Towards Highly Available Clos-Based WAN Routers

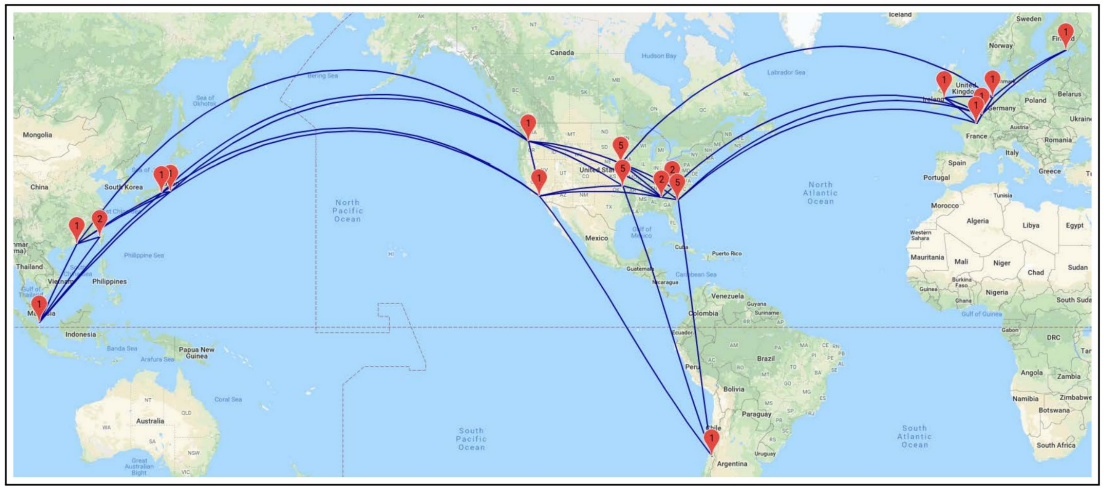
Sucha Supittayapornpong, Barath Raghavan, Ramesh Govindan
University of Southern California

SIGCOMM 2019

USC

Google’s Wide Area Network

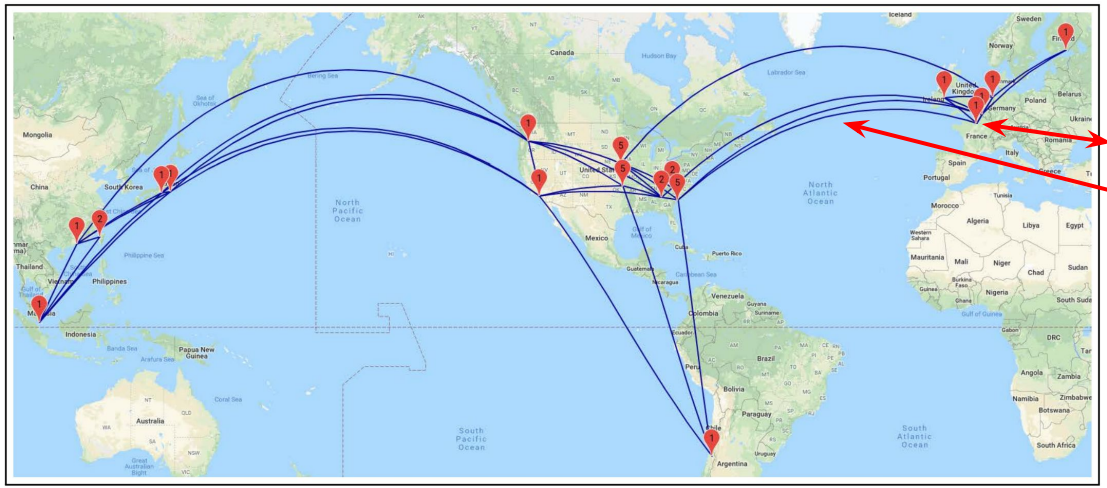
This network connects datacenters, so it has to be highly available.



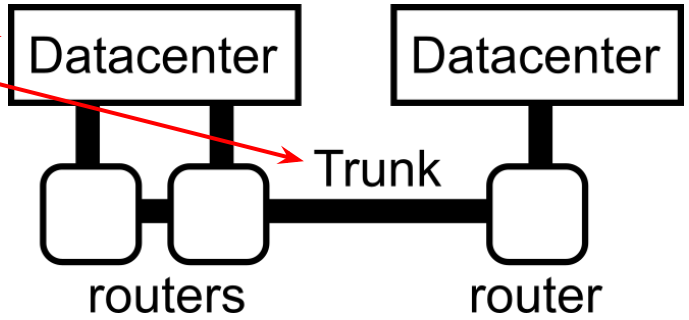
from B4 and After SIGCOMM’18

Google's Wide Area Network

Each datacenter has one or more routers, and each router is connected by trunks.



from B4 and After SIGCOMM'18



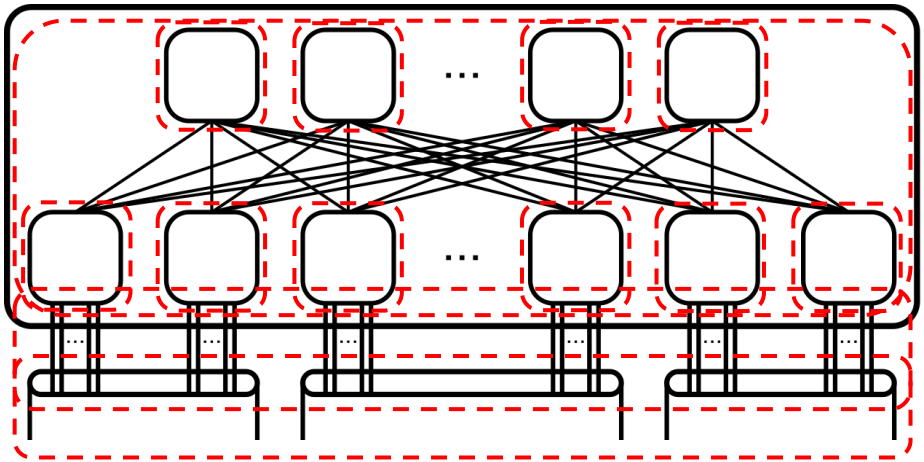
A Trunk Contains Many Optical Links

https://www.sd-wan-experts.com/blog/undersea-cables/

WAN Router

Trunk's links are wired to the router.

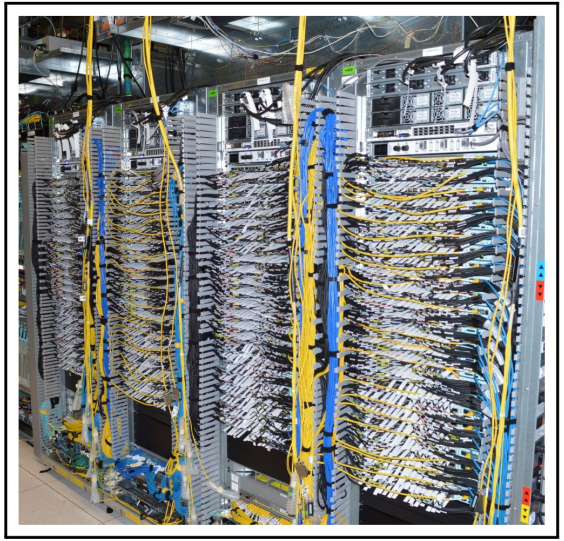
Real routers have 128 or 512 ports.



Router

Wiring

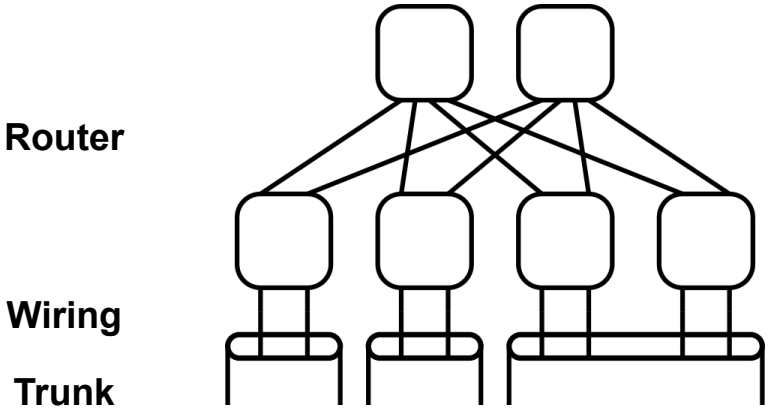
Trunk



from B4 and After SIGCOMM'18

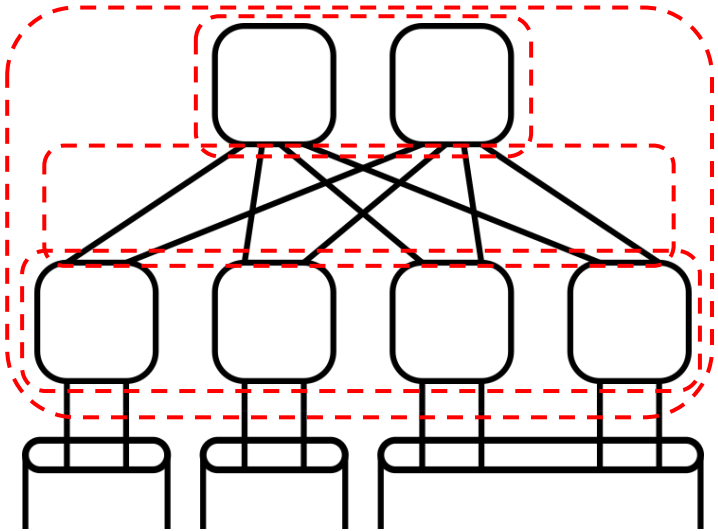
WAN Router

Let’s use a toy router to develop intuitions.



Clos-Based WAN Router

A router is built as a Clos topology.



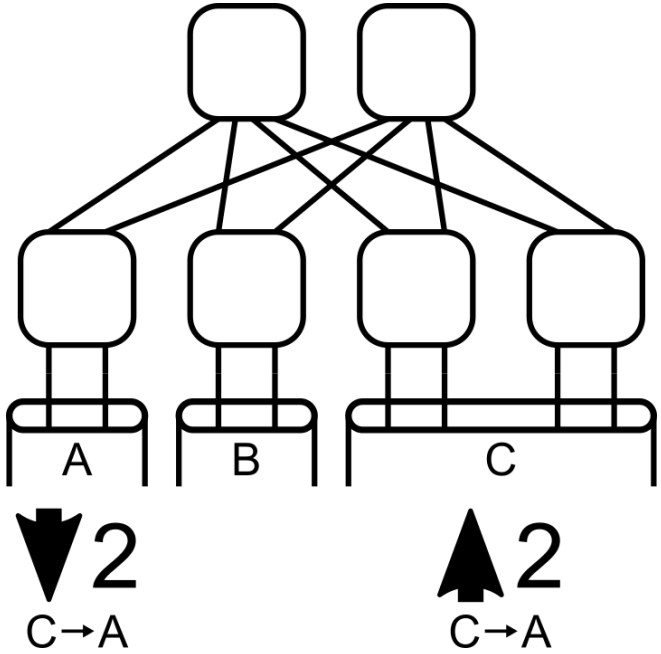
Upper stage

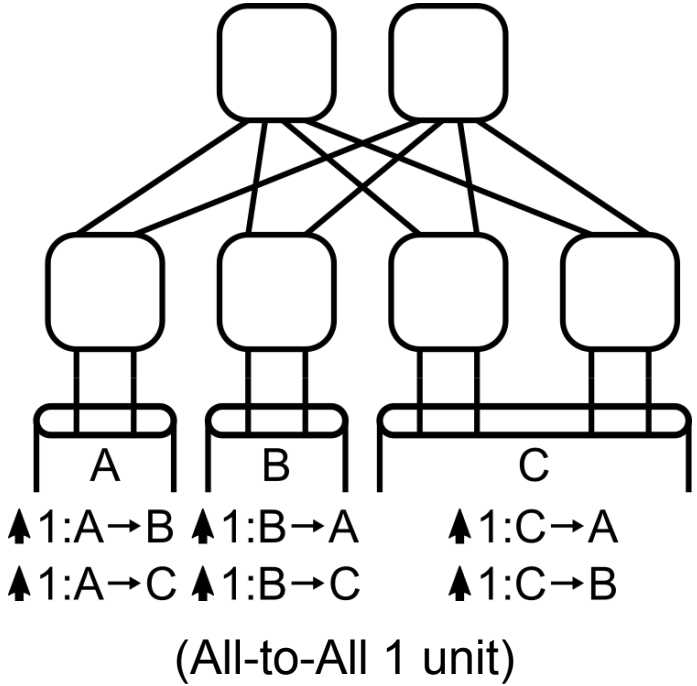
Internal Link

Lower stage

Clos is Non-Blocking

It can handle any traffic matrix without loss.

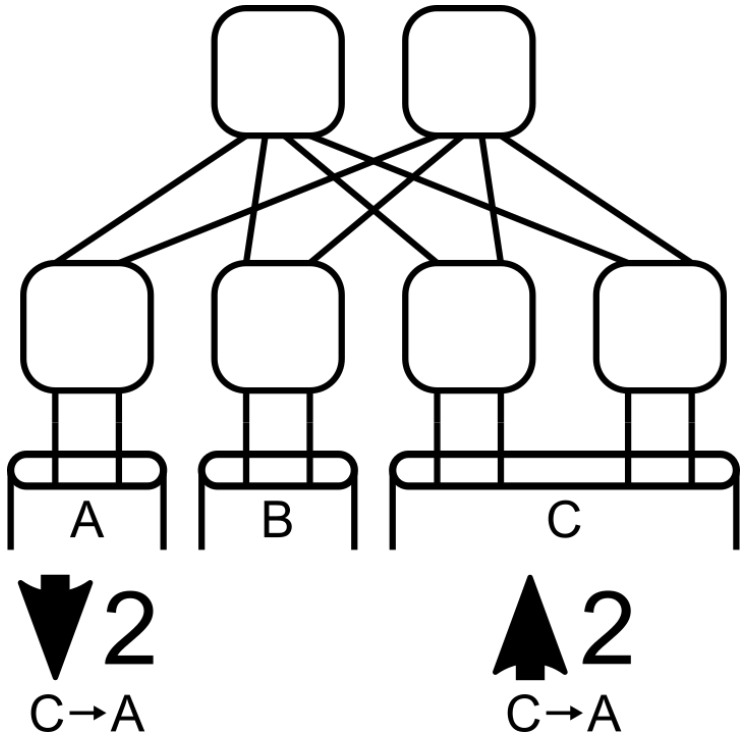




(All-to-All 1 unit)

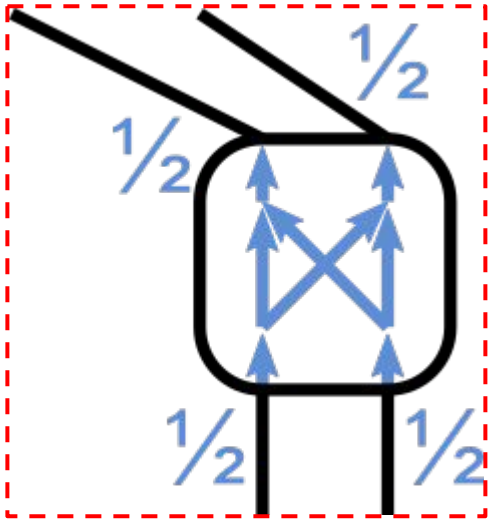
Clos is Non-Blocking

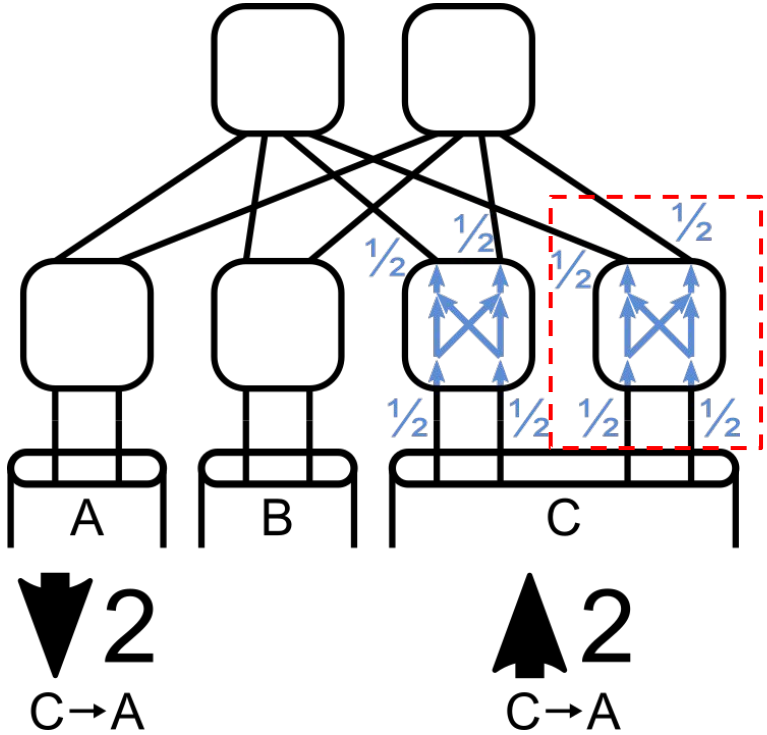
Equal-cost multipath (ECMP) routing can achieve the non-blocking property.



Achieving Non-Blocking Property via ECMP

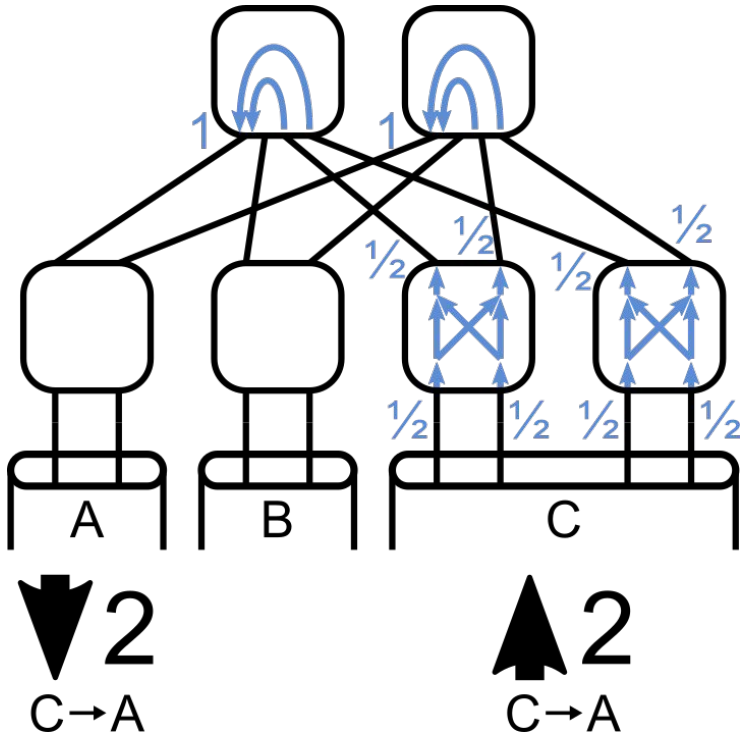
ECMP splits traffic equally to nexthops.





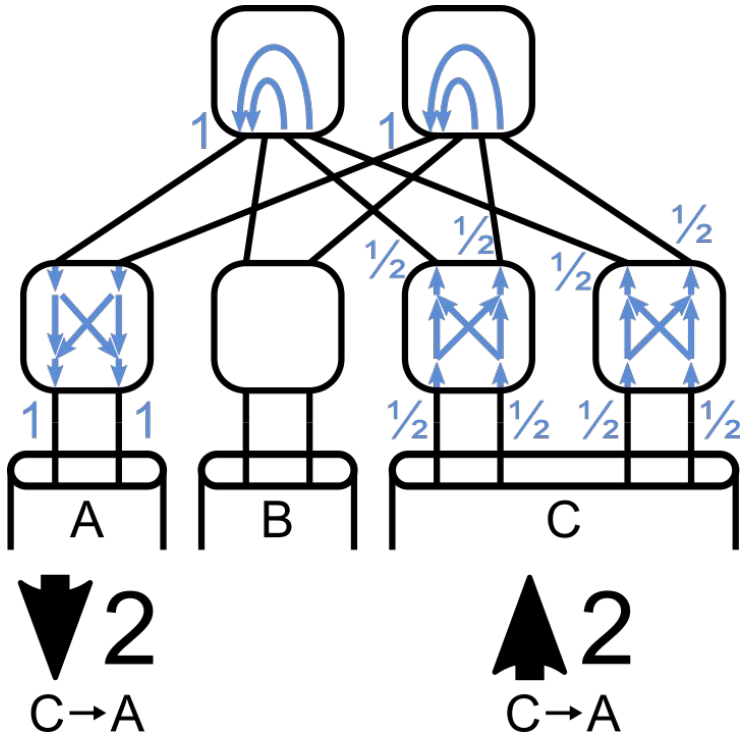
Achieving Non-Blocking Property via ECMP

ECMP splits traffic equally to nexthops.



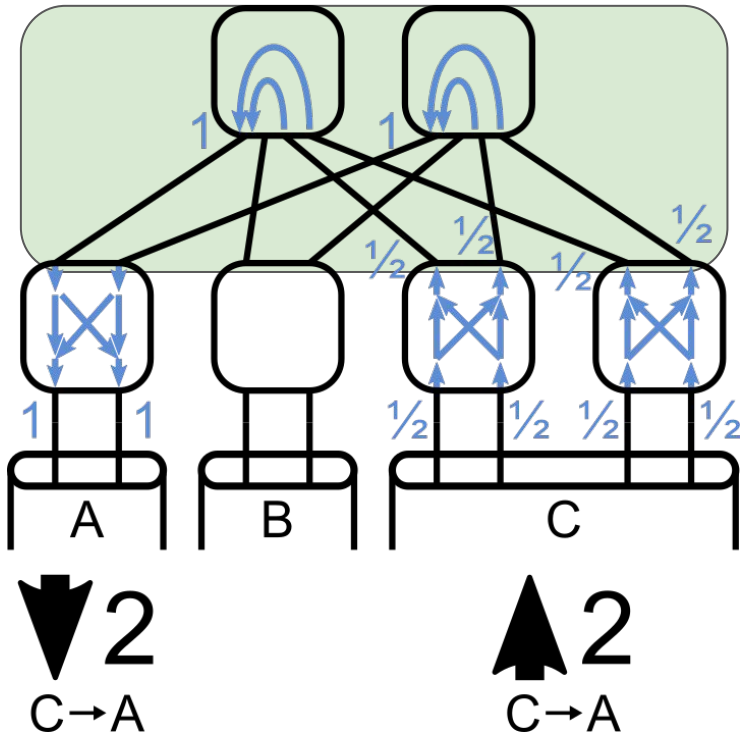
Achieving Non-Blocking Property via ECMP

ECMP splits traffic equally to nexthops.

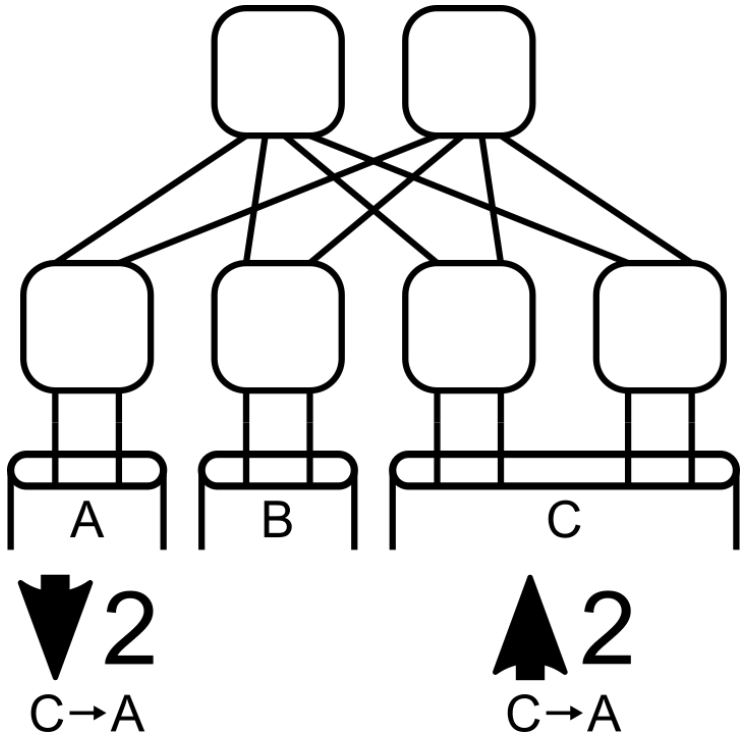


Implication of Non-Blocking Property

There is sufficient internal capacity to route traffic between lower and upper stages.

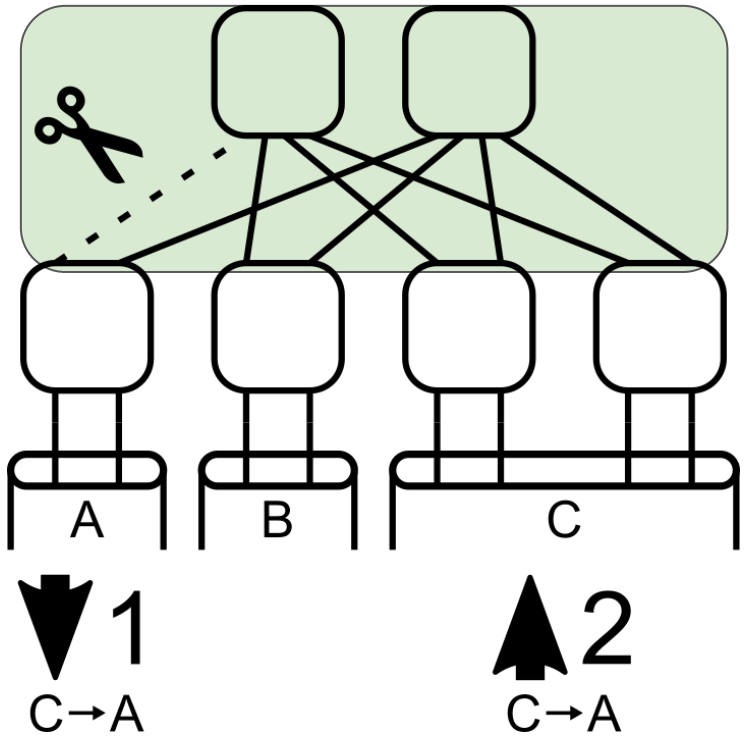


What Happens If There are Failures?



What Happens If There are Failures?

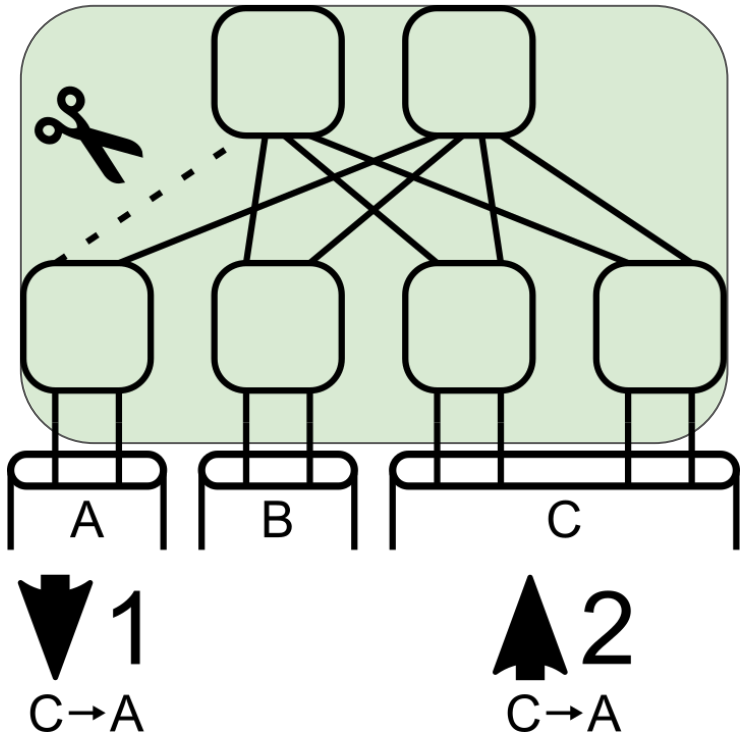
A single failure reduces internal capacity.



What Happens If There are Failures?

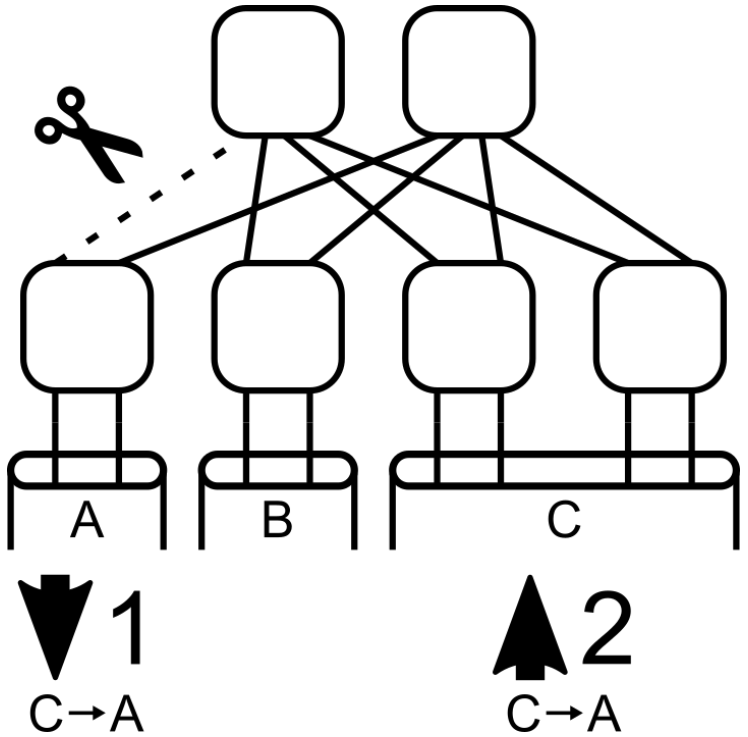
A single failure reduces internal capacity.

Overall capacity can reduce by half when ECMP is used.



Key Question

Can we completely mask internal link and switch failures?

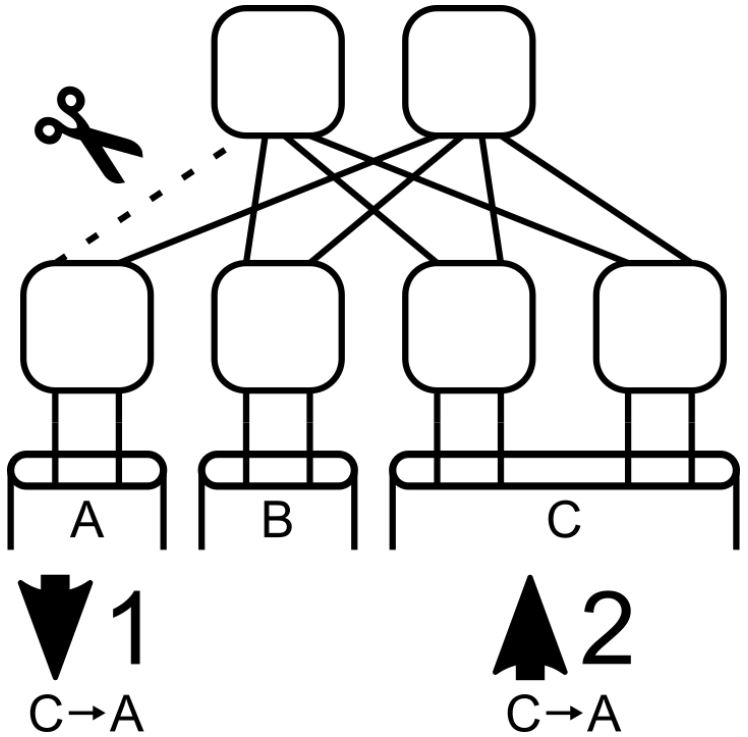


Key Question

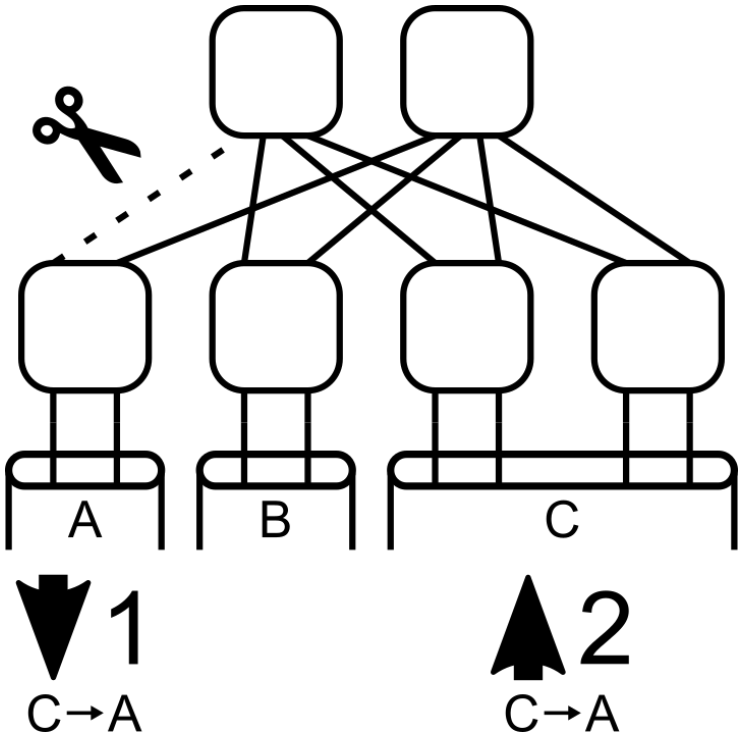
Can we completely mask internal link and switch failures?

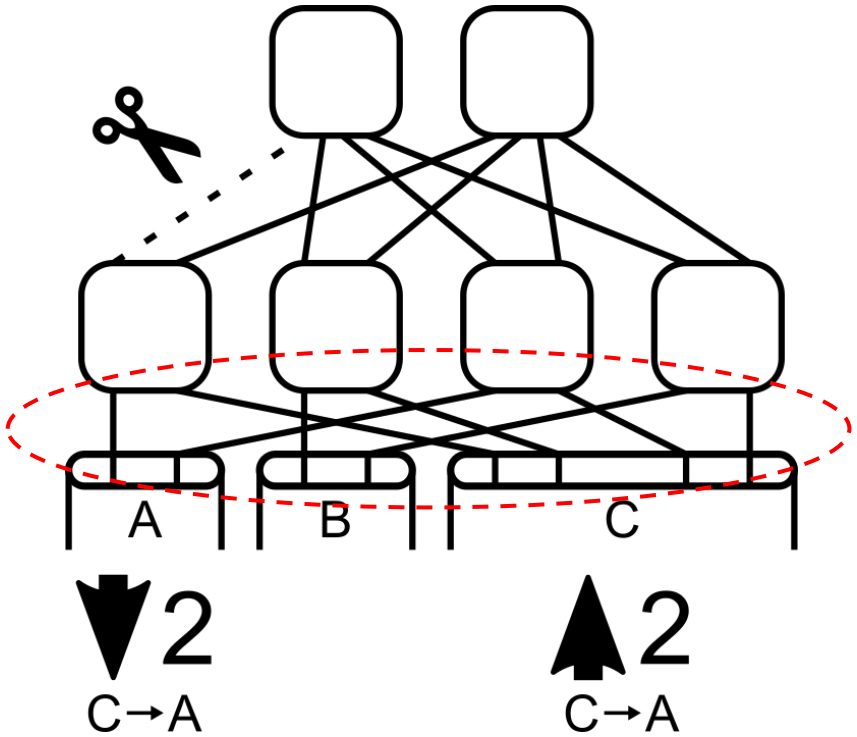
If not, can we degrade gracefully?

Existing approaches do neither of these.



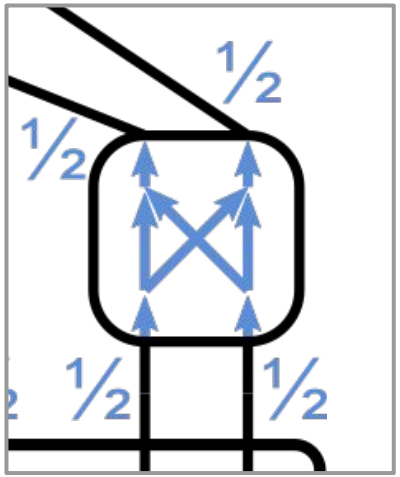
Key Insight: Wiring trunks to maximize early forwarding



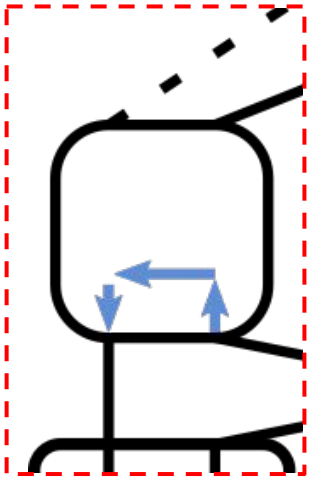


Key Insight: Wiring trunks to maximize early forwarding

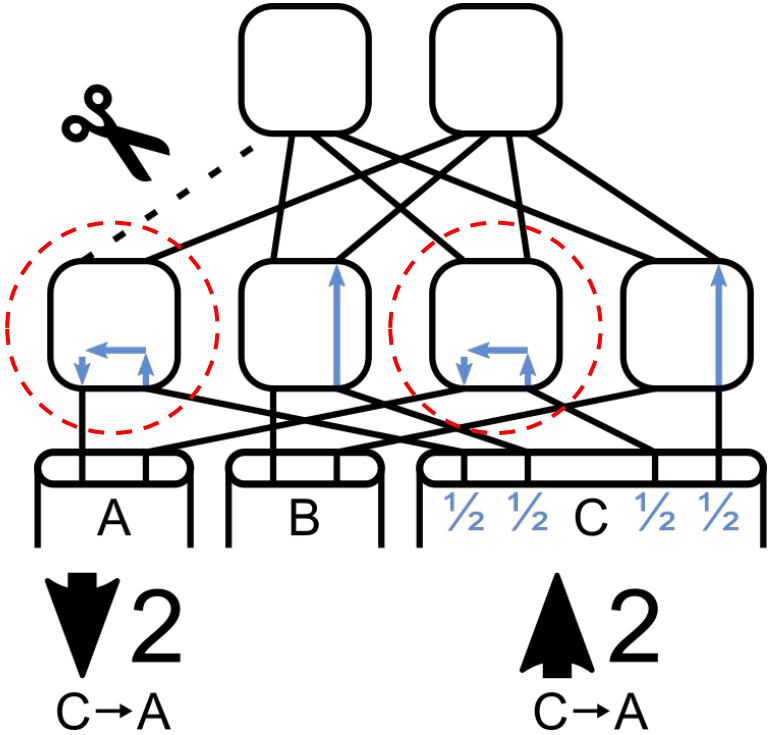
Careful wiring enables early forwarding.



Previous

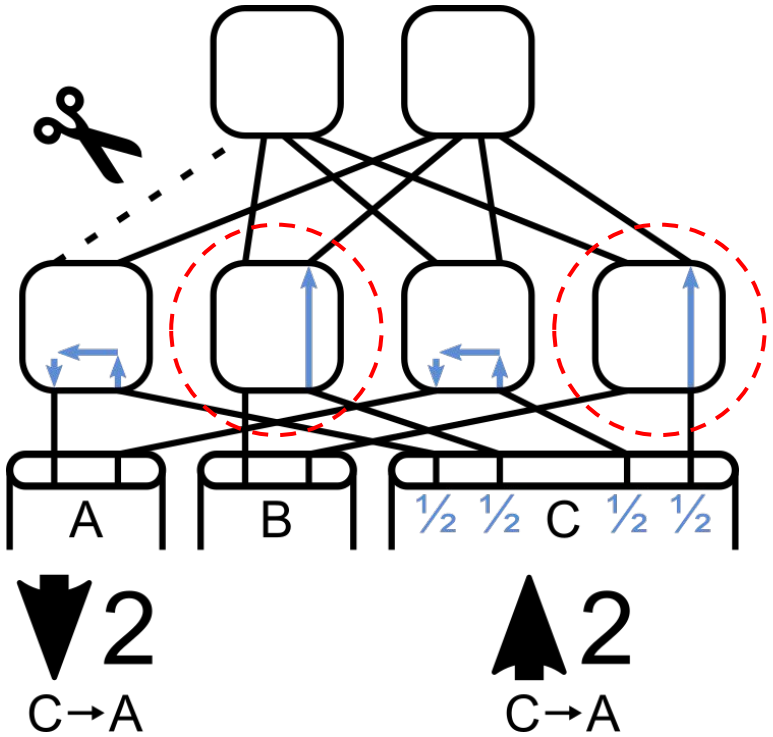


Early forwarding



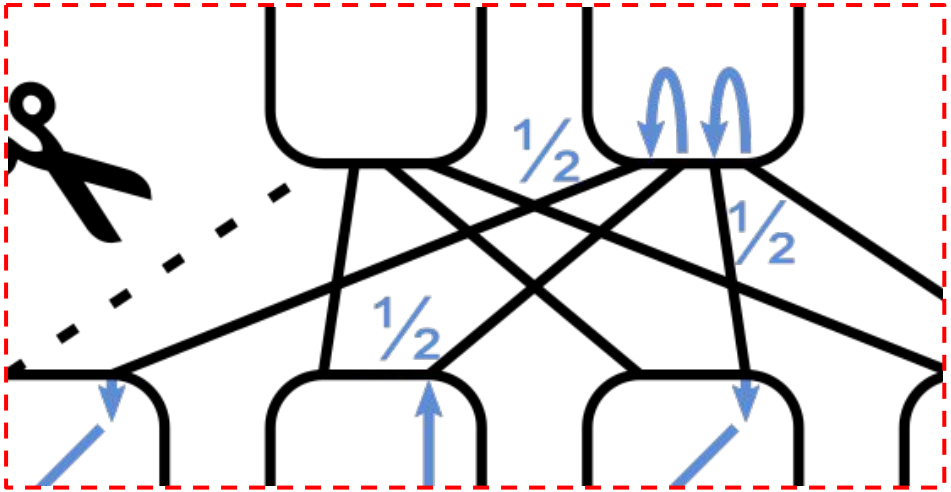
Key Insight: Wiring trunks to maximize early forwarding

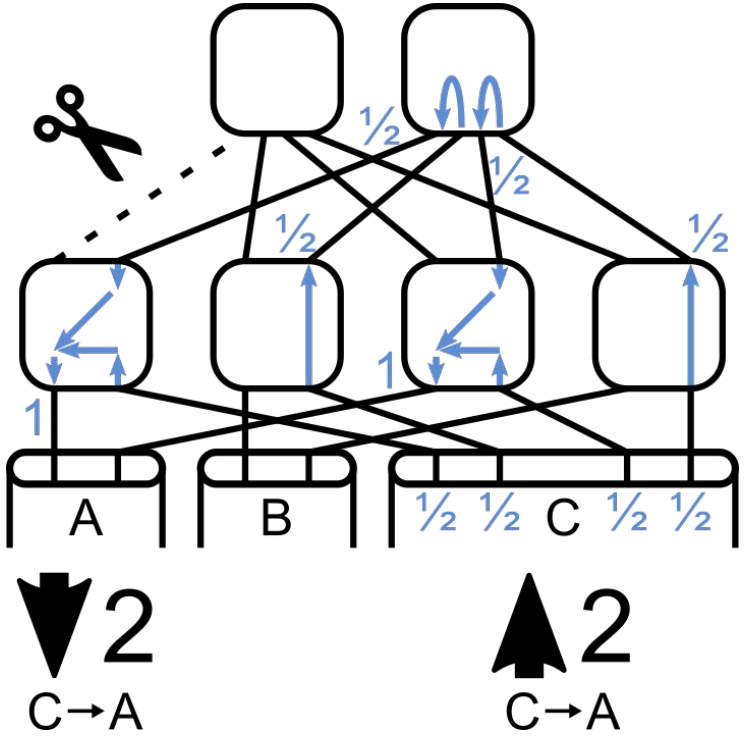
Early forwarding can reduce upflow.



Key Insight: Wiring trunks to maximize early forwarding

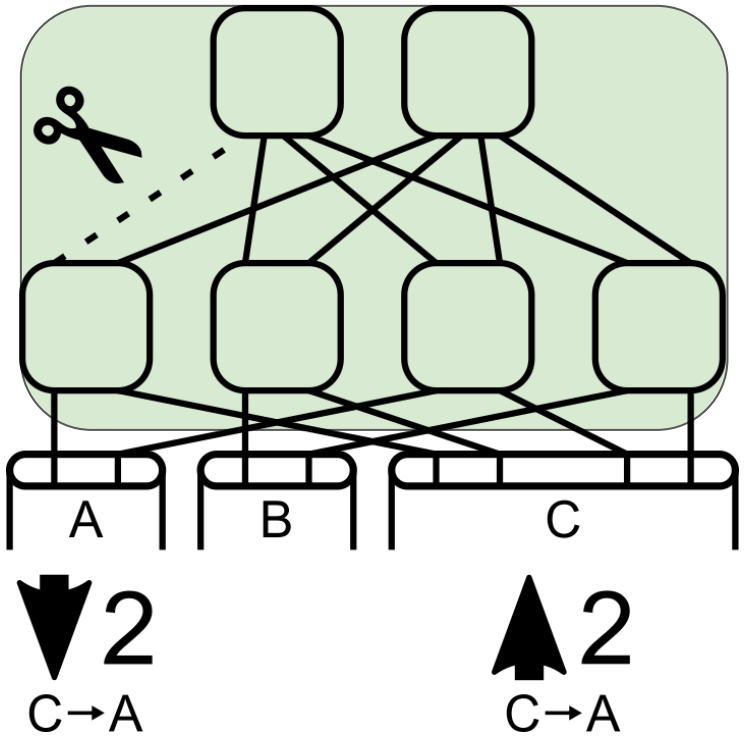
Early forwarding can reduce upflow.



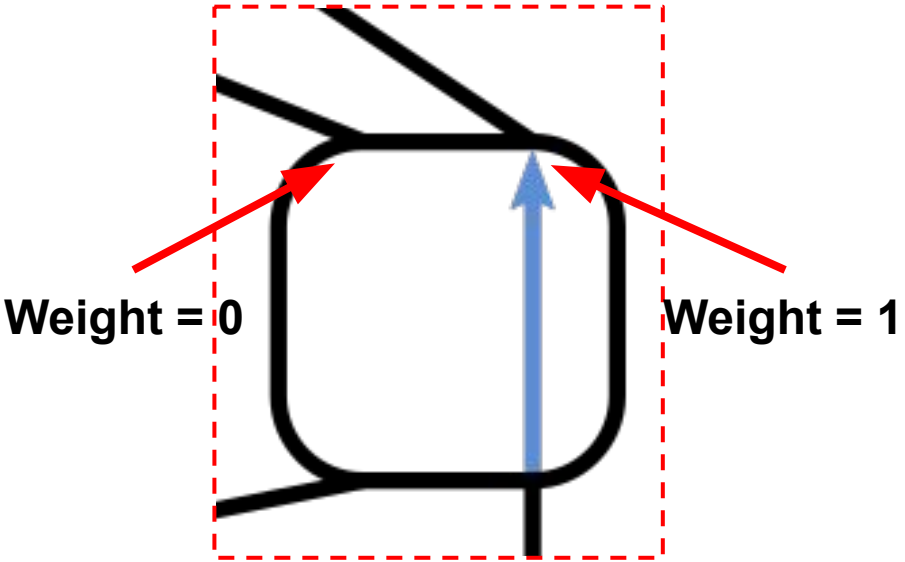


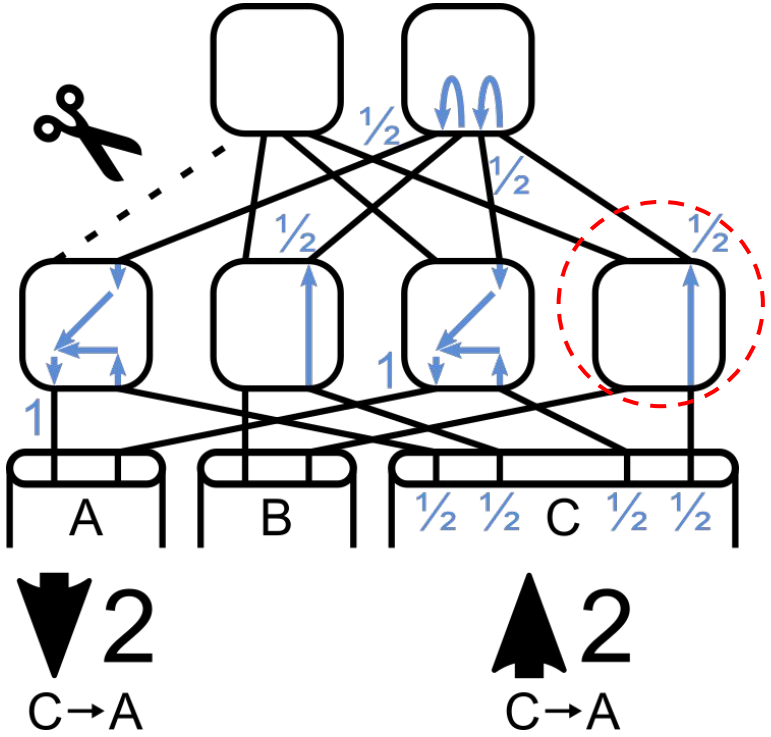
Key Insight: Wiring trunk to maximize early forwarding

The router can recover full capacity in this example. (We completely mask the failure.)

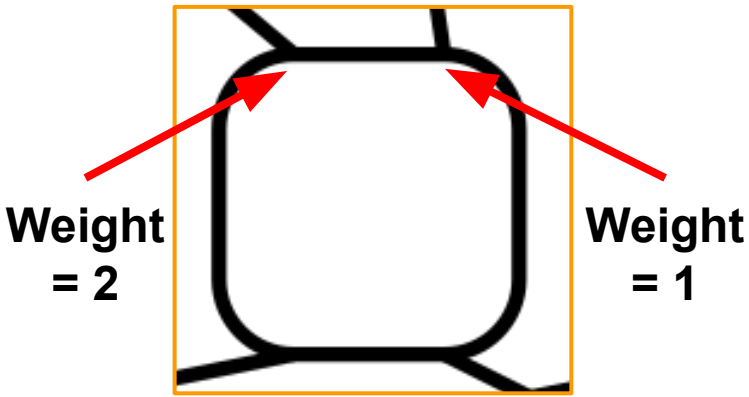


Early forwarding needs weighted version of ECMP

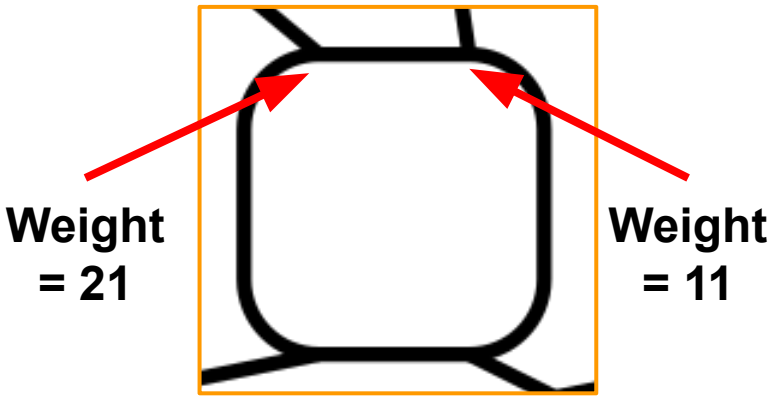




WCMP can increase table sizes

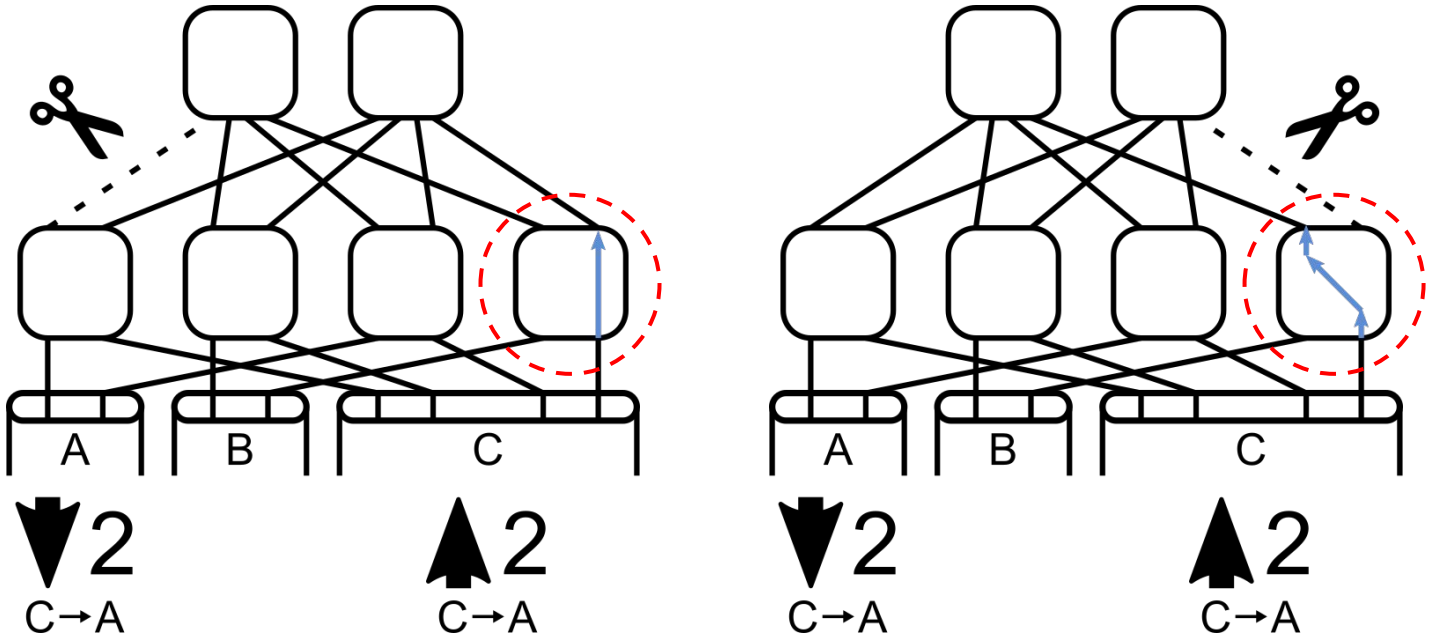


Use 2+1 = 3
weight entries



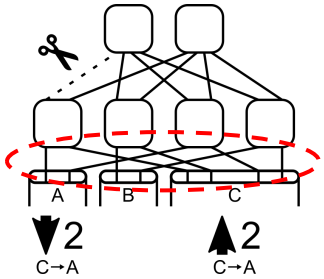
Use 21+11 = 32
weight entries

WCMP weights can depend on failure pattern

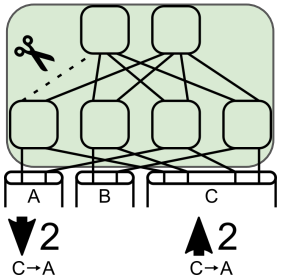


Challenges

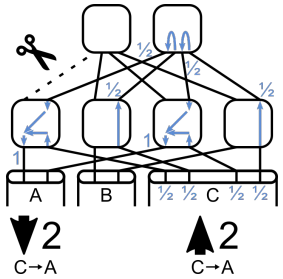
What wiring minimizes upflow?



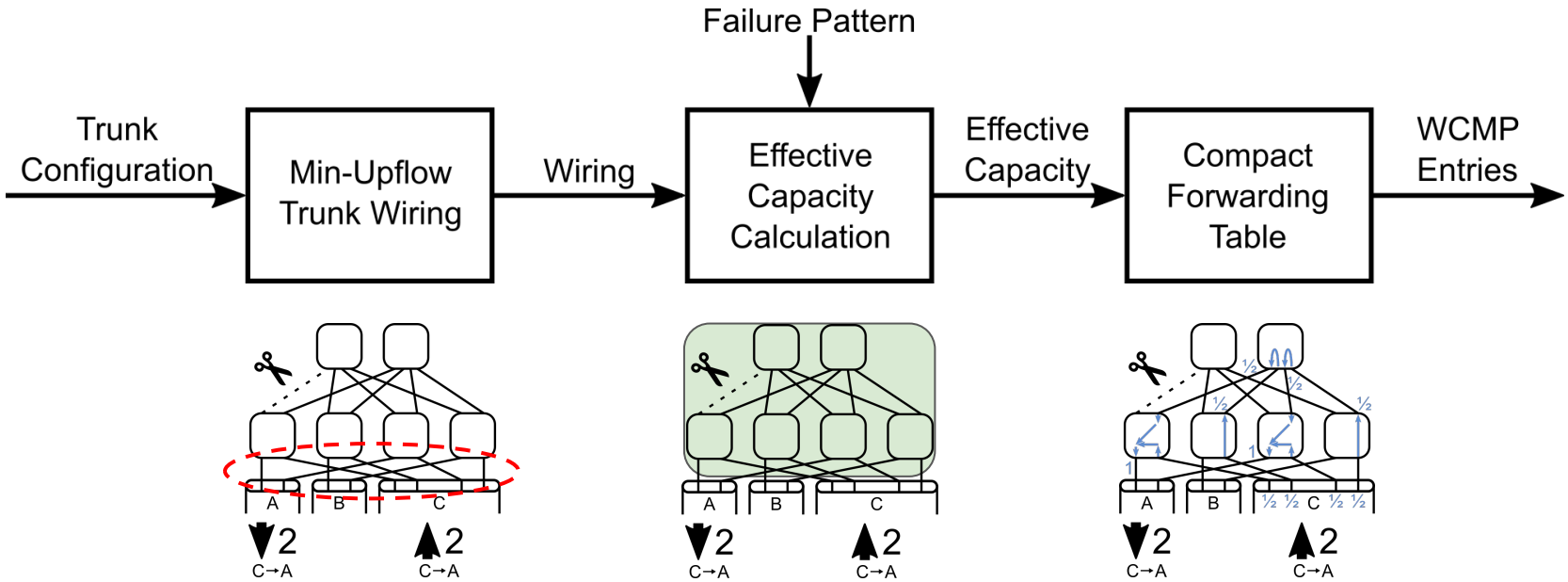
What is the effective capacity for a failure pattern?



What is the most space-efficient set of WCMP weights?

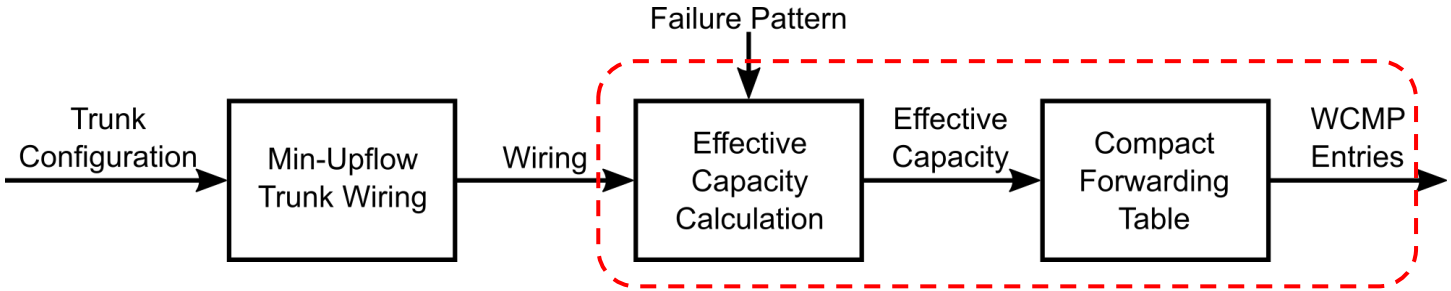


Contributions



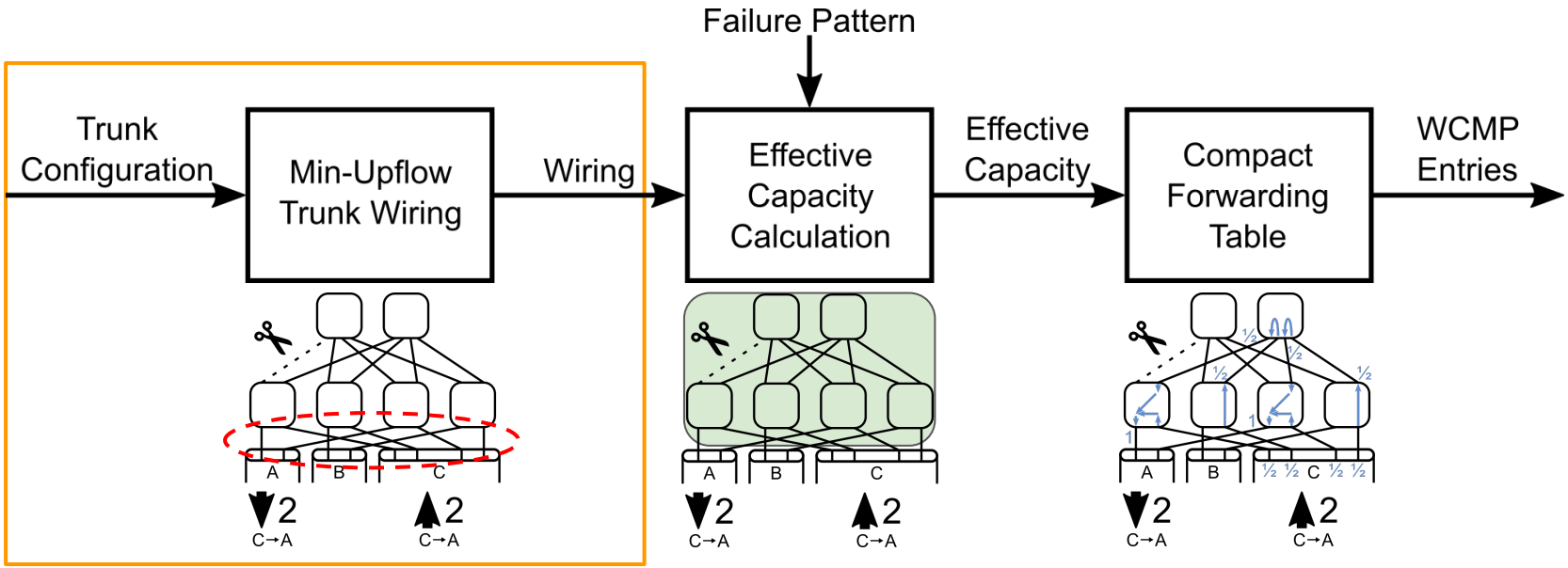
The Entire Pipeline is Offline

Computing routing table is expensive and cannot be done after failure happens.
So, we must precompute tables for every possible pattern.

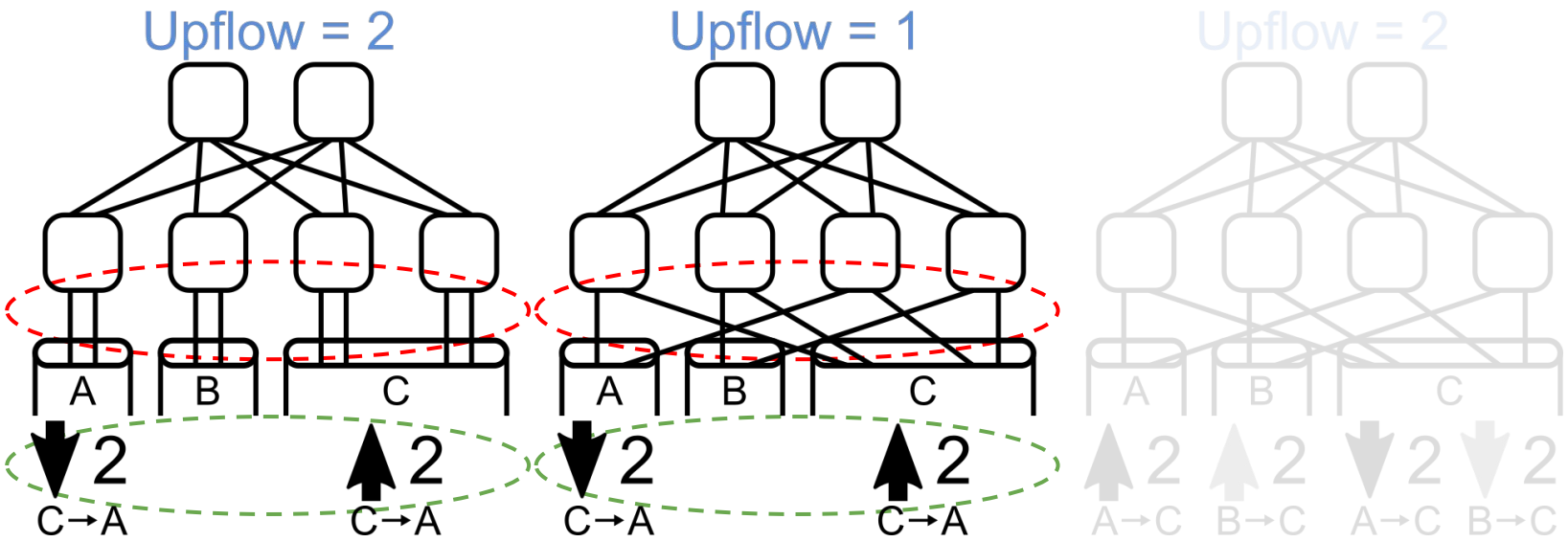


Challenge: All of these steps must scale to very large routers.

Finding Optimal Wiring

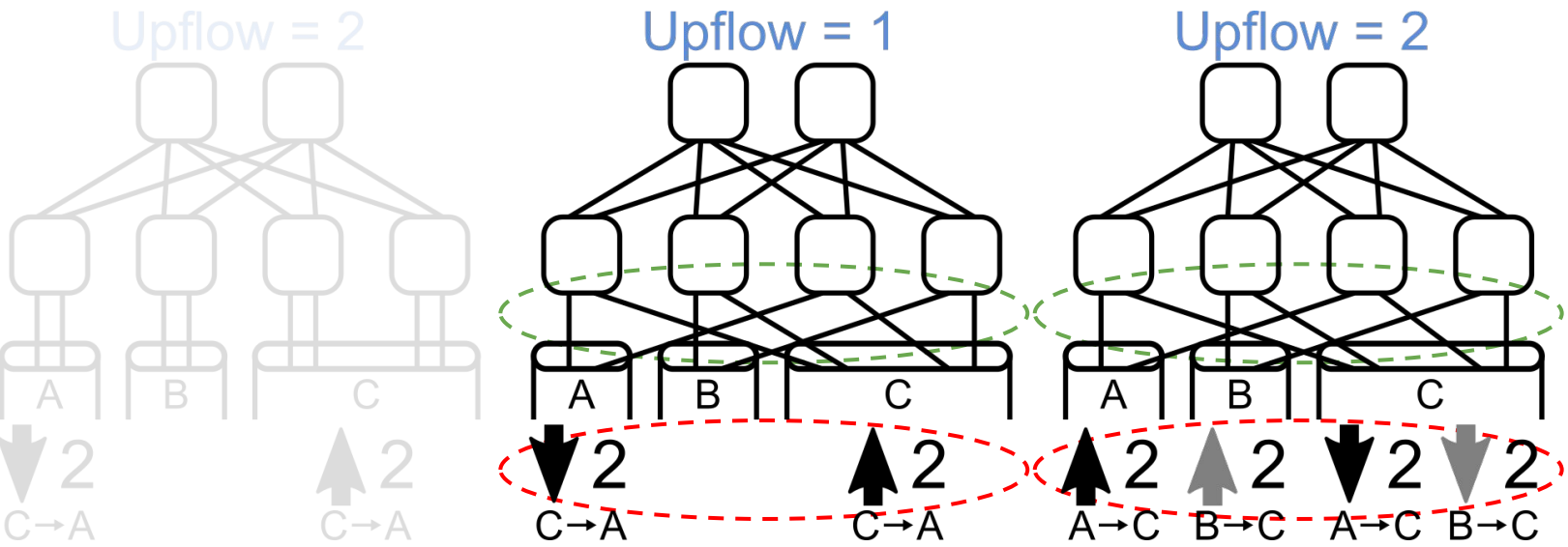


Upflow depends on both trunk wiring and traffic



Same traffic, **Different wiring**

Upflow depends on both trunk wiring and traffic



Different traffic, **Same wiring**

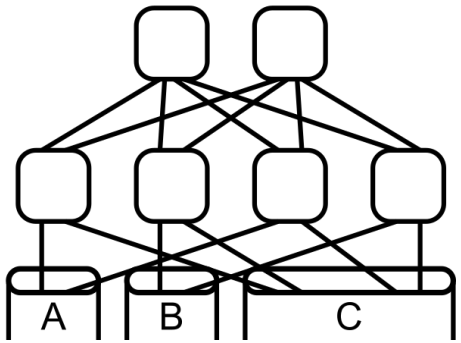
Upflow depends on both trunk wiring and traffic

Upflow is a function of wiring and traffic matrix $U(w, T)$**.**

Each wiring has its worst-case traffic

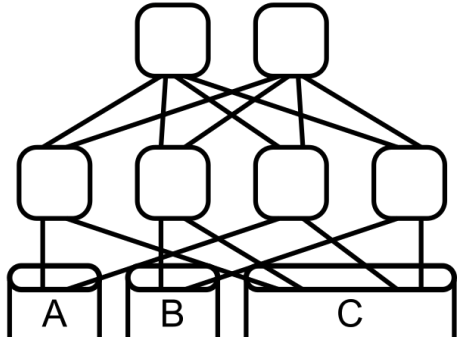
$$\max_{T \in \mathcal{T}} U(w, T)$$

Upflow = 2



Traffic matrix 1

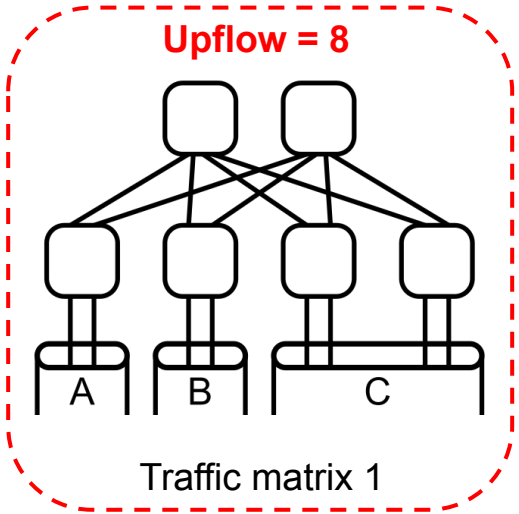
Upflow = 4

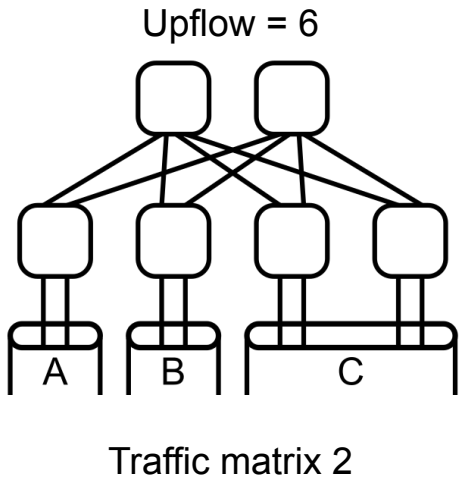


Traffic matrix 2

Each wiring has its worst-case traffic

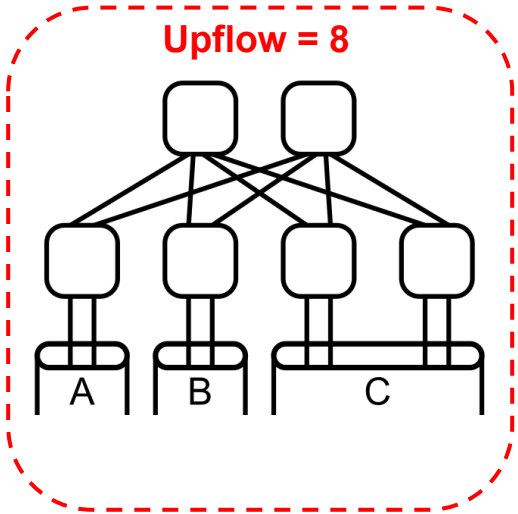
$$\max_{T \in \mathcal{T}} U(w, T)$$

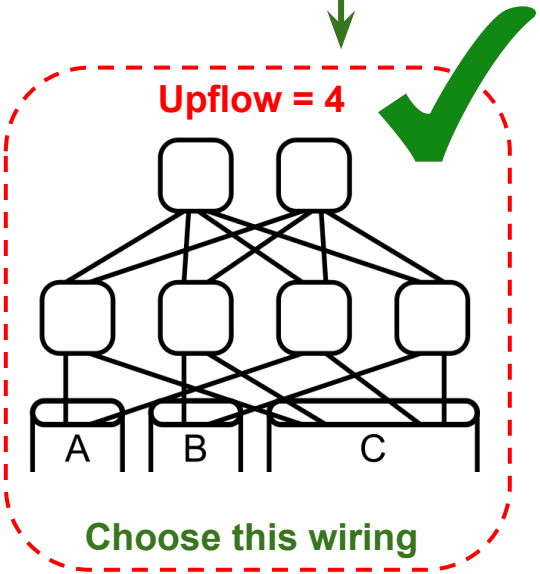




Optimal wiring minimizes the worst-case upflow

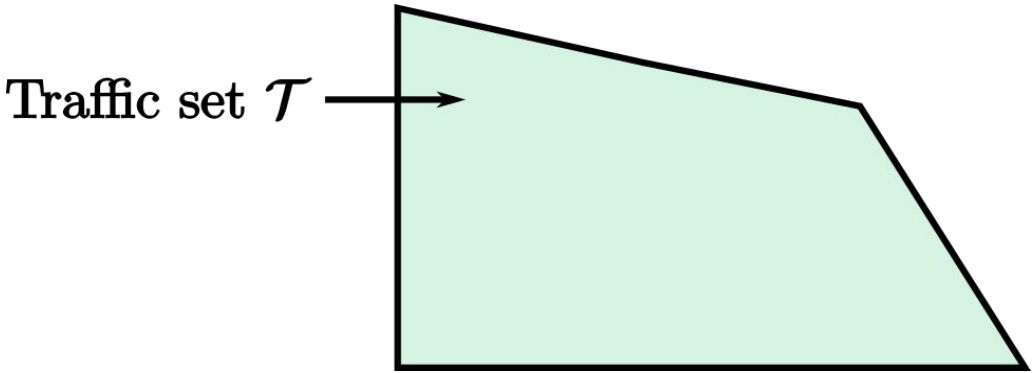
$$\min_{w \in \mathcal{W}} \max_{T \in \mathcal{T}} U(w, T)$$





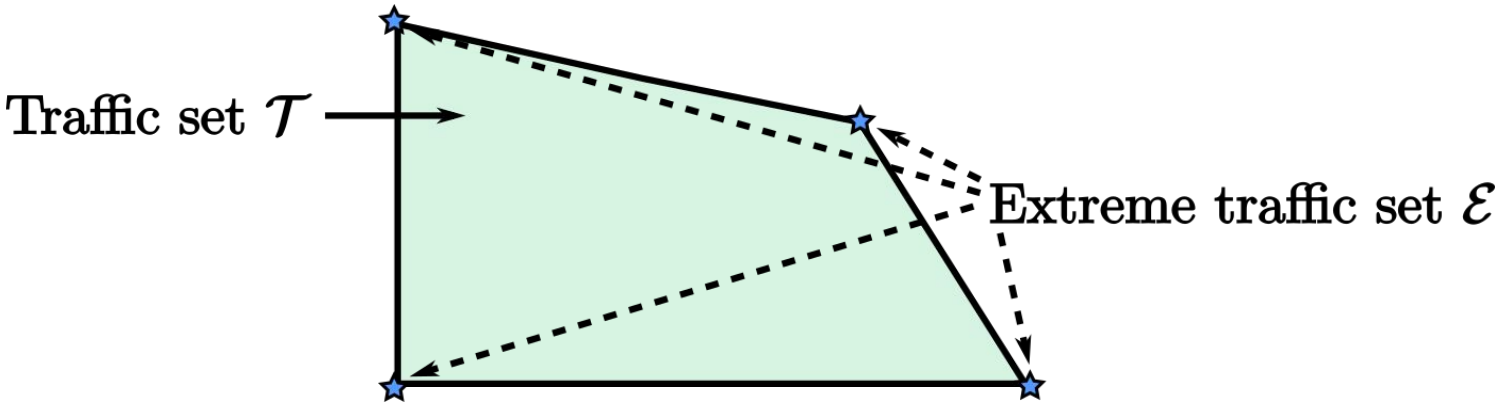
Challenge: There are infinitely many traffic matrices!

$$\min_{w \in \mathcal{W}} \max_{T \in \mathcal{T}} U(w, T)$$



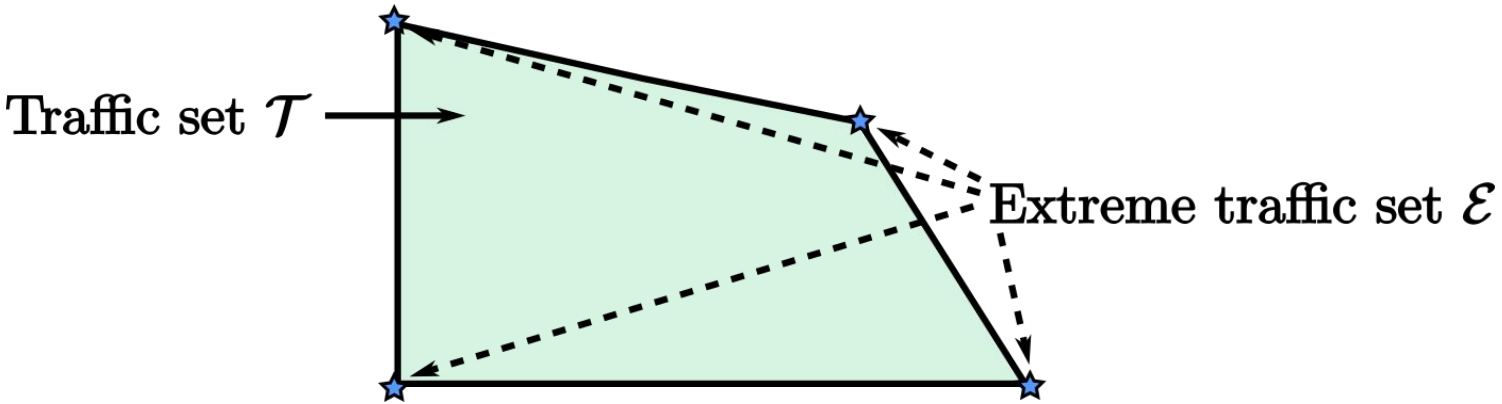
Solution: Extreme traffic matrices are sufficient

$$\min_{w \in \mathcal{W}} \max_{T \in \mathcal{T}} U(w, T)$$

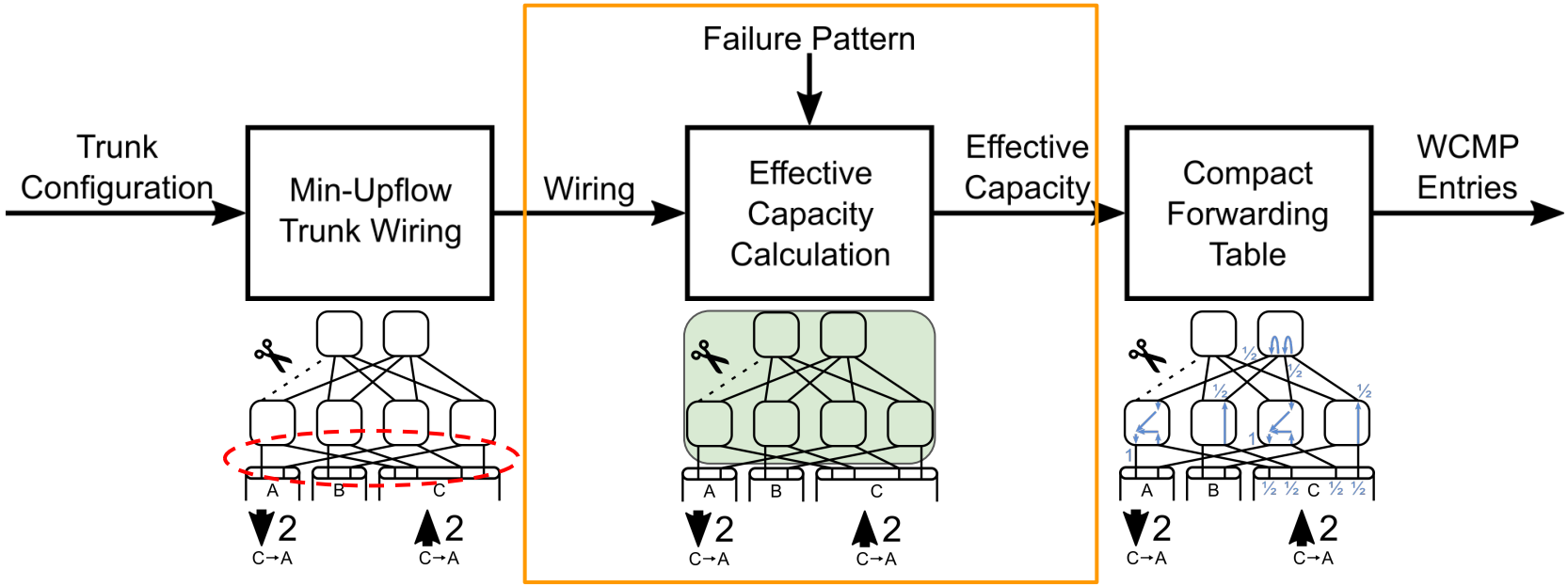


Finding optimal wiring becomes MILP

$$\min_{w \in \mathcal{W}} \max_{T \in \mathcal{T}} U(w, T) = \min_{w \in \mathcal{W}} \max_{T \in \mathcal{E}} U(w, T)$$

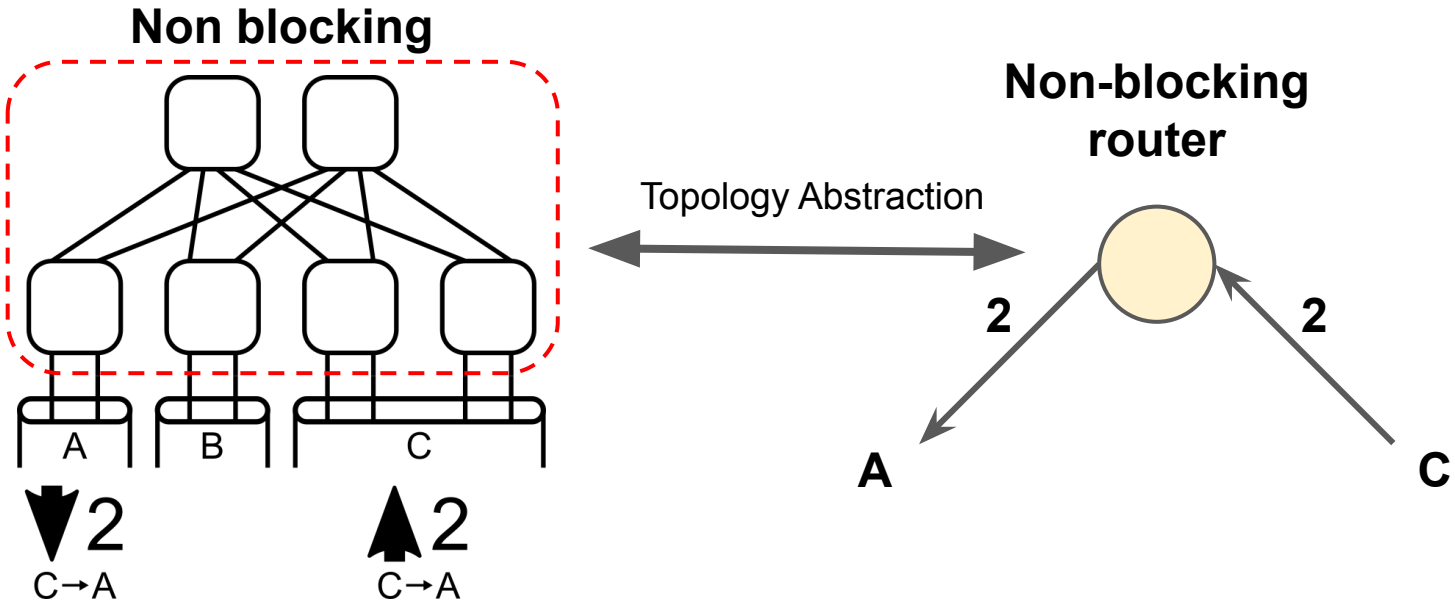


Calculating Effective Capacity



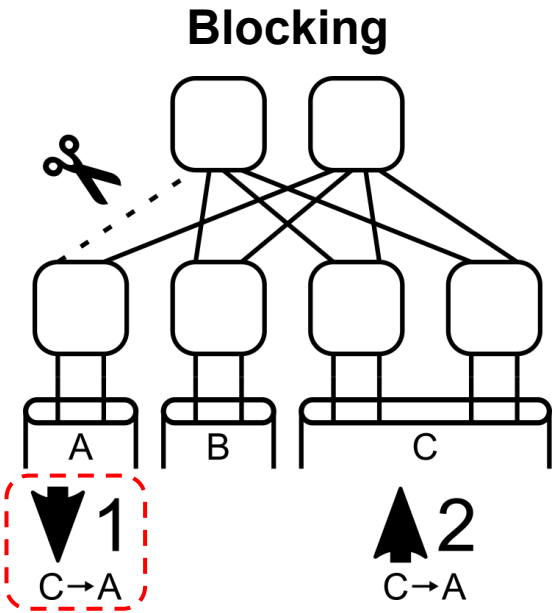
A Non-Blocking Router Allows Topology Abstraction

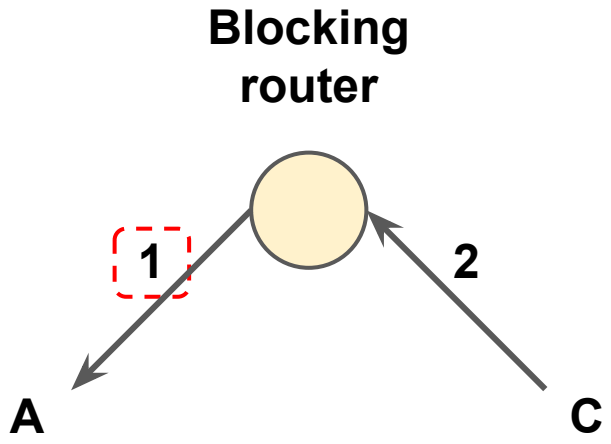
Abstraction simplifies traffic engineering.



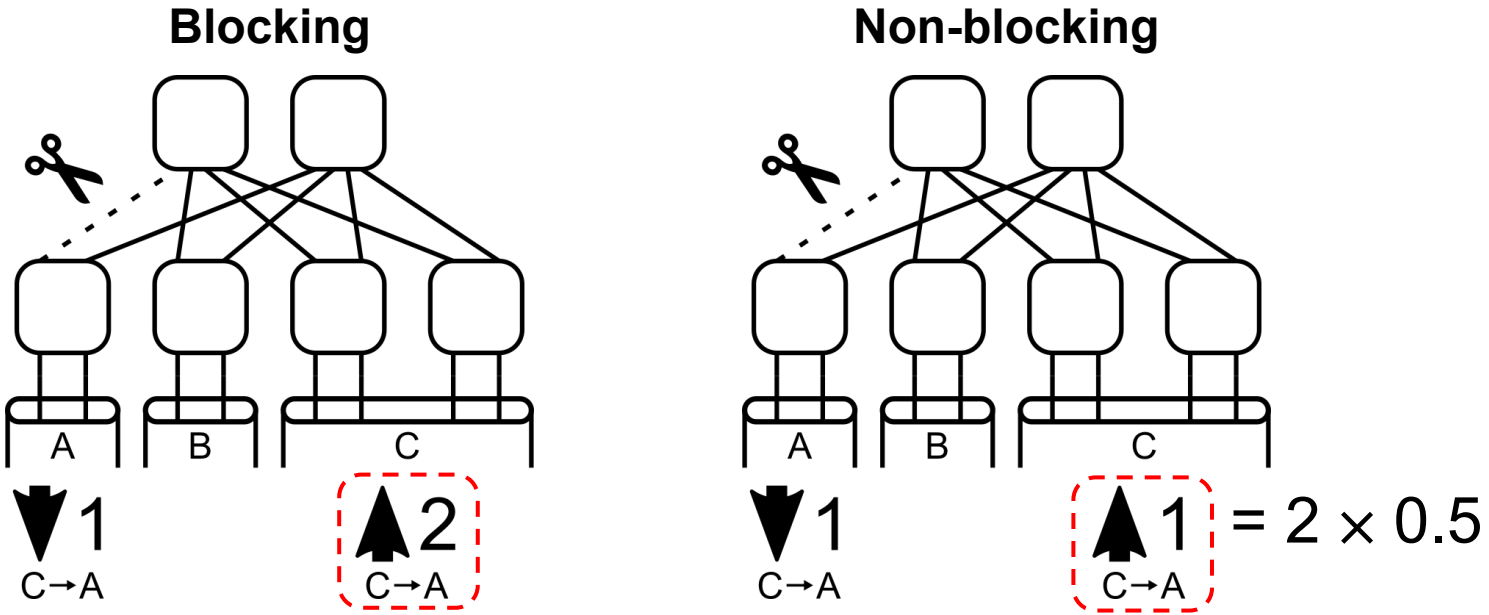
A Blocking router breaks the Topology Abstraction

The router cannot be abstracted by a simple node with flow conservation anymore.



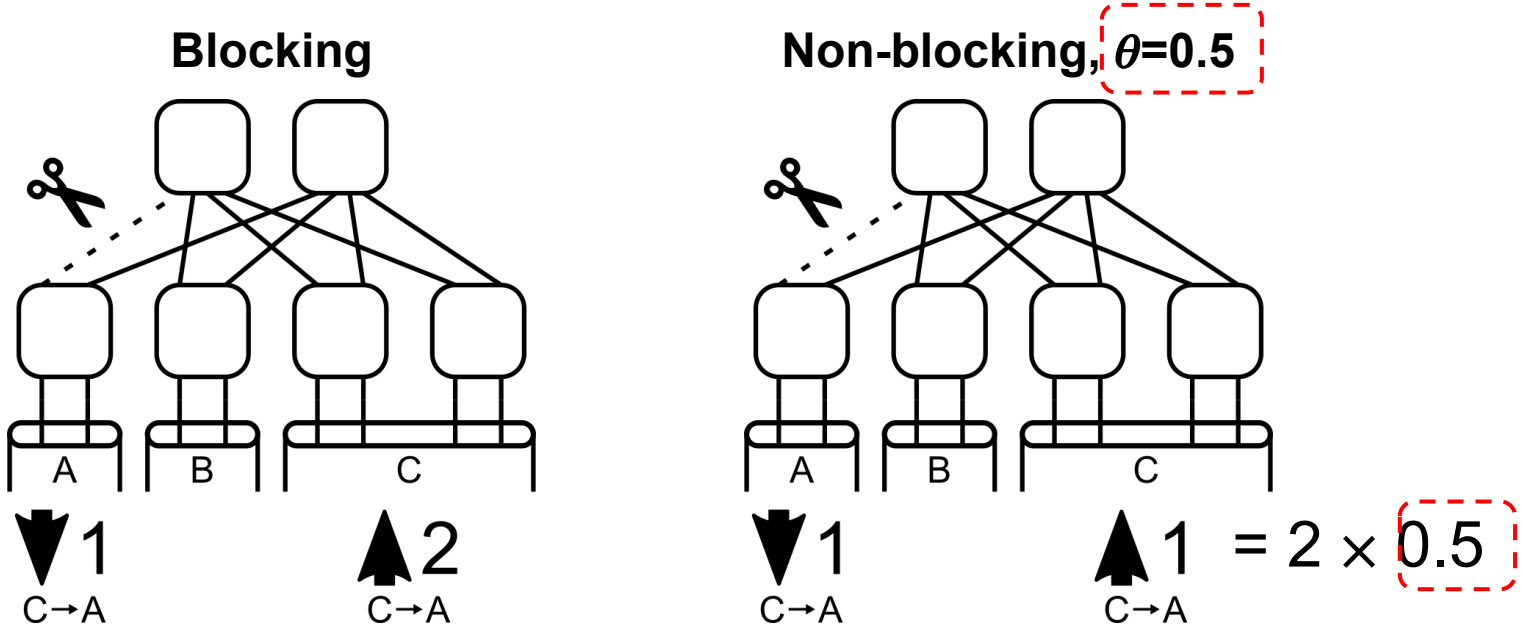


Upon Failure, Scale Demand to Ensure Non-Blocking



Effective Capacity for Non-Blocking Design

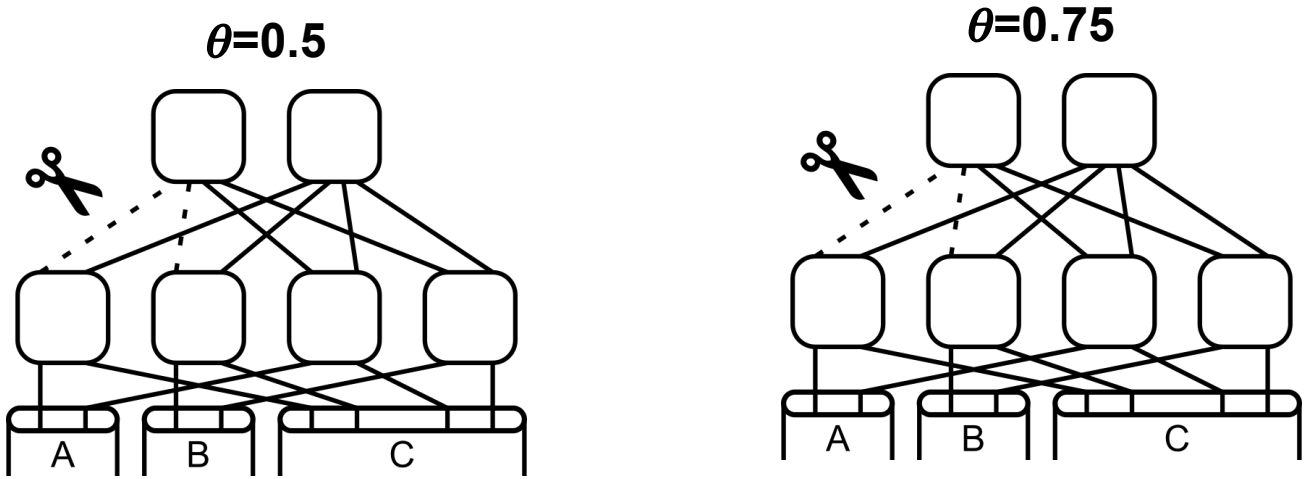
Effective capacity is the largest scaling factor $\theta \in [0, 1]$ that a router is non-blocking under a given failure pattern.



Computing Effective Capacity

Under a failure pattern, finding effective capacity is a linear program per traffic matrix.

$$\max_{\theta \in \Theta(F,T)} \theta$$



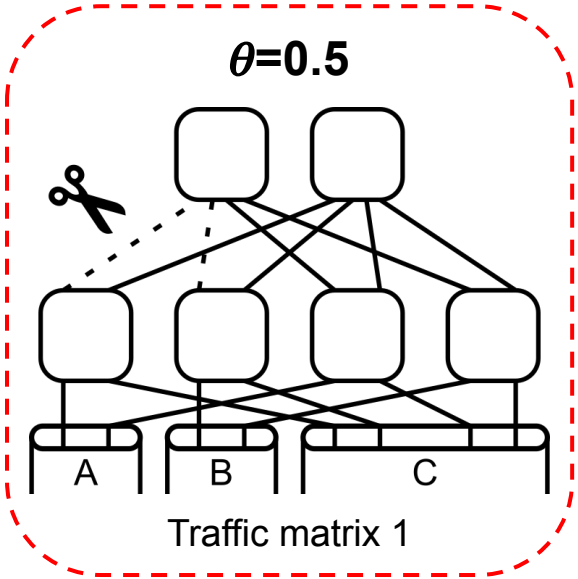
Traffic matrix 1

Traffic matrix 2

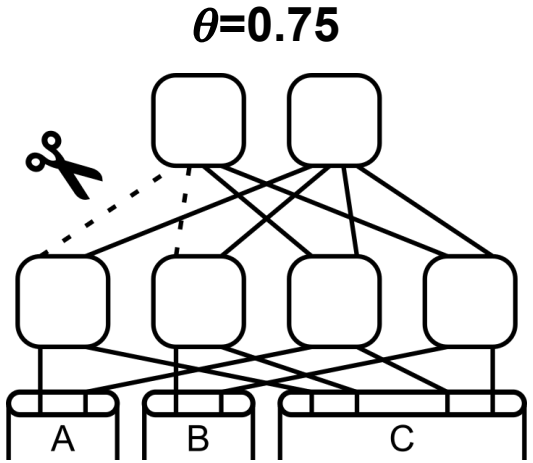
Effective Capacity under Failure and Traffic

The effective capacity is the minimum value.

$$\min_{T \in \mathcal{T}} \max_{\theta \in \Theta(F,T)} \theta = \min_{T \in \mathcal{E}} \max_{\theta \in \Theta(F,T)} \theta$$

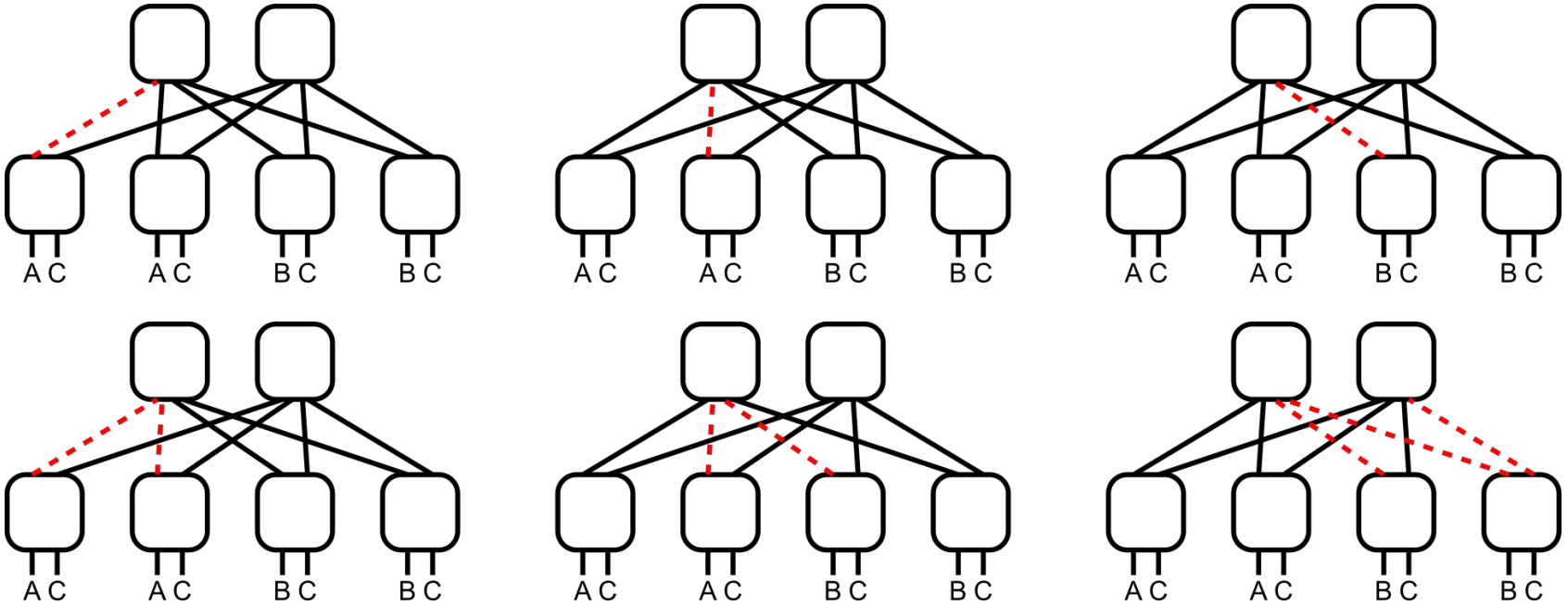


Traffic matrix 1

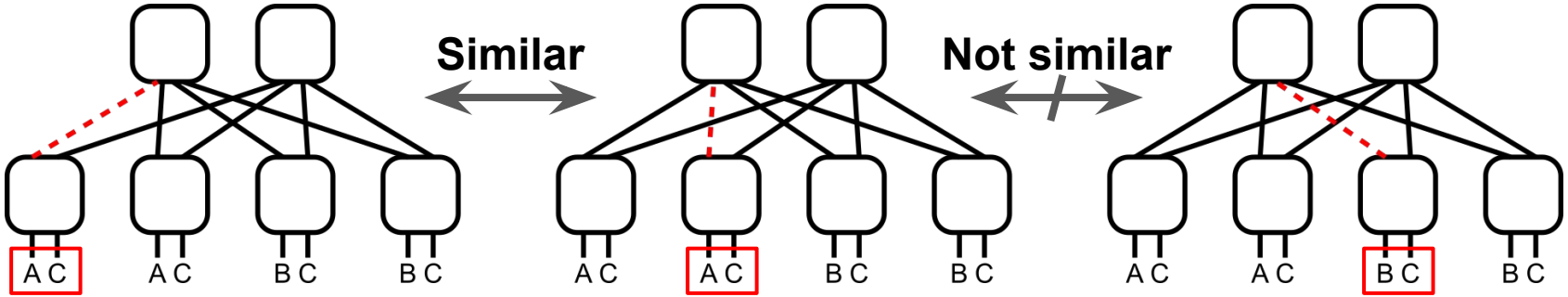


Traffic matrix 2

Challenge: Exponential Number of Failure Patterns

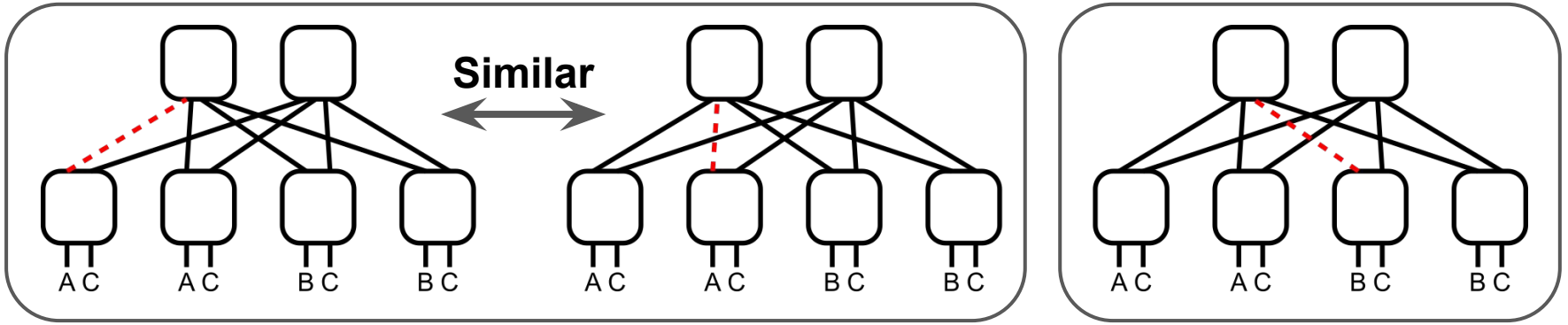


Challenge: Exponential Number of Failure Patterns



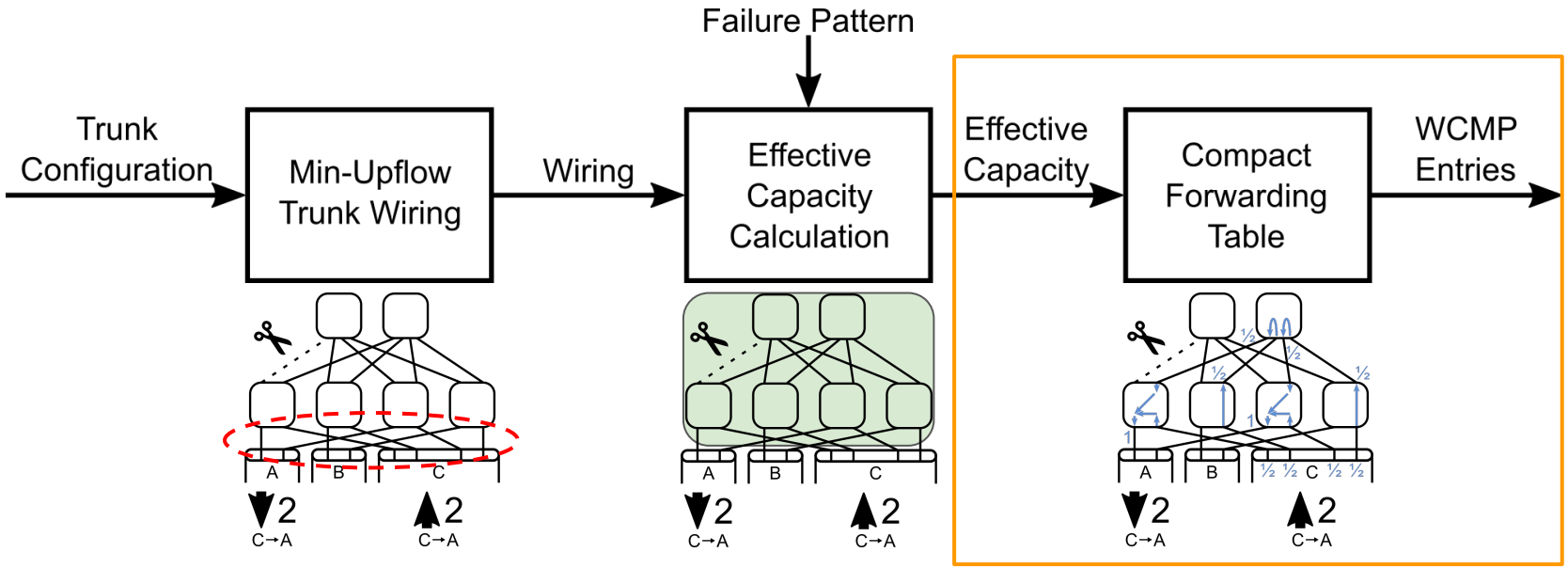
Challenge: Exponential Number of Failure Patterns

Solution: Group similar failure patterns using a graph canonicalization algorithm



Calculate effective capacity for each canonical pattern

Compacting Routing Table



Please see this part in the paper.

Evaluation

Resilience of 128-port router

Comparison to alternative strategies

Scalability to 512-port router

Routing table sizes

Impact of optimizations

Evaluation

Resilience of 128-port router

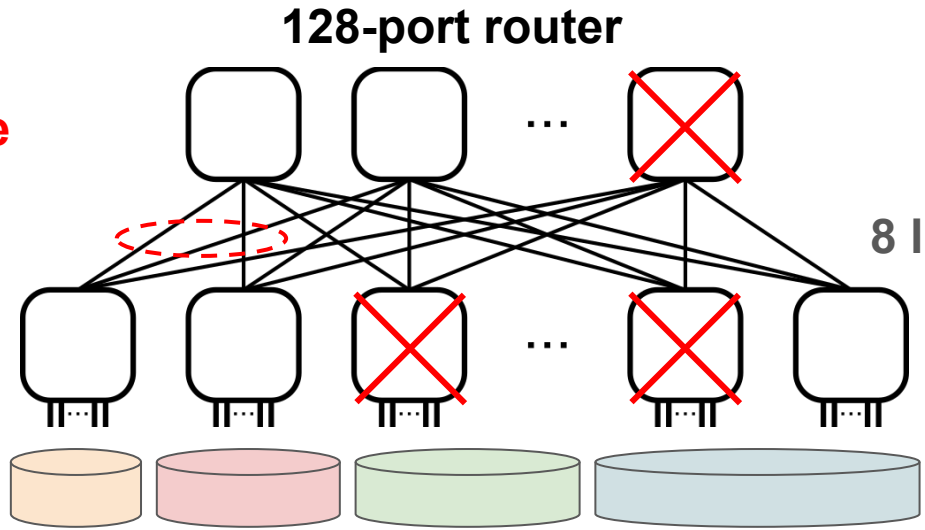
Comparison to alternative strategies

Scalability to 512-port router

Routing table sizes

Impact of optimizations

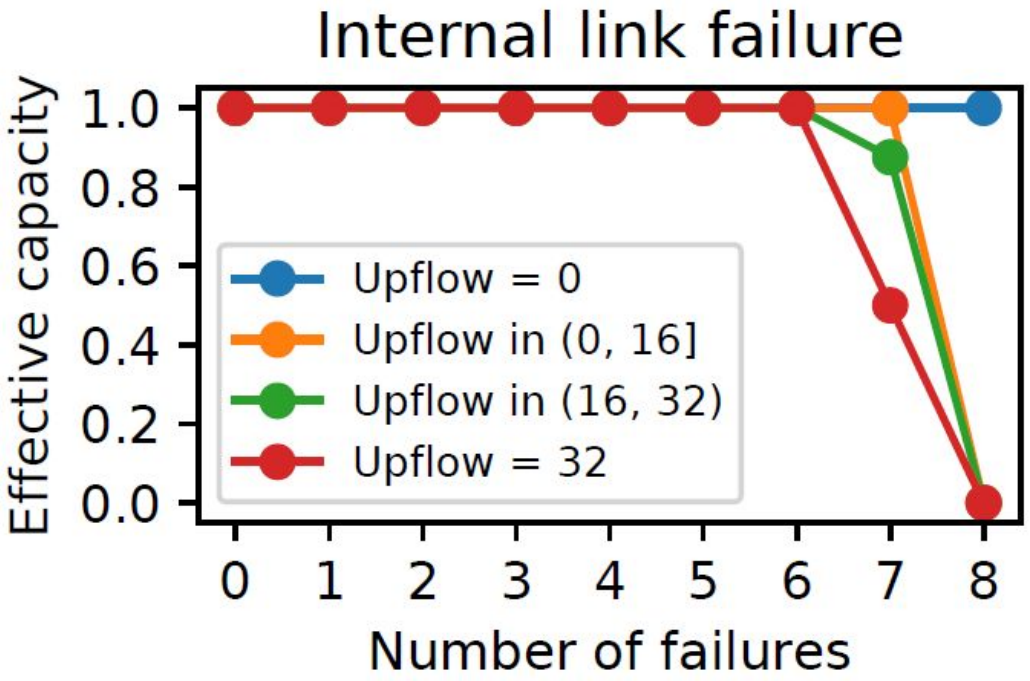
Methodology

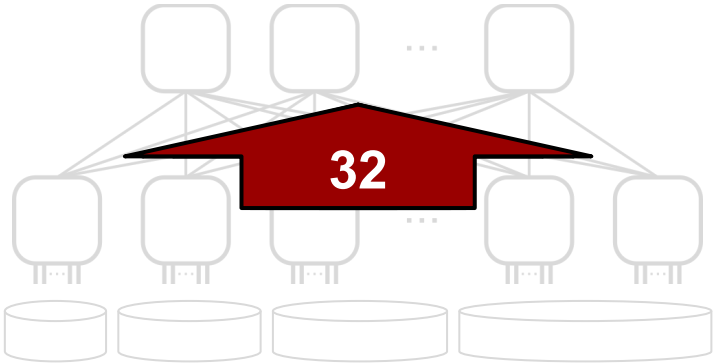


We enumerate all multiple-of-8 trunk sizes. (34 combinations)

We compute the effective capacity under all possible failure conditions.

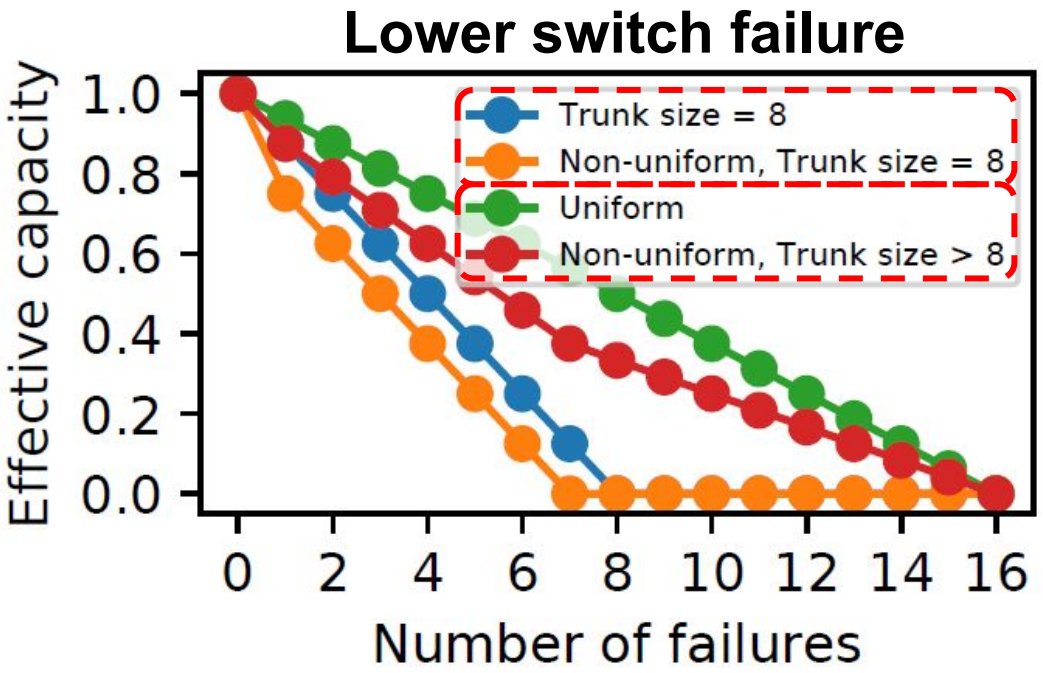
Effective Capacity - Link Failure: 128-Port Router





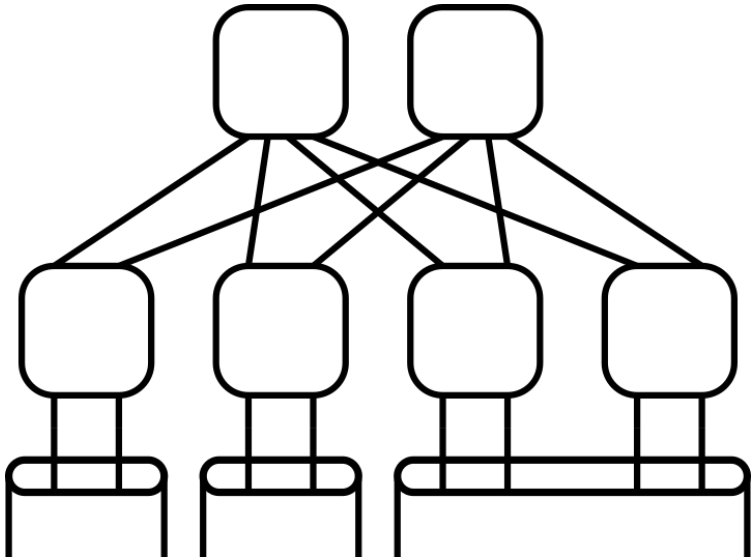
Our approach can mask up to 6 concurrent link failures.

Lower Switch Failure: 128-Port Router

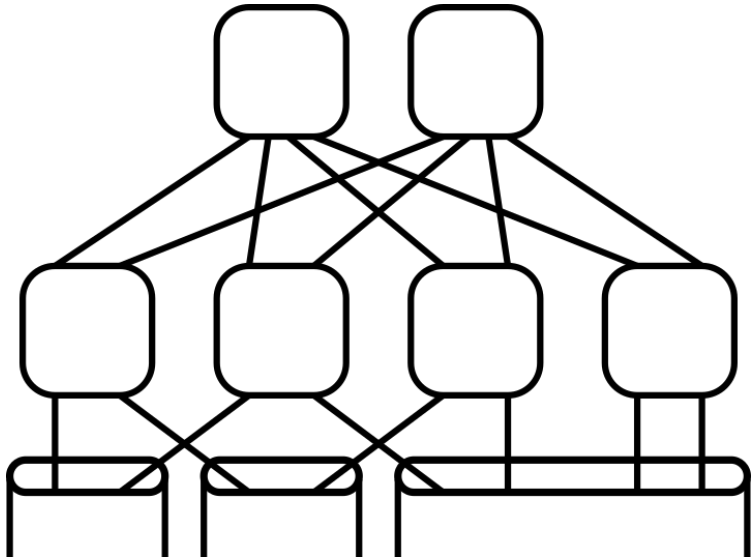


Capacity degrades gracefully.

Comparison to Alternative Wiring Strategies



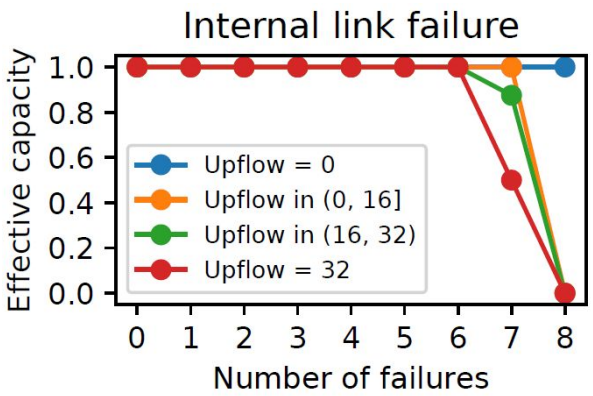
Baseline Wiring



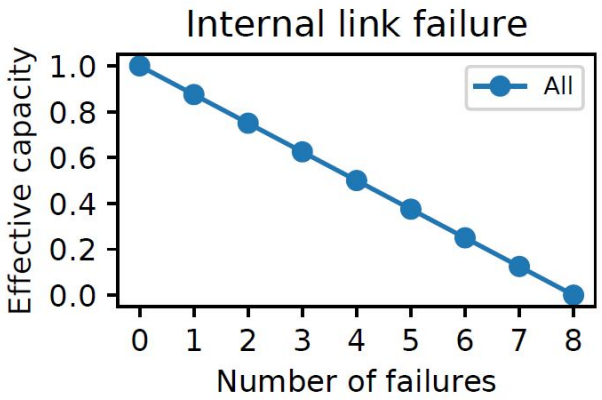
Random Wiring

Minimal-Upflow Wiring Yields Superior Resilience

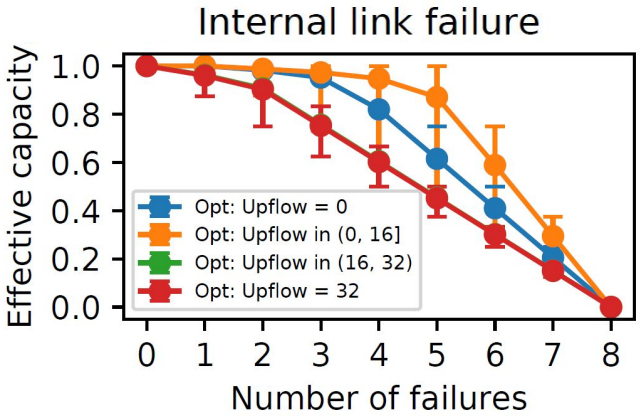
Minimal-Upflow Wiring



Baseline Wiring



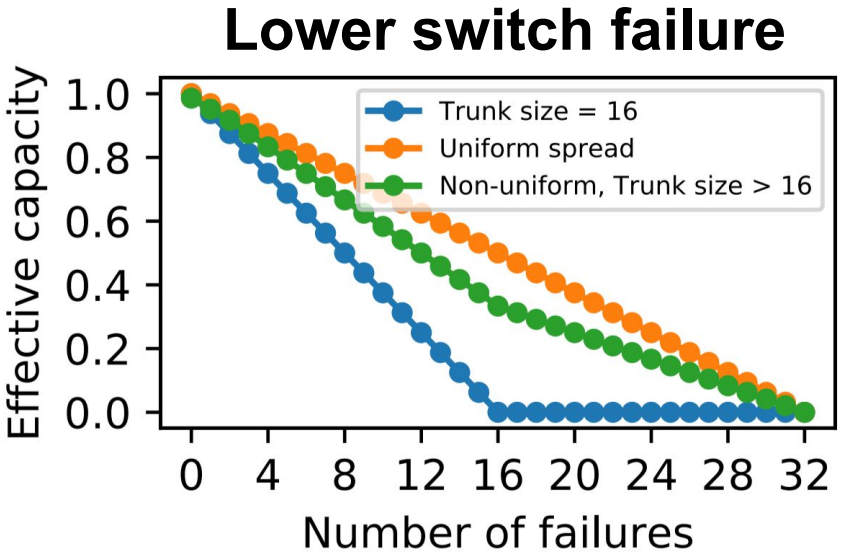
Random Wiring

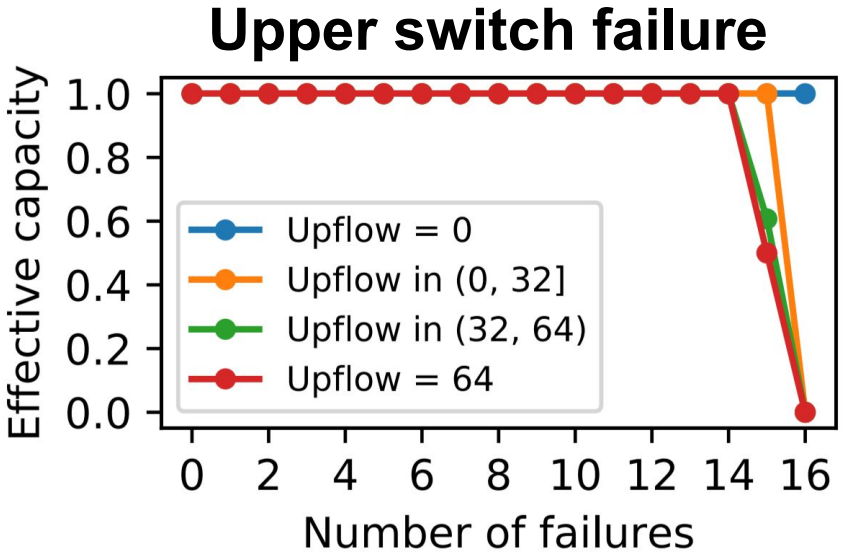


No other approach can mask even a single link failure

Scalability: 512-Port Router

The pipeline can scale to the 512-port router.



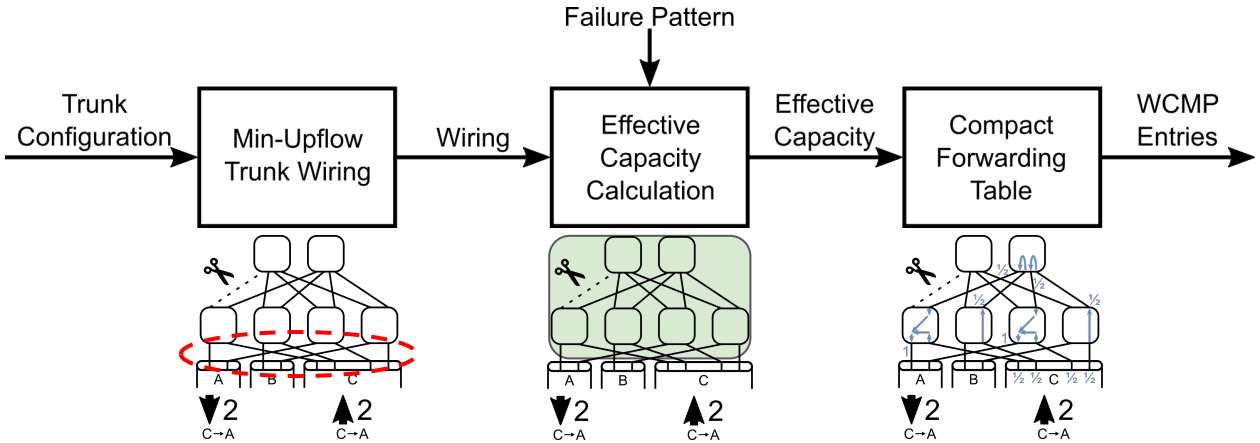


Conclusion

Min-upflow wiring and early forwarding can mask significant number of failures.

It improves the availability of WAN routers.

It can be used to reduce the cost of WAN routers.



https://github.com/USC-NSL/Highly-Available-WAN-Router

(Available Oct. 2019)