SMORE: Semi-Oblivious Traffic Engineering

Praveen Kumar* Yang Yuan* Chris Yu‡ Nate Foster*
Robert Kleinberg* Petr Lapukhov# Chiun Lin Lim# Robert Soulé §

* Cornell ‡ CMU # Facebook § USI Lugano

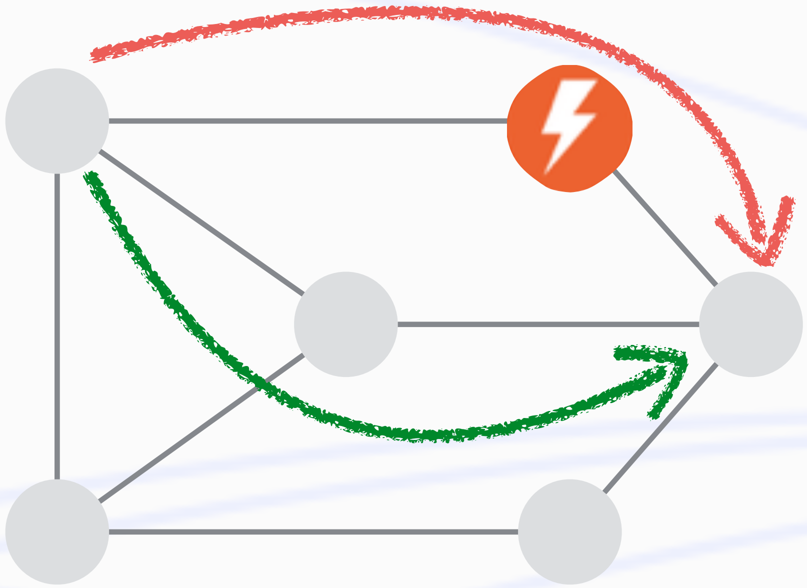
ANRW 2018

WAN Traffic Engineering

WAN Traffic Engineering

Objectives

Performance



Robustness

Latency

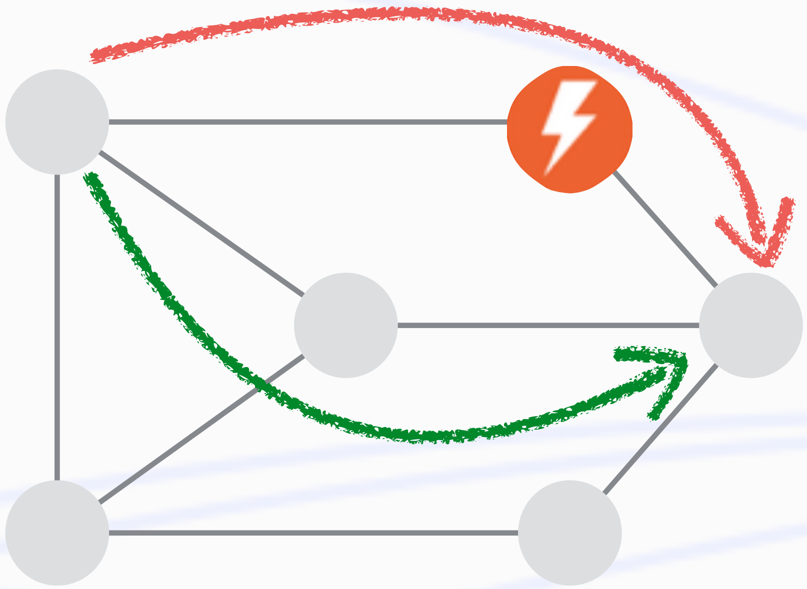
Operational simplicity

Challenges

WAN Traffic Engineering

Objectives

Performance



Robustness

Latency

Operational simplicity

Challenges

Unstructured topology

Heterogeneous capacity

Unexpected failures

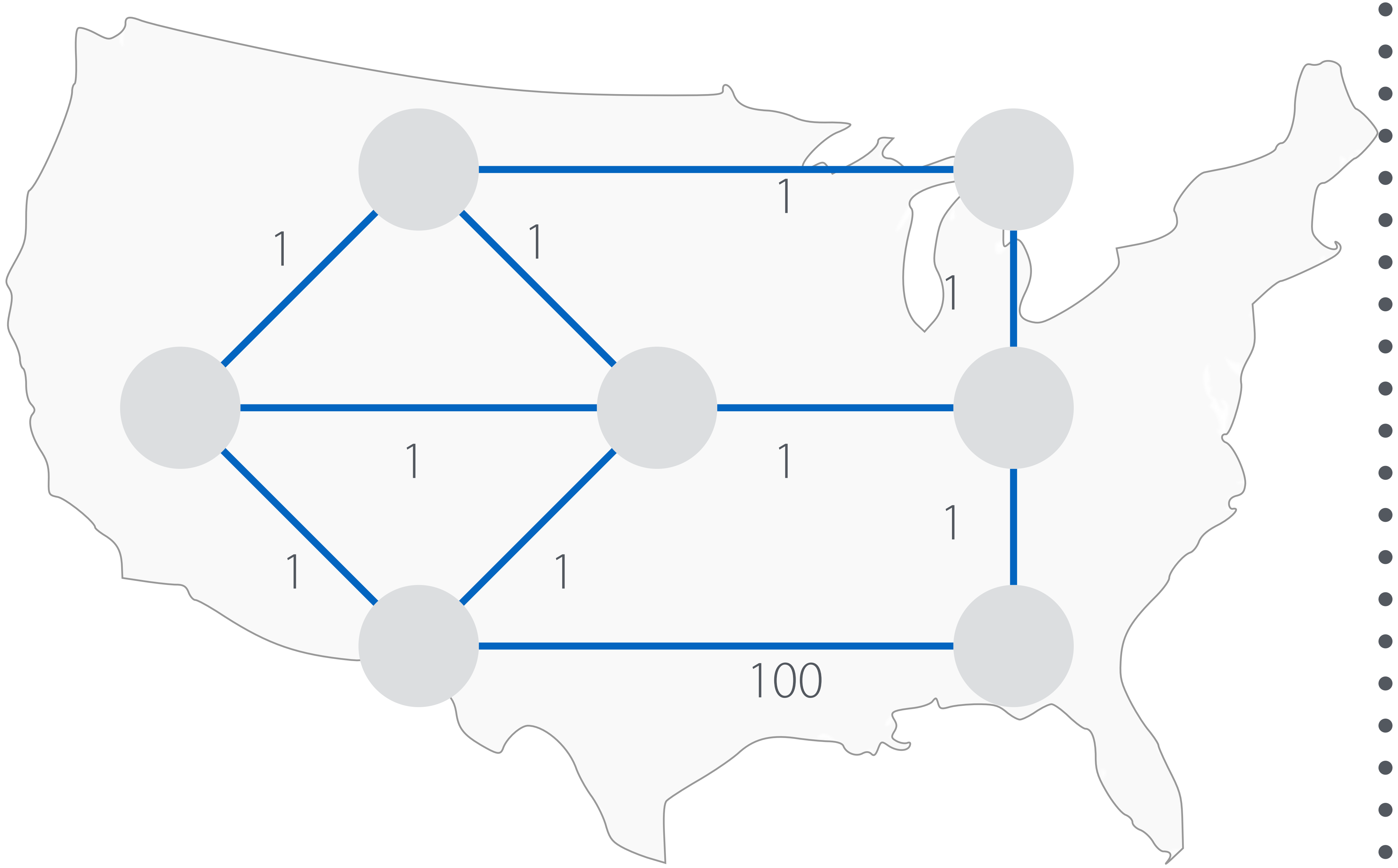
Misprediction & Traffic Bursts

Device limitations

Update overheads

TE Approaches

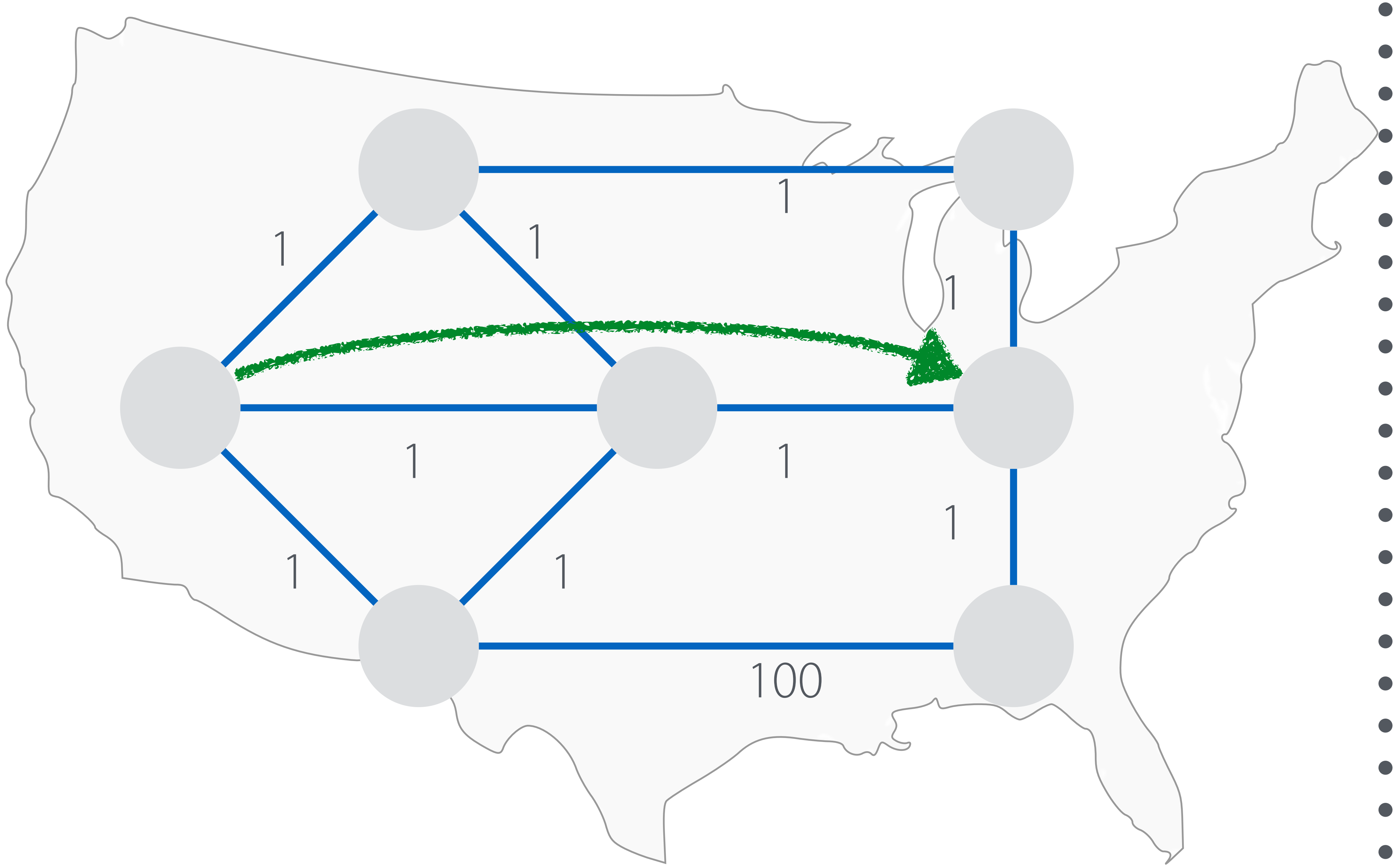
Traditional
Distributed



SDN-Based
Centralized

TE Approaches

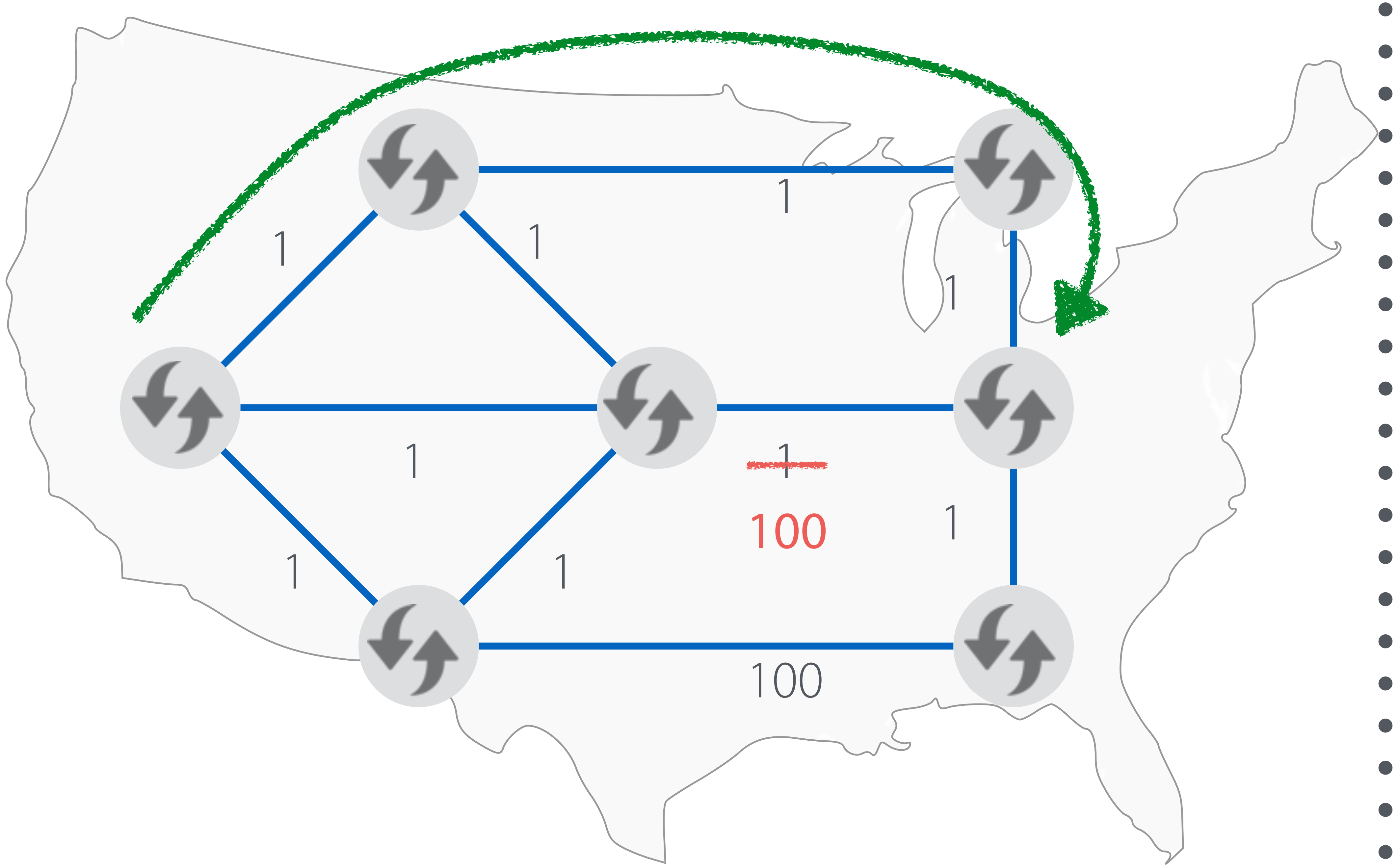
Traditional Distributed



SDN-Based Centralized

TE Approaches

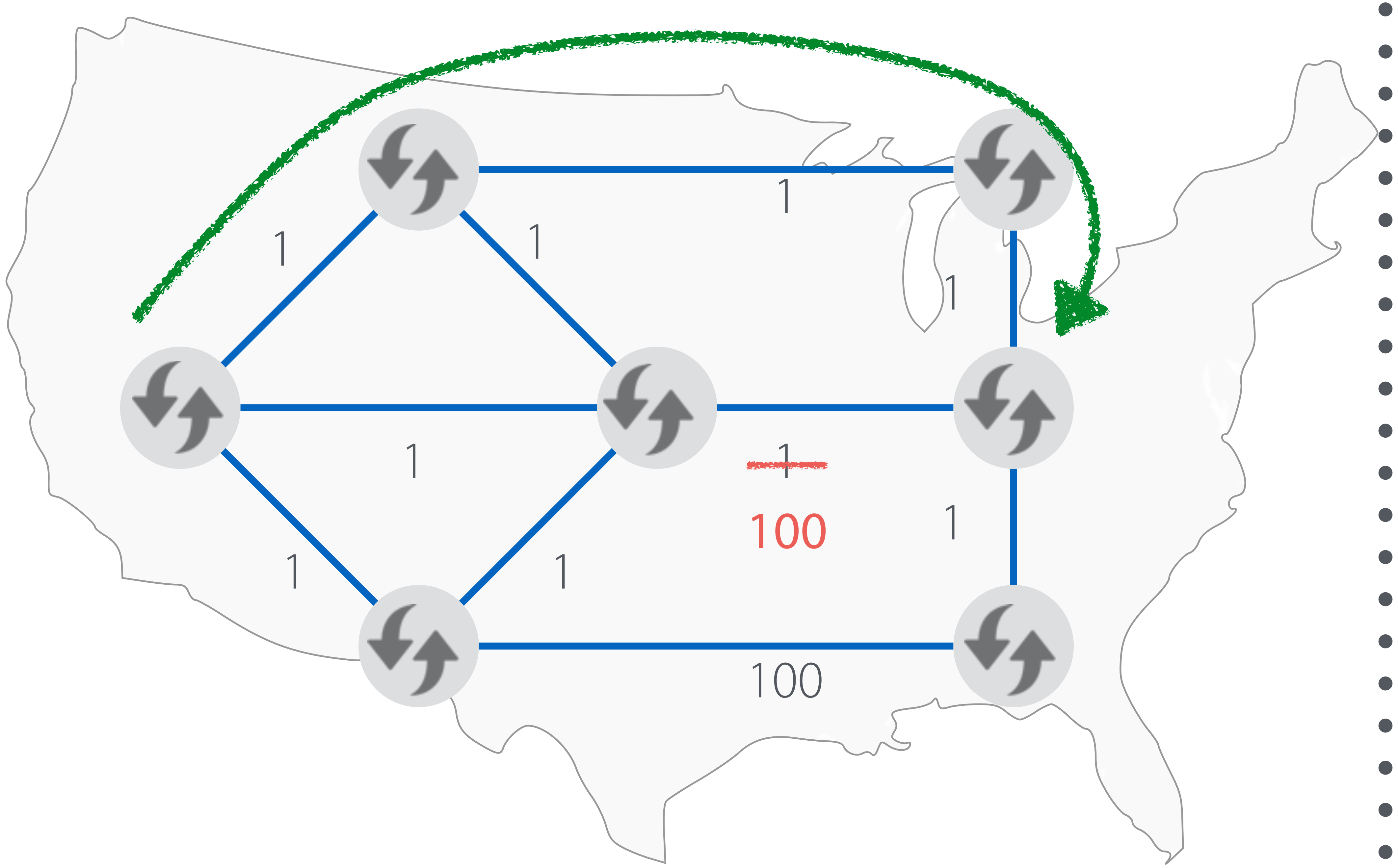
Traditional Distributed



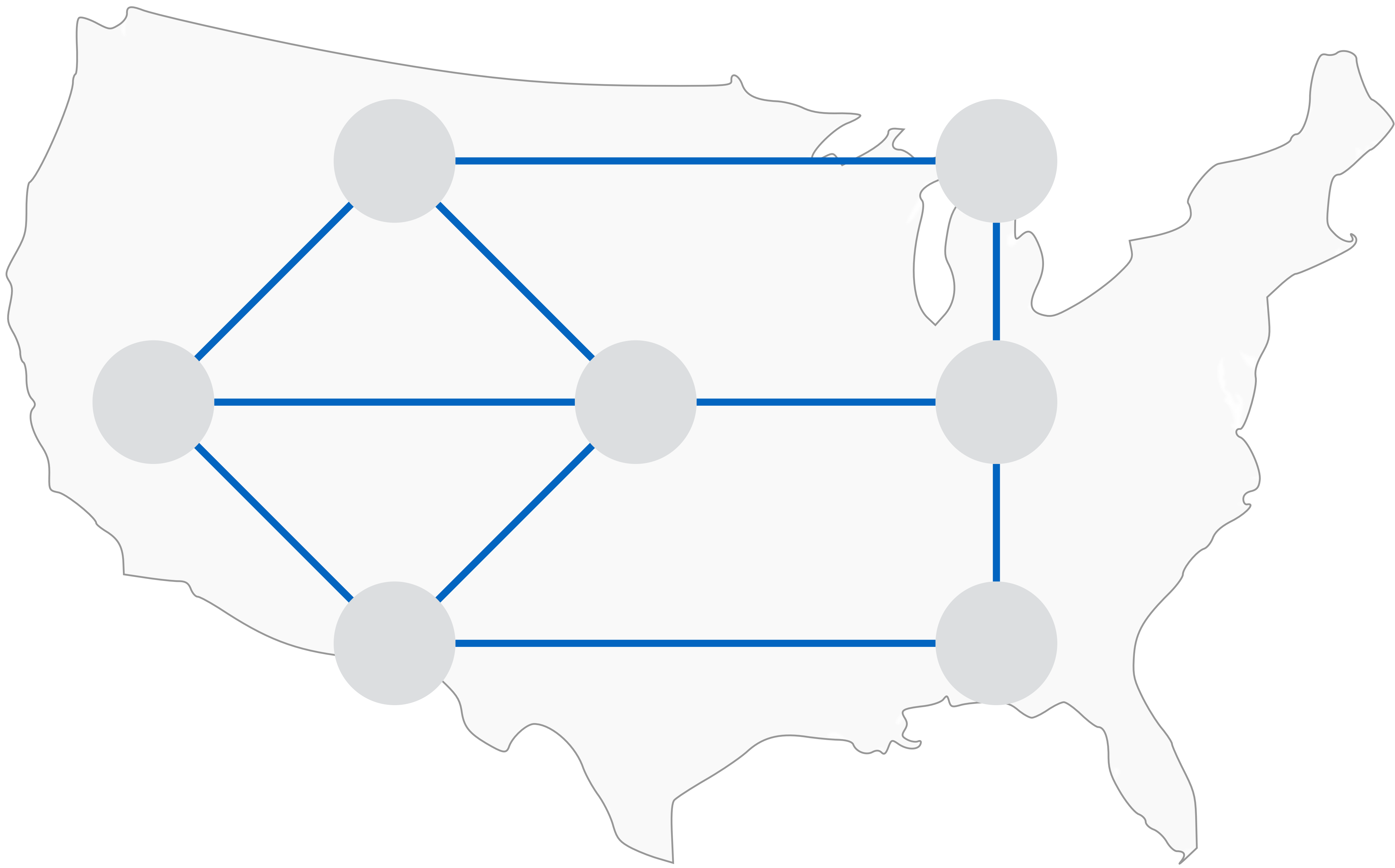
SDN-Based Centralized

TE Approaches

Traditional Distributed

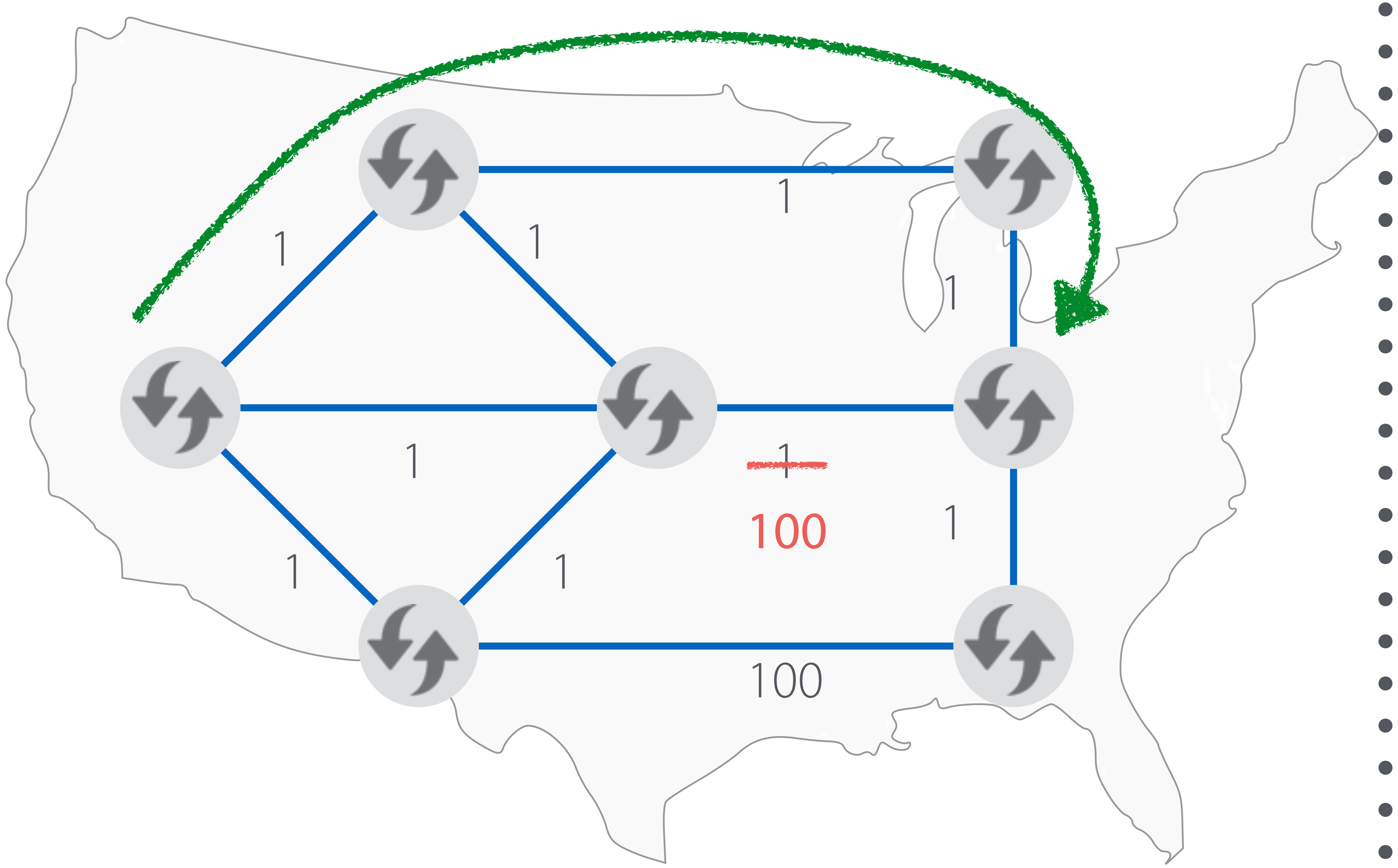


SDN-Based Centralized

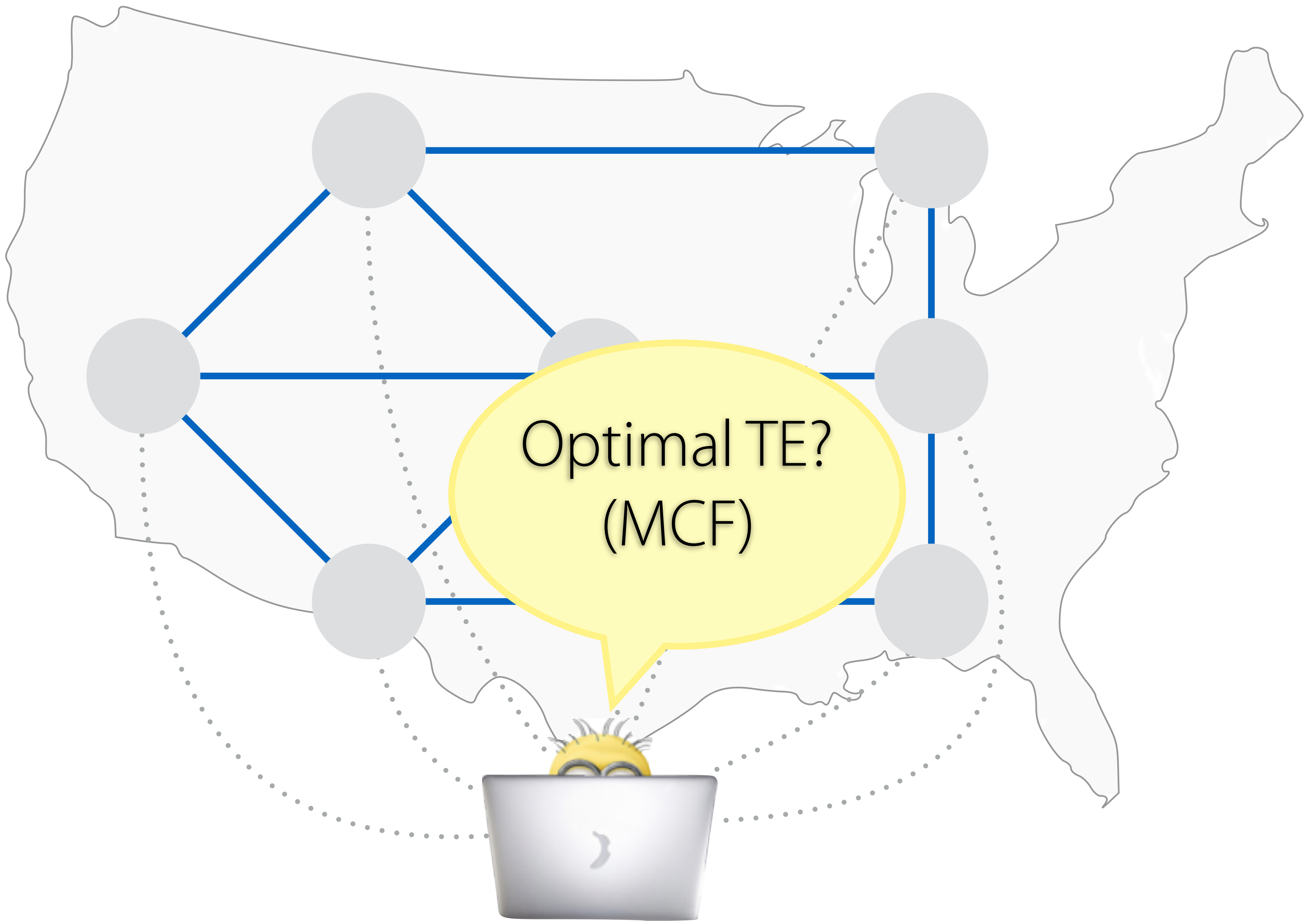


TE Approaches

Traditional Distributed

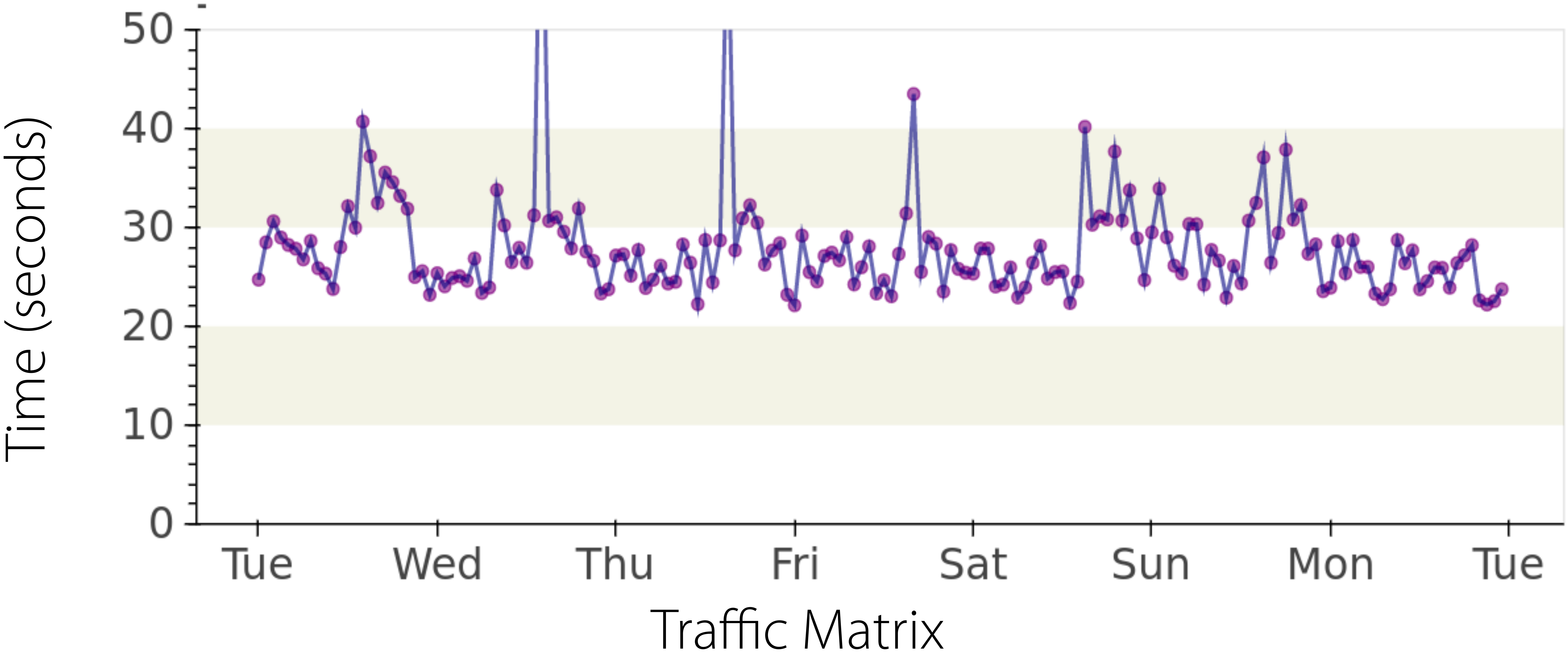


SDN-Based Centralized



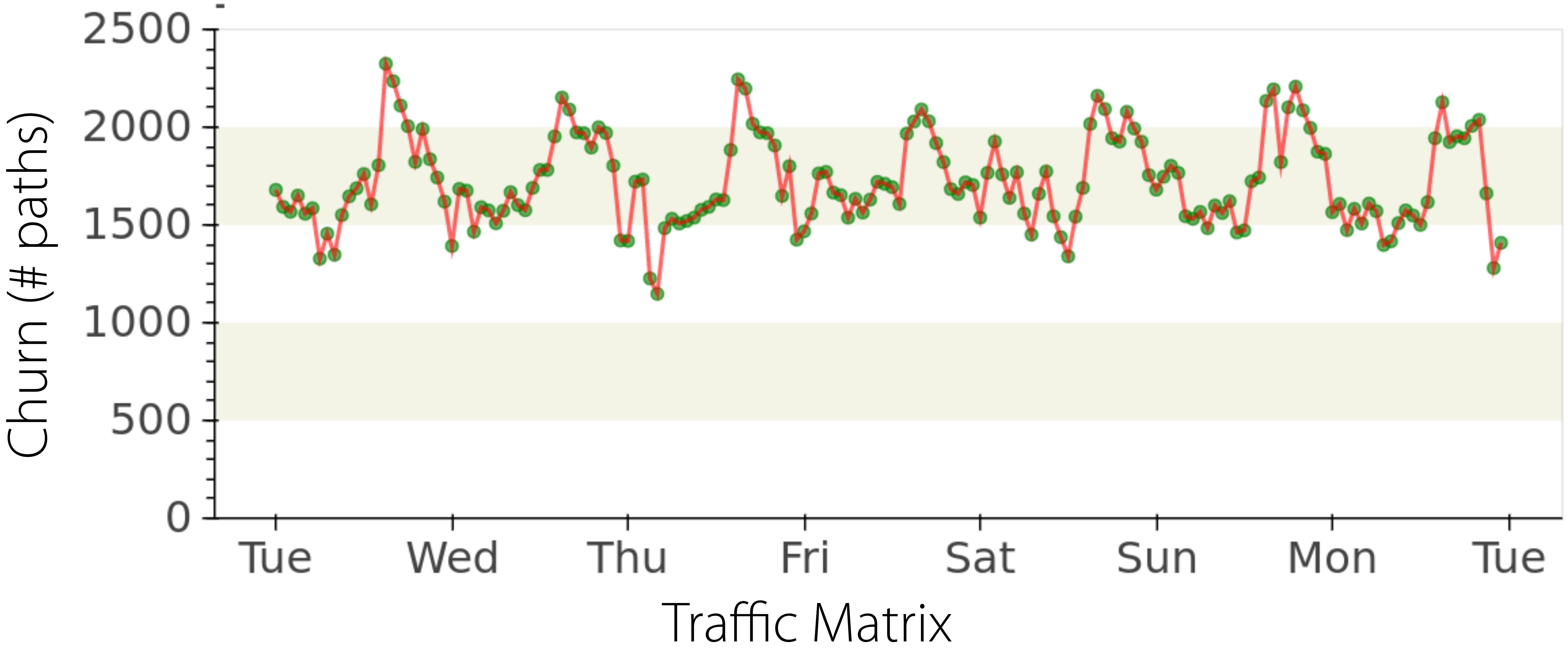
Operational Cost of Optimality

Solver Time



Operational Cost of Optimality

Path Churn



Towards a Practical Model

Topology
(+ demands)

Path Selection

1

Paths

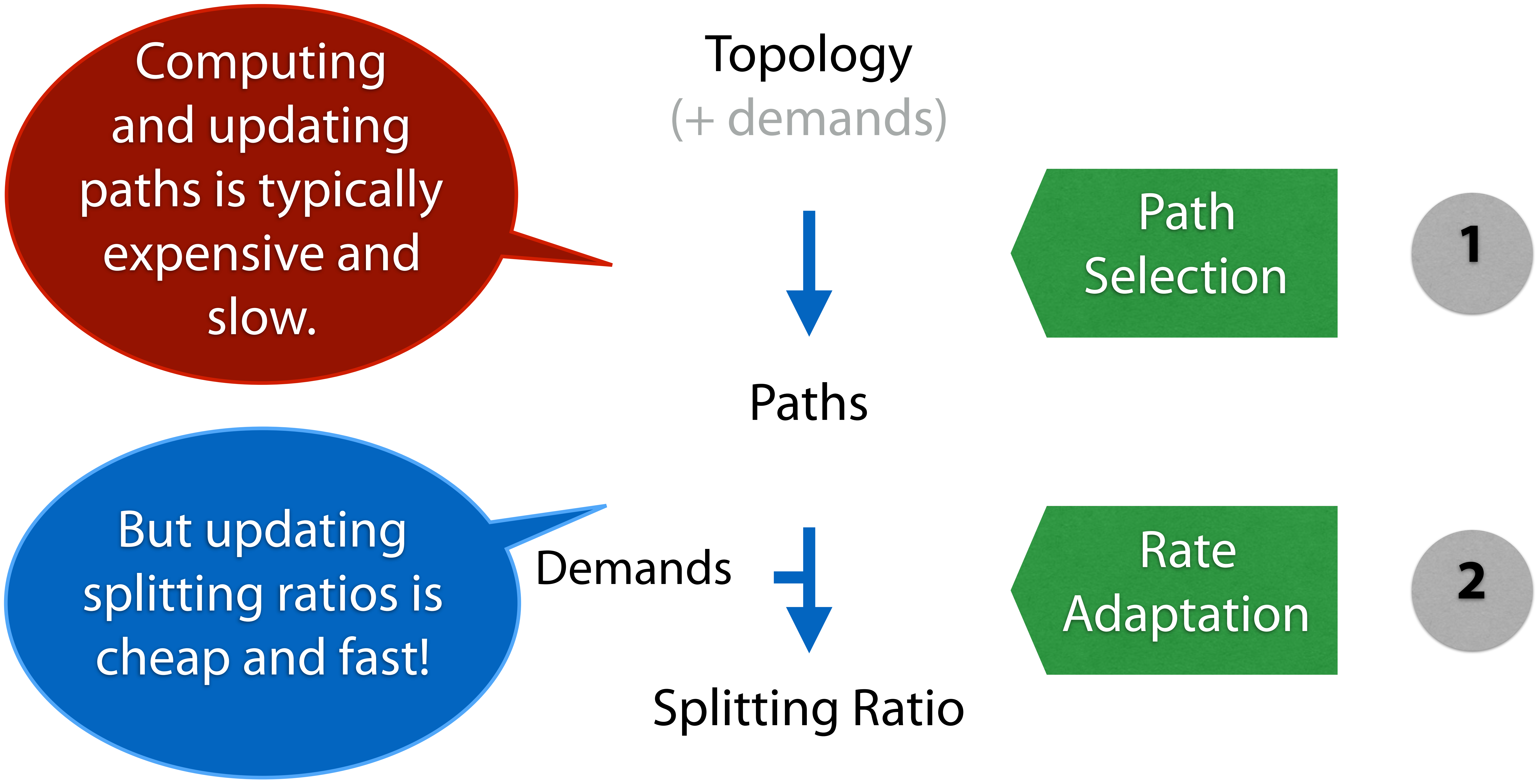
Demands

Rate Adaptation

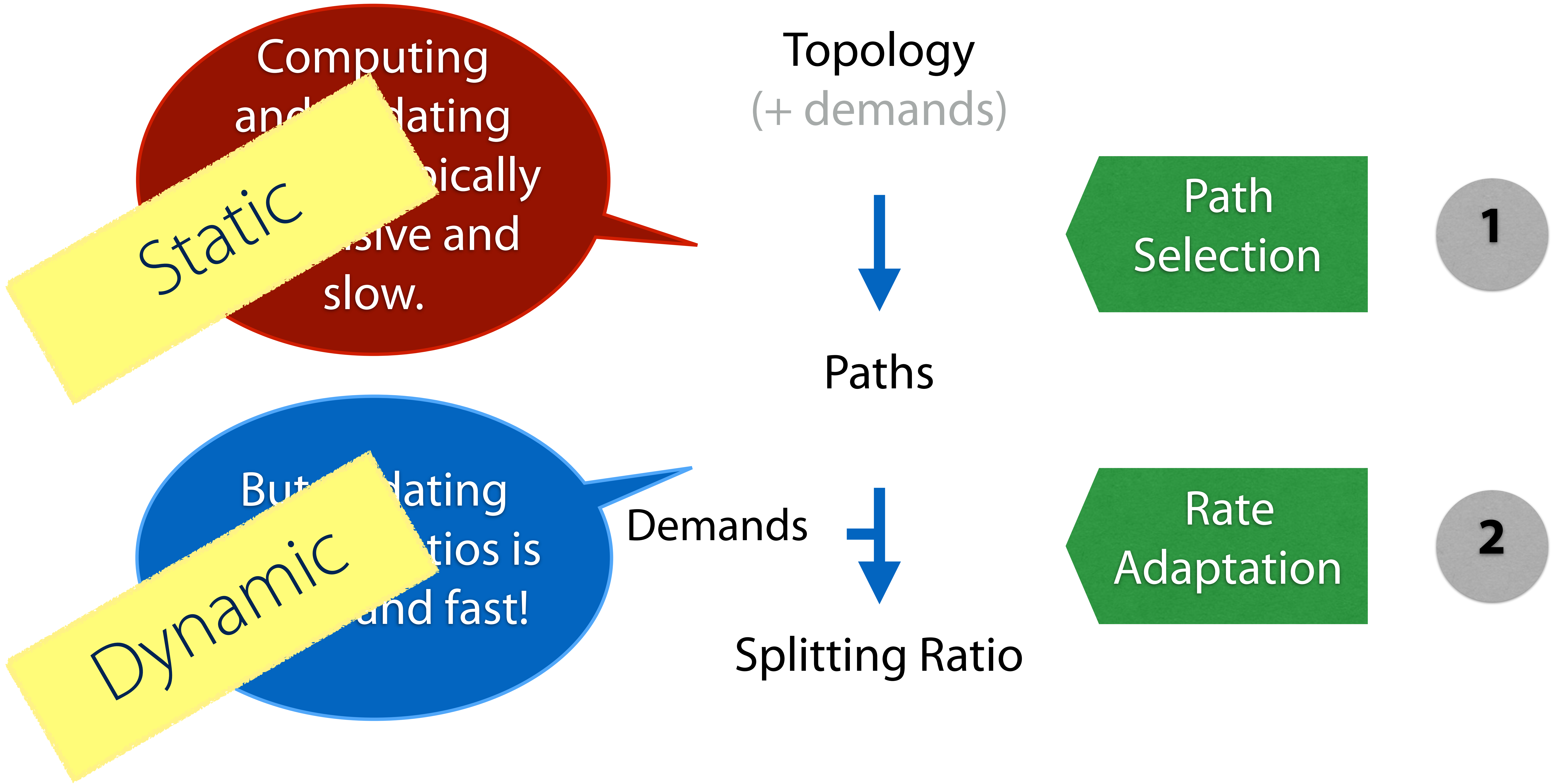
2

Splitting Ratio

Towards a Practical Model



Towards a Practical Model



Path Selection Challenges

- Selecting a good set of paths is tricky!
  - **Route** the demands (ideally, with competitive **latency**)
  - React to **changes in demands** (diurnal changes, traffic bursts, etc.)
  - Be robust under **mis-prediction** of demands
  - Have sufficient extra capacity to route demands in presence of **failures**
  - and more …

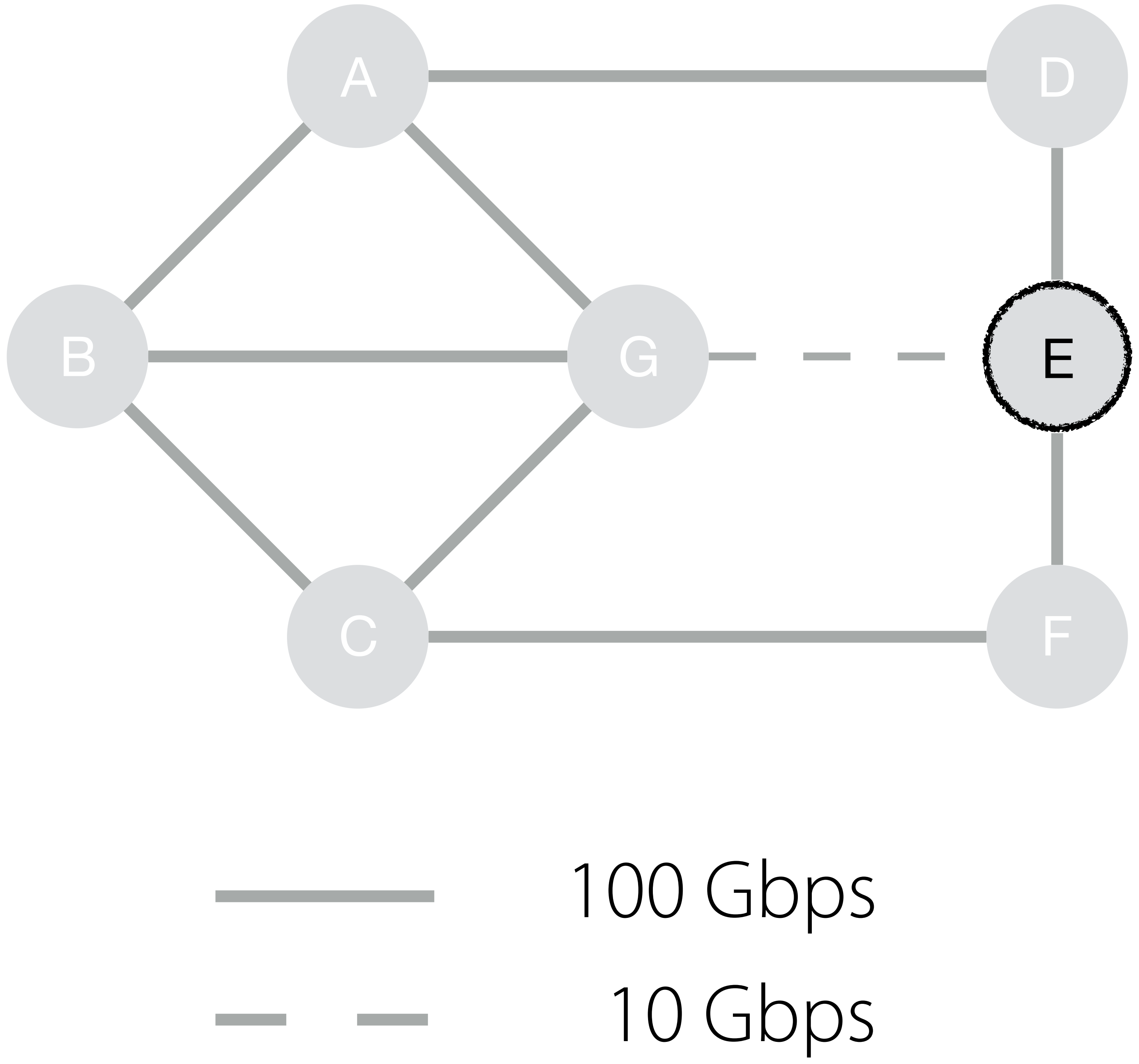
Approach

A static set of cleverly-constructed paths can provide near-optimal performance and robustness!

Desired path properties:

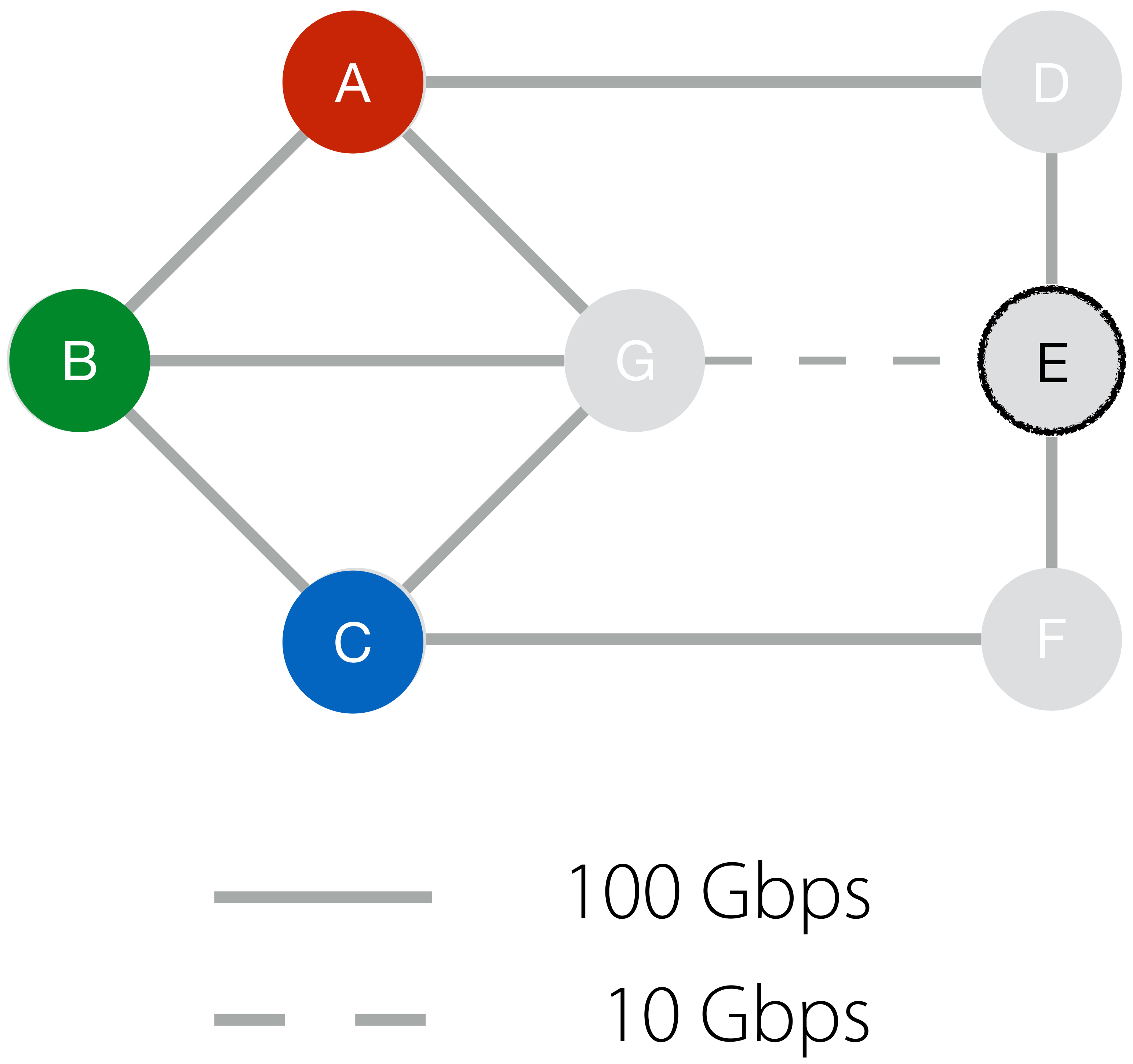
- ***Low stretch*** for minimizing latency
- ***High diversity*** for ensuring robustness
- ***Good load balancing*** for performance
  - Capacity aware
  - Globally optimized

Path Properties: Capacity Aware



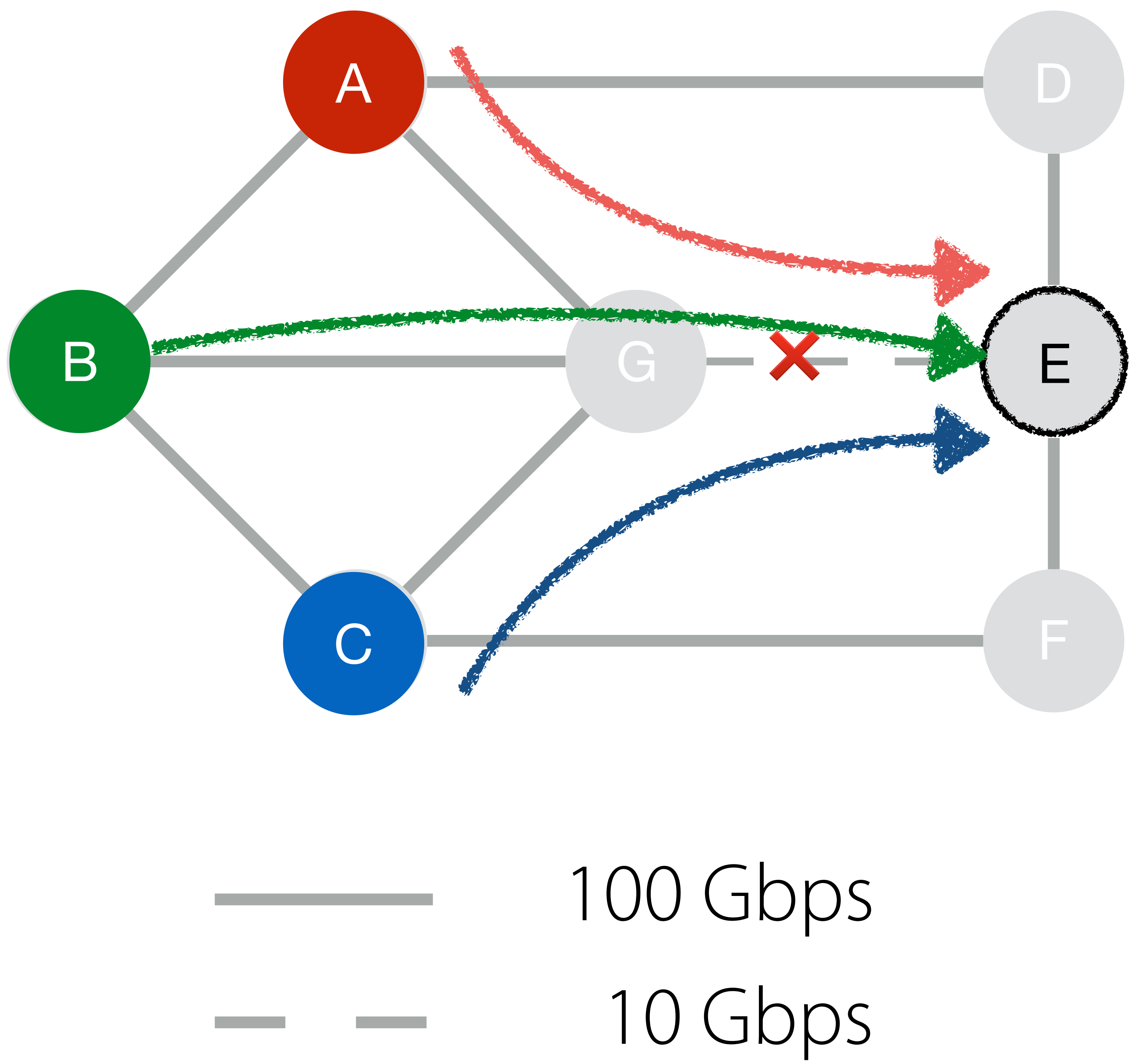
- Traditional approaches to routing based on shortest paths (e.g., ECMP, KSP) are generally not capacity aware

Path Properties: Capacity Aware



- Traditional approaches to routing based on shortest paths (e.g., ECMP, KSP) are generally not capacity aware

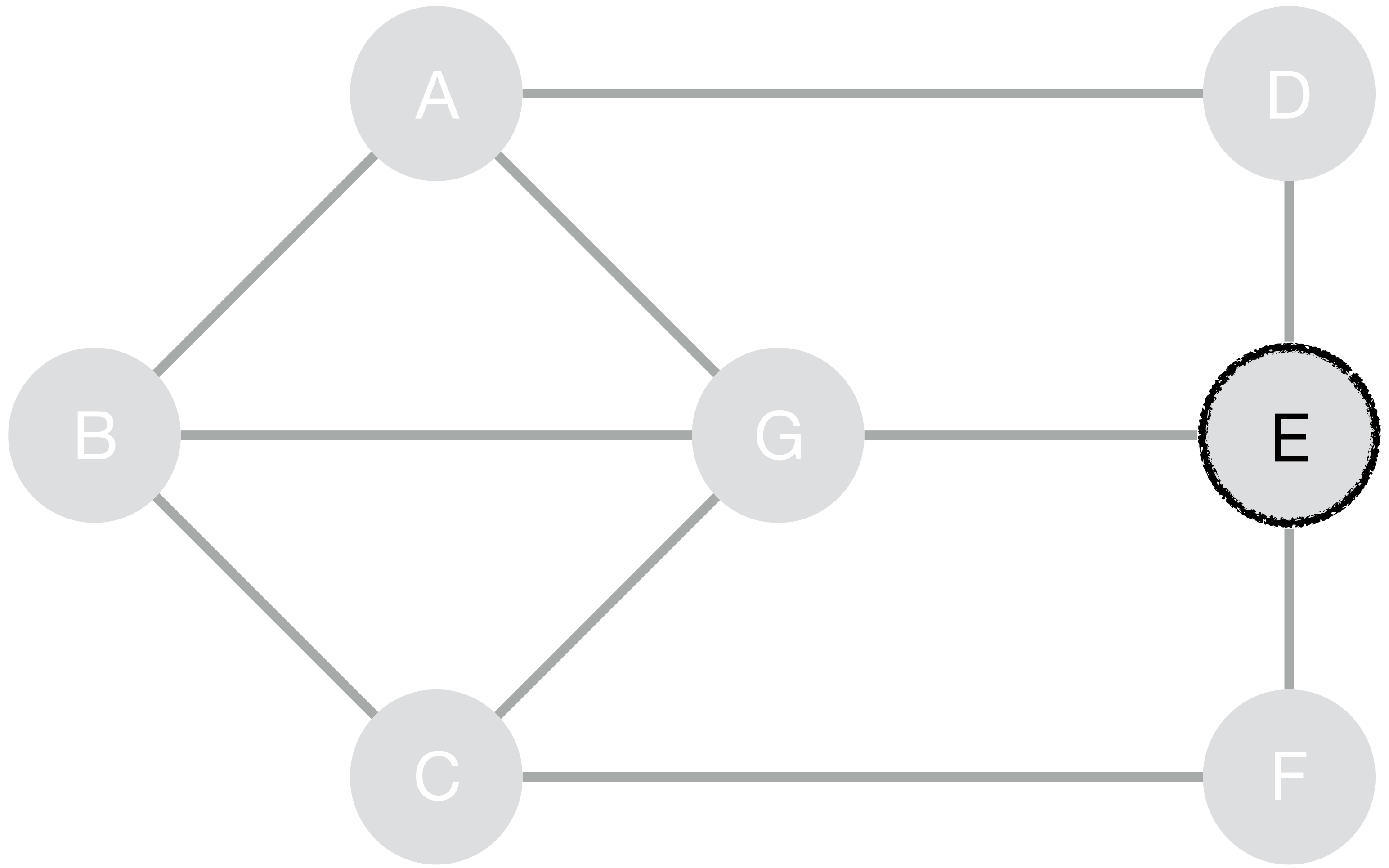
Path Properties: Capacity Aware



- Traditional approaches to routing based on shortest paths (e.g., ECMP, KSP) are generally not capacity aware

Path Properties: Globally Optimal

Other approaches based on greedy algorithms are capacity aware, but are still not globally optimal

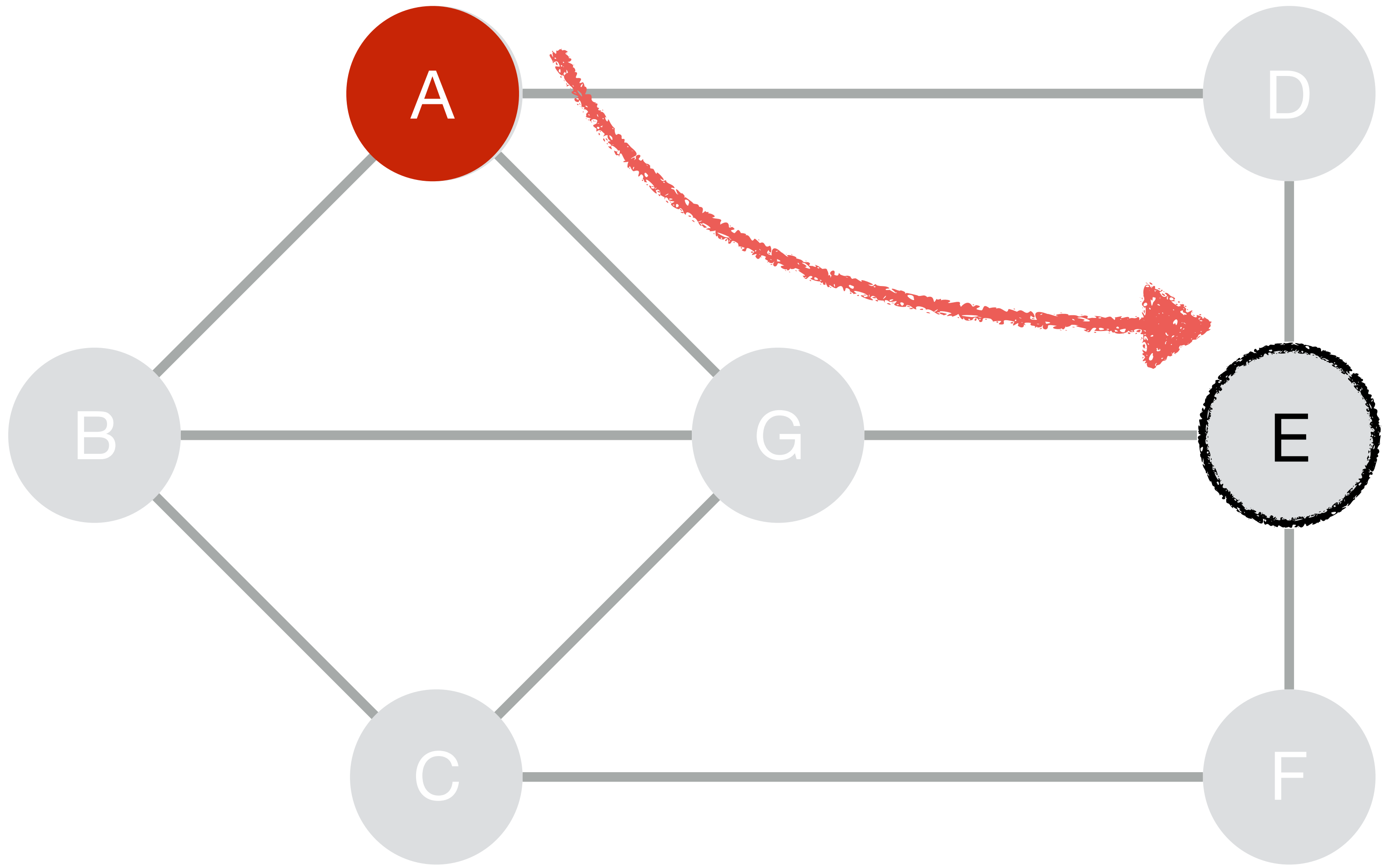


CSPF

Globally optimal

Path Properties: Globally Optimal

Other approaches based on greedy algorithms are capacity aware, but are still not globally optimal

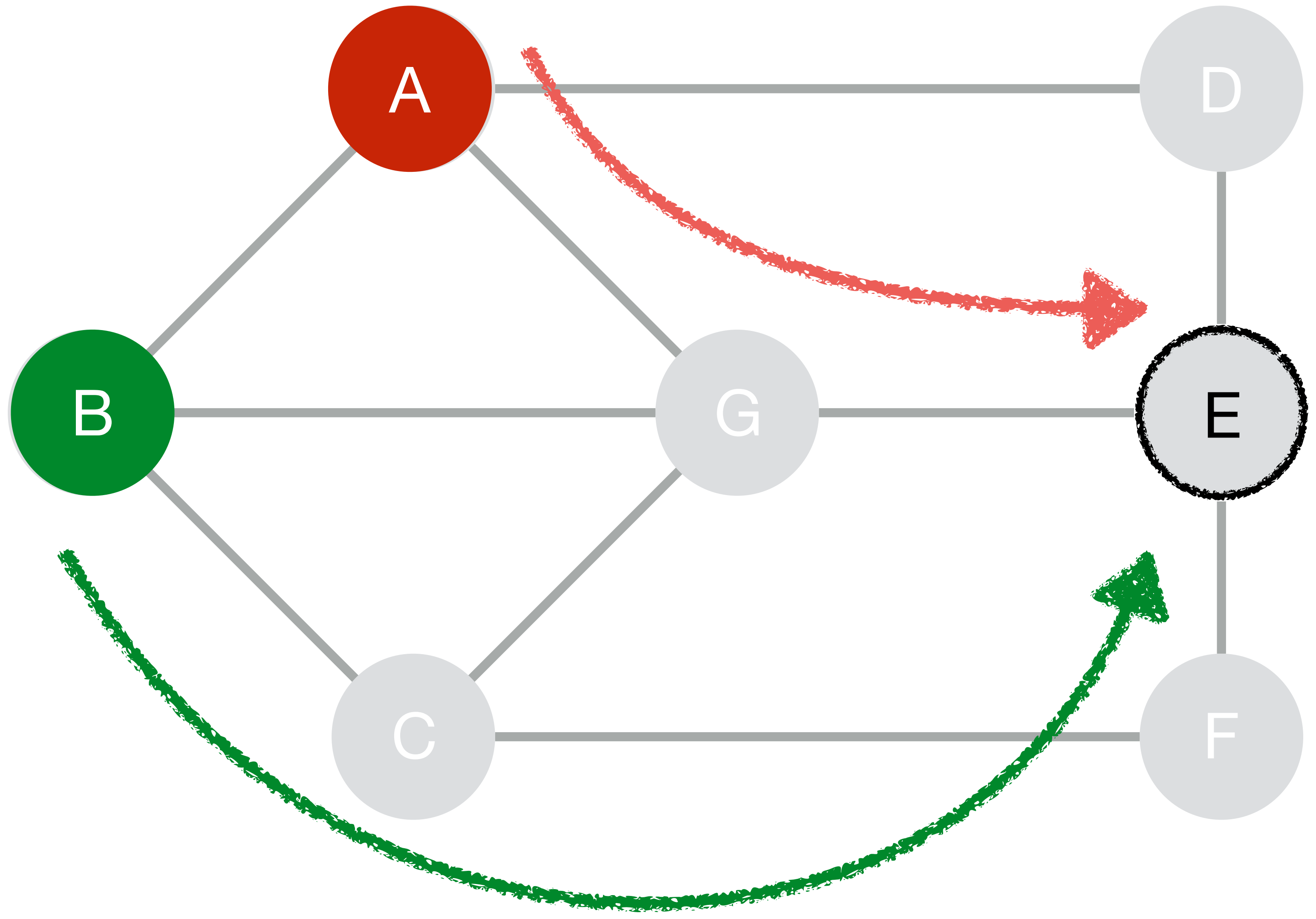


CSPF

Globally optimal

Path Properties: Globally Optimal

Other approaches based on greedy algorithms are capacity aware, but are still not globally optimal

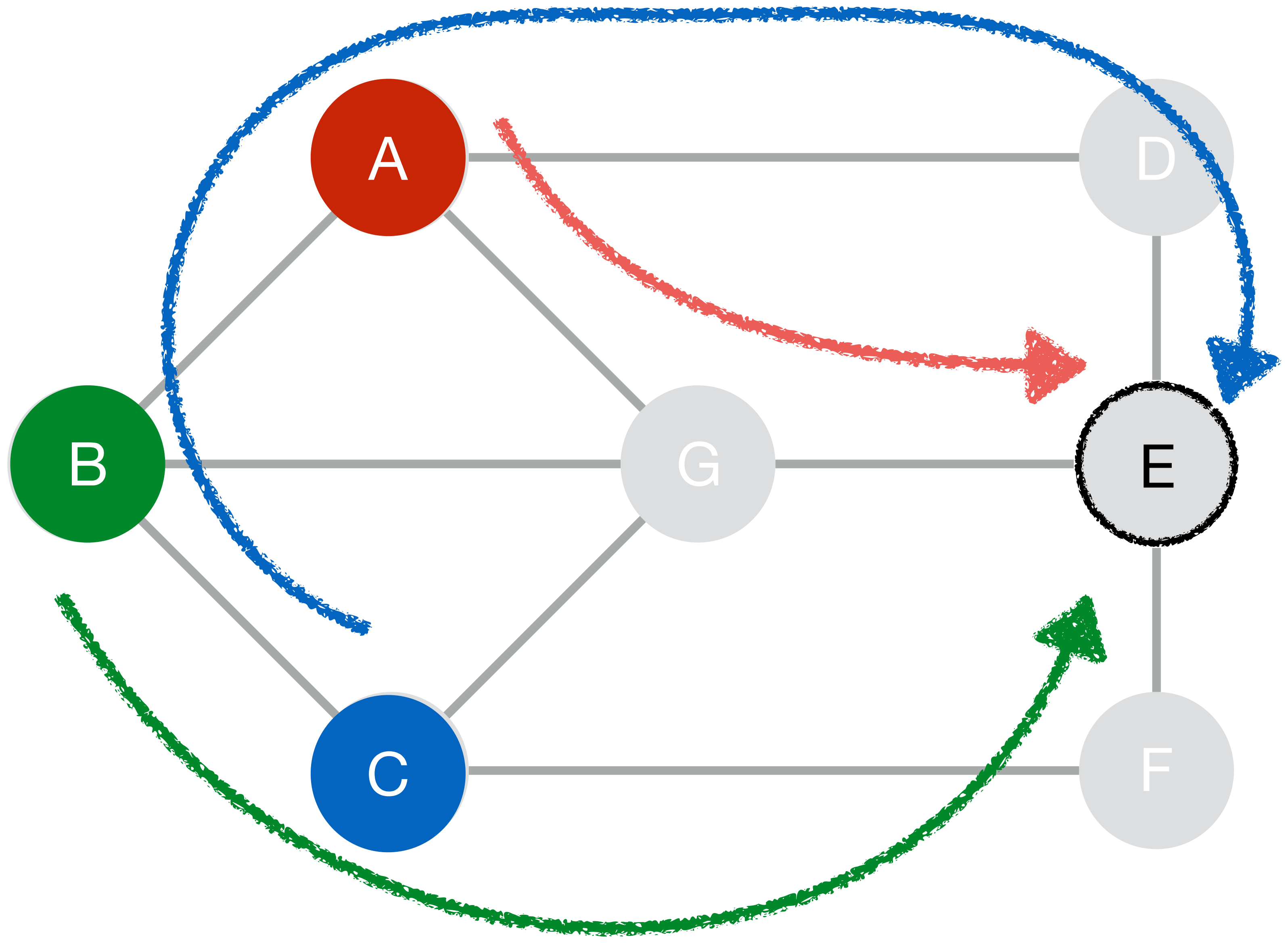


CSPF

Globally optimal

Path Properties: Globally Optimal

Other approaches based on greedy algorithms are capacity aware, but are still not globally optimal

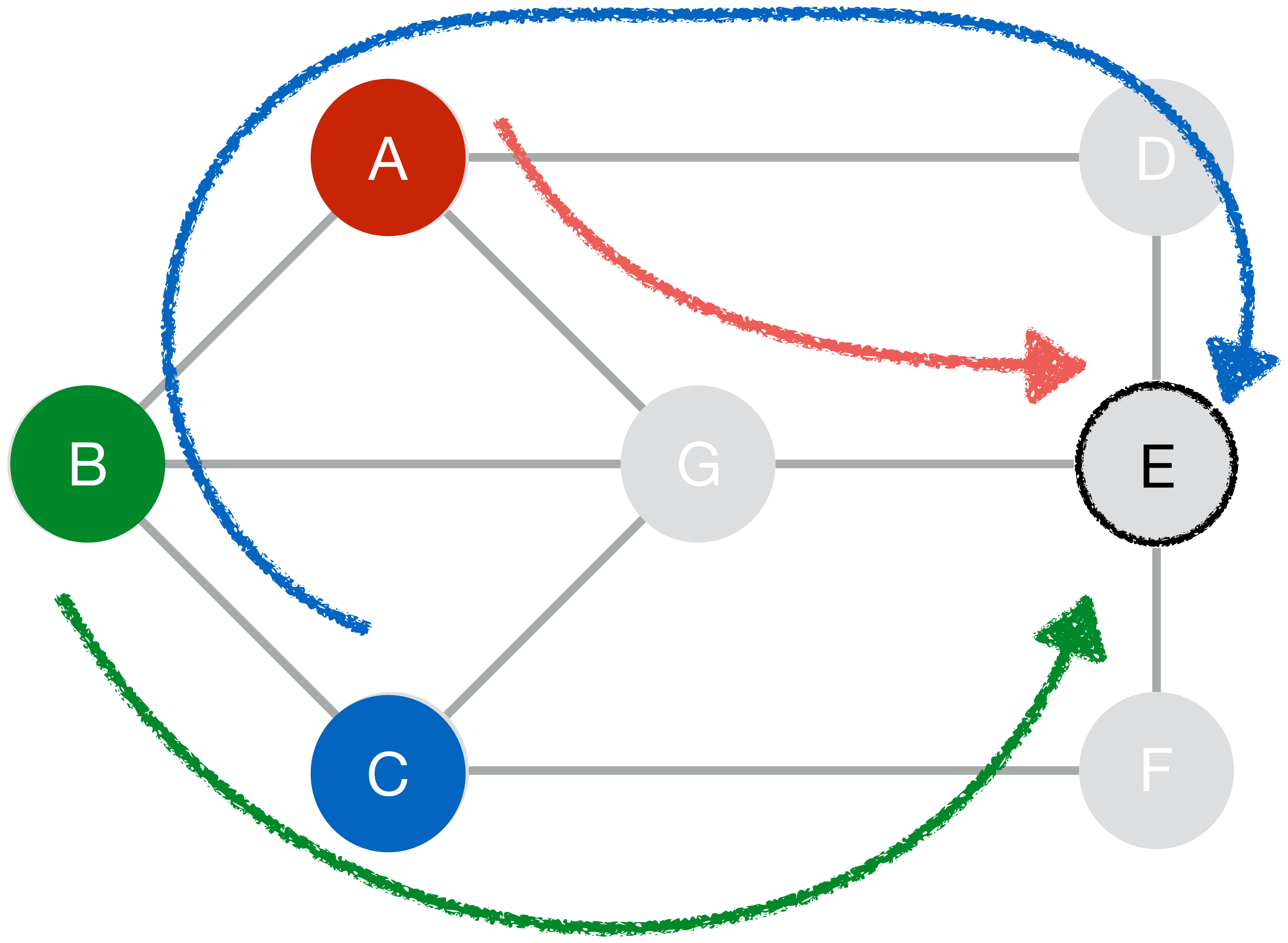


CSPF

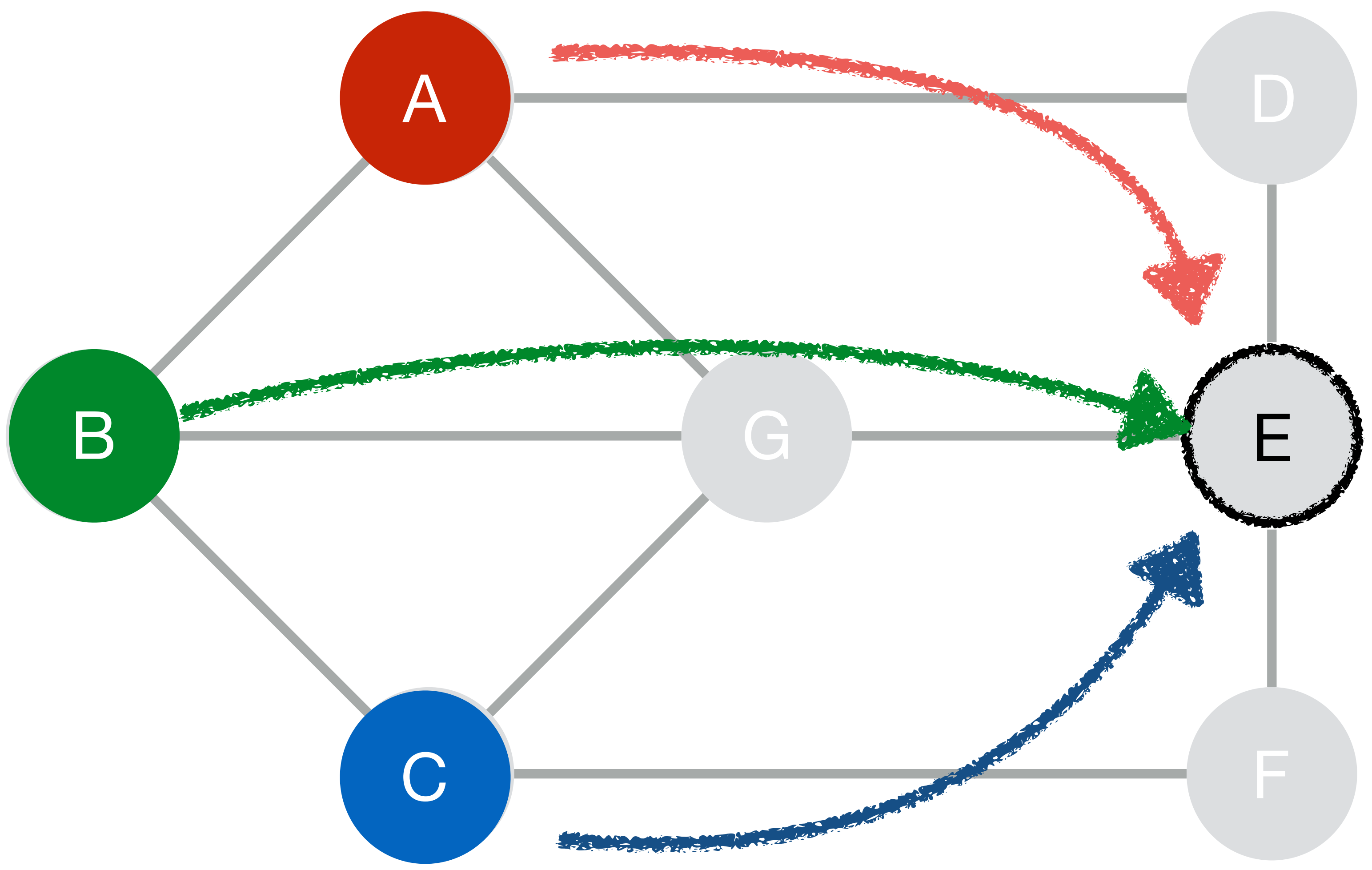
Globally optimal

Path Properties: Globally Optimal

Other approaches based on greedy algorithms are capacity aware, but are still not globally optimal



CSPF



Globally optimal

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✘	✘	✘	✔
CSPF	✔	✘	✘	✔
k-shortest paths	✘	✘	?	✔
Edge-disjoint KSP	✘	✘	✔	✔
MCF	✔	✔	✘	✘
VLB	✘	✘	✔	✘
B4	✔	✔	✘	?

? - Difficult to generalize

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?

? - Difficult to generalize

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?

? - Difficult to generalize

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?

? - Difficult to generalize

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?

? - Difficult to generalize

Path Selection

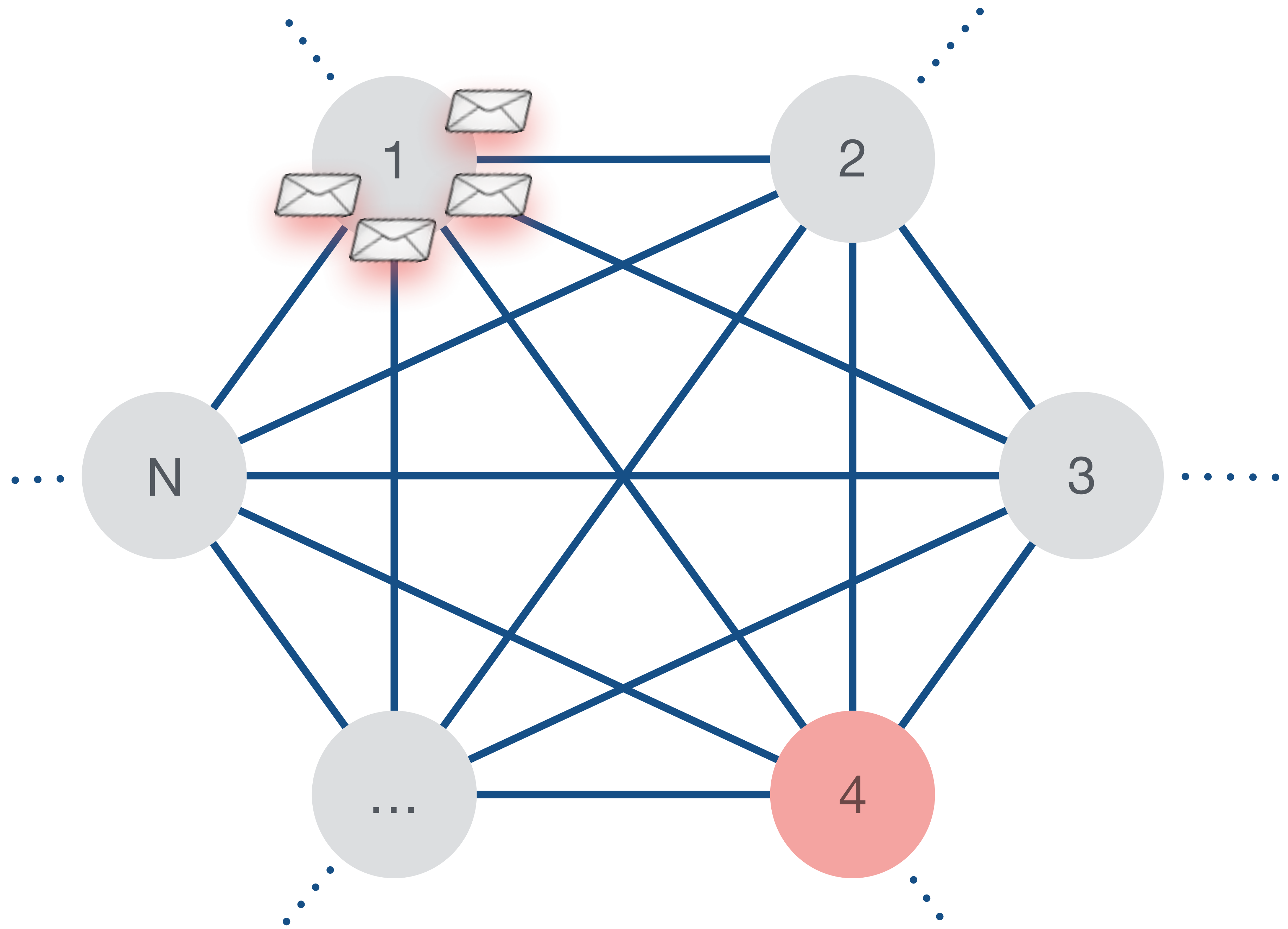
Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?

? - Difficult to generalize

Oblivious Routing

VLB

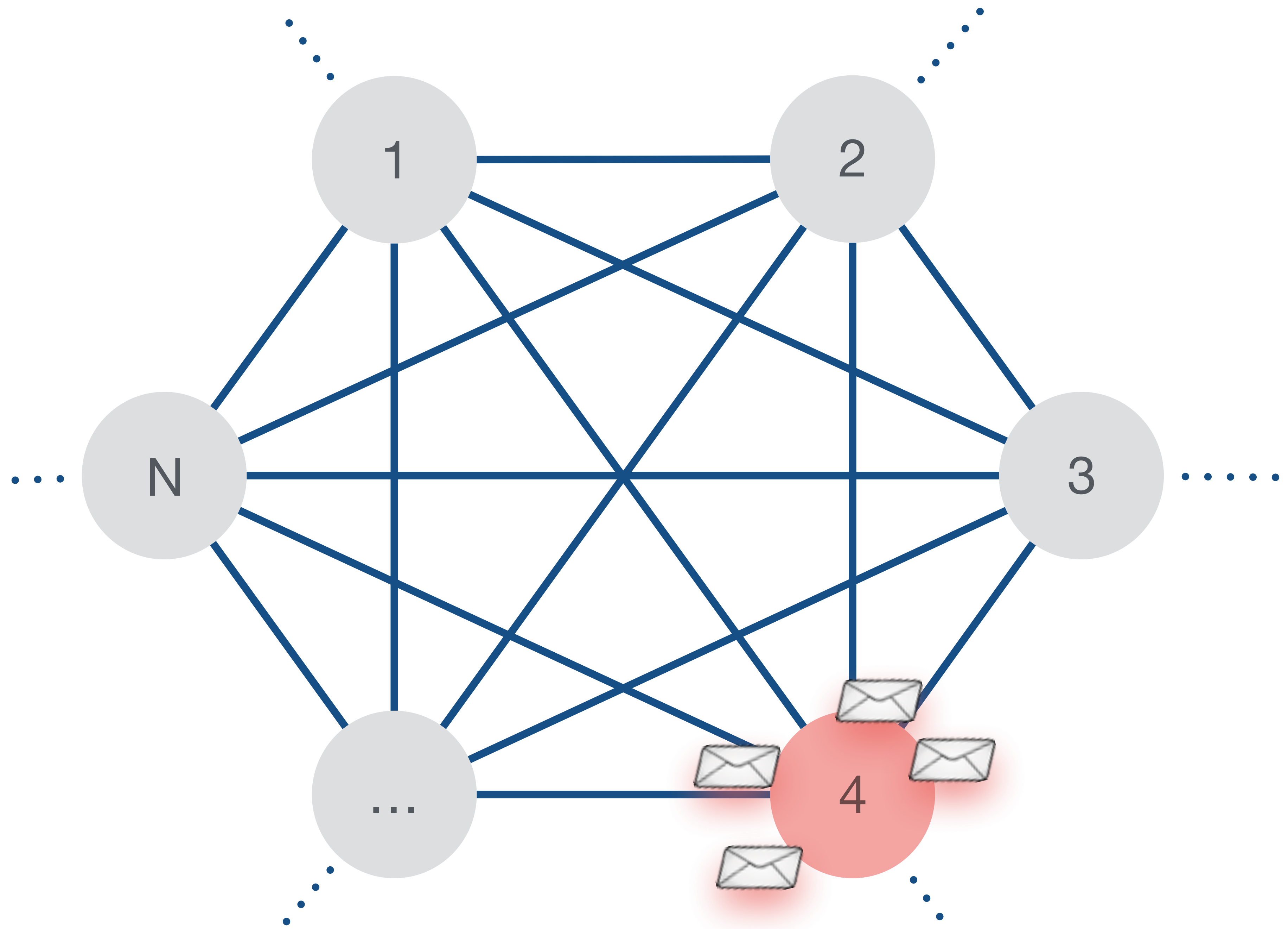
Mesh



- Route through random intermediate node
- Works well for mesh topologies
- WANs are not mesh-like
    - Good resilience
    - Poor performance & latency

VLB

Mesh



- Route through random intermediate node
- Works well for mesh topologies
- WANs are not mesh-like
    - Good resilience
    - Poor performance & latency

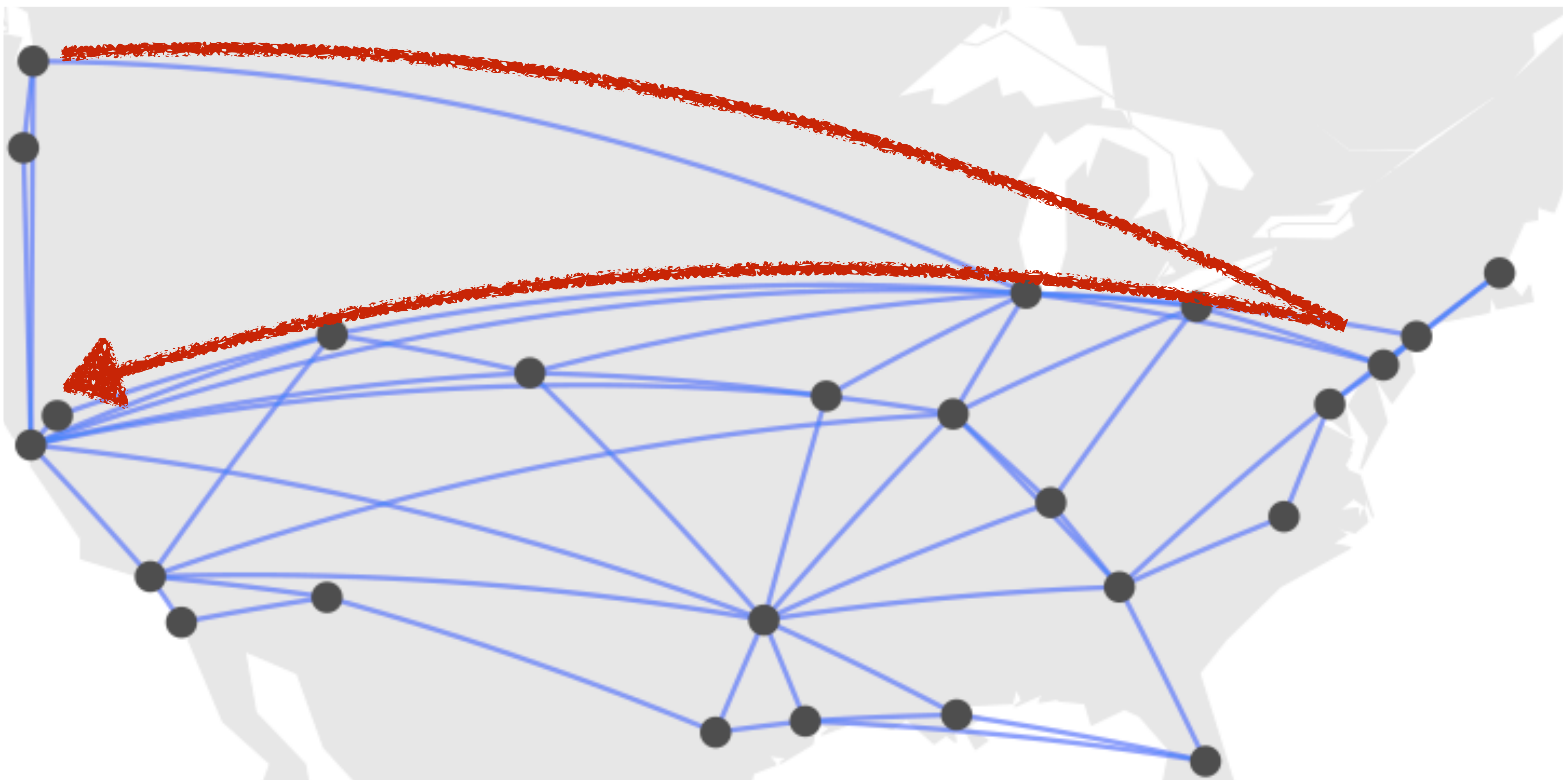
VLB

Not Mesh

- Route through random intermediate node
- Works well for mesh topologies
- WANs are not mesh-like
  - Good resilience
  - Poor performance & latency

VLB

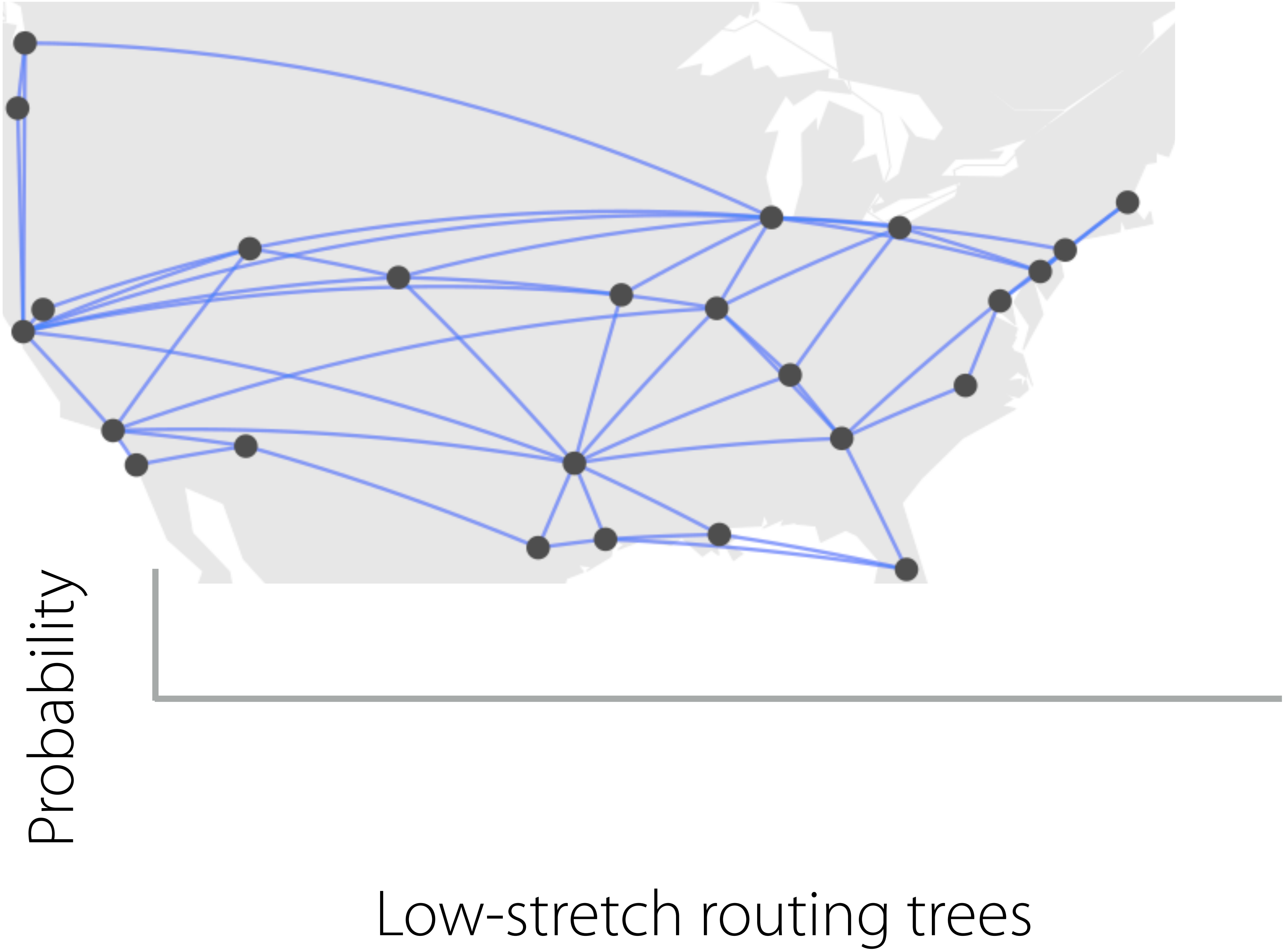
Not Mesh



- Route through random intermediate node
- Works well for mesh topologies
- WANs are not mesh-like
  - Good resilience
  - Poor performance & latency

Oblivious [Räcke '08]

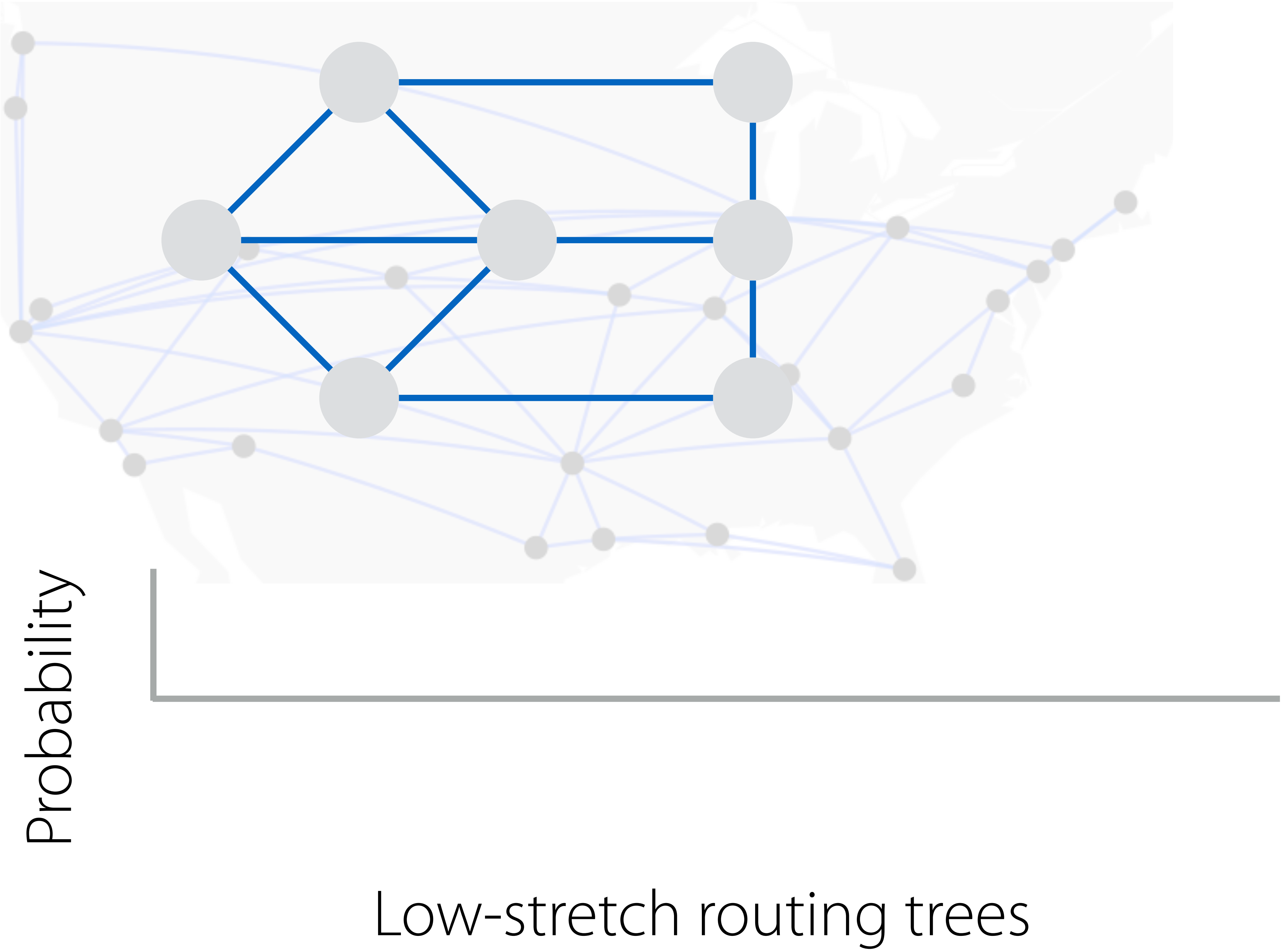
Not Mesh



- Generalizes VLB to non-mesh
- Distribution over routing trees
  - Approximation algorithm for low-stretch trees [FRT '04]
  - Penalize links based on usage
- *O(log n)* competitive

Oblivious [Räcke '08]

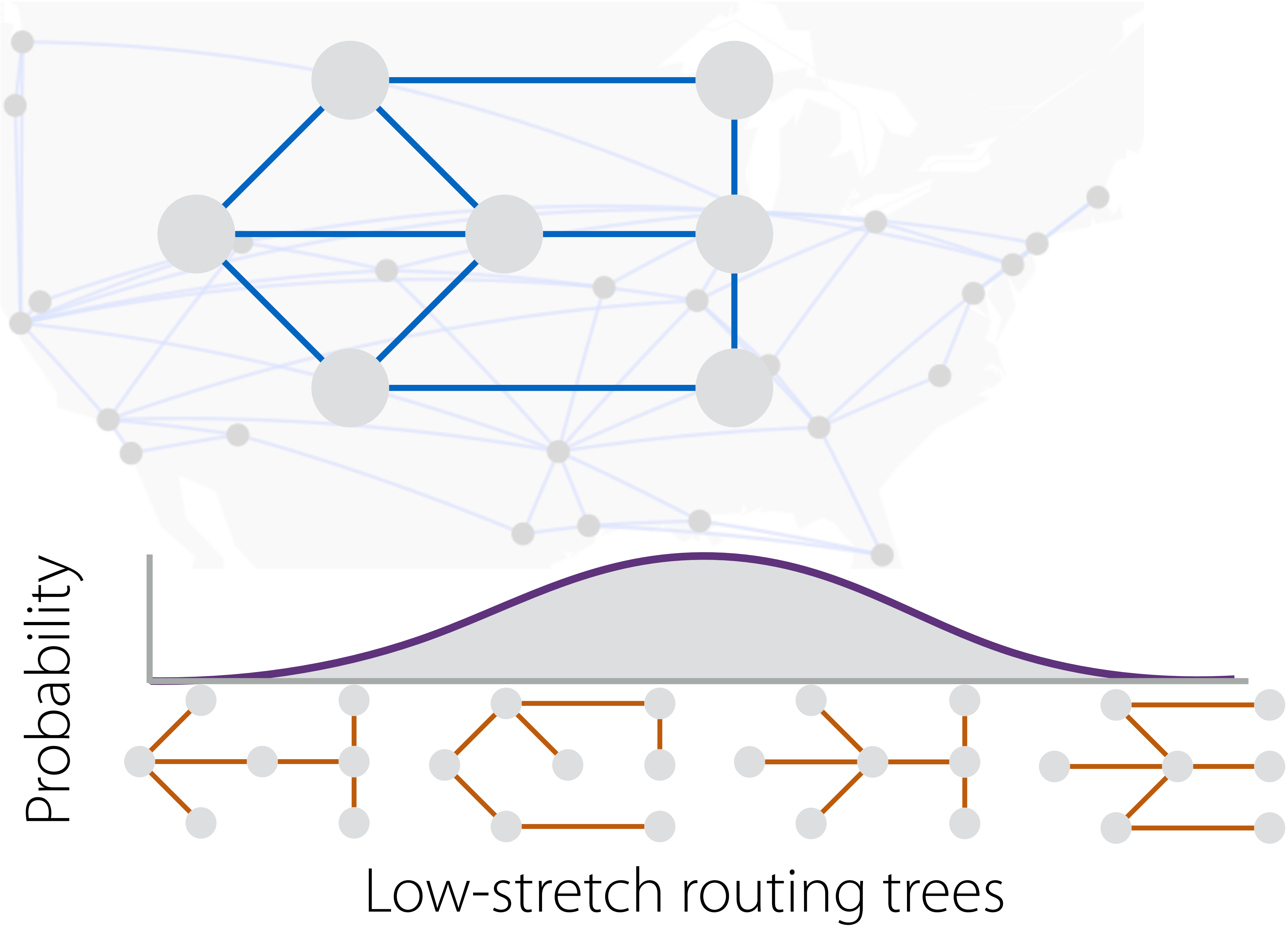
Not Mesh



- Generalizes VLB to non-mesh
- Distribution over routing trees
  - Approximation algorithm for low-stretch trees [FRT '04]
  - Penalize links based on usage
- *O(log n)* competitive

Oblivious [Räcke '08]

Not Mesh



- Generalizes VLB to non-mesh
- Distribution over routing trees
  - Approximation algorithm for low-stretch trees [FRT '04]
  - Penalize links based on usage
- *O(log n)* competitive

Path Selection

Algorithm	Load balanced: Capacity aware	Load balanced: Globally Optimized	Diverse	Low-stretch
SPF / ECMP	✗	✗	✗	✔
CSPF	✔	✗	✗	✔
k-shortest paths	✗	✗	?	✔
Edge-disjoint KSP	✗	✗	✔	✔
MCF	✔	✔	✗	✗
VLB	✗	✗	✔	✗
B4	✔	✔	✗	?
SMORE / Oblivious	✔	✔	✔	✔

SMORE: Semi-Oblivious Routing

Oblivious Routing computes a set of paths which are low-stretch, robust and have good load balancing properties

Path Selection

LP Optimizer balances load by dynamically adjusting splitting ratios used to map incoming traffic flows to paths

Rate Adaptation

Semi-Oblivious Traffic Engineering: The Road Not Taken [NSDI '18]

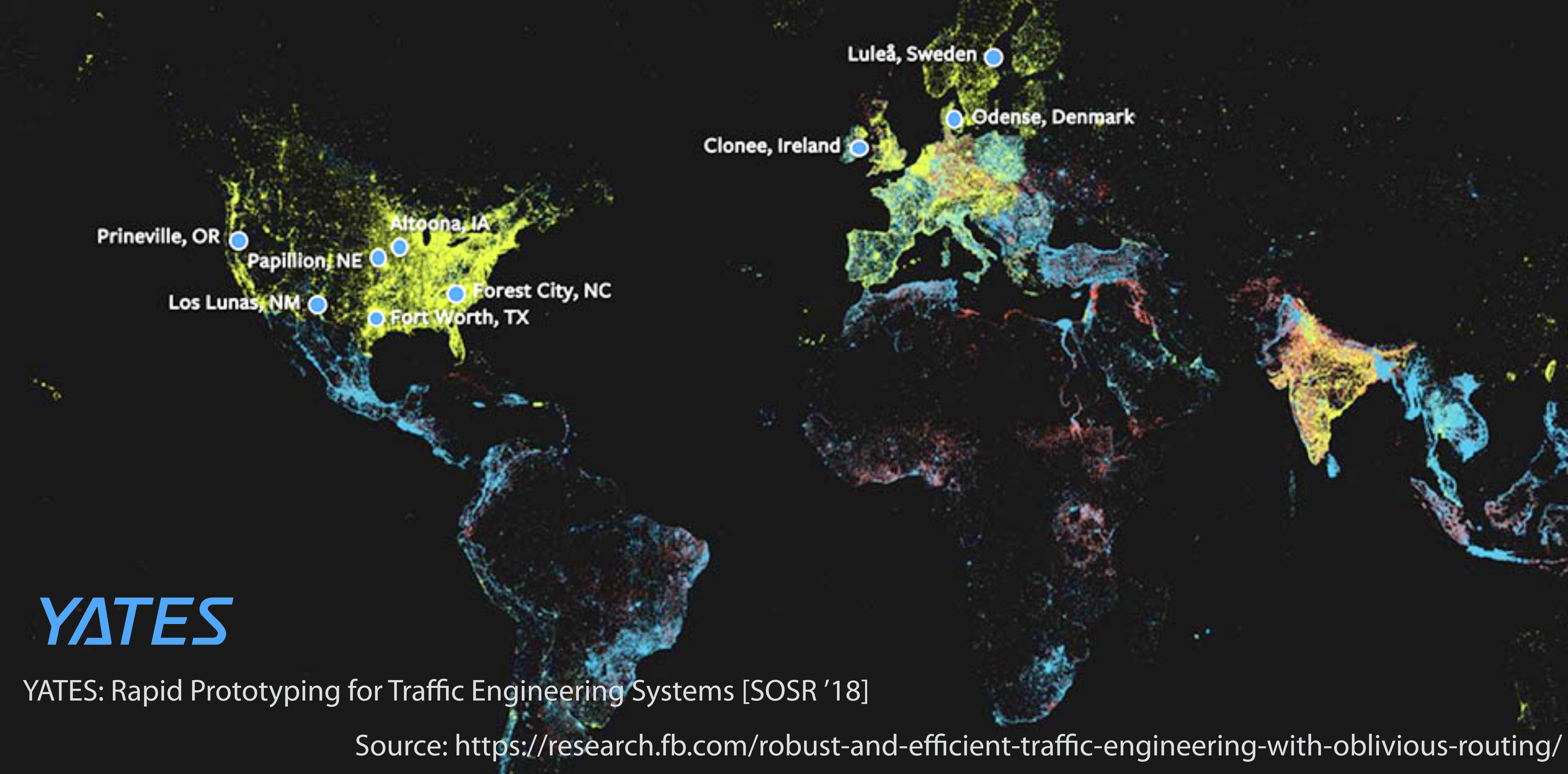
Semi-Oblivious Routing in Practice?

- ▼ Previous work [Hajiaghayi et al.] established a worst-case competitive ratio that is not much better than oblivious routing: $\Omega(\log(n)/\log(\log(n)))$
- ▲ But the real-world does not typically exhibit worst-case scenarios
- ▲ Implicit correlation between demands and link capacities

Question: How well does semi-oblivious routing perform in practice?

Evaluation

Facebook's Backbone Network

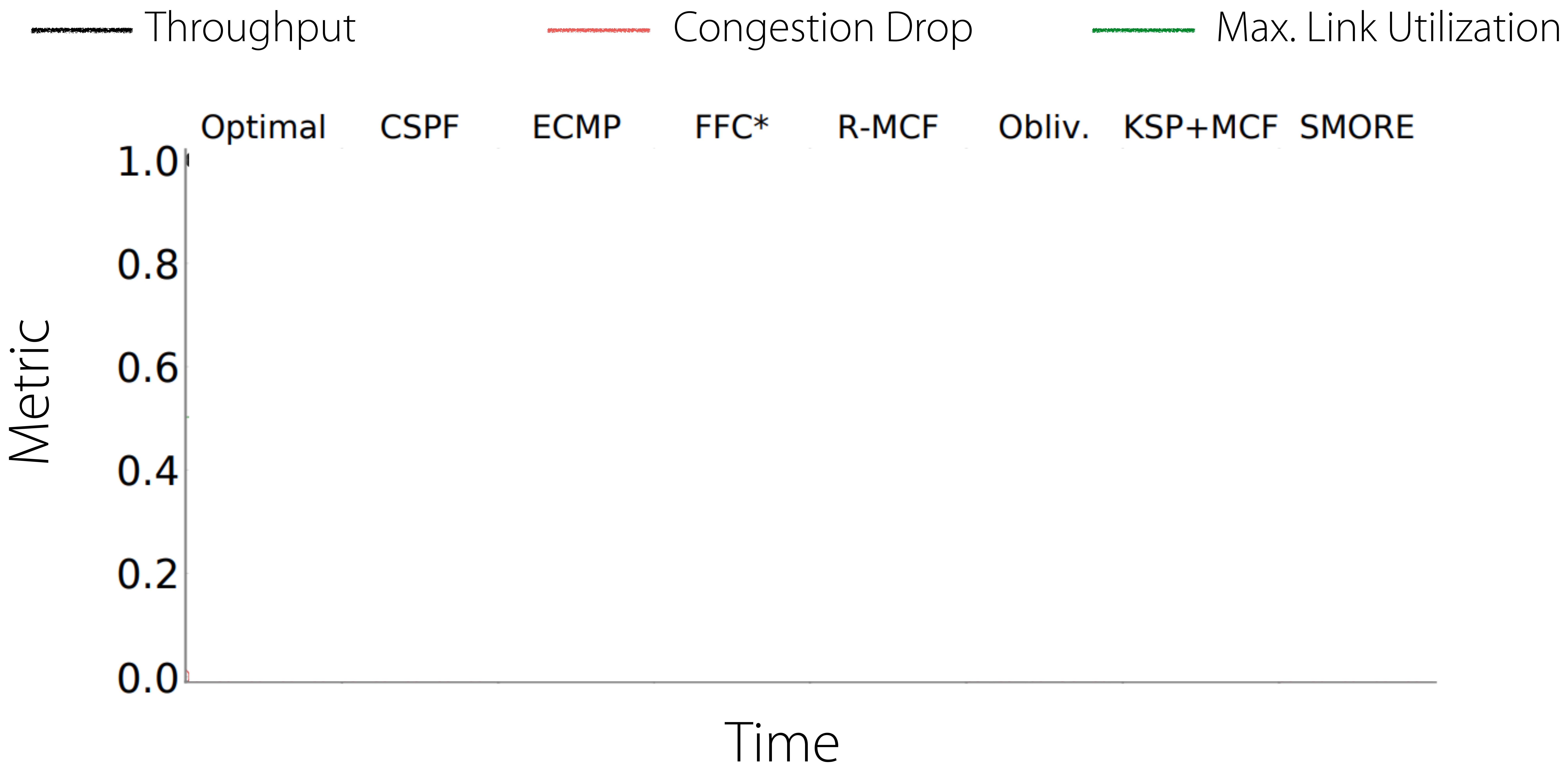


YATES

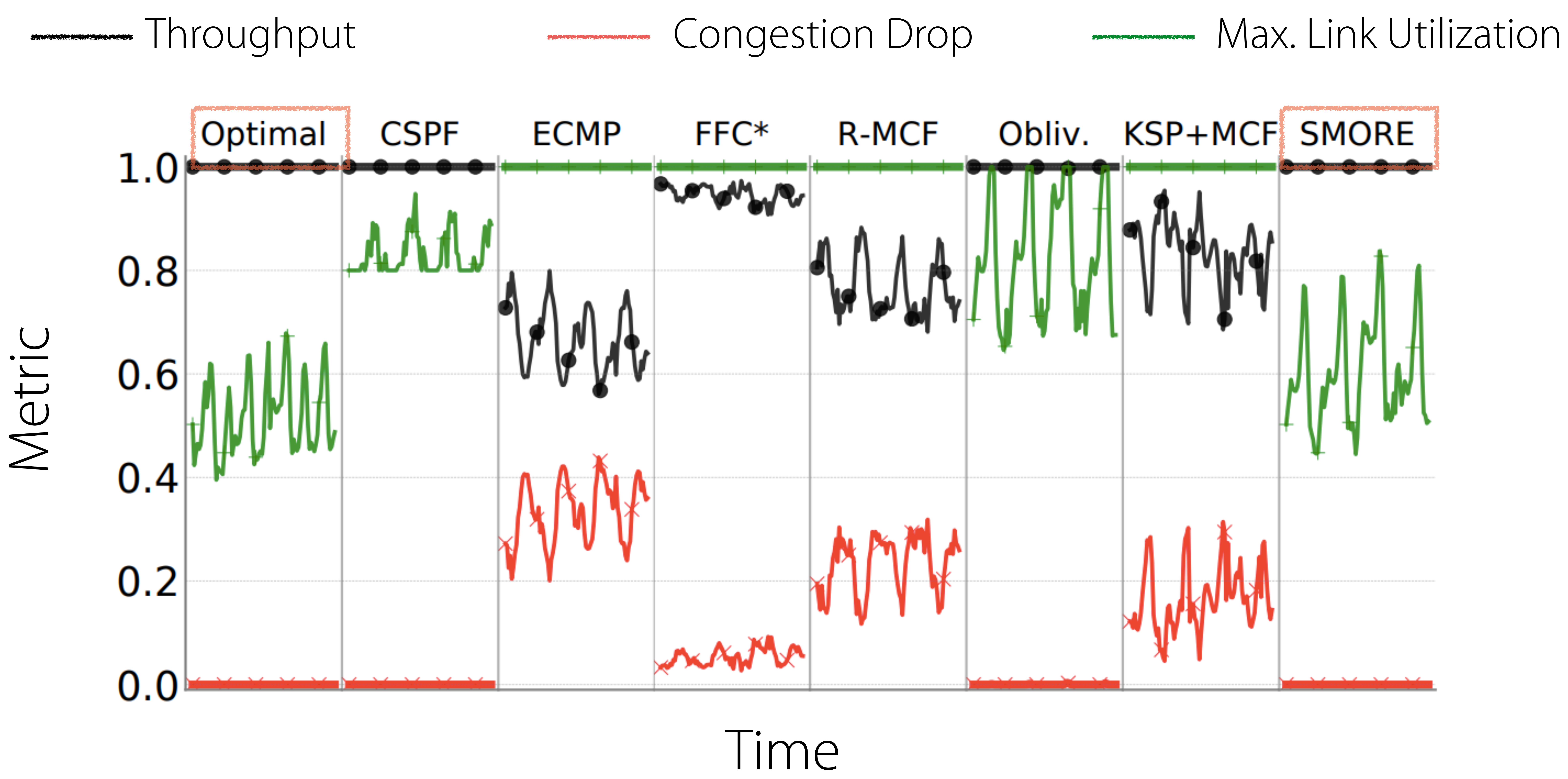
YATES: Rapid Prototyping for Traffic Engineering Systems [SOSR '18]

Source: https://research.fb.com/robust-and-efficient-traffic-engineering-with-oblivious-routing/

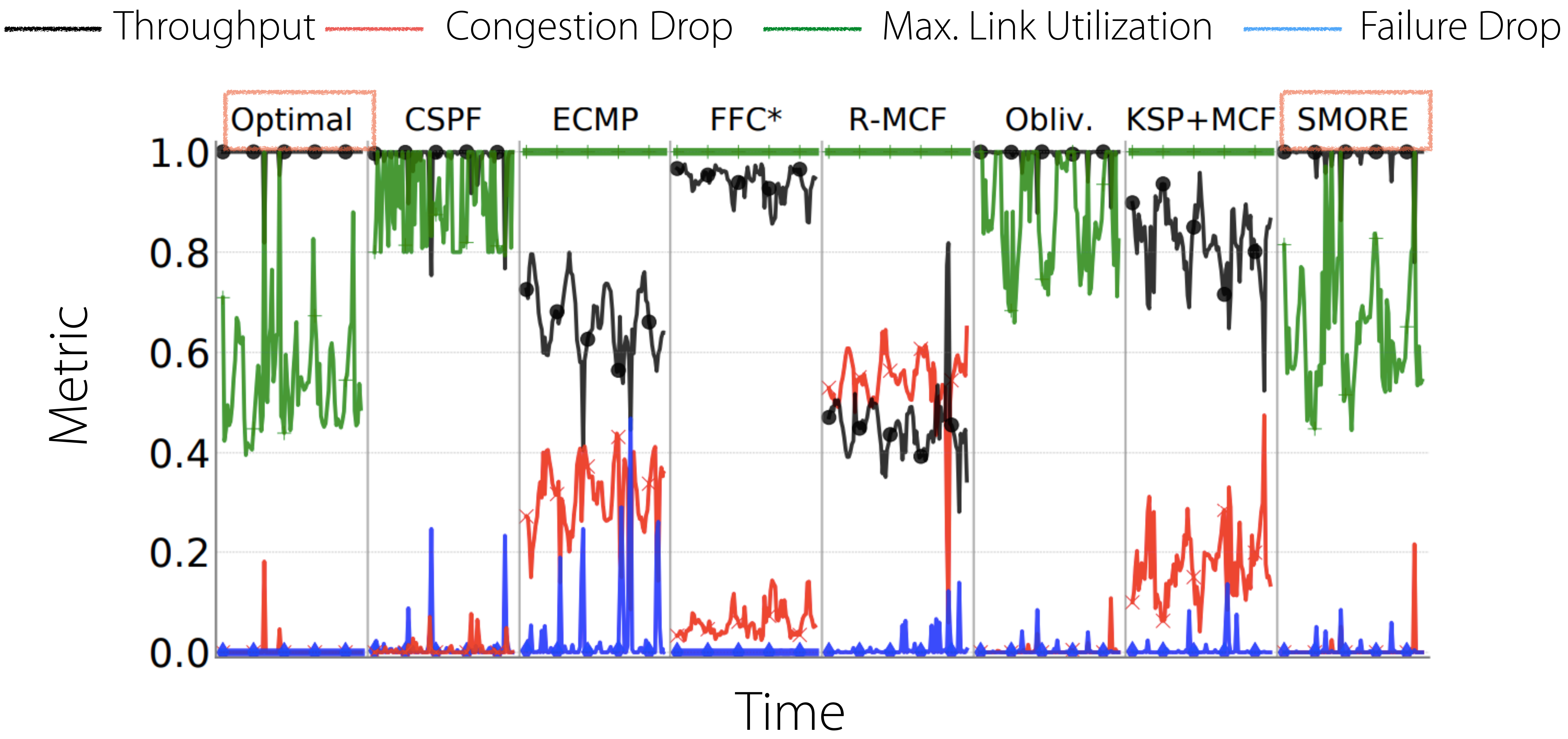
Performance



Performance

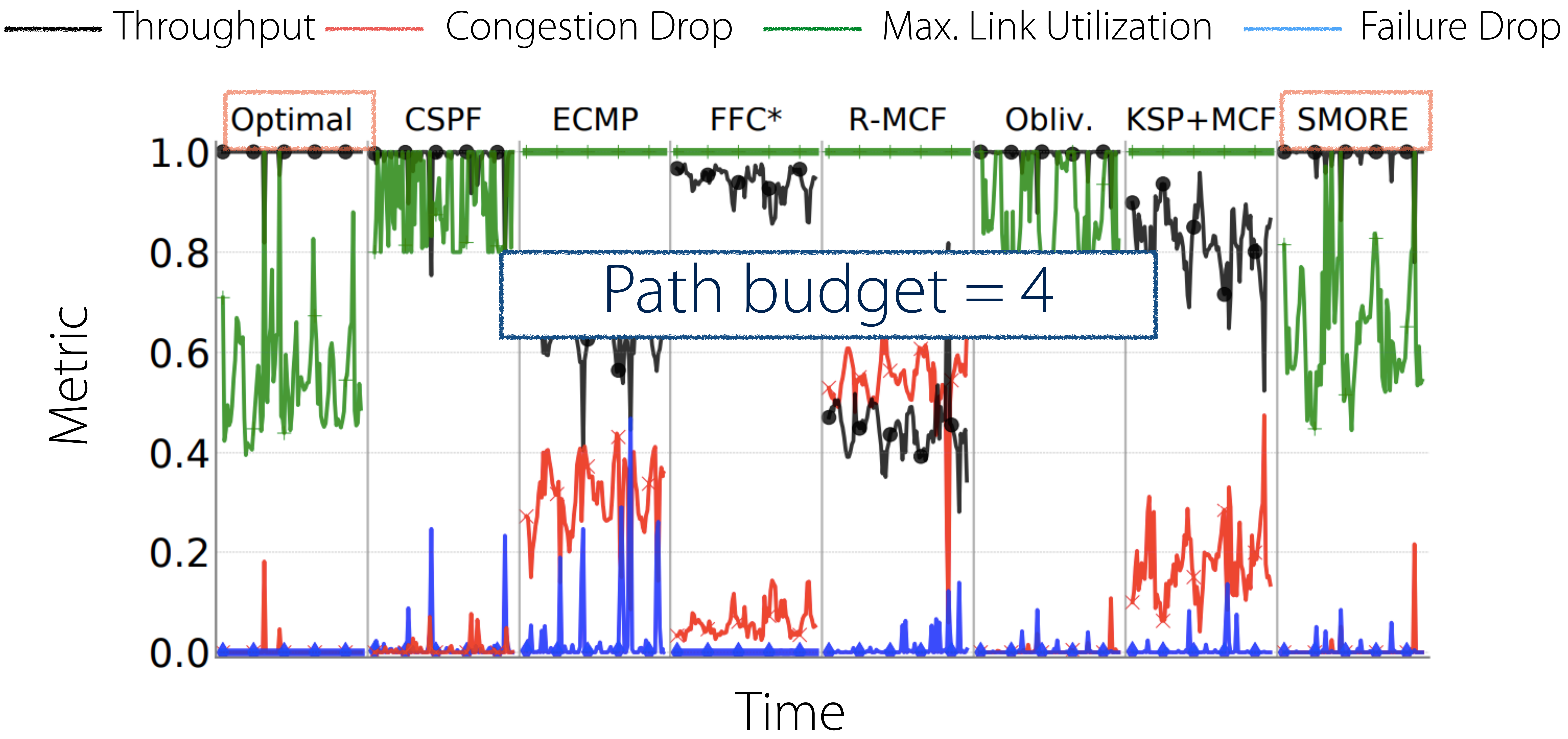


Robustness

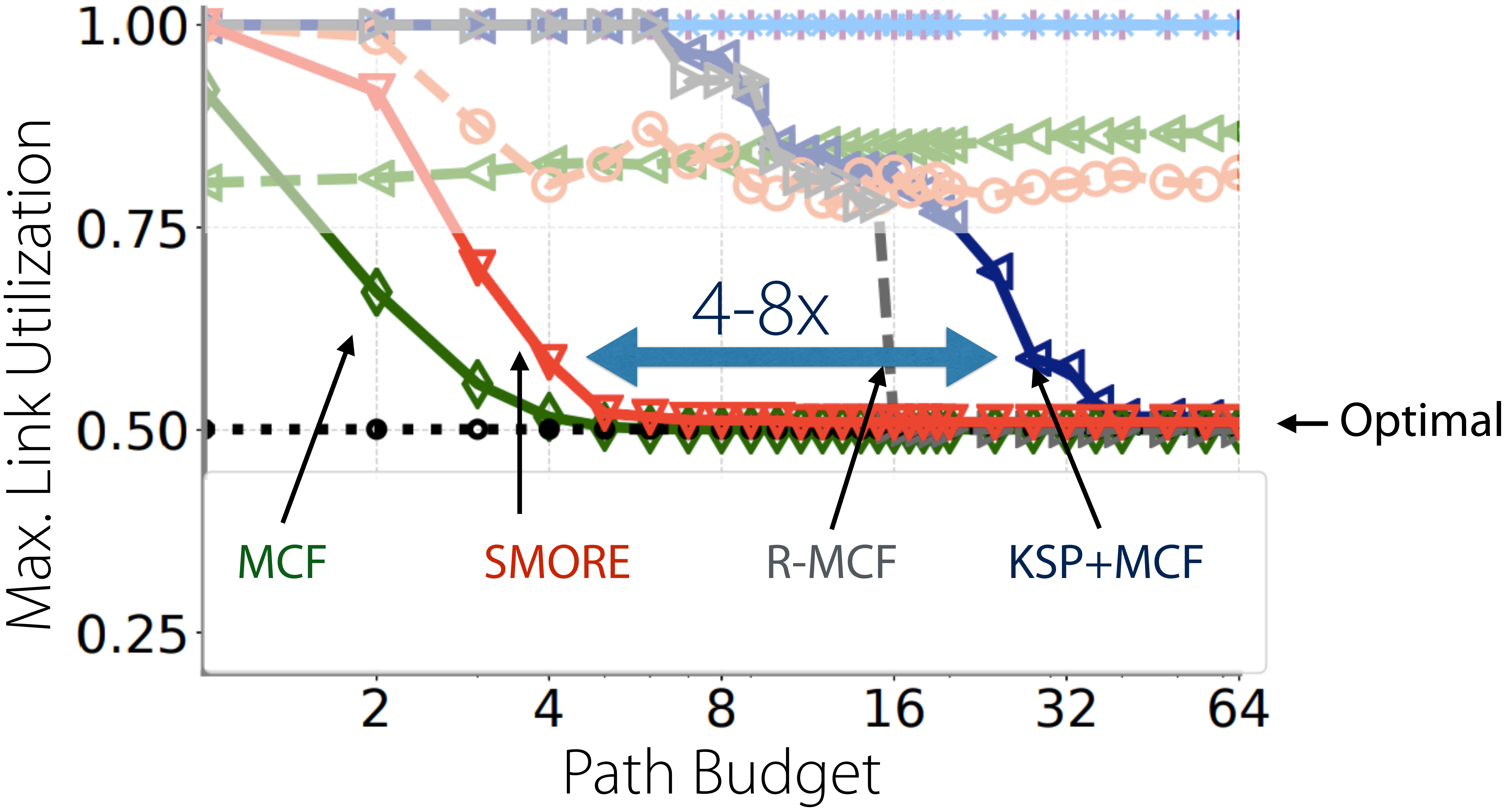


Throughput
Congestion Drop
Max. Link Utilization
Failure Drop
Optimal
CSPF
ECMP
FFC*
R-MCF
Obliv.
KSP+MCF
SMORE
1.0
0.8
0.6
0.4
0.2
0.0
Metric
Time

Robustness


Throughput
Congestion Drop
Max. Link Utilization
Failure Drop
Optimal
CSPF
ECMP
FFC*
R-MCF
Obliv.
KSP+MCF
SMORE
1.0
0.8
0.6
0.4
0.2
0.0
Metric
Time
Path budget = 4

Operational Constraints - Path Budget

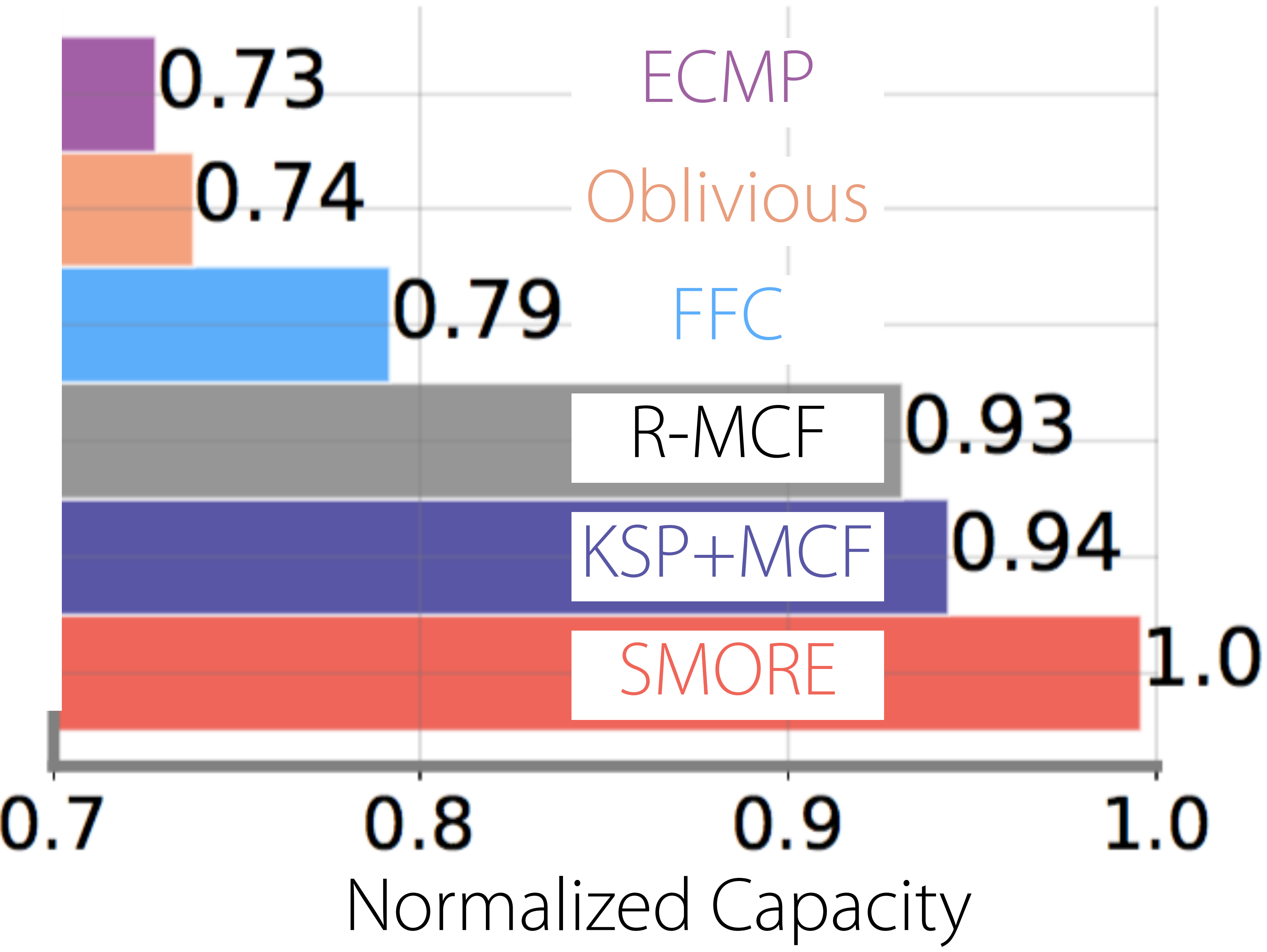


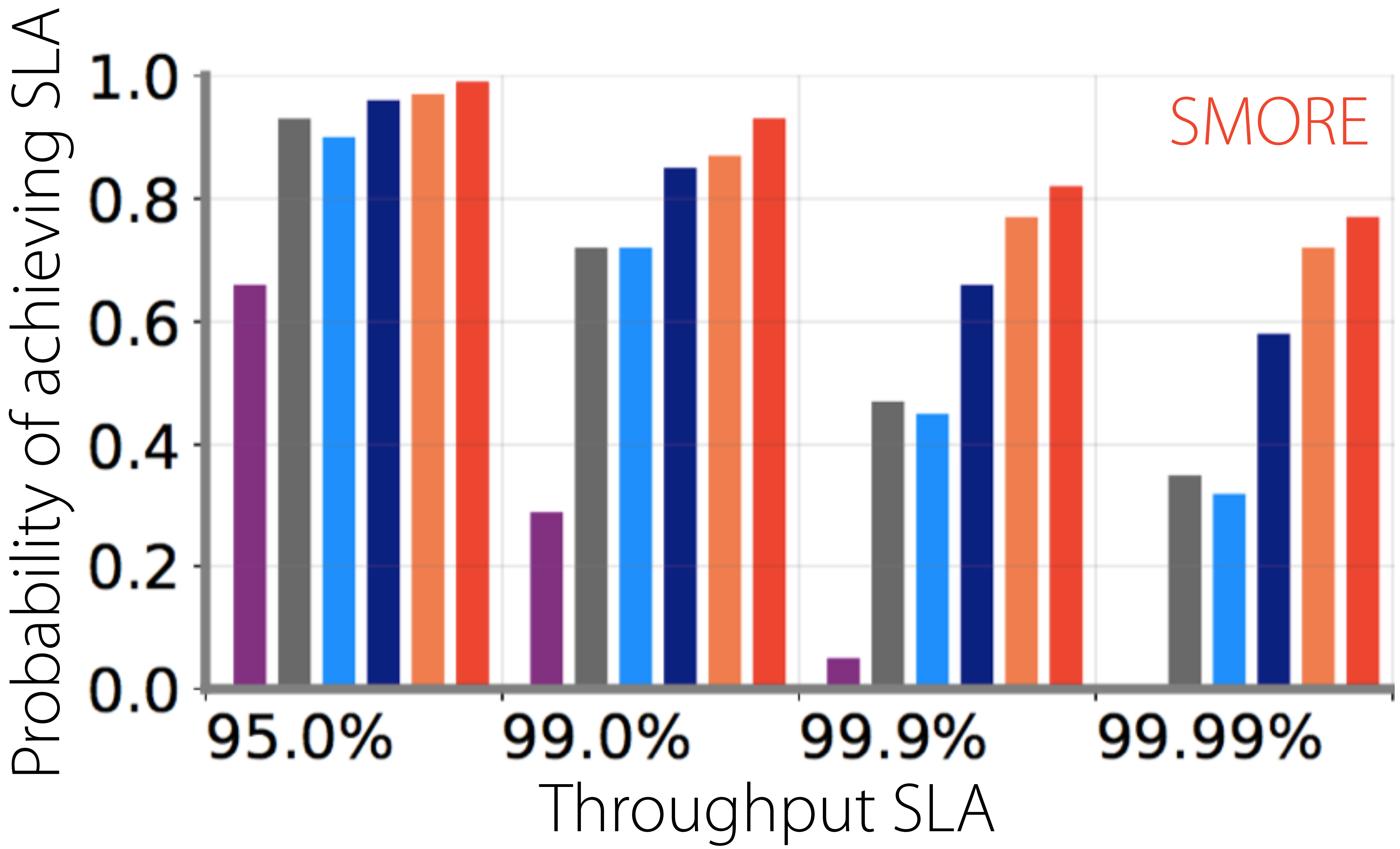
Large Scale Simulations

- Conducted larger set of simulations on Internet Topology Zoo
- 30 topologies from ISPs and content providers
- Multiple traffic matrices (gravity model), failure models and operational conditions

Do these results generalize?

Yes*





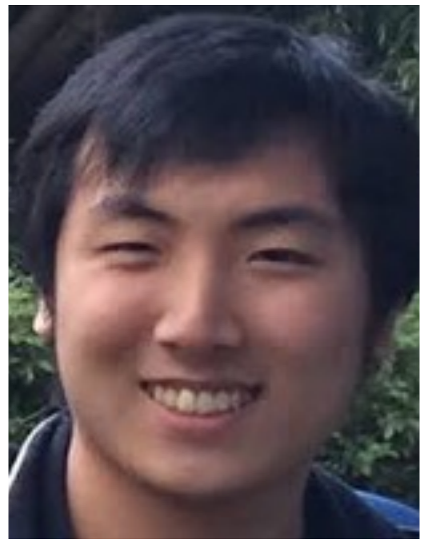
Takeaways

- **Path selection** plays an outsized role in the performance of TE systems
- **Semi-oblivious TE** meets the competing objectives of performance and robustness in modern networks
    - **Oblivious routing** for path selection + **Dynamic load-balancing**
- Ongoing and future-work:
    - Apply to other networks (e.g. non-Clos DC topologies)
    - SR-based implementations and deployments

Thank You!

SMORE: Oblivious routing + Dynamic rate adaptation

Yang Yuan
Cornell



Chris Yu
CMU

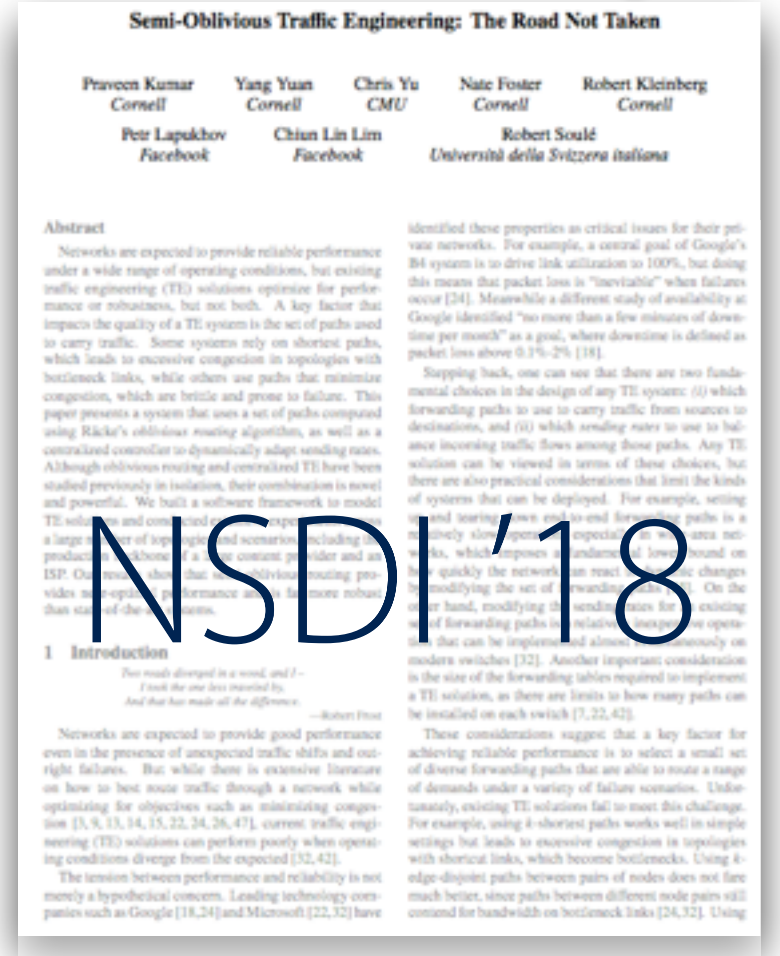
Nate Foster
Cornell

Bobby Kleinberg
Cornell

Petr Lapukhov
Facebook

Chiun Lin Lim
Facebook

Robert Soule
Lugano



Semi-Oblivious Traffic Engineering: The Road Not Taken

Praveen Kumar *Cornell* · Yang Yuan *Cornell* · Chris Yu *CMU* · Nate Foster *Cornell* · Robert Kleinberg *Cornell*

Petr Lapukhov *Facebook* · Chiun Lin Lim *Facebook* · Robert Soulé *Università della Svizzera italiana*

Abstract

NSDI '18

Code: github.com/cornell-netlab/yates

Learn more: www.cs.cornell.edu/~praveenk/smore/