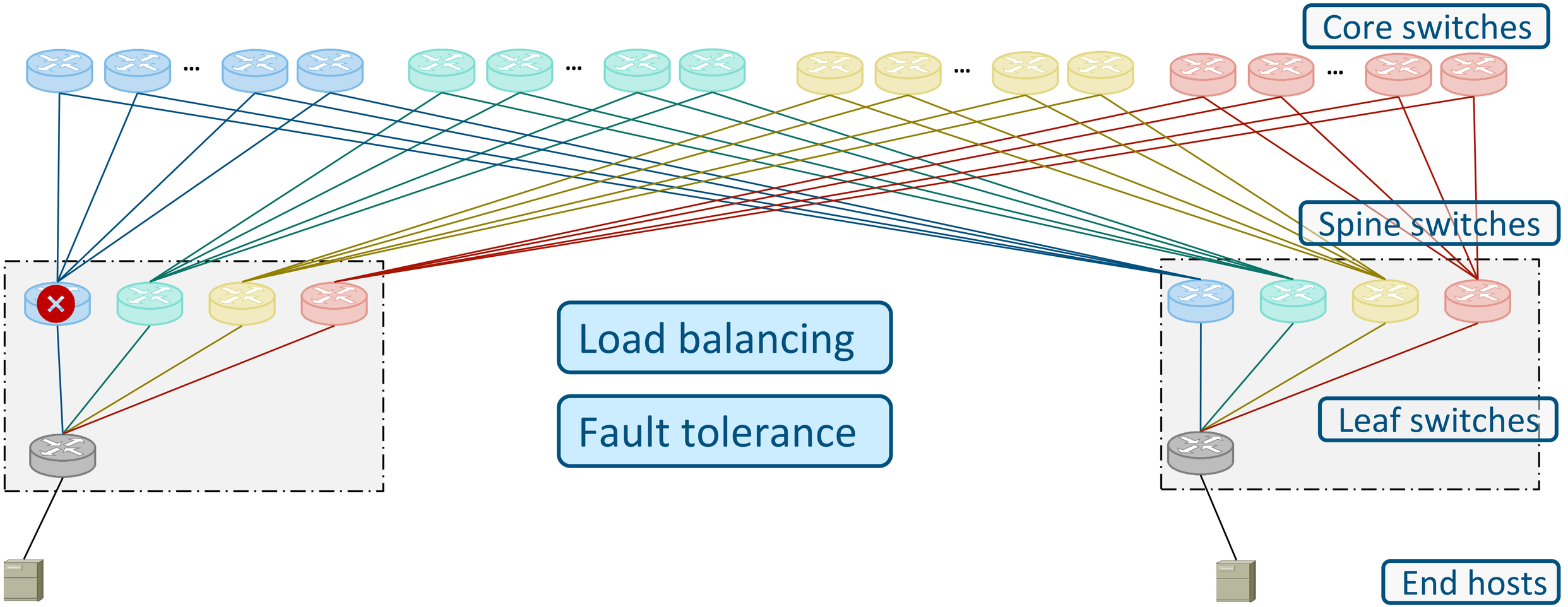
Rich path diversity in data centers



Rich path diversity in data centers



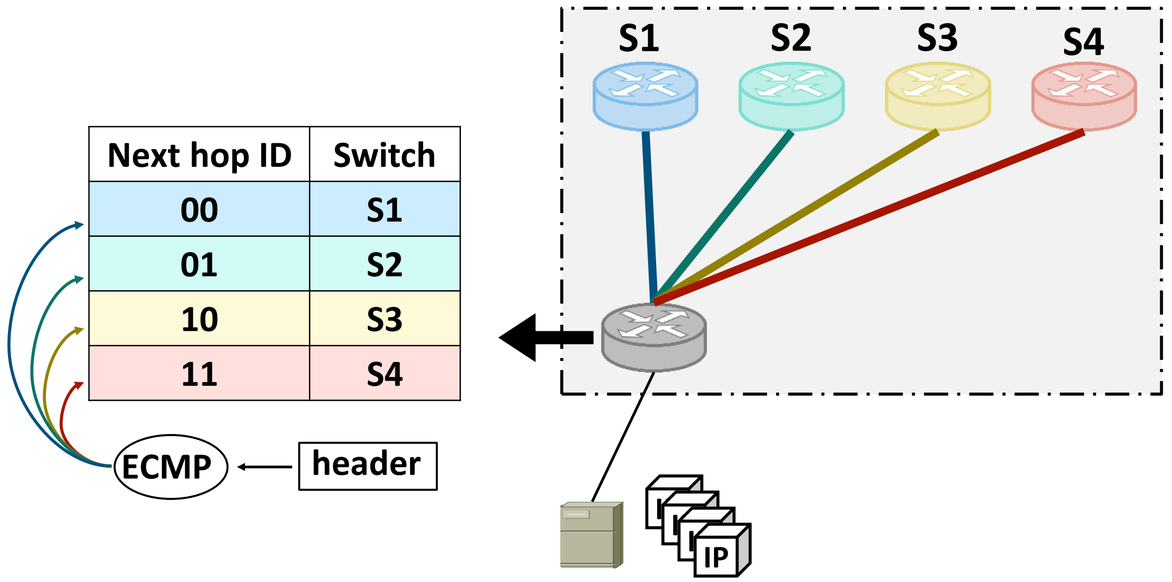
Rich path diversity in data centers



Problems in leveraging path diversity

Common practice: ECMP

- Load balancing: select the next hop based on hashing

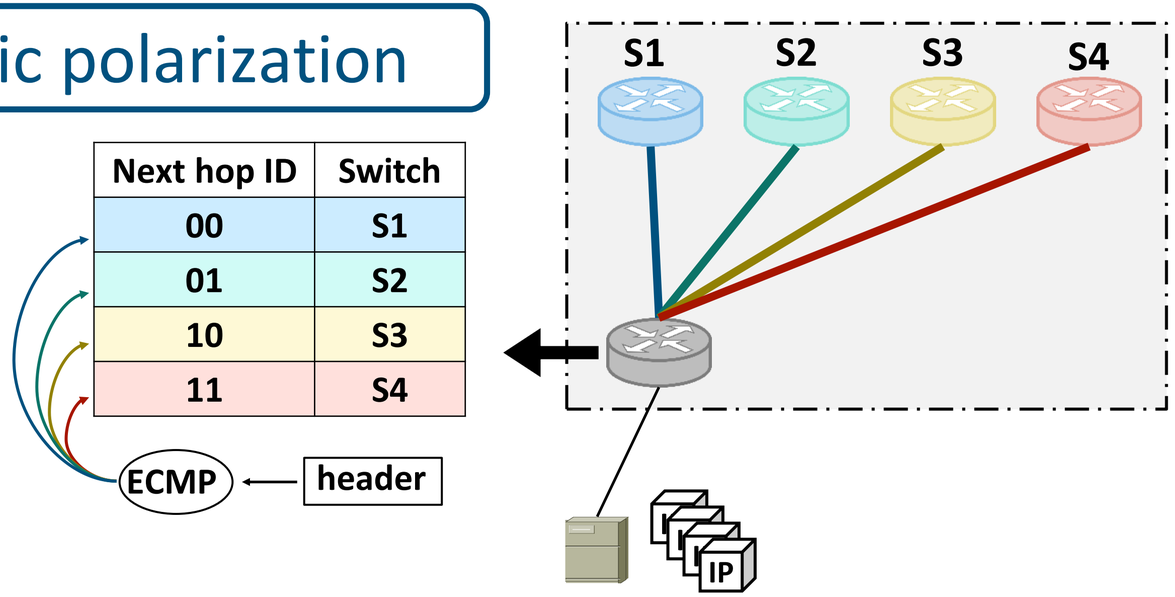


Problems in leveraging path diversity

Common practice: ECMP

- Load balancing: select the next hop based on hashing

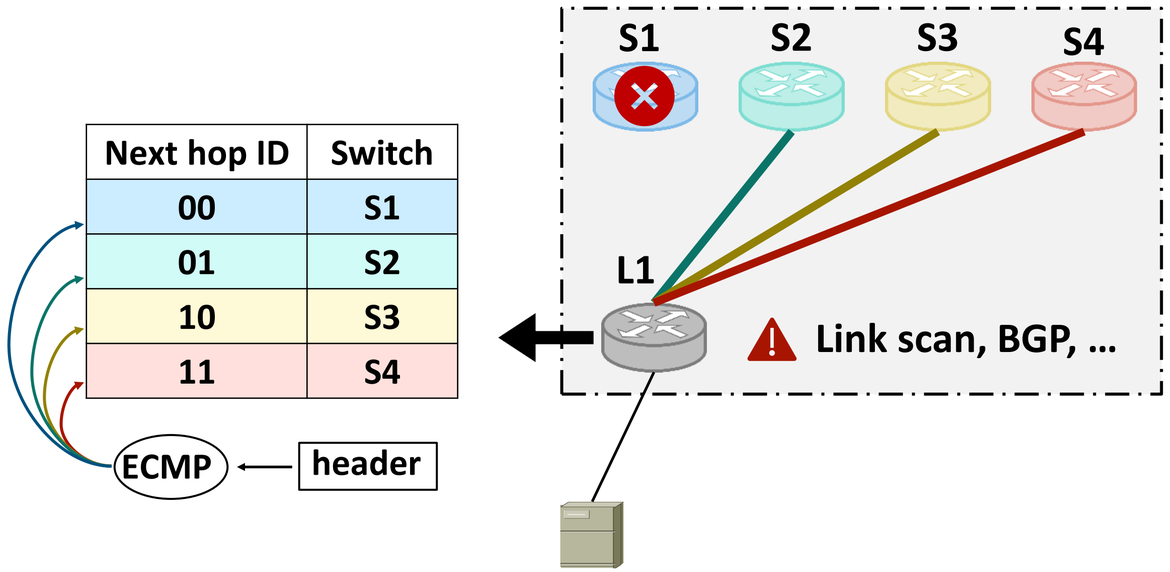
Limitation: traffic polarization



Problems in leveraging path diversity

Common practice: ECMP

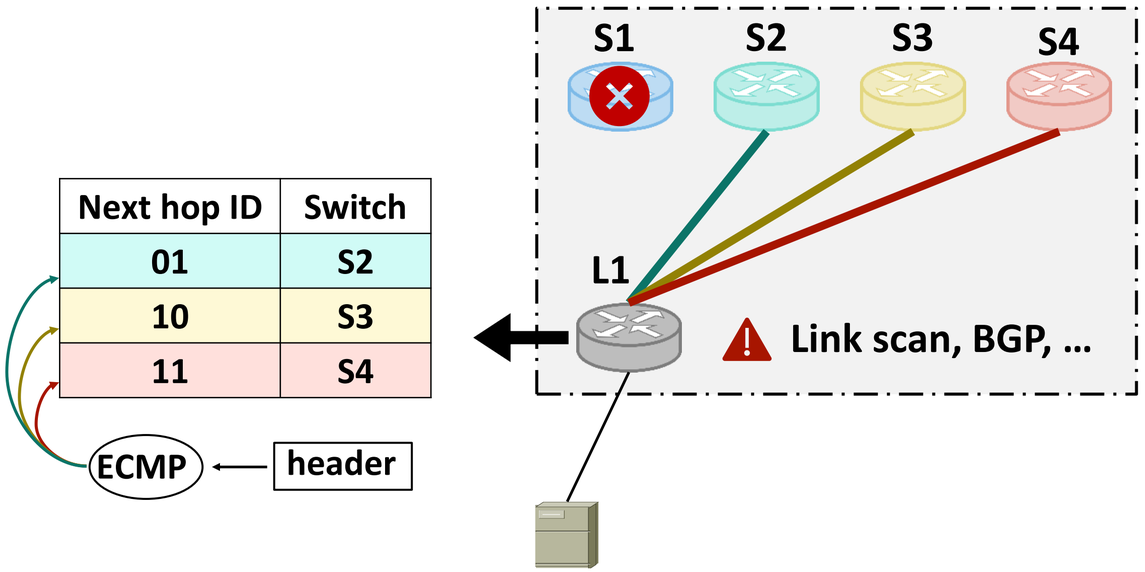
- Load balancing: select the next hop based on hashing
- Fault tolerance: remove the next hop upon detected failure



Problems in leveraging path diversity

Common practice: ECMP

- Load balancing: select the next hop based on hashing
- Fault tolerance: remove the next hop upon detected failure

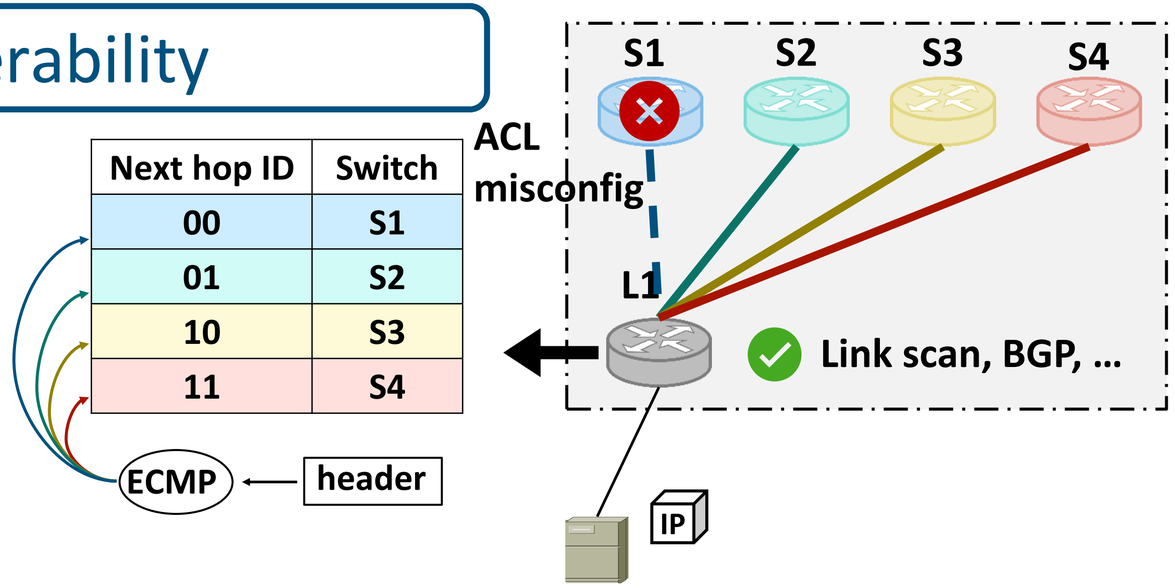


Problems in leveraging path diversity

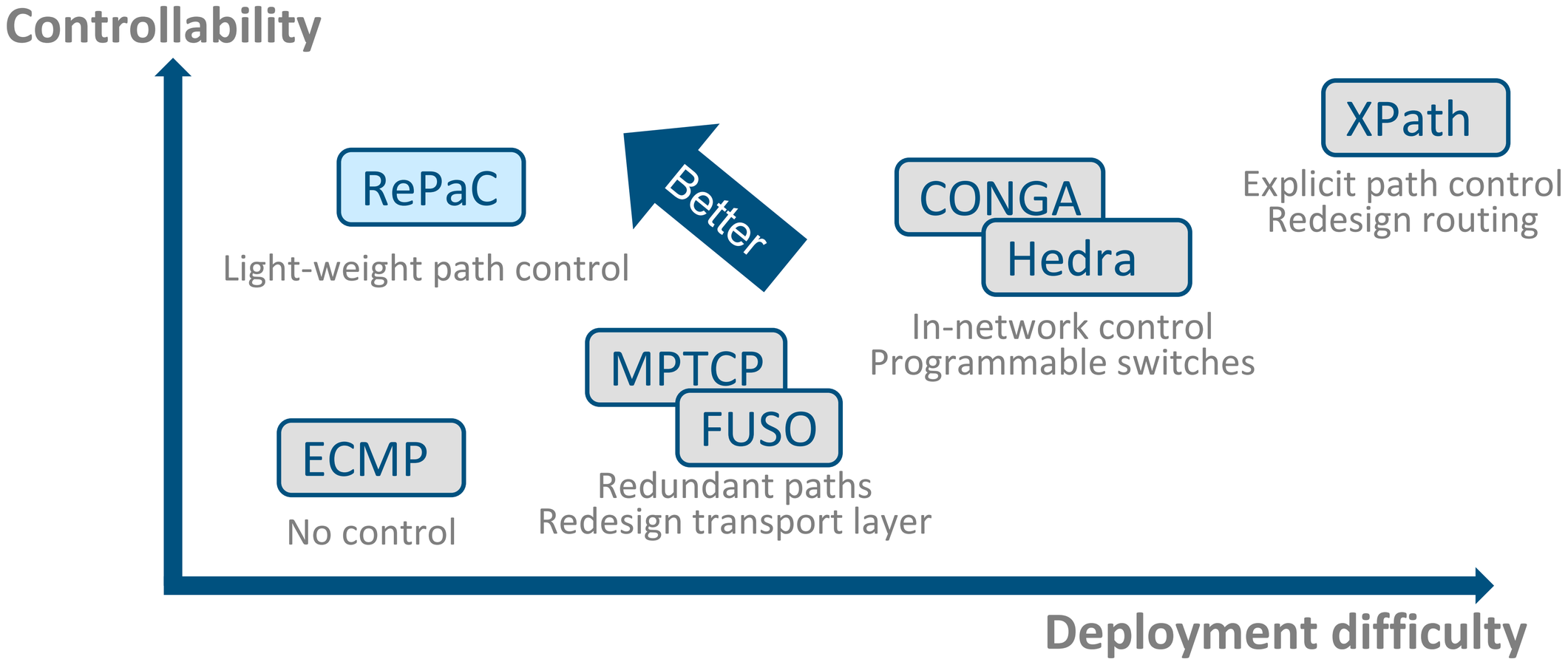
Common practice: ECMP

- Load balancing: select the next hop based on hashing
- Fault tolerance: remove the next hop upon detected failure

Limitation: coverability



Existing approaches



RePaC: Relative Path Control

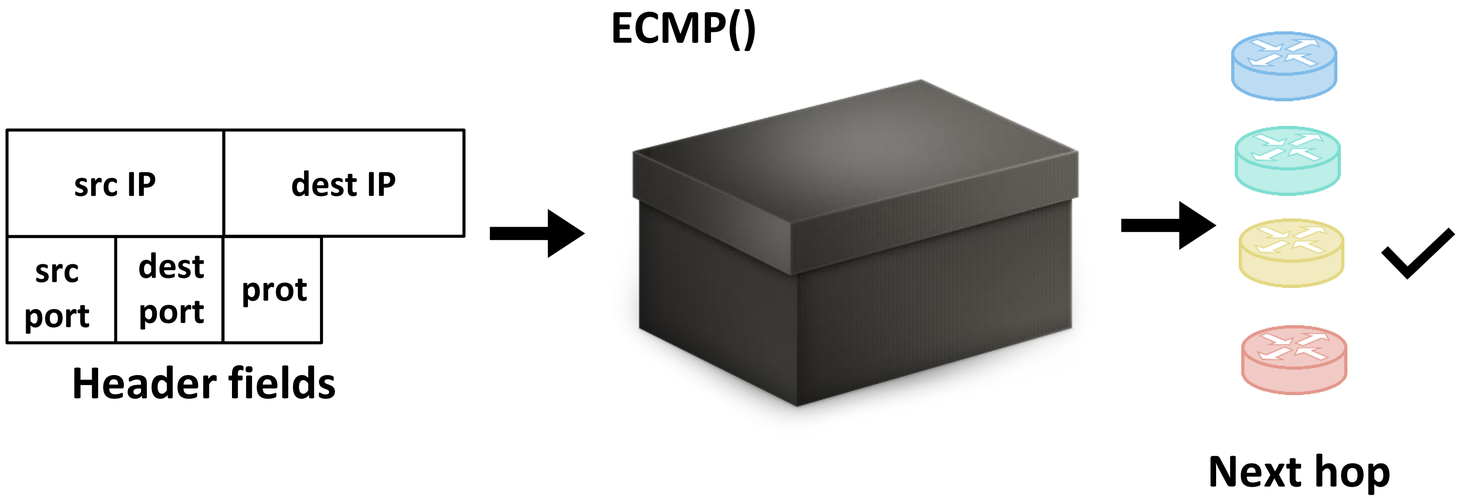
Insight: leverage hashing linearity in ECMP

What is hashing linearity?

How does RePaC leverage hashing linearity?

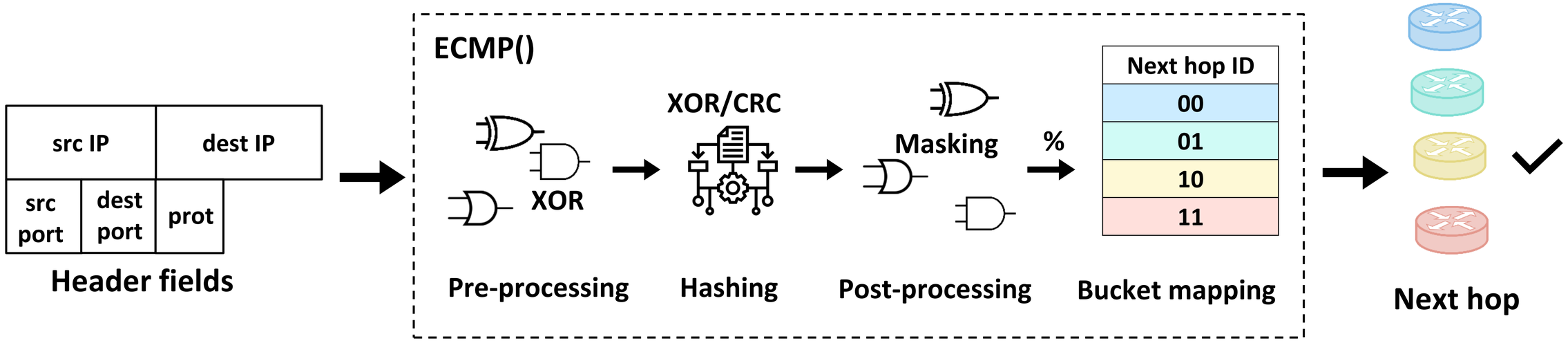
How do end-hosts apply RePaC?

Conventional perception of ECMP



Hashing linearity of ECMP

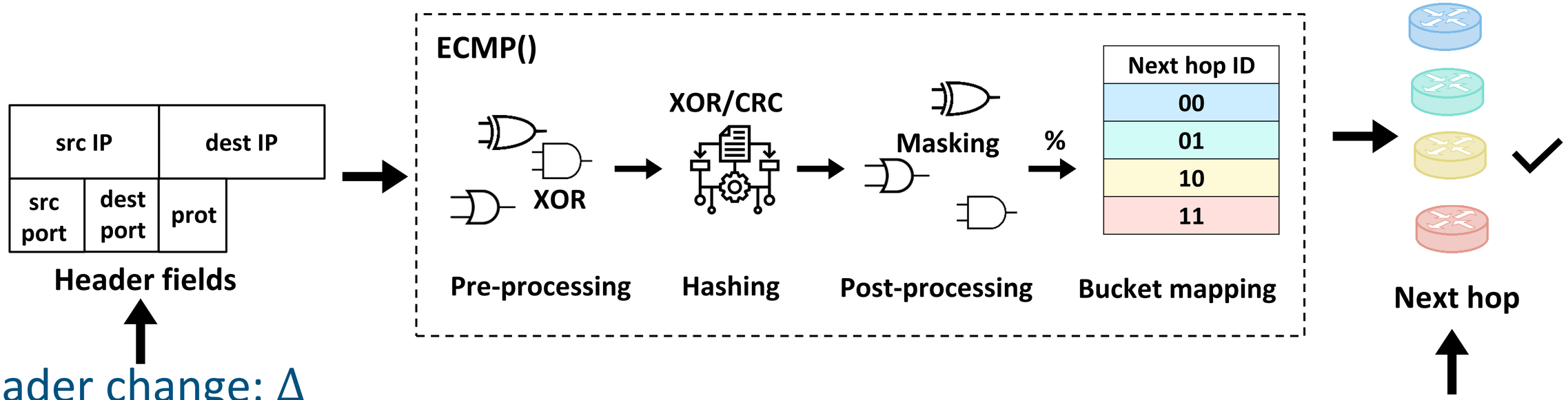
SAI (Switch Abstriction Interface)



Hashing linearity of ECMP

Linearity:

$$ECMP(hdr \oplus \Delta) = ECMP(hdr) \oplus ECMP(0) \oplus ECMP(\Delta)$$



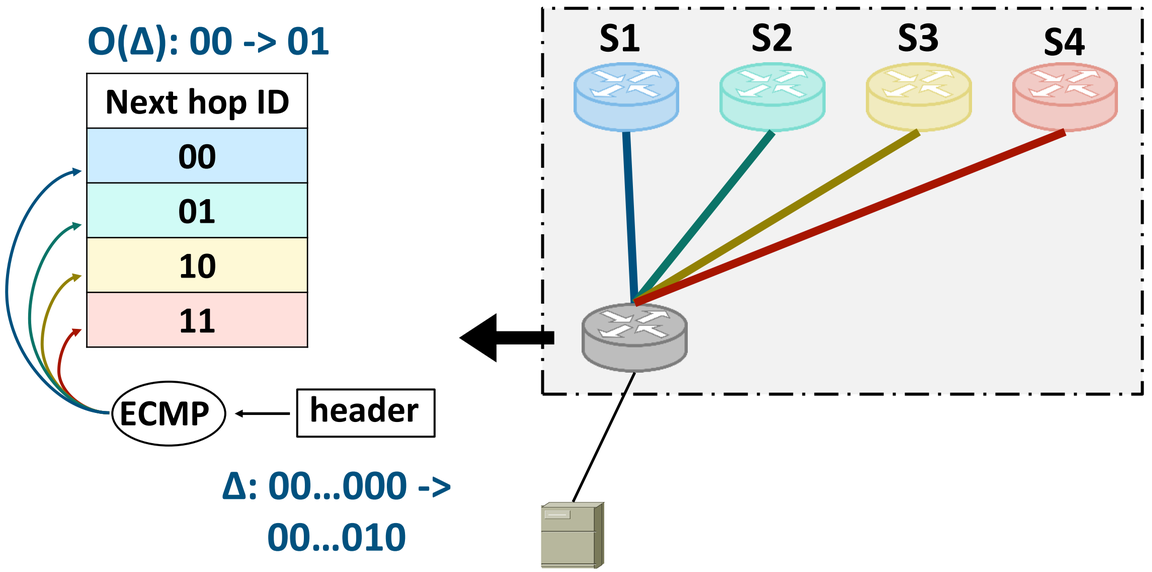
Header change: Δ

Relative path change $O(\Delta)$: $ECMP(0) \oplus ECMP(\Delta)$

Model linearity with pathmap

Pathmap: mapping between header changes Δ and path changes O(Δ)

Naïve approach: model the entire header space (2^112 keys)



Naïve map (2^112 * 2)

Header change (Δ)	Path change O(Δ)
00…001	00
00…010	01
00…011	01
…	
11…110	10
11…111	10

Model linearity with pathmap

Pathmap: mapping between header changes Δ and path changes $O(\Delta)$

Our approach: mapping **basis** header changes (exponential -> linear)

Pathmap (112 * 2)

Header change (Δ)	Path change O(Δ)
00…001	00
00…010	01
…	00
01…000	01
10…000	10

O(00…001)
⊕ O(00…010)
= O(00…011)

Naïve map (2^112 * 2)

Header change (Δ)	Path change O(Δ)
00…001	00
00…010	01
00…011	01
…	
11…110	10
11…111	10

Path control with pathmap

Key idea: decide the relative path change O(Δ)

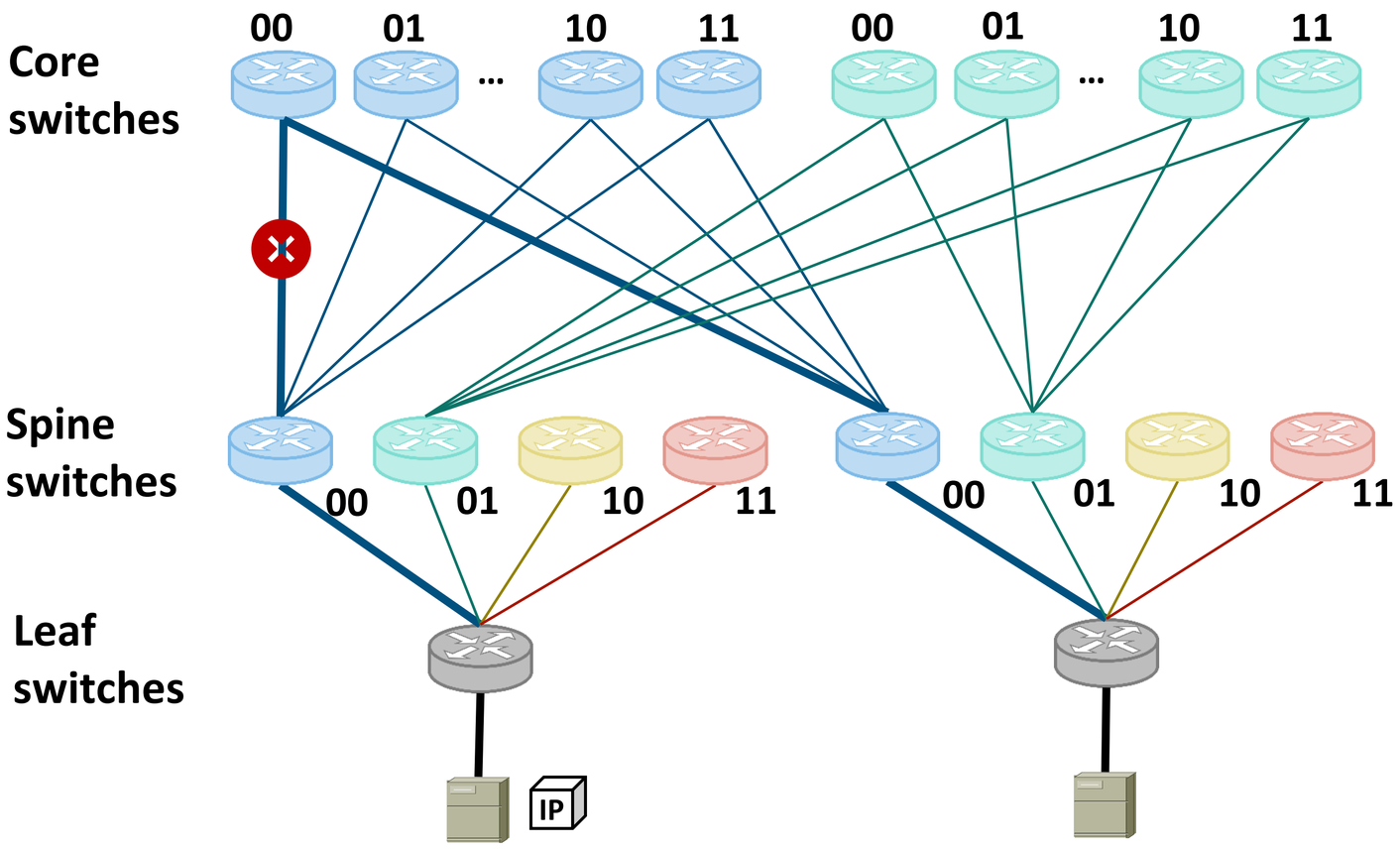
Fault tolerance: redirect flows from the failed path to a new path

Utility function: O(Δ) != 0

Load balancing: distribute loads on different paths

Utility function: O(Δ) with least load

Showcase: fault tolerance

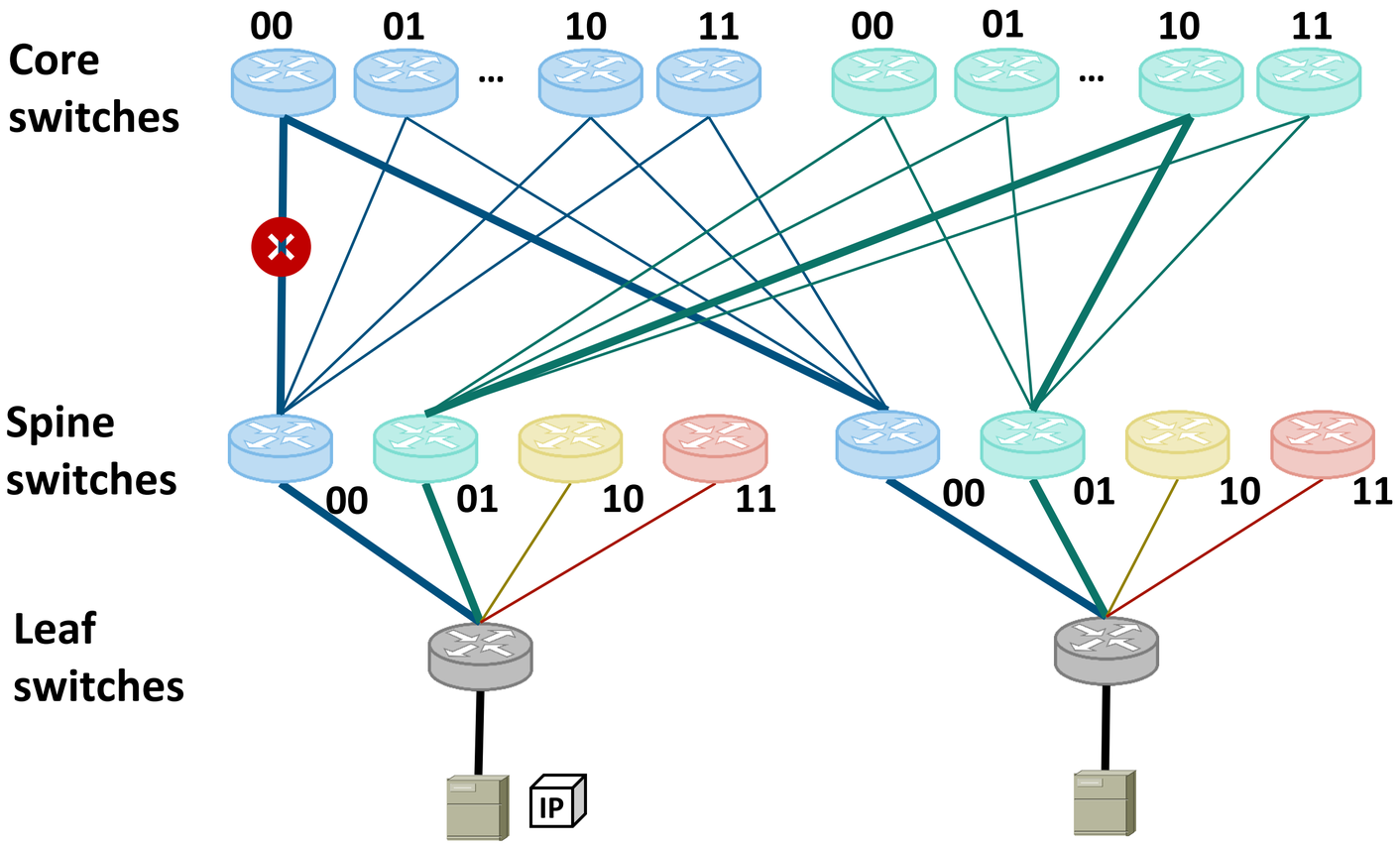


Spine switch: 00 -> 01

Core switch: 00 -> 10

Header change (Δ)	Next hop change at spine O(Δ)	Next hop change at core O(Δ)
00001	00	01
00010	01	10
00100	00	00
01000	01	00
10000	10	00

Showcase: fault tolerance



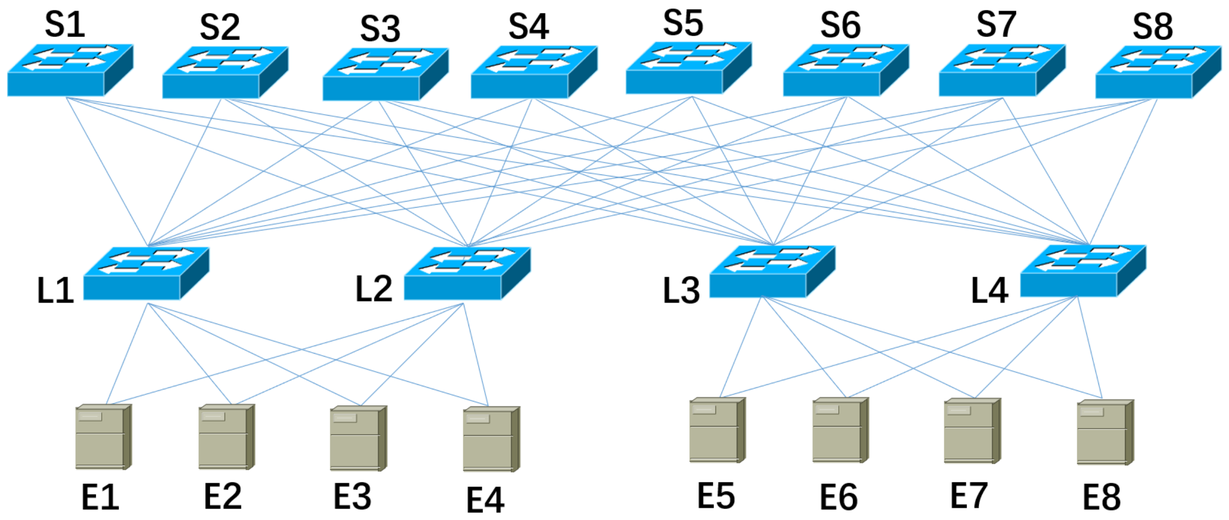
Spine switch: 00 -> 01

Core switch: 00 -> 10

Header change (Δ)	Next hop change at spine O(Δ)	Next hop change at core O(Δ)
00001	00	01
00010	01	10
00100	00	00
01000	01	00
10000	10	00

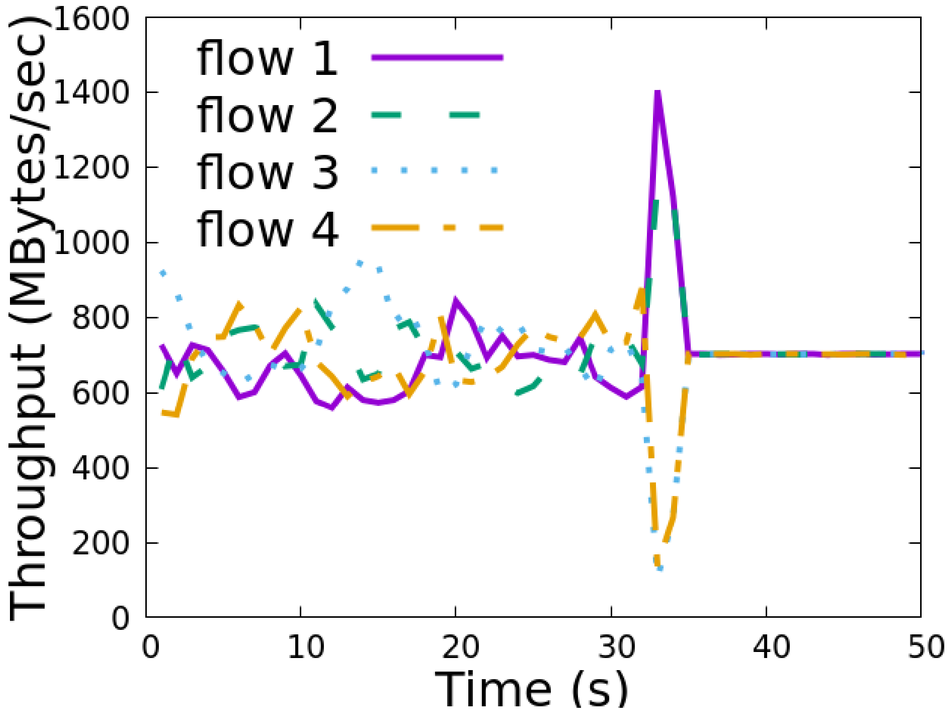
Evaluation

- A test pod with Broadcom Trident 3 and Broadcom Tomahawk 3
- Same configurations as production data centers

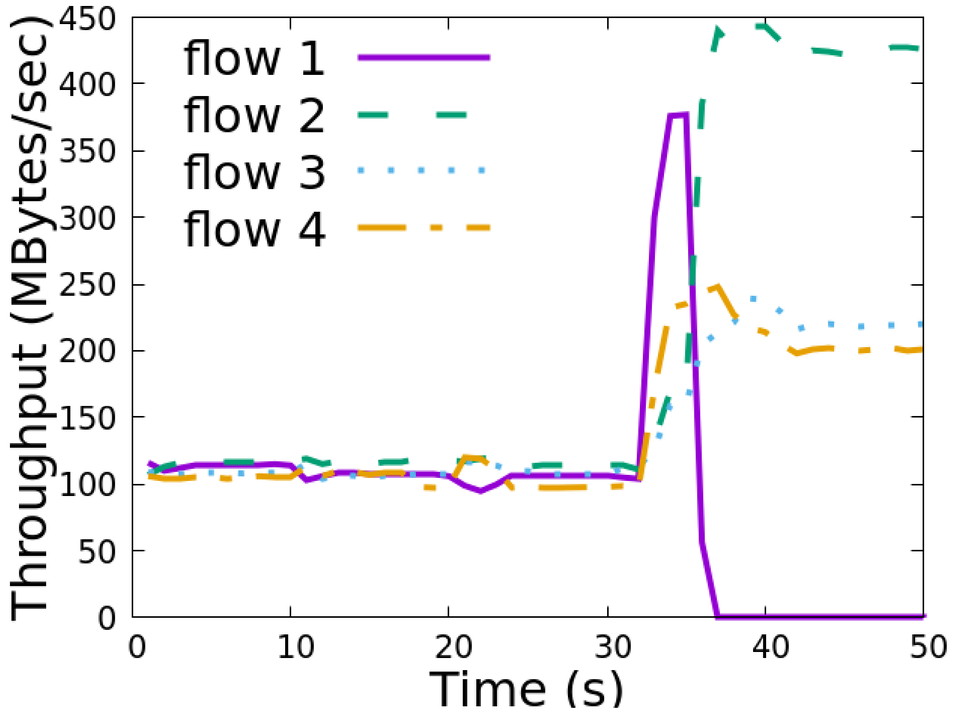


Fault tolerance performance

RePaC successfully recovers from link failure but MPTCP cannot.



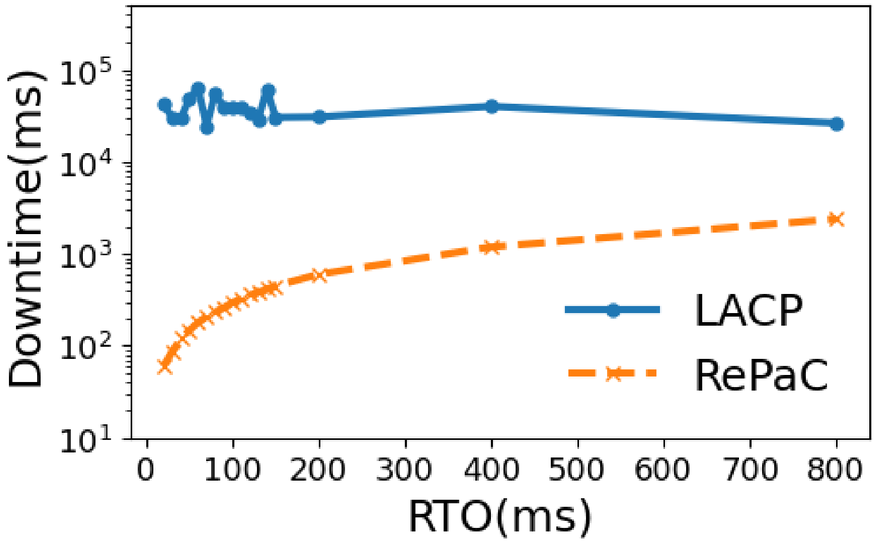
RePaC



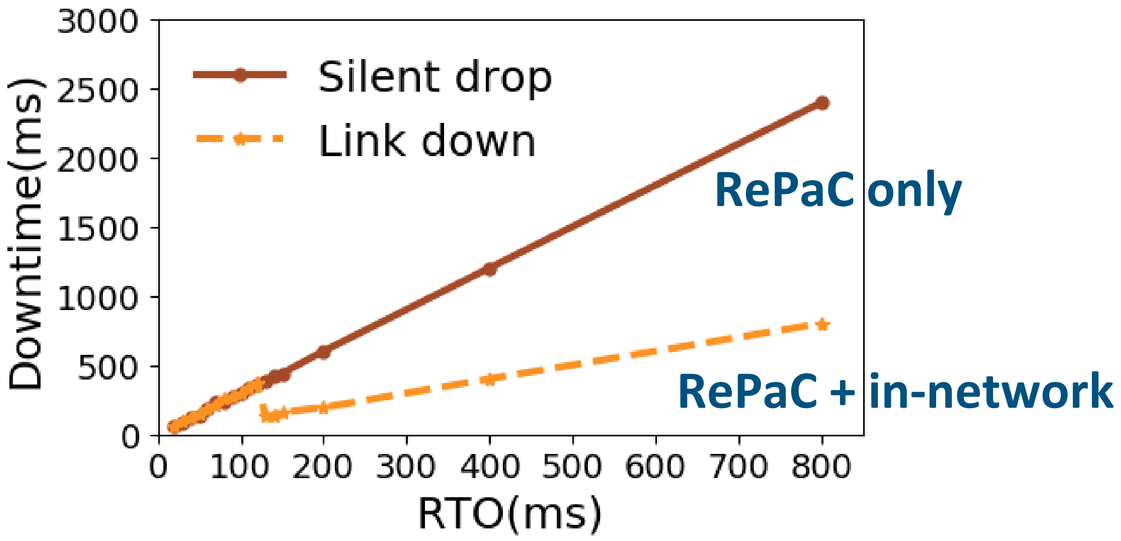
MPTCP

Fault tolerance performance

- RePaC reduces downtime compared with in-network failure recovery.
- RePaC cooperates with in-network schemes well



RePaC v.s. in-network schemes



RePaC with in-network schemes (link scan, etc.)

Takeaways

Insight:
Leverage low layer properties for network control

RePaC: leverage hashing linearity to control relative path change

Applications with relative path control:
- Showcases: fault tolerance, load balancing
- More in the future

Thank you!

Contact: zhehui@cs.ucla.edu

UCLA

Alibaba Group