On low-latency-capable topologies, and their impact on the design of intra-domain routing

Nikola Gvozdiev, Stefano Vissicchio, Brad Karp, Mark Handley

University College London (UCL)

We want low latency!

We want low latency!

In the datacenter

We want low latency!

In the datacenter

In the enterprise WAN [B4, BwE, SWAN …]

- operator controls both WAN and sources
- … so demands are predictable

We want low latency!

In the datacenter

In the enterprise WAN [B4, BwE, SWAN …]

- operator controls both WAN and sources
- … so demands are predictable

In the ISP ⟵ this talk

- ISP operator does not control sources

How do we get low latency in a loaded ISP network?

How do we get low latency in a loaded ISP network?

The topology?

Topology must offer diverse low-latency paths…

How do we get low latency in a loaded ISP network?

The topology?

Topology must offer diverse low-latency paths…

The routing?

…and routing system must make good use of those low-latency paths

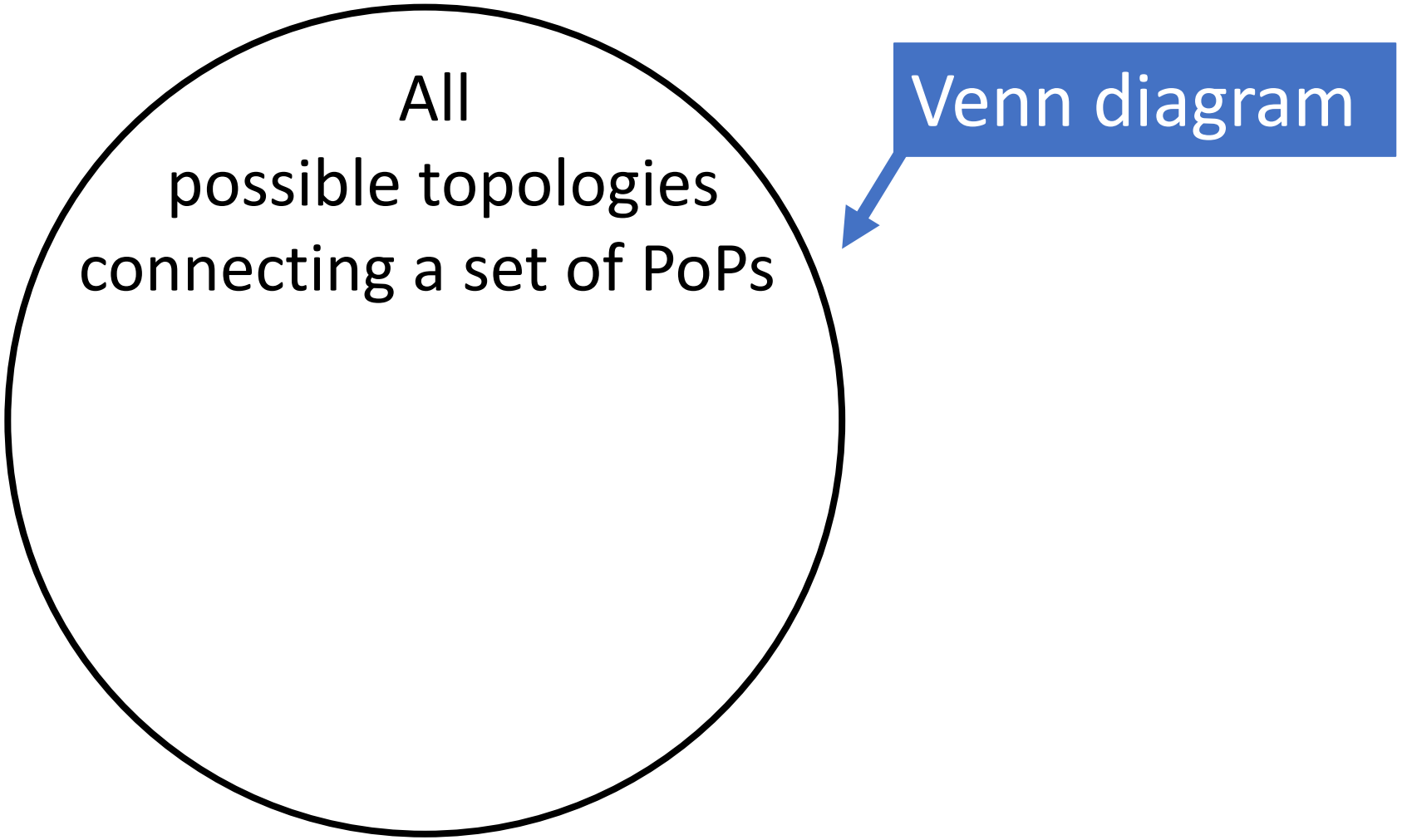
How do we get low latency in a loaded ISP network?

The topology?

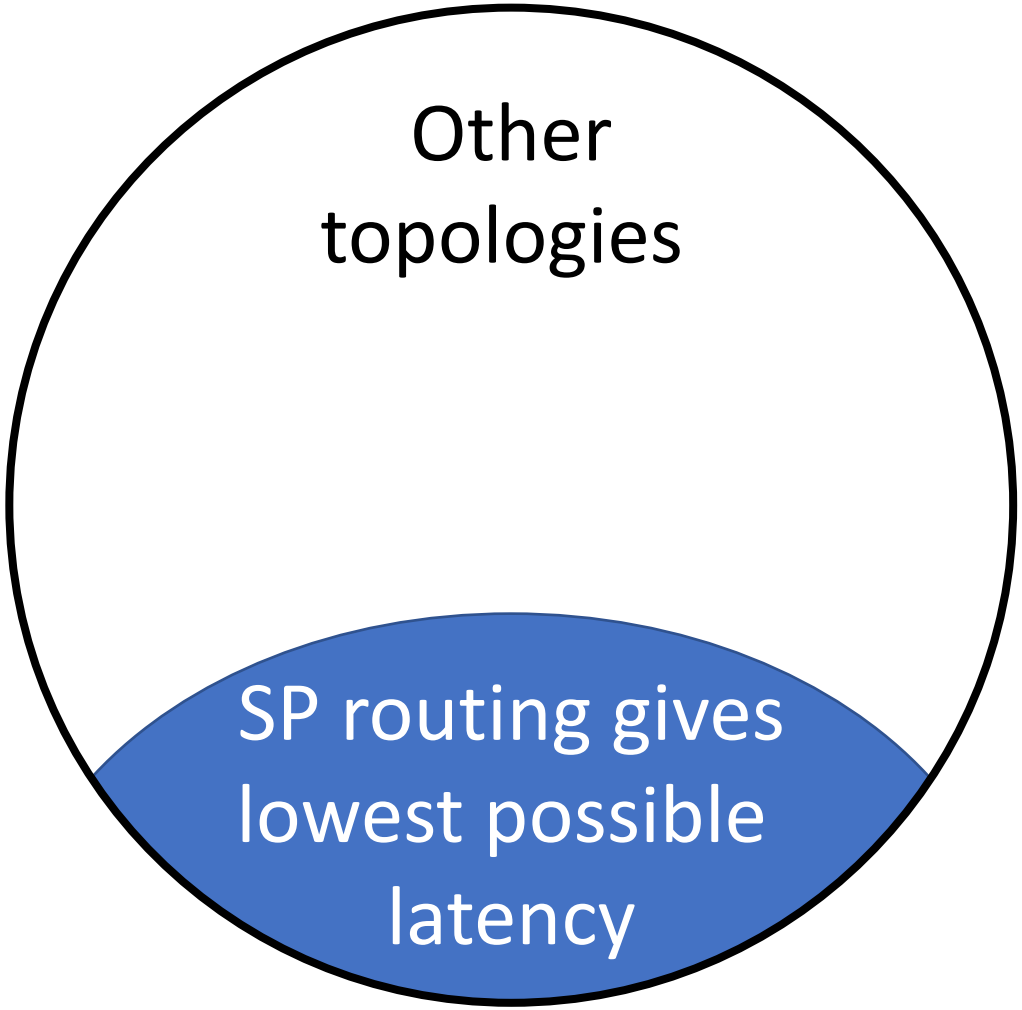
The routing?

Topology must offer diverse low-latency paths…

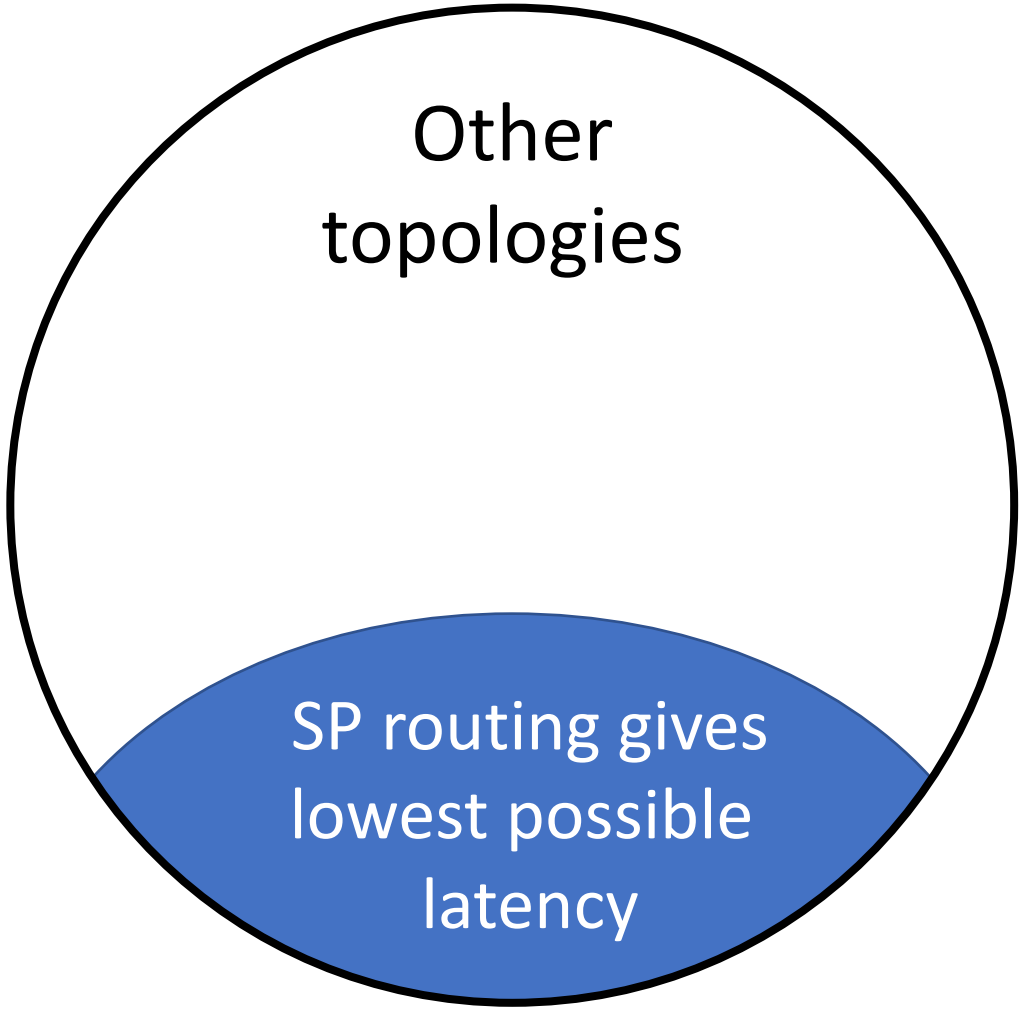
…and routing system must make good use of those low-latency paths


All
possible topologies
connecting a set of PoPs
Venn diagram

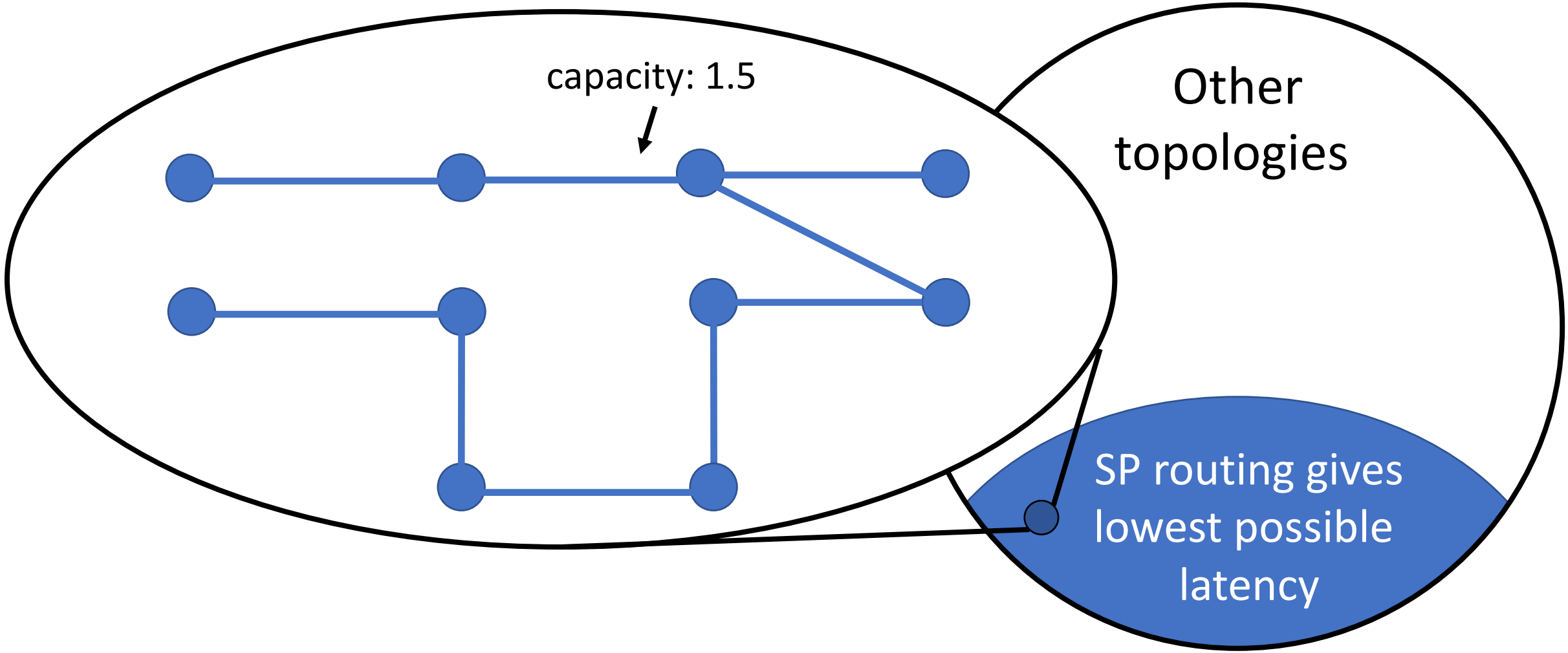
Shortest-path routing doesn’t yield lowest latency on all topologies



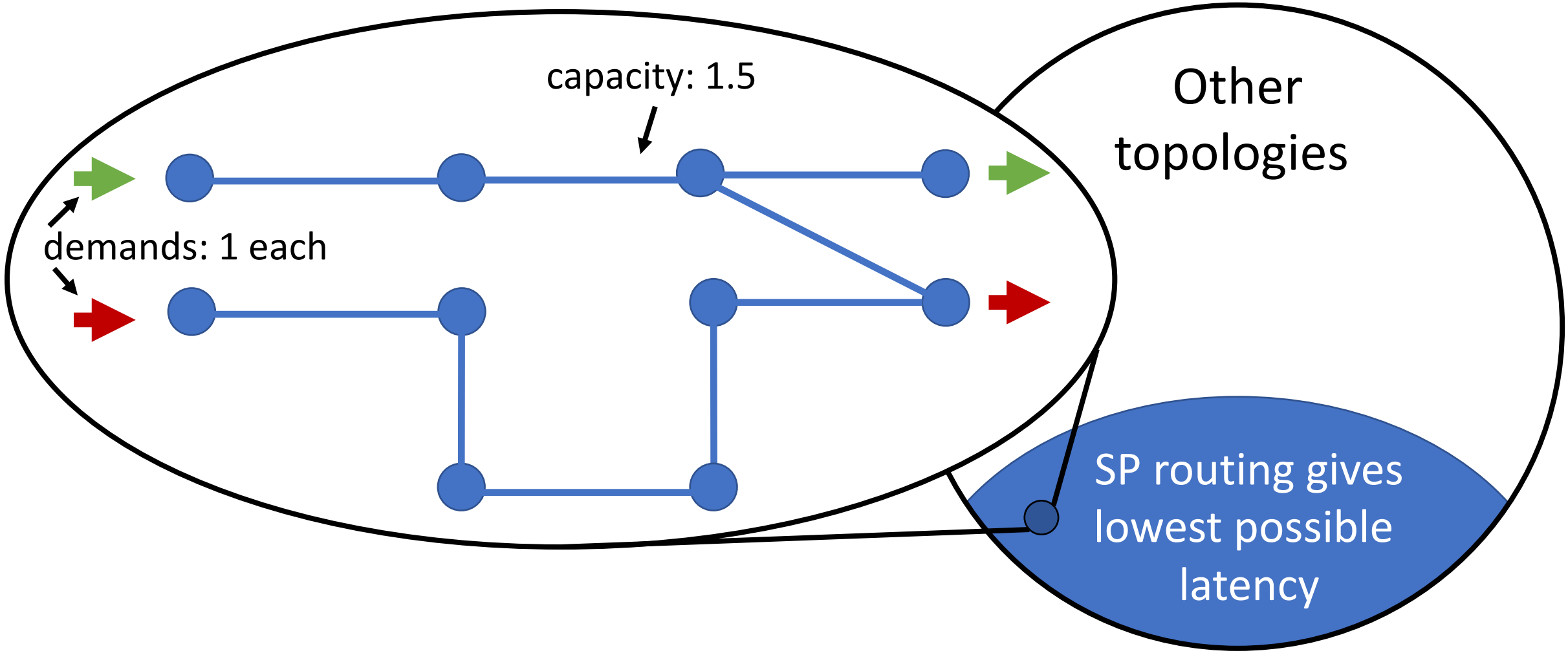
Shortest-path routing doesn’t yield lowest latency on all topologies



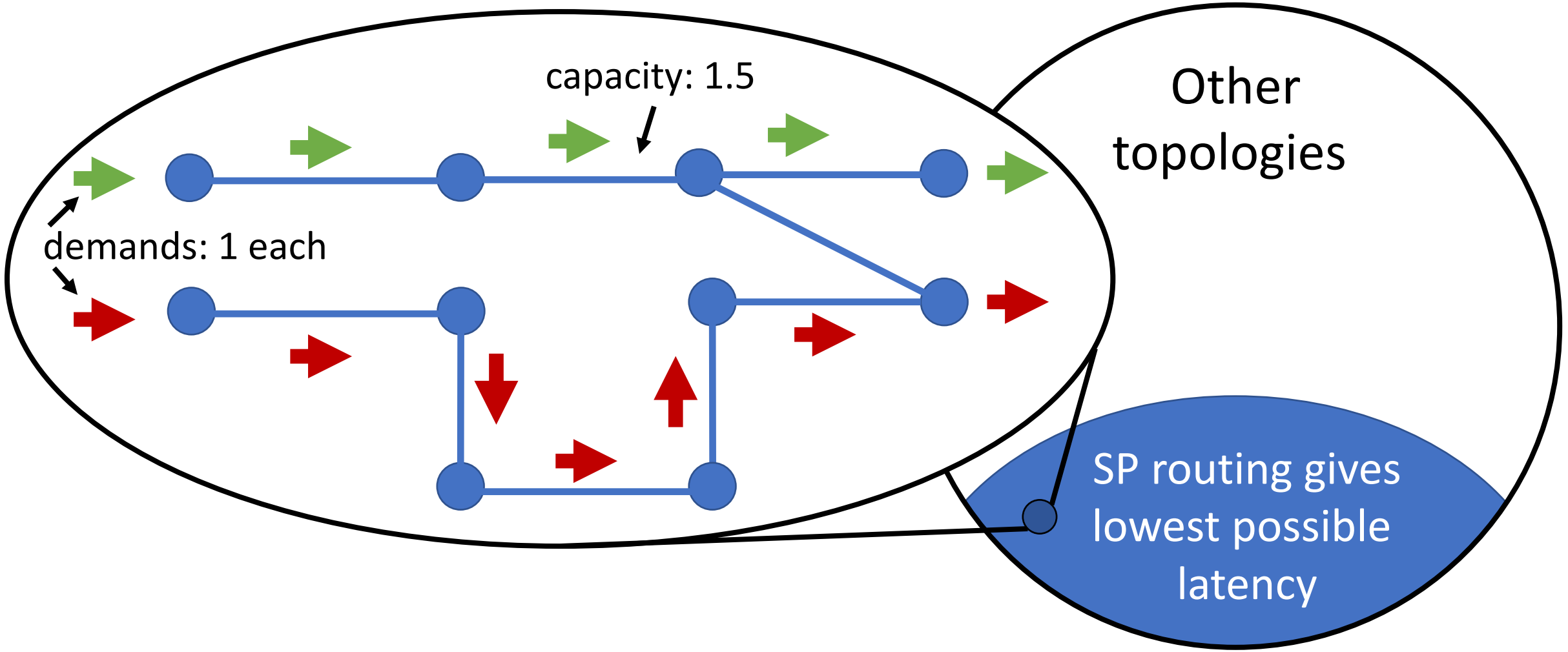
Shortest-path routing doesn't yield lowest latency on all topologies



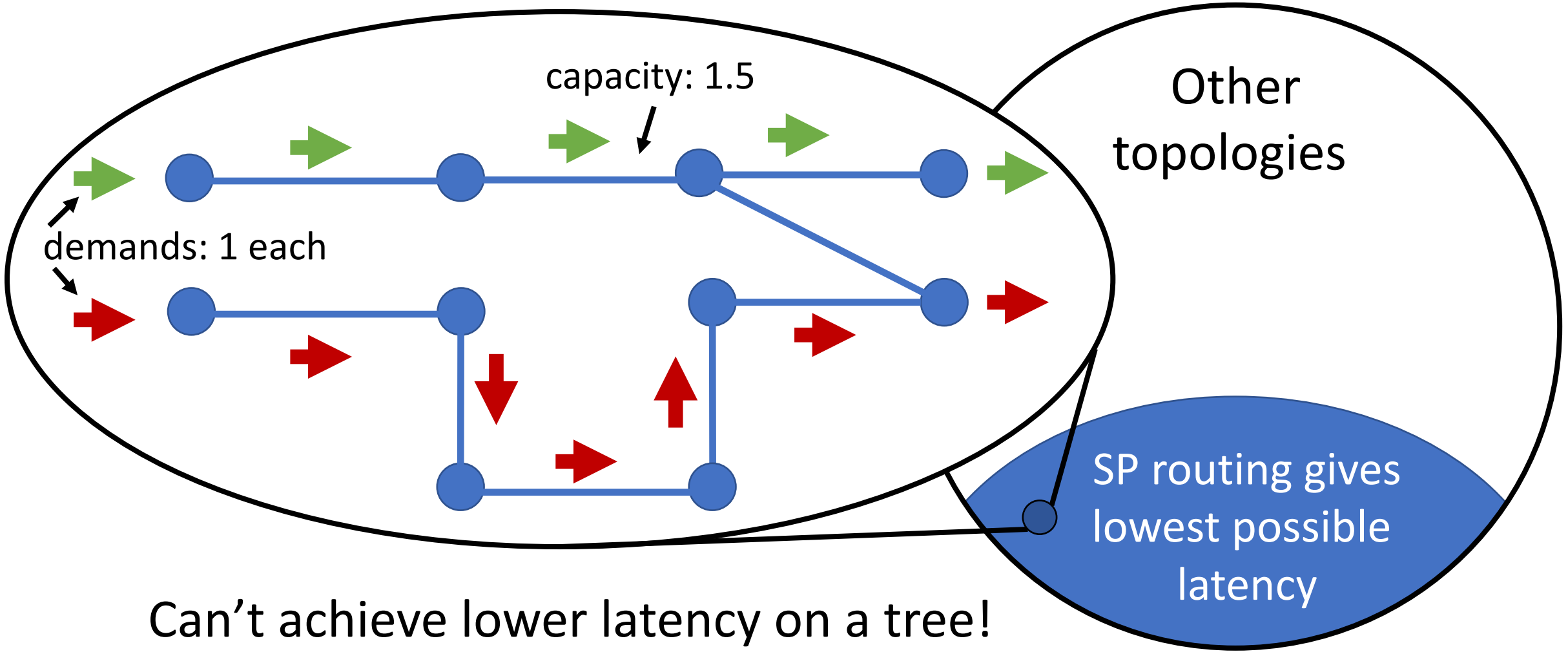
Shortest-path routing doesn't yield lowest latency on all topologies



Shortest-path routing doesn't yield lowest latency on all topologies

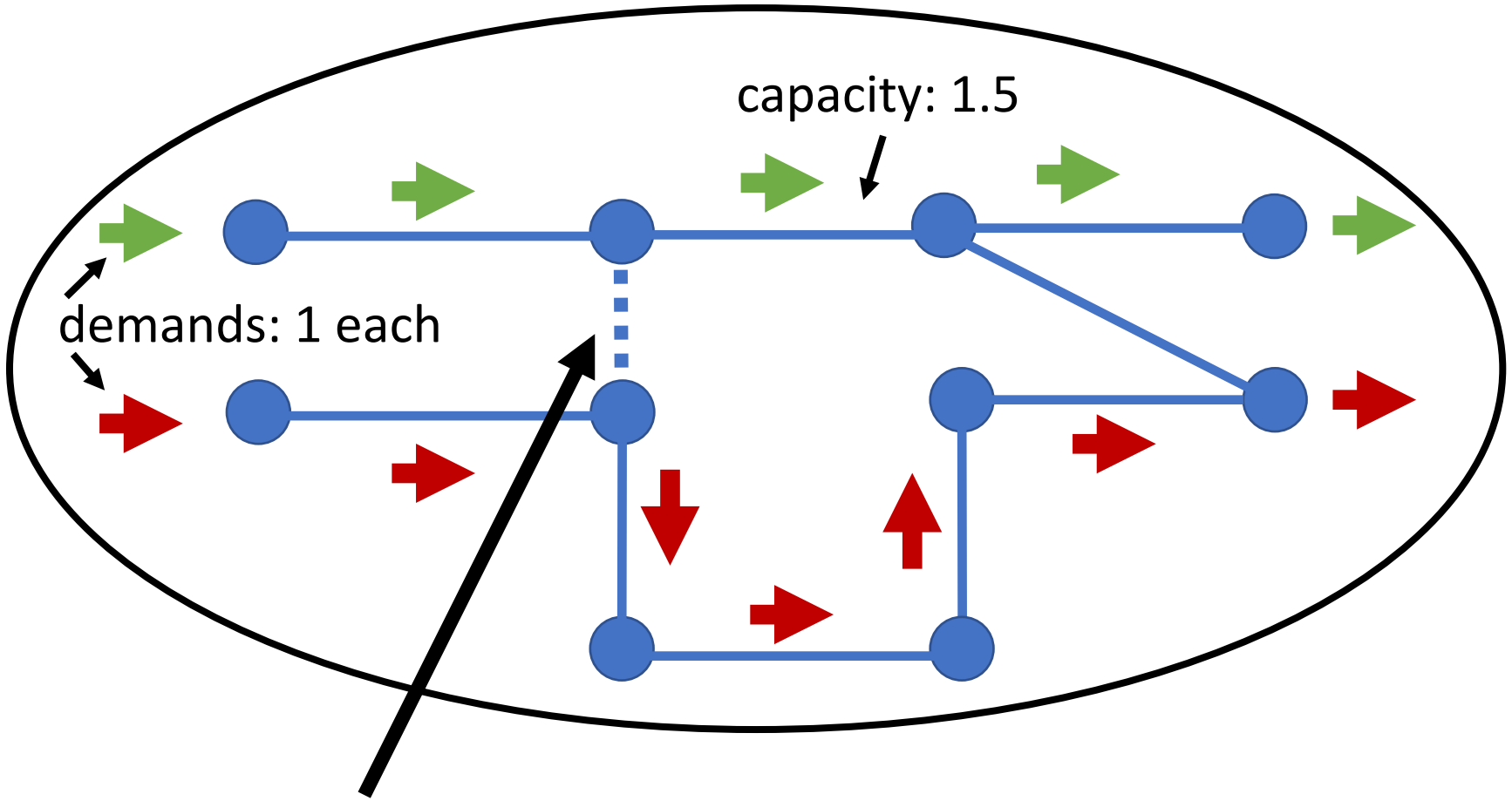


Shortest-path routing doesn't yield lowest latency on all topologies



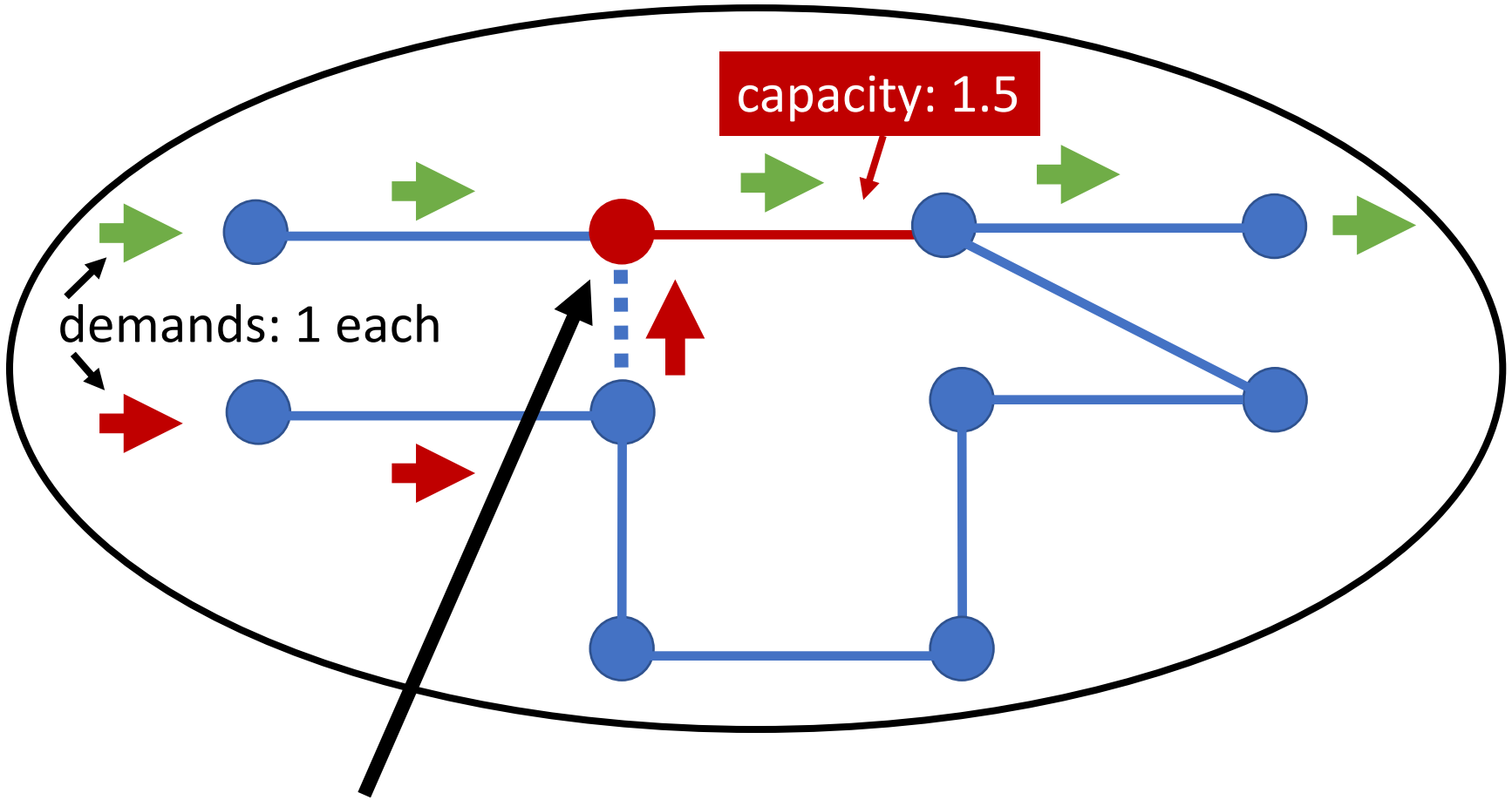
Can't achieve lower latency on a tree!

Better Topologies May Need Better Routing



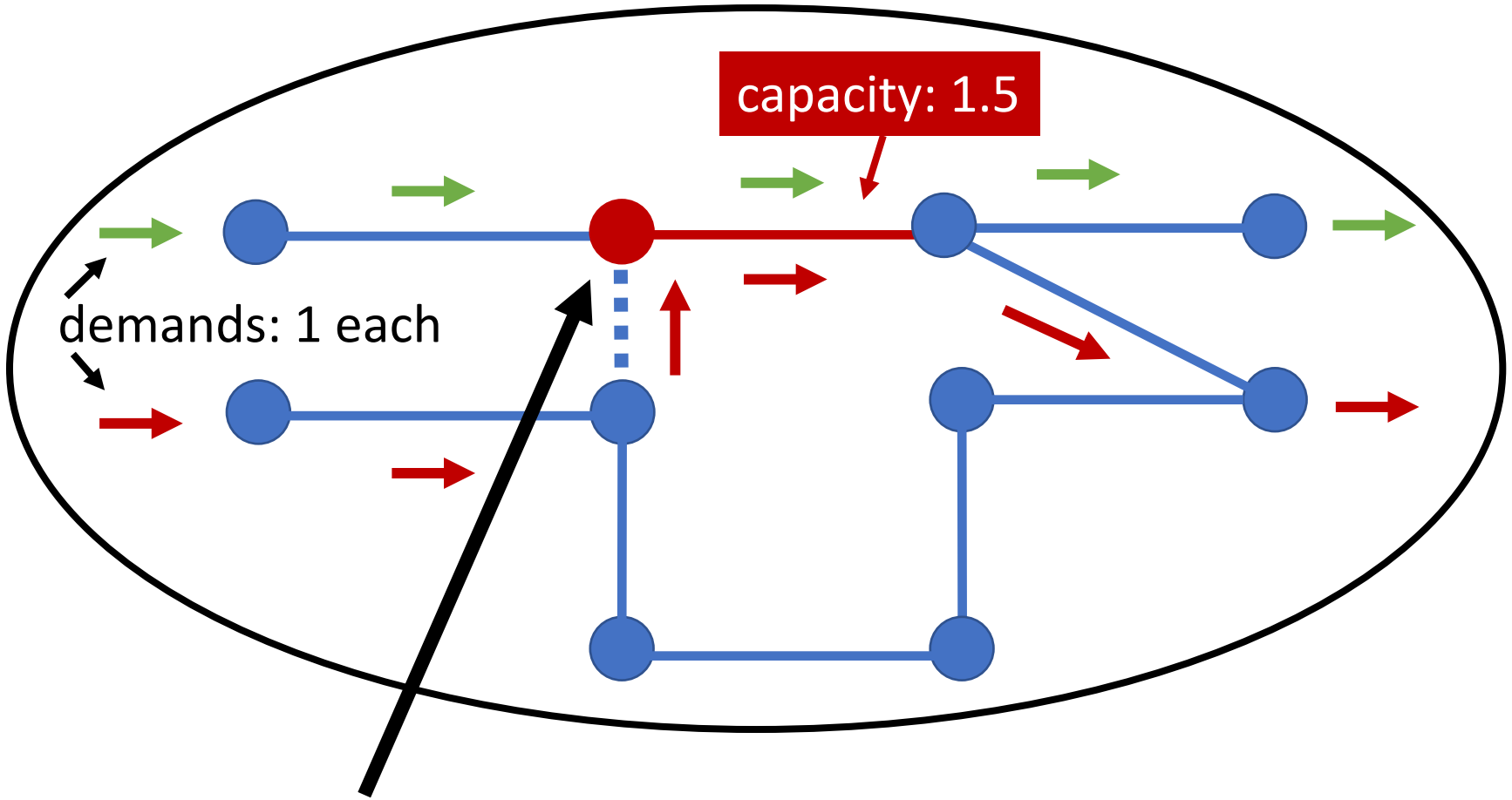
Let’s improve the topology, let’s add redundancy!

Better Topologies May Need Better Routing



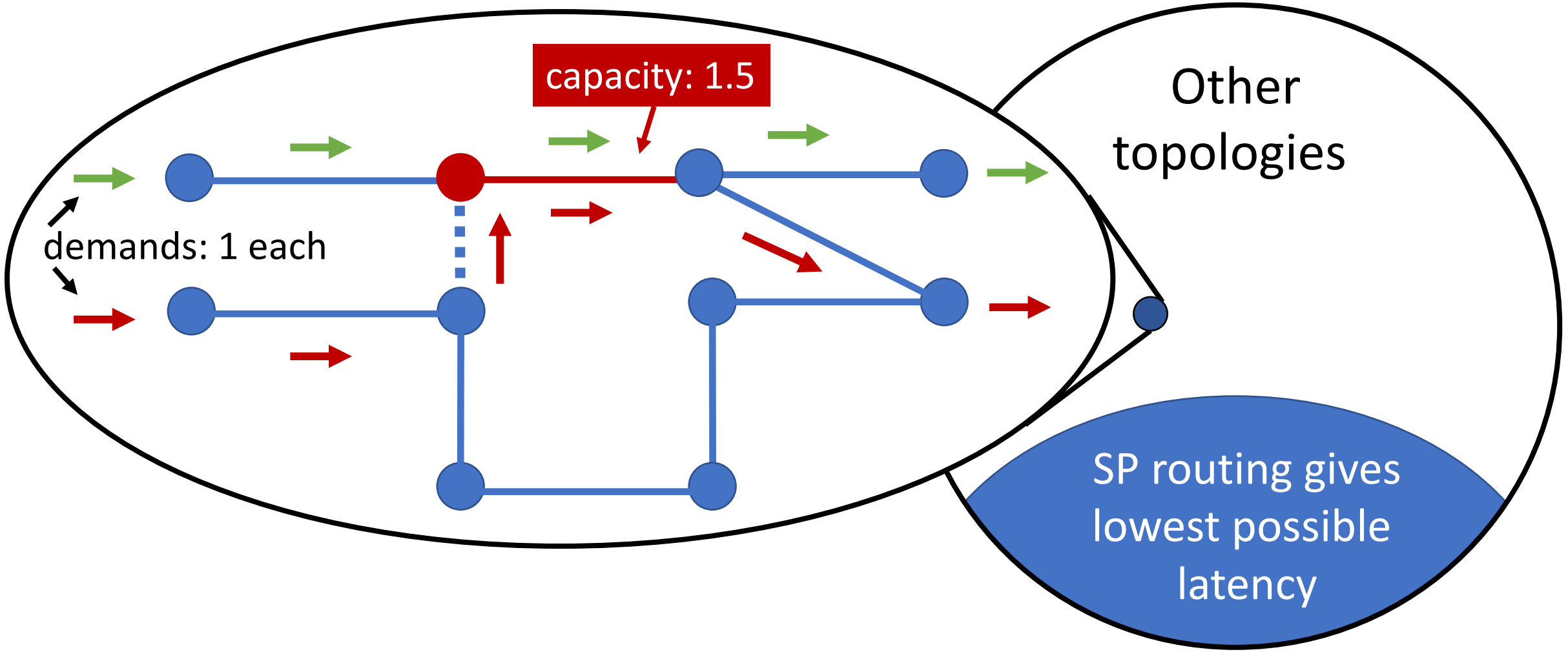
Congestion inflates latency if both aggregates don’t fit.

Better Topologies May Need Better Routing

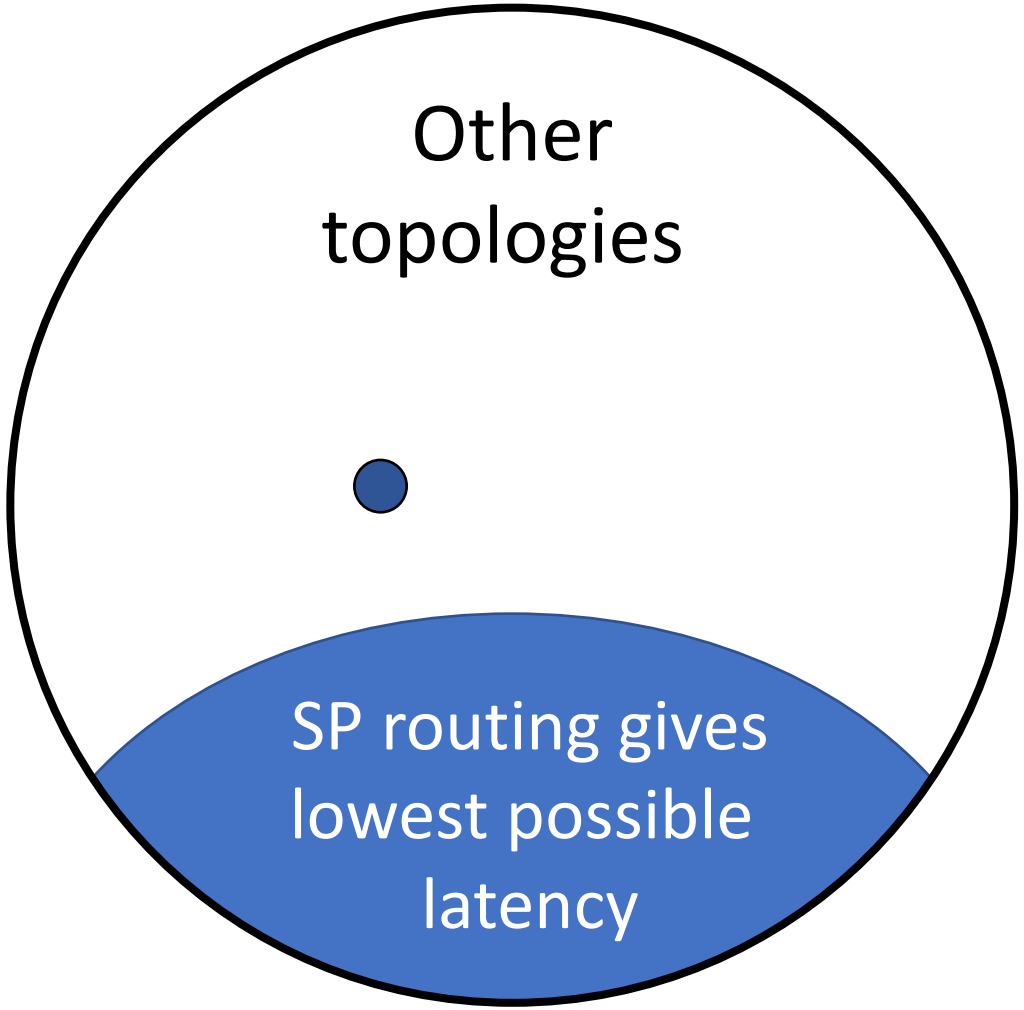


Congestion control makes aggregates fit; hurts throughput

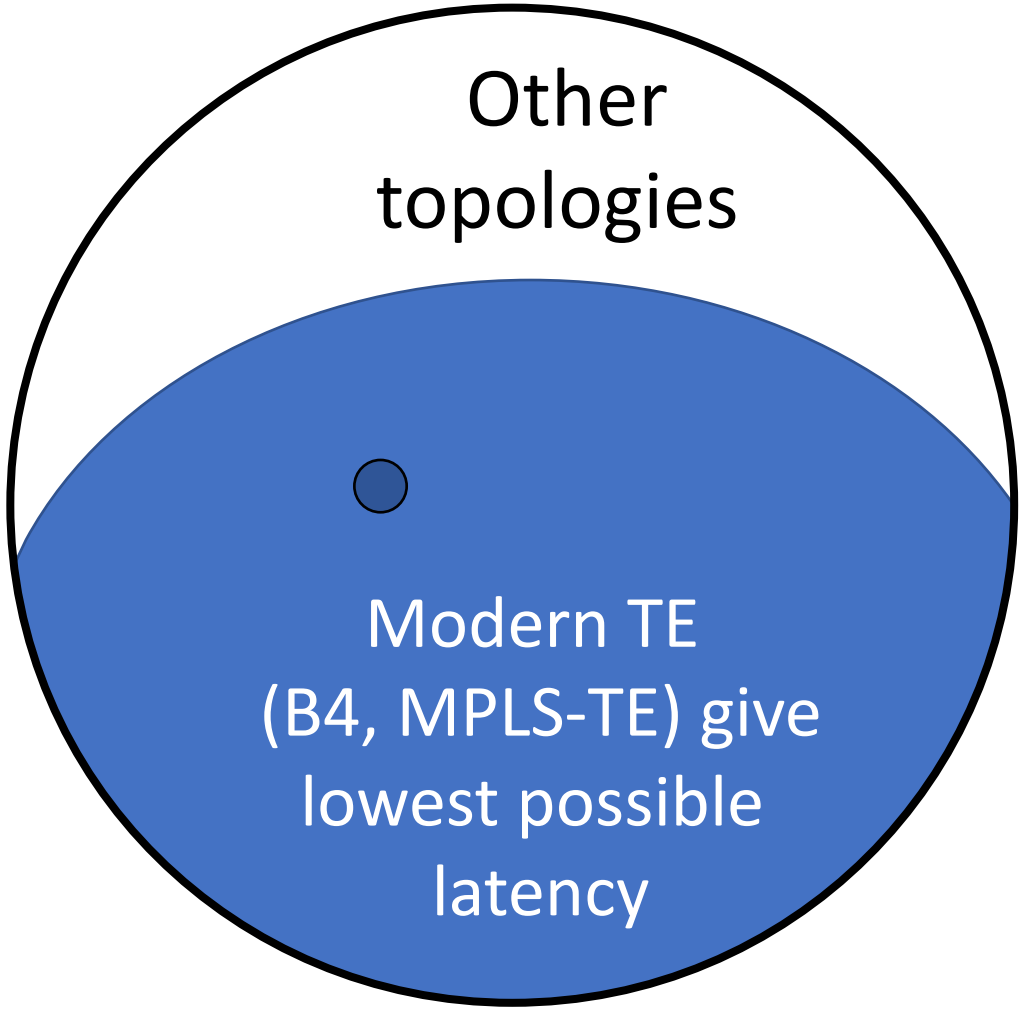
Better Topologies May Need Better Routing



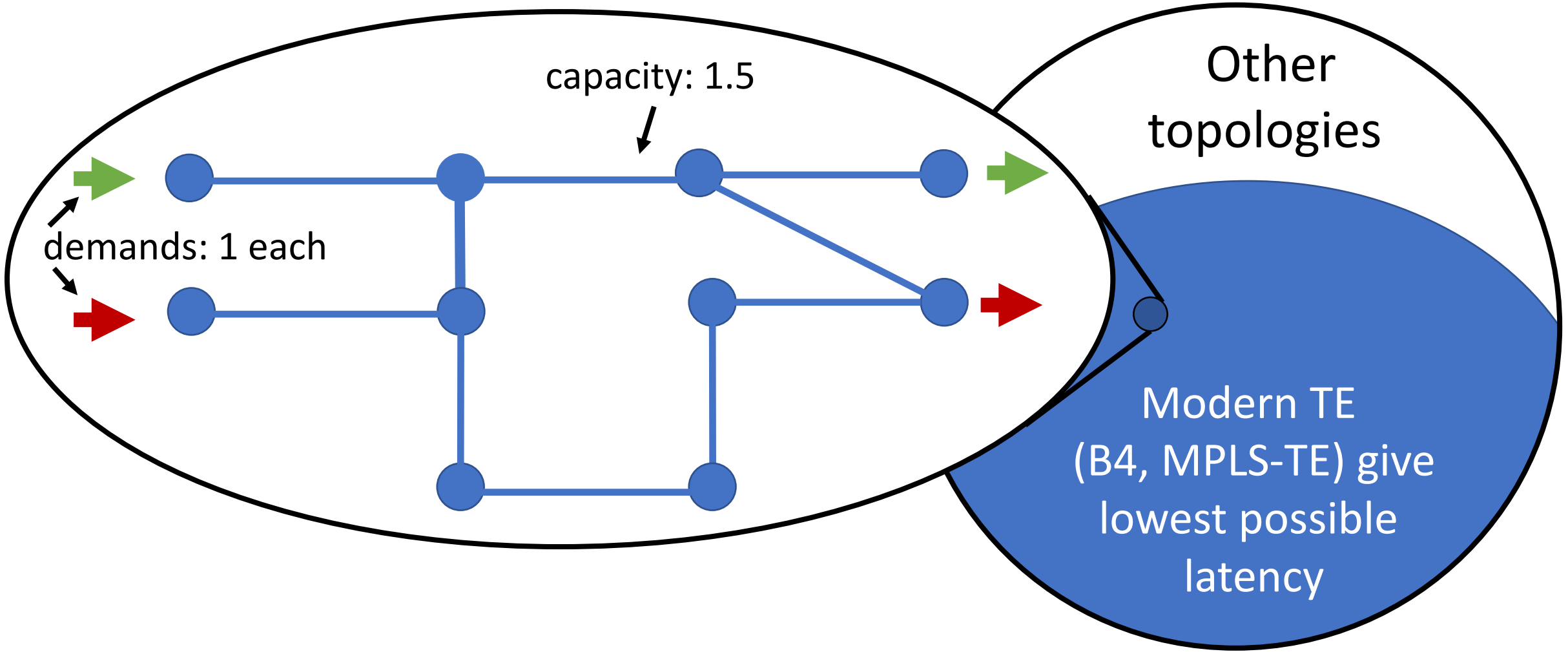
Better Topologies May Need Better Routing



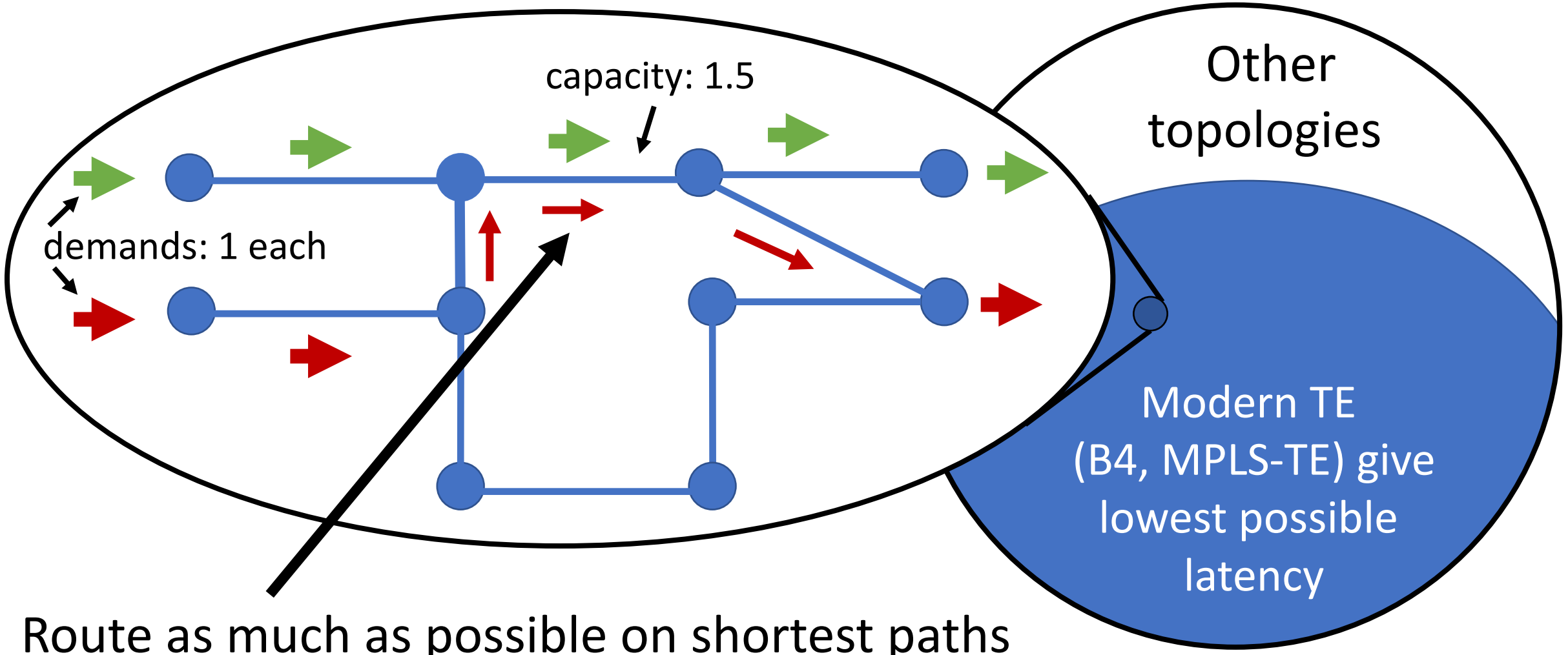
Better Topologies May Need Better Routing



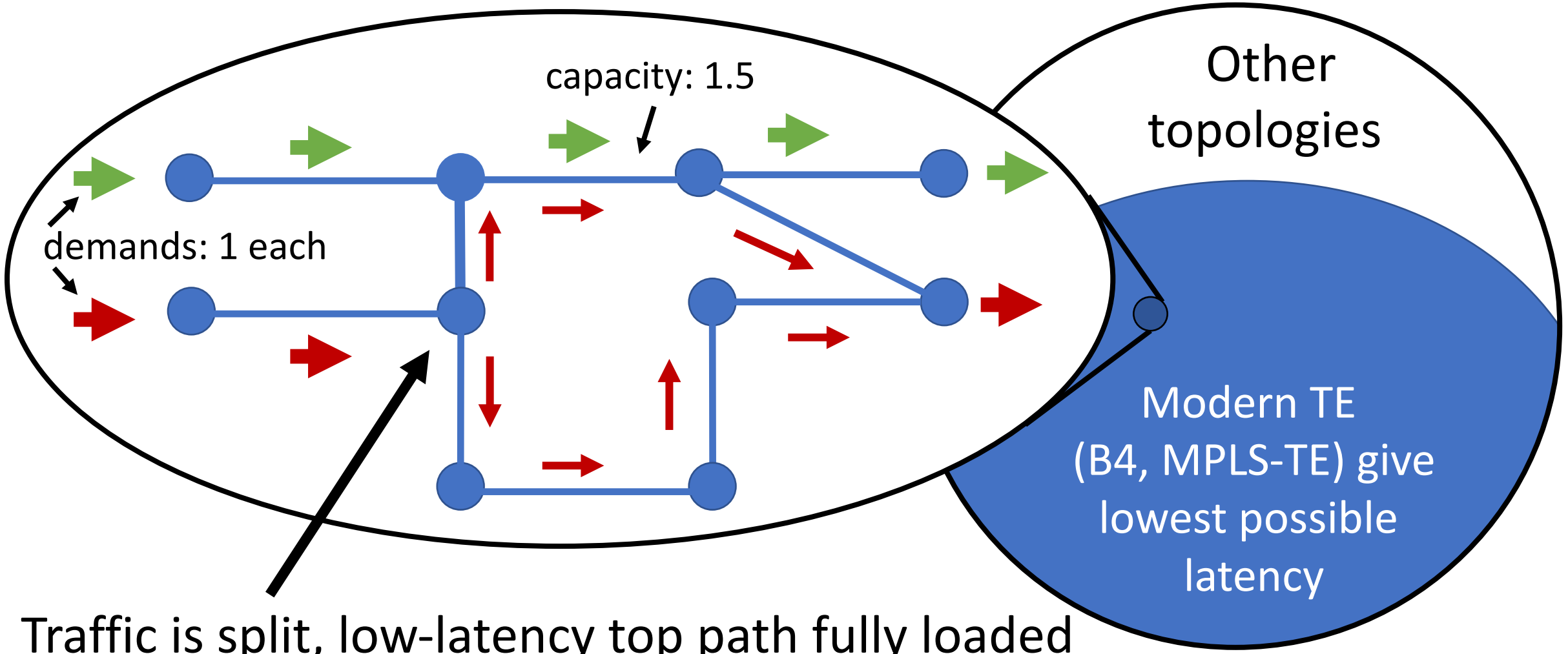
Better Topologies May Need Better Routing



Better Topologies May Need Better Routing

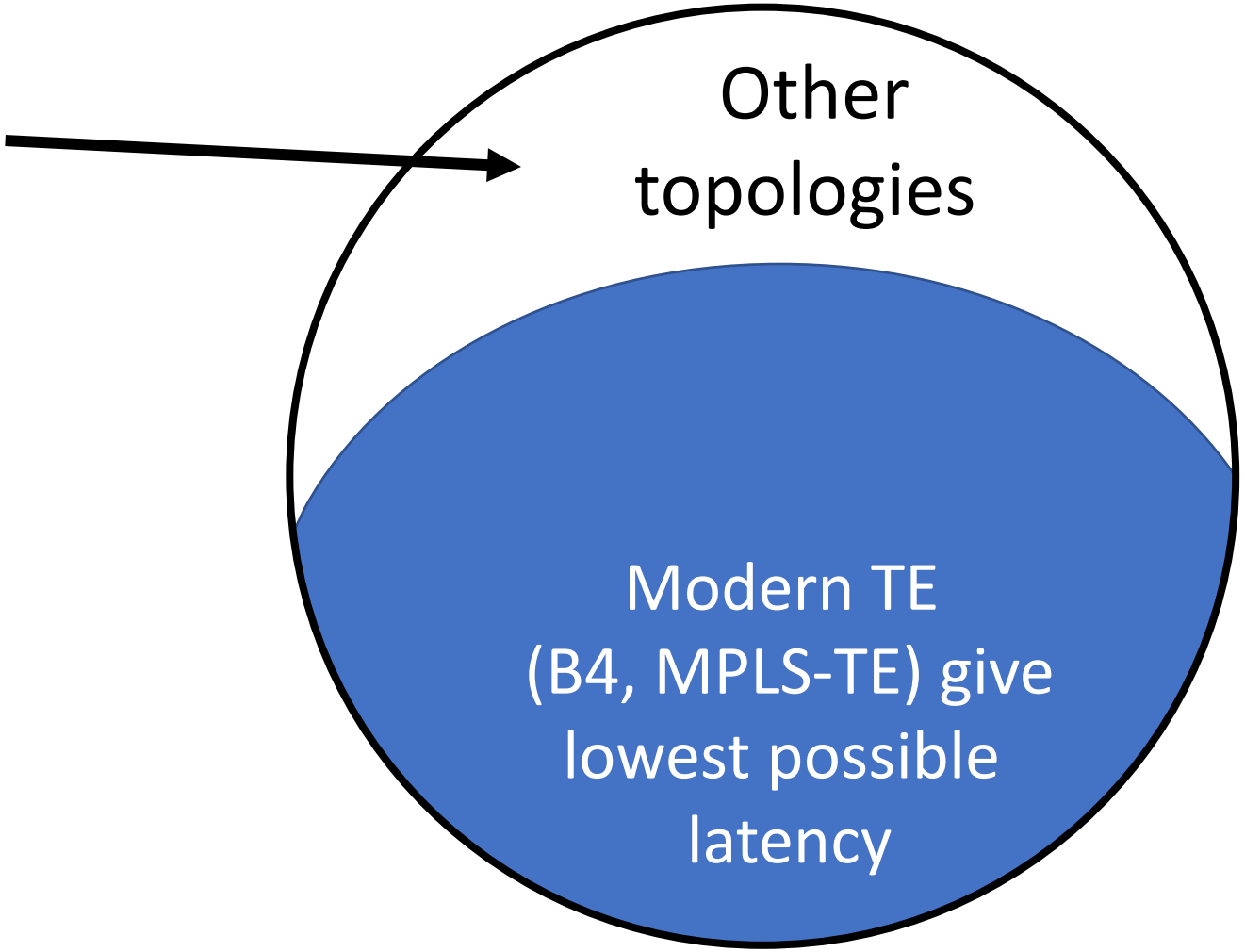


Better Topologies May Need Better Routing



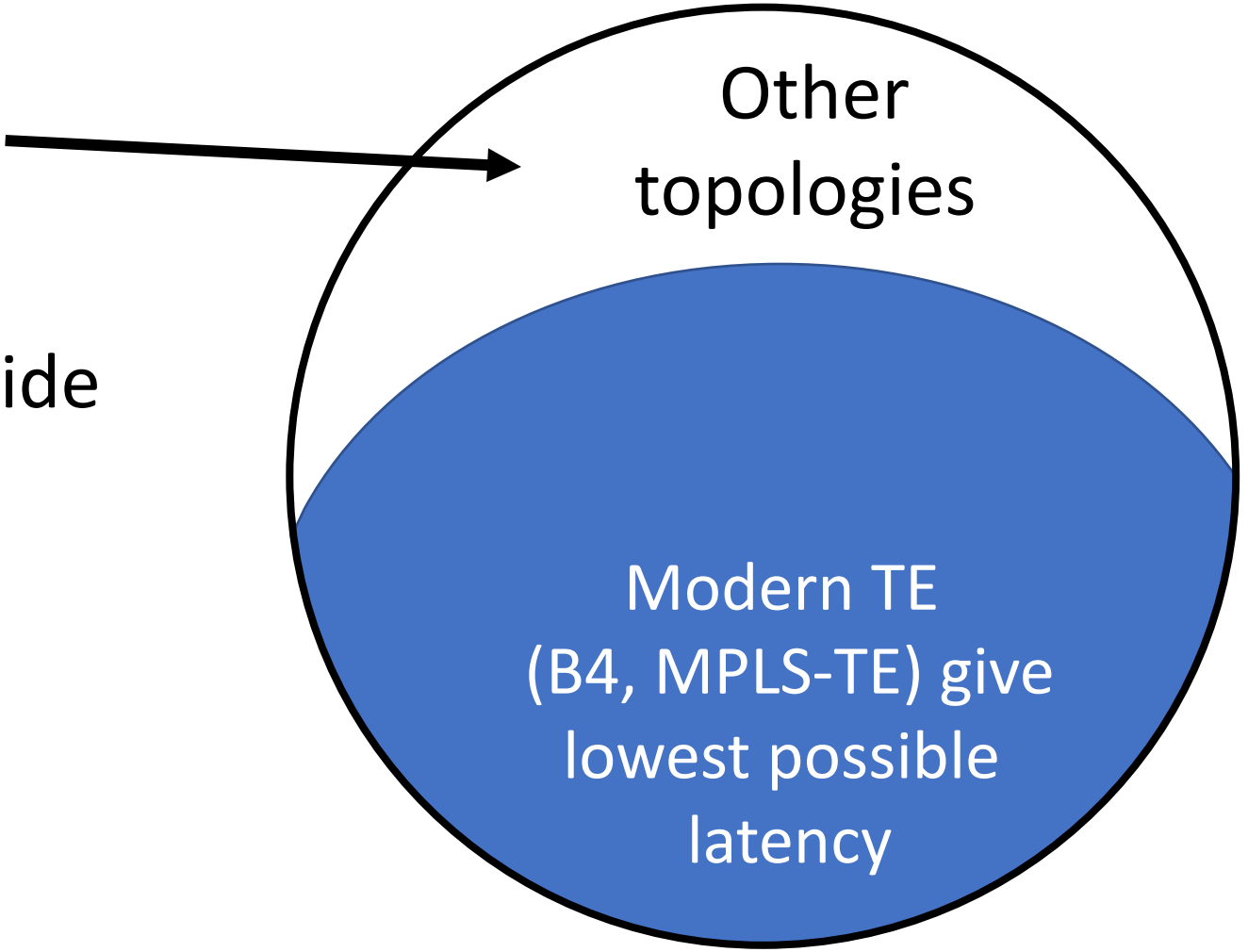
Do Even Better Topologies Need Even Better Routing?

- Do any topologies fall in this region?



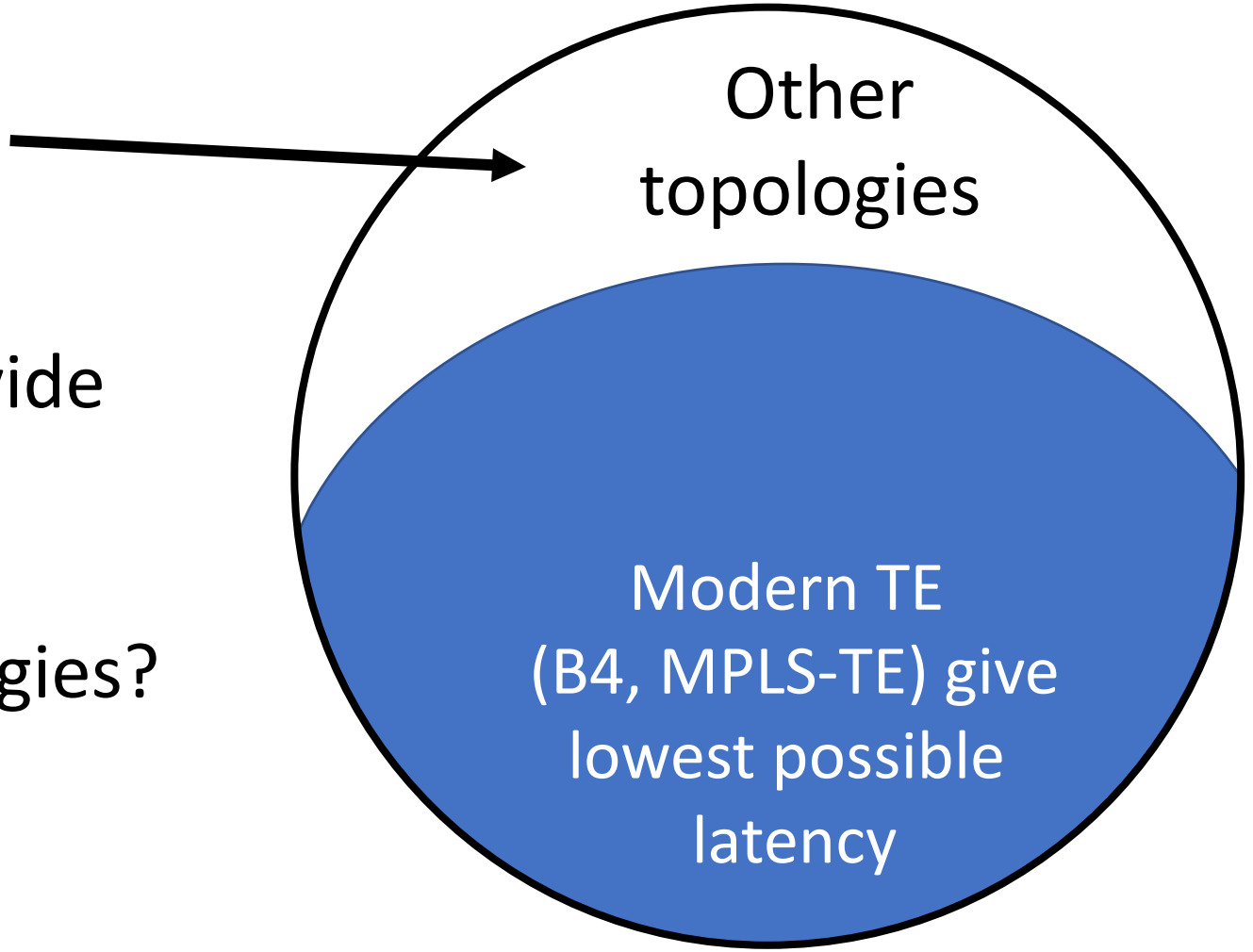
Do Even Better Topologies Need Even Better Routing?

- Do any topologies fall in this region?
- If so, do any of them have a greater potential to provide low latency?



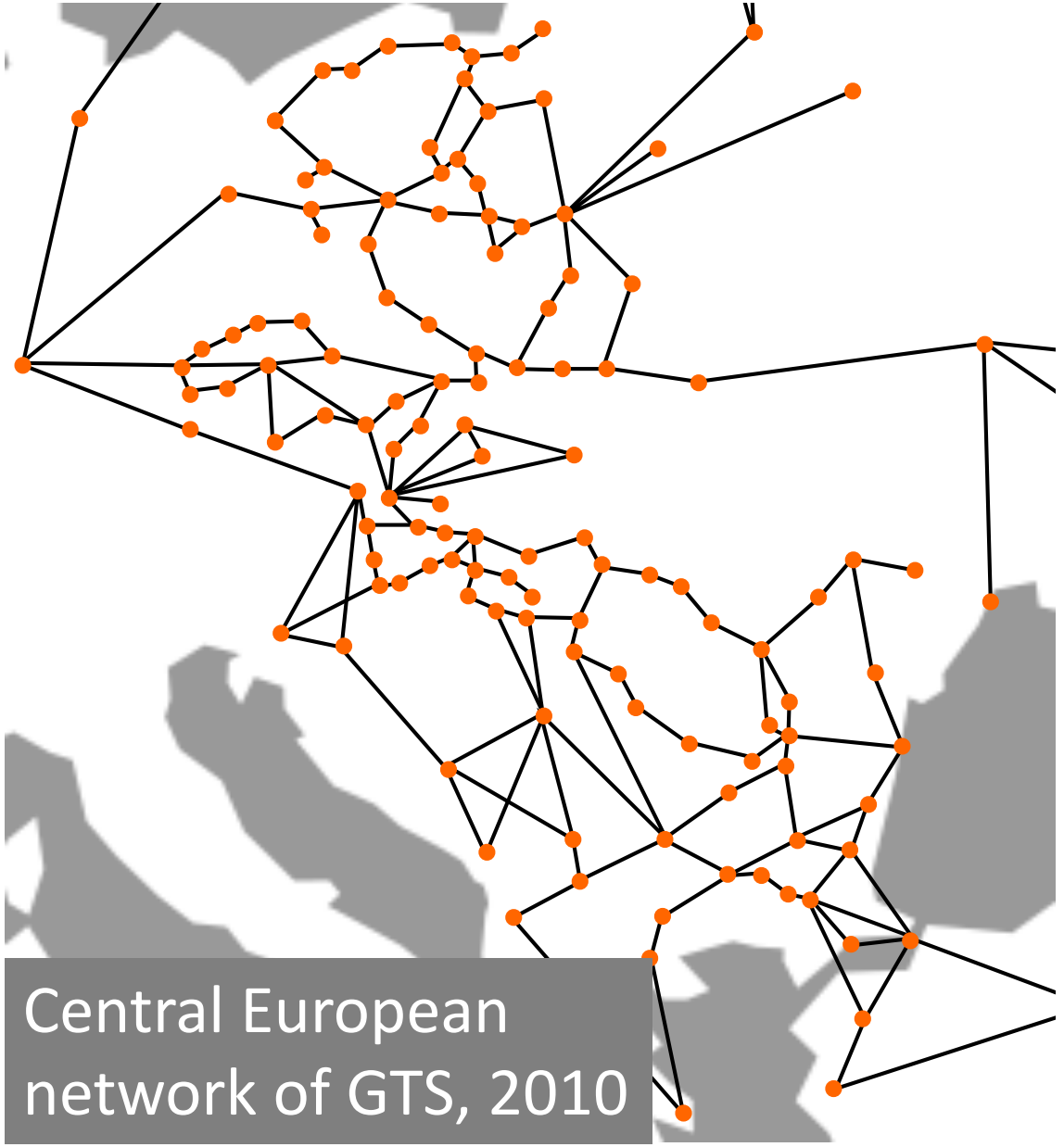
Do Even Better Topologies Need Even Better Routing?

- Do any topologies fall in this region?
- If so, do any of them have a greater potential to provide low latency?
- Why does current routing do poorly on those topologies?

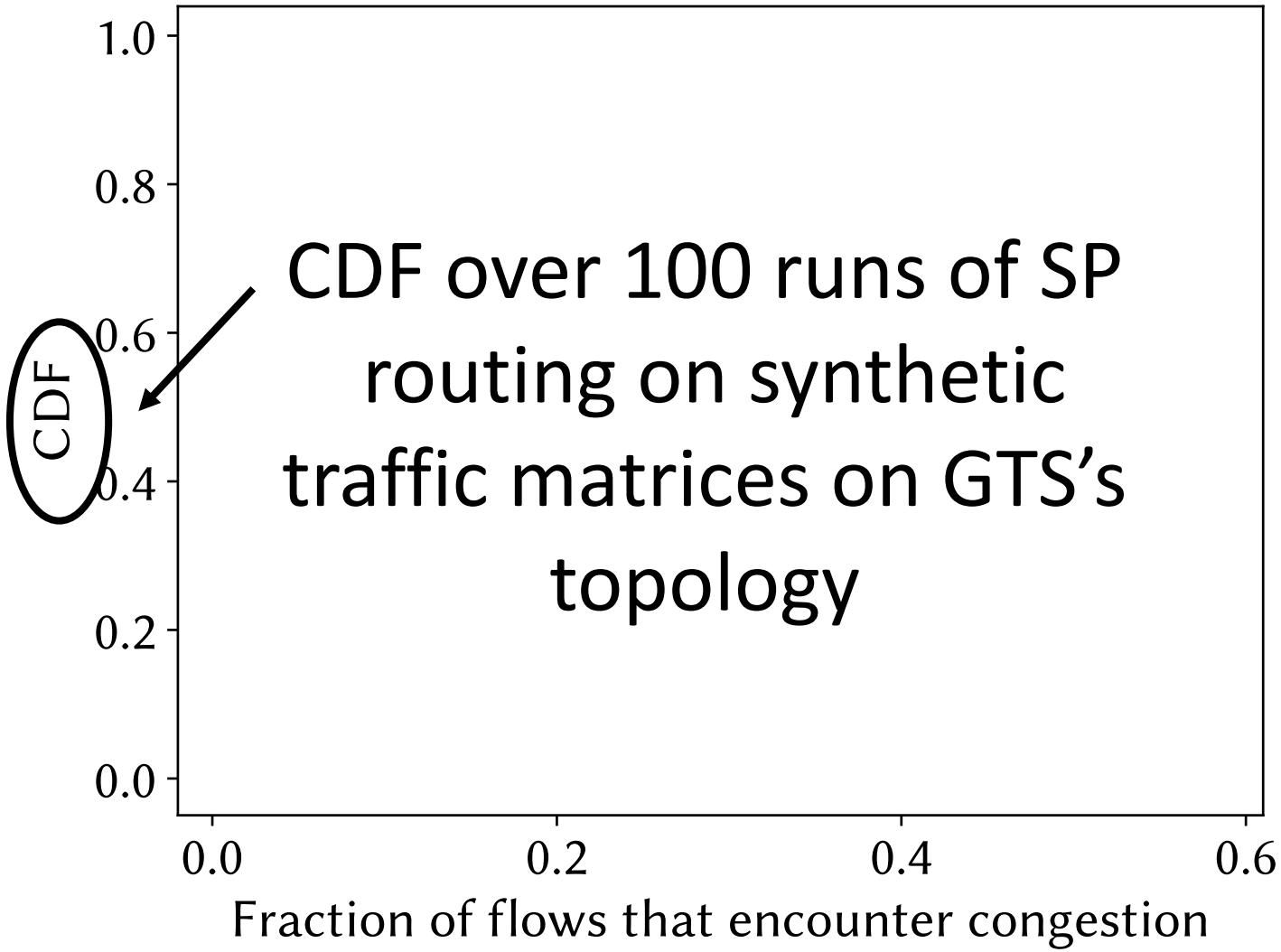


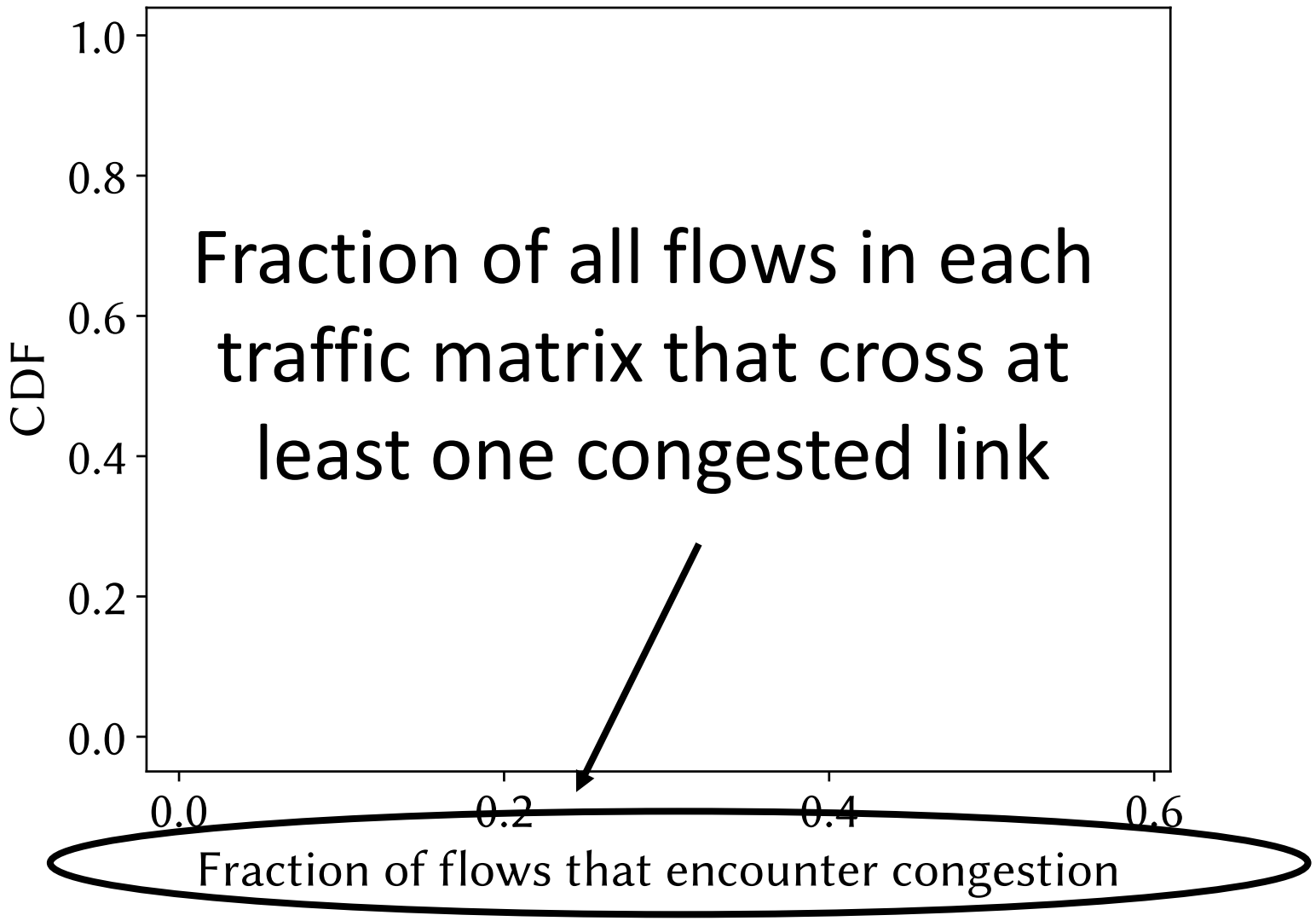
Limitations of Today's Routing

Proof by example. Consider this real-world ISP topology…

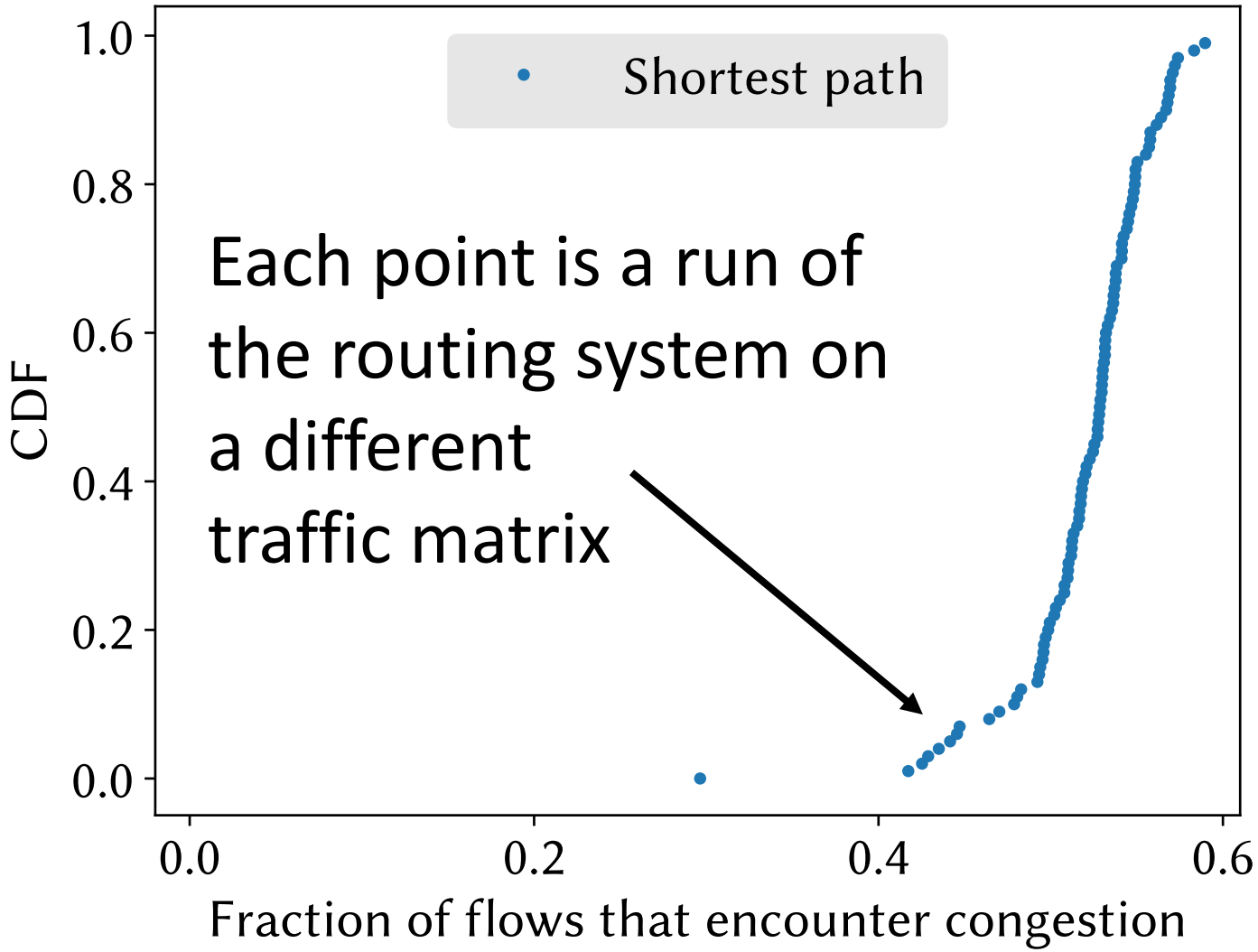


Central European network of GTS, 2010


1.0
0.8
0.6
0.4
0.2
0.0
CDF
CDF over 100 runs of SP routing on synthetic traffic matrices on GTS's topology
0.0
0.2
0.4
0.6
Fraction of flows that encounter congestion

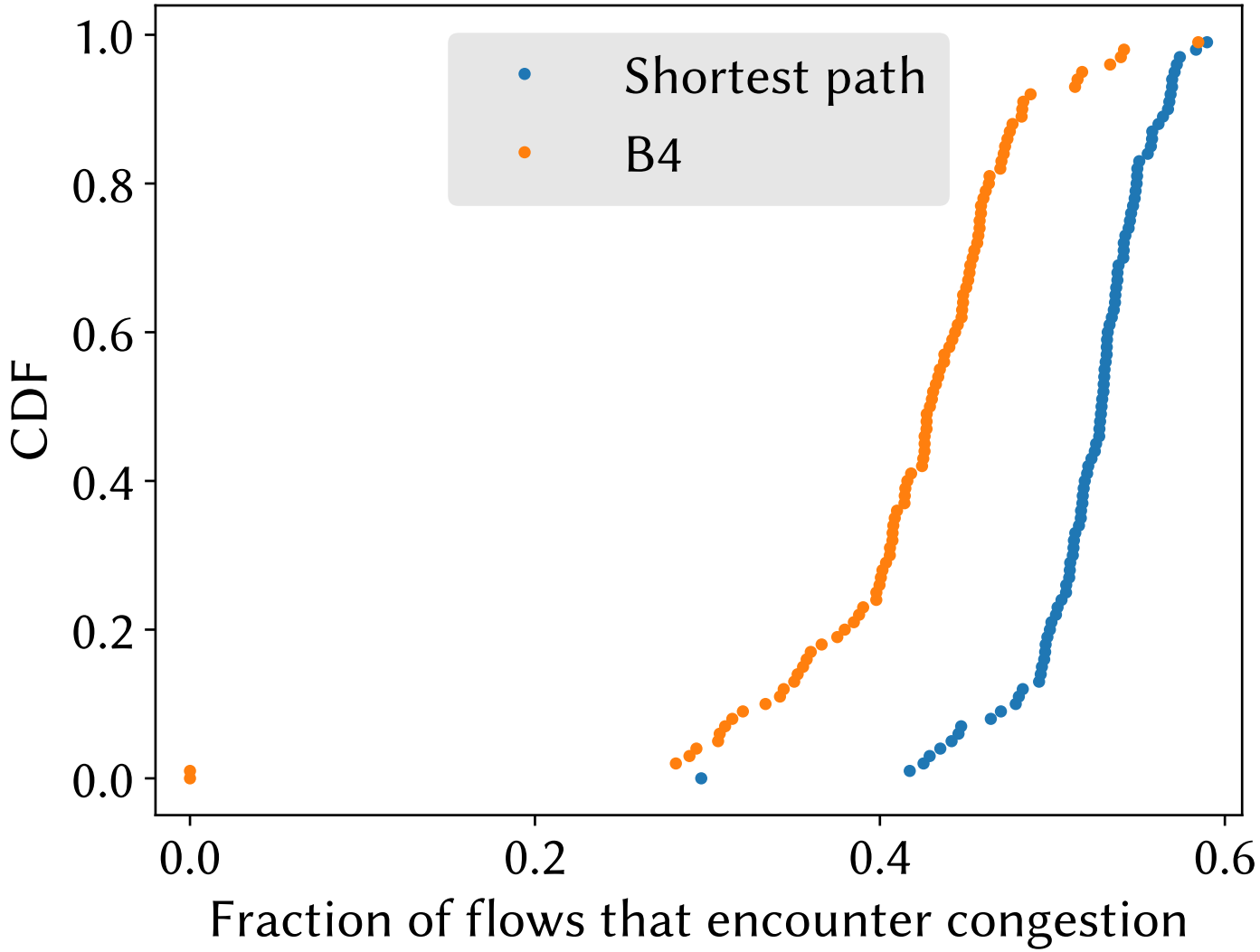

1.0
0.8
0.6
0.4
0.2
0.0
CDF
Fraction of all flows in each traffic matrix that cross at least one congested link
0.0
0.2
0.4
0.6
Fraction of flows that encounter congestion

SP does poorly, as expected

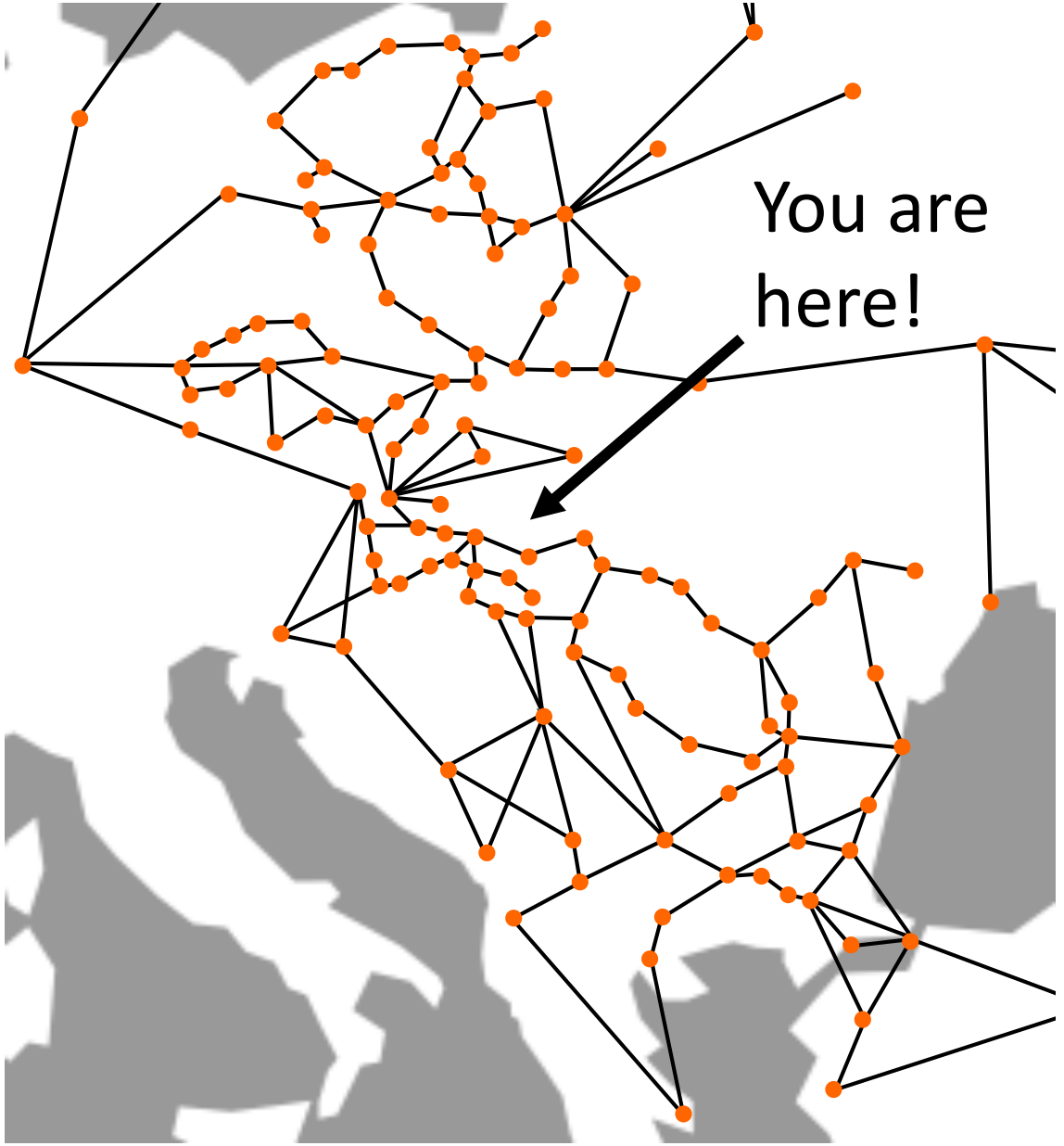


B4 for the win … sort of

Jain, Sushant, et al. "B4: Experience with a globally-deployed software defined WAN." *ACM SIGCOMM 2013*

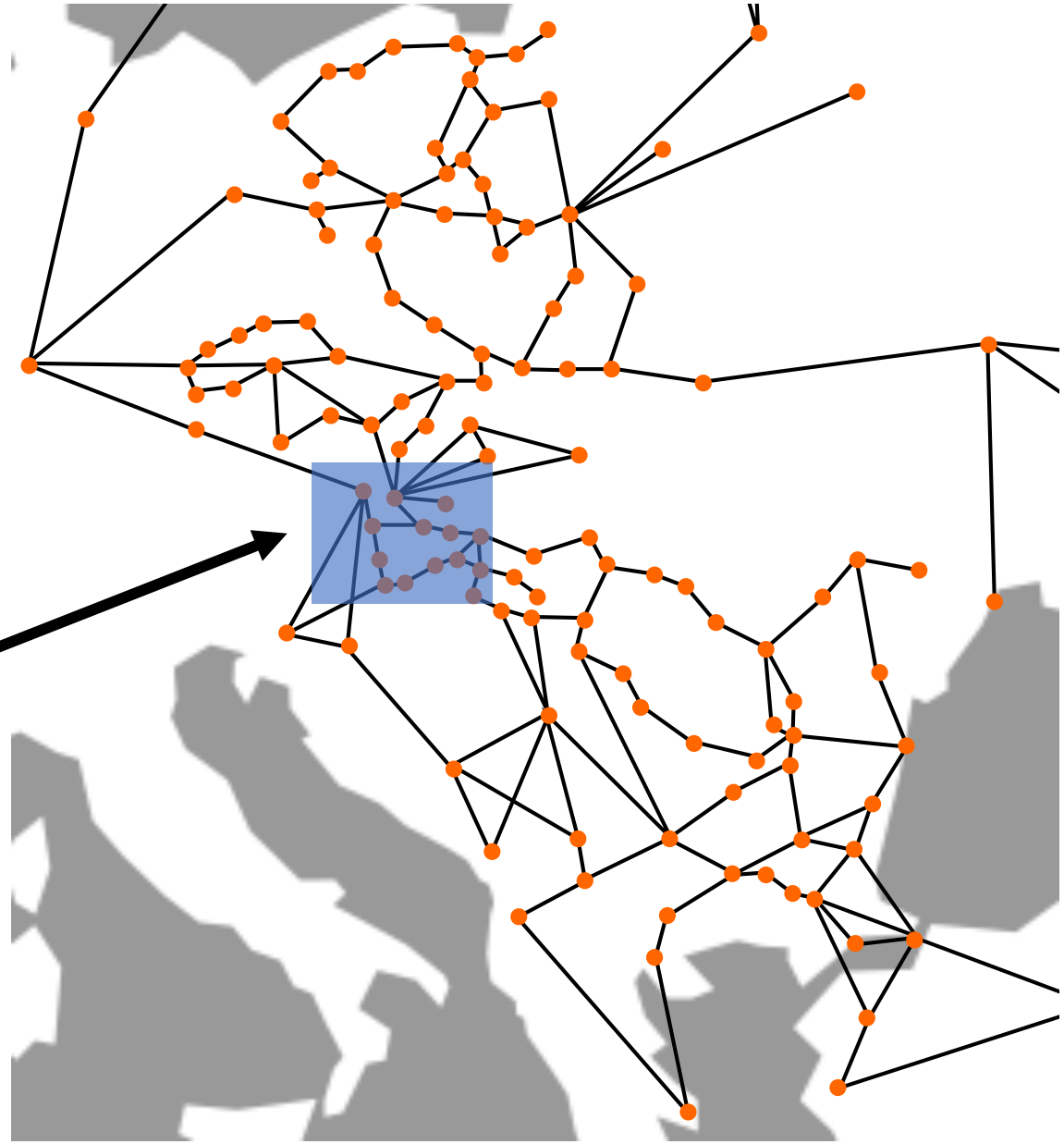


Where does greedy routing such as B4 go wrong?

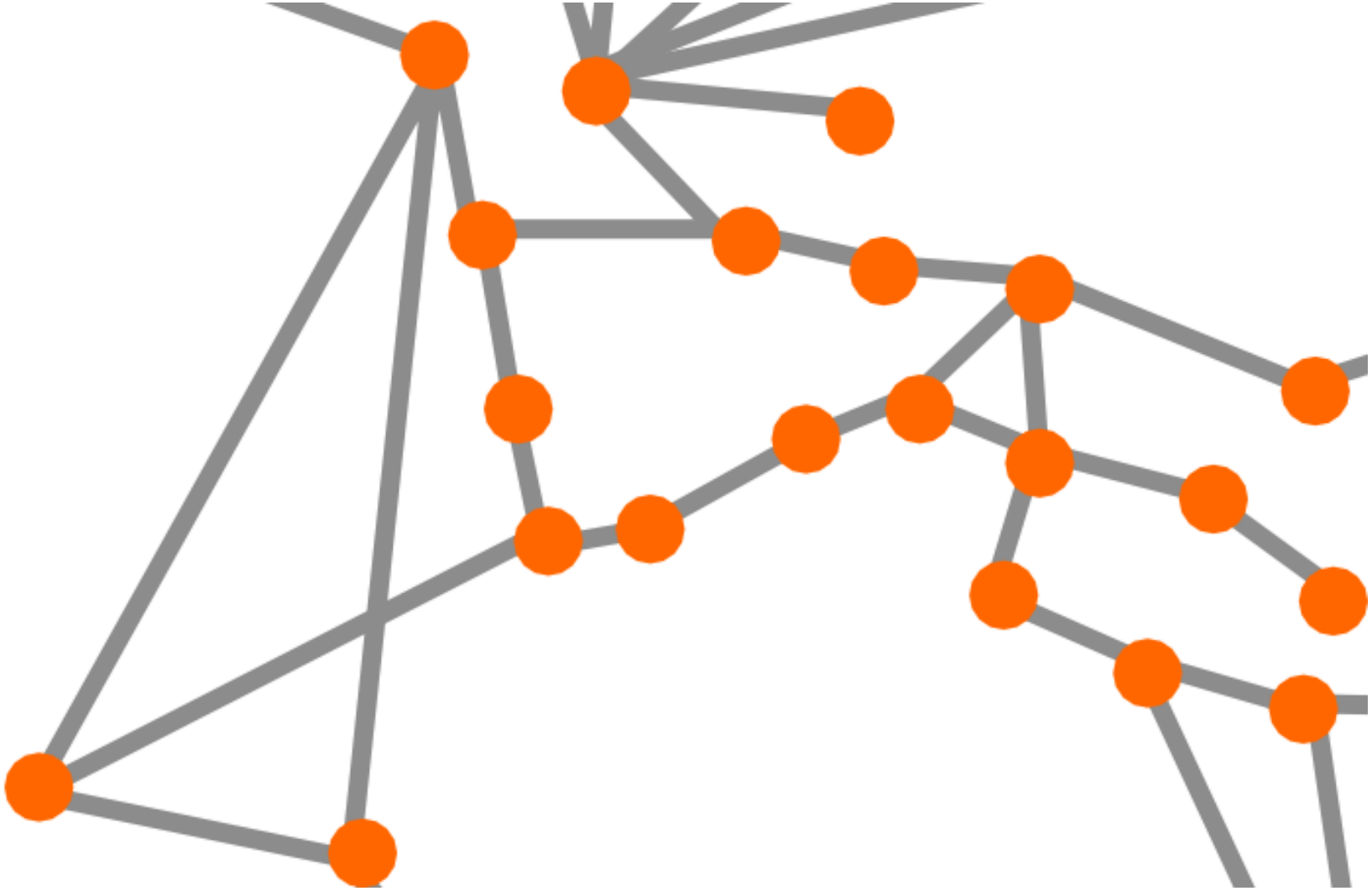


Where does greedy routing such as B4 go wrong?

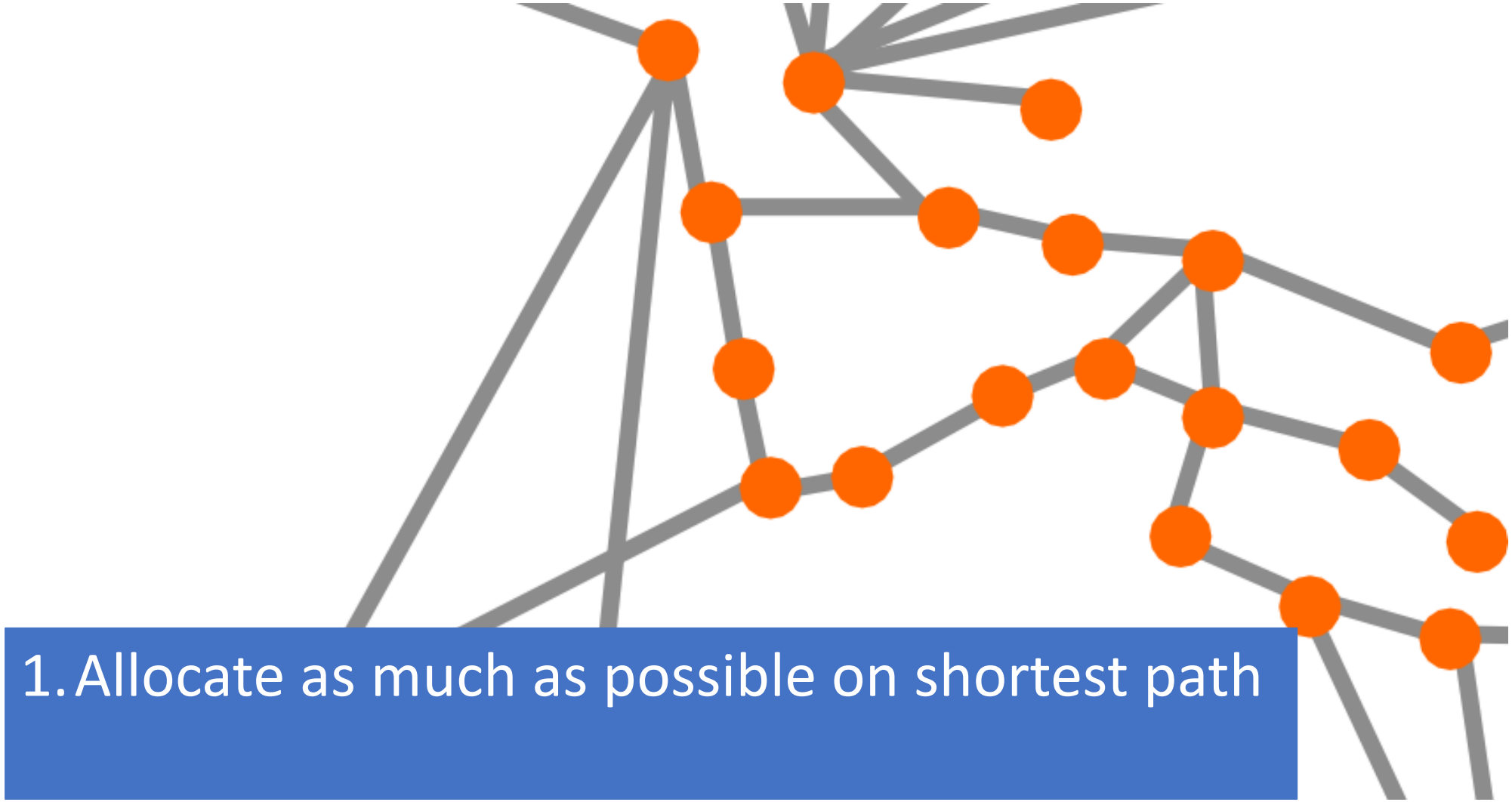
Let’s focus on a small part of the network



Limitations of greedy routing

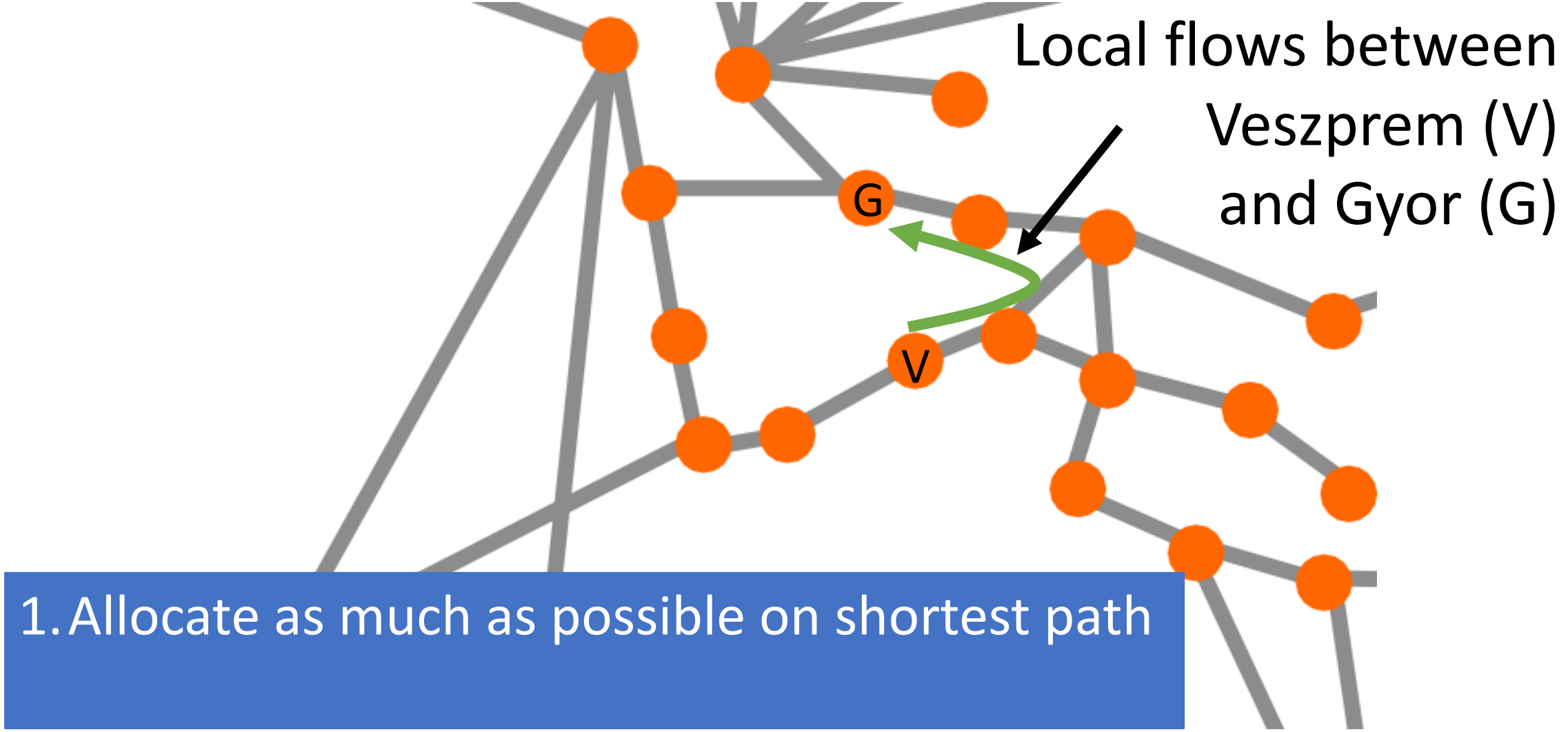


Limitations of greedy routing



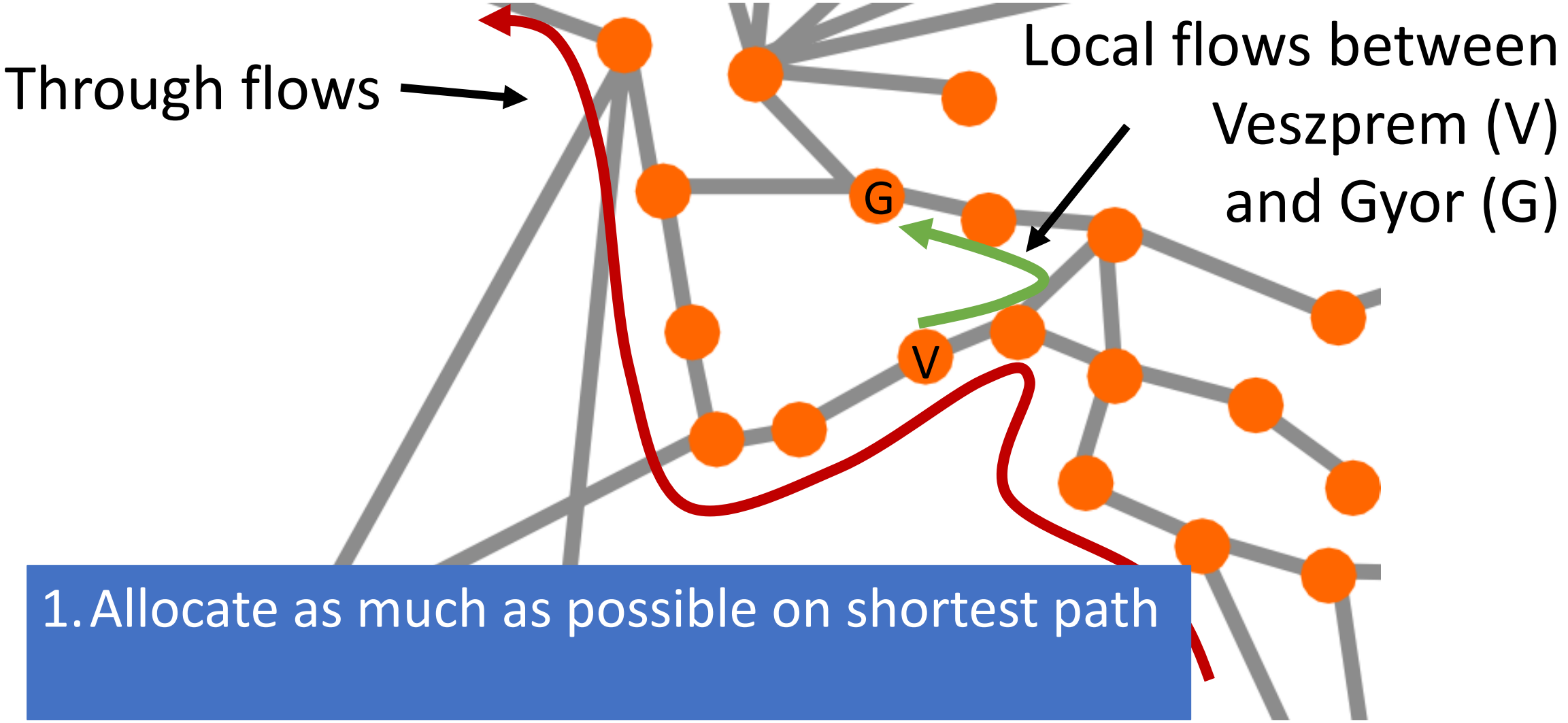
1. Allocate as much as possible on shortest path

Limitations of greedy routing



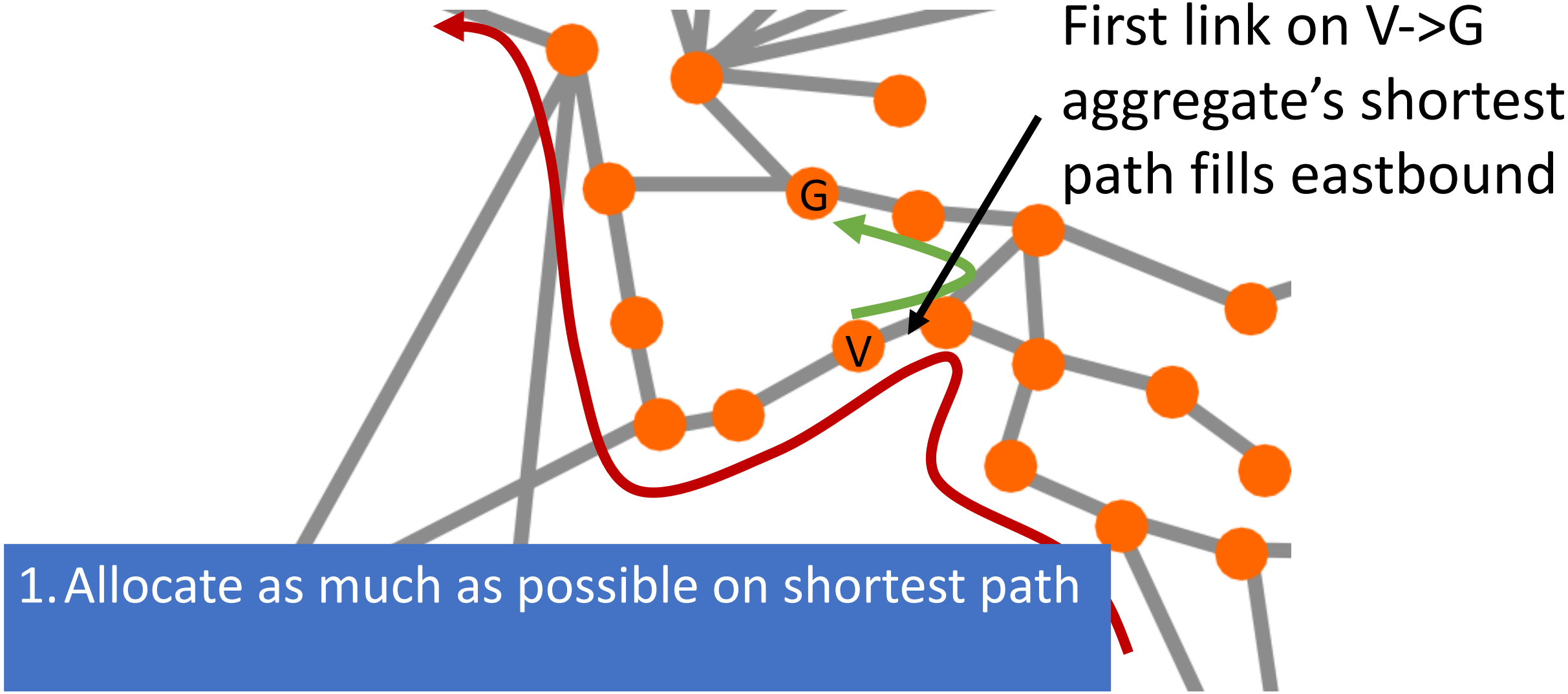
1. Allocate as much as possible on shortest path

Limitations of greedy routing



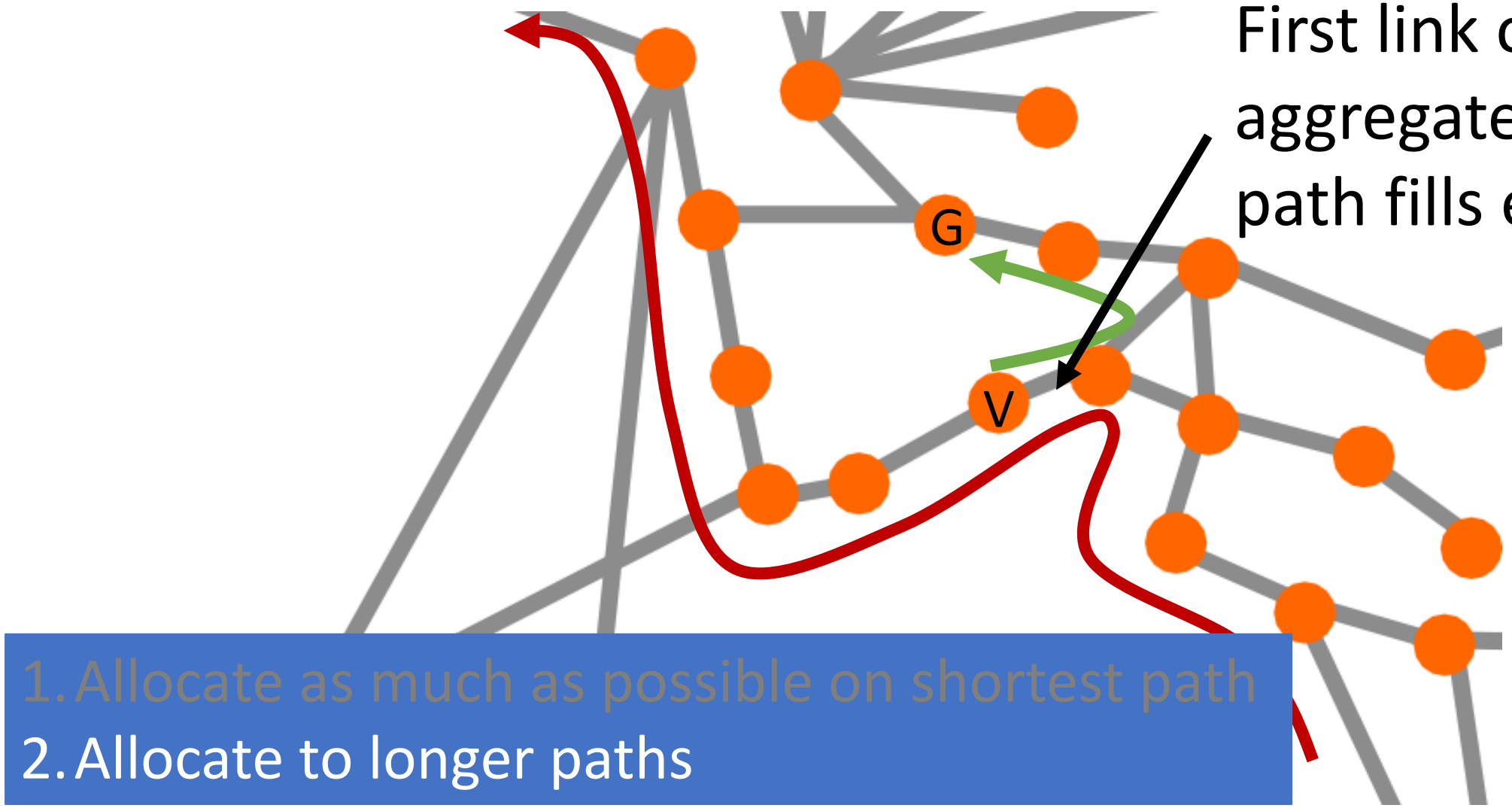
1. Allocate as much as possible on shortest path

Limitations of greedy routing



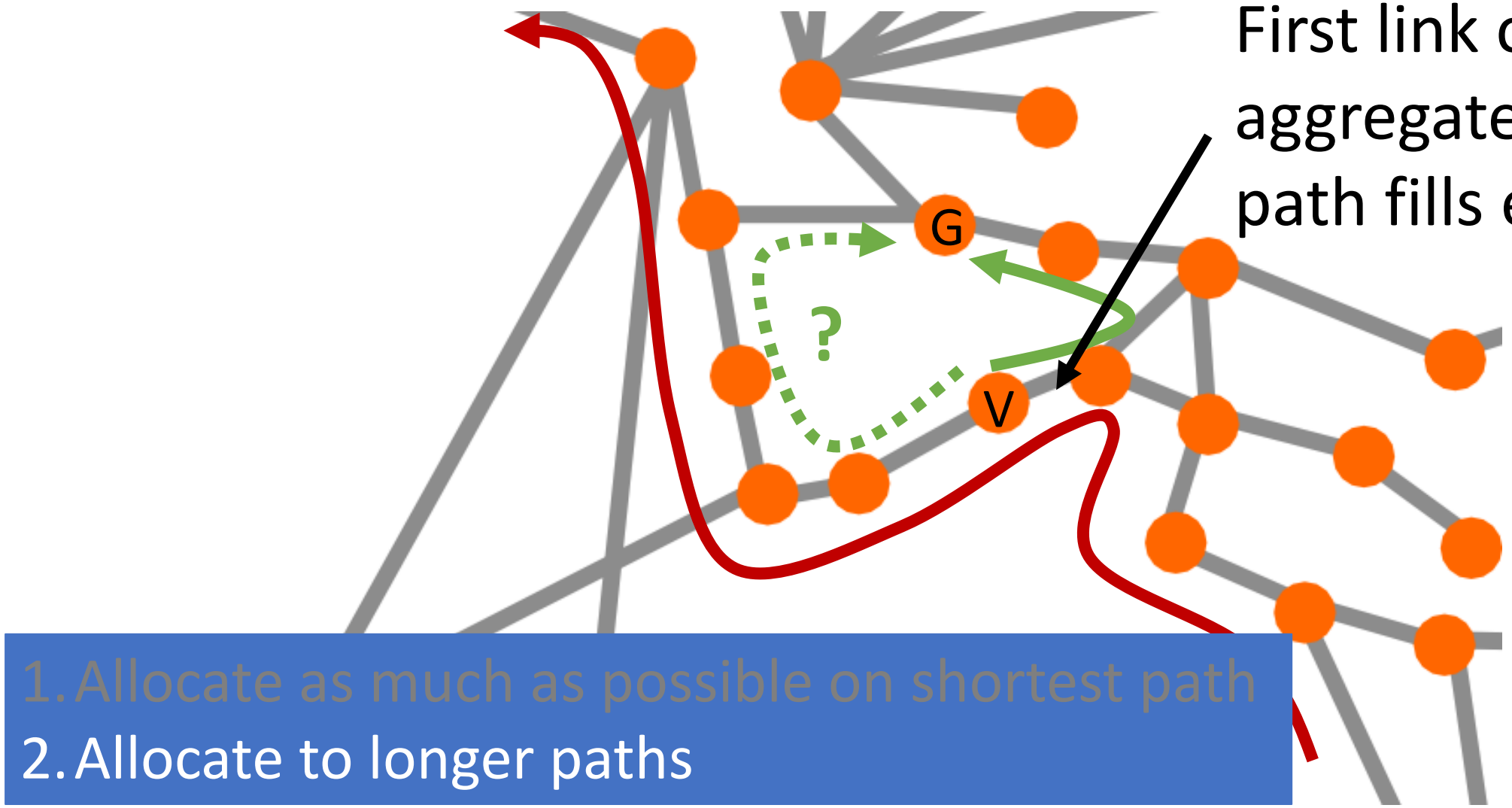
1. Allocate as much as possible on shortest path

Limitations of greedy routing



1. Allocate as much as possible on shortest path
2. Allocate to longer paths

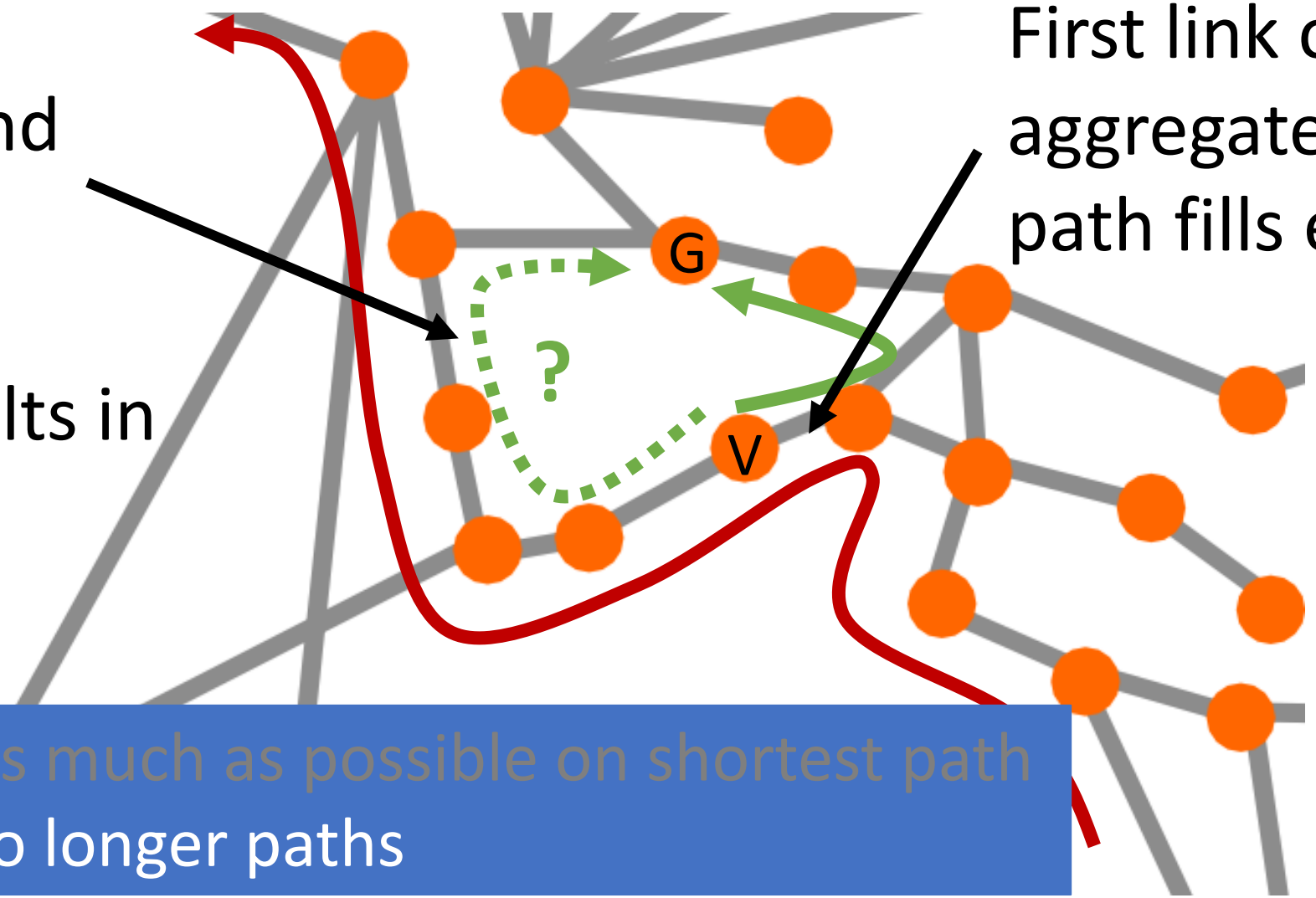
Limitations of greedy routing



1. Allocate as much as possible on shortest path
2. Allocate to longer paths

Limitations of greedy routing

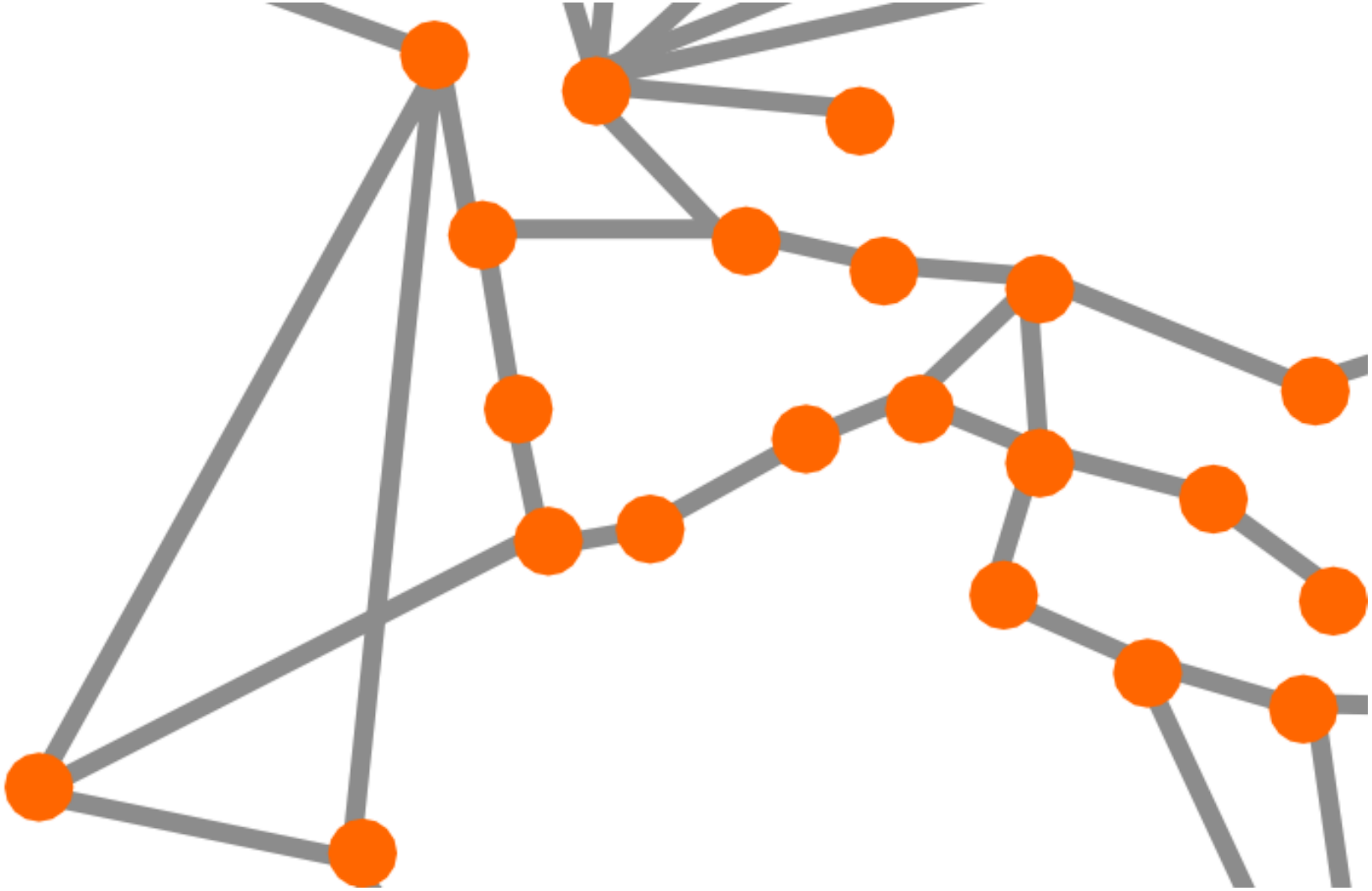
V->G's second best path is already full, using it results in congestion!



First link on V->G aggregate's shortest path fills eastbound

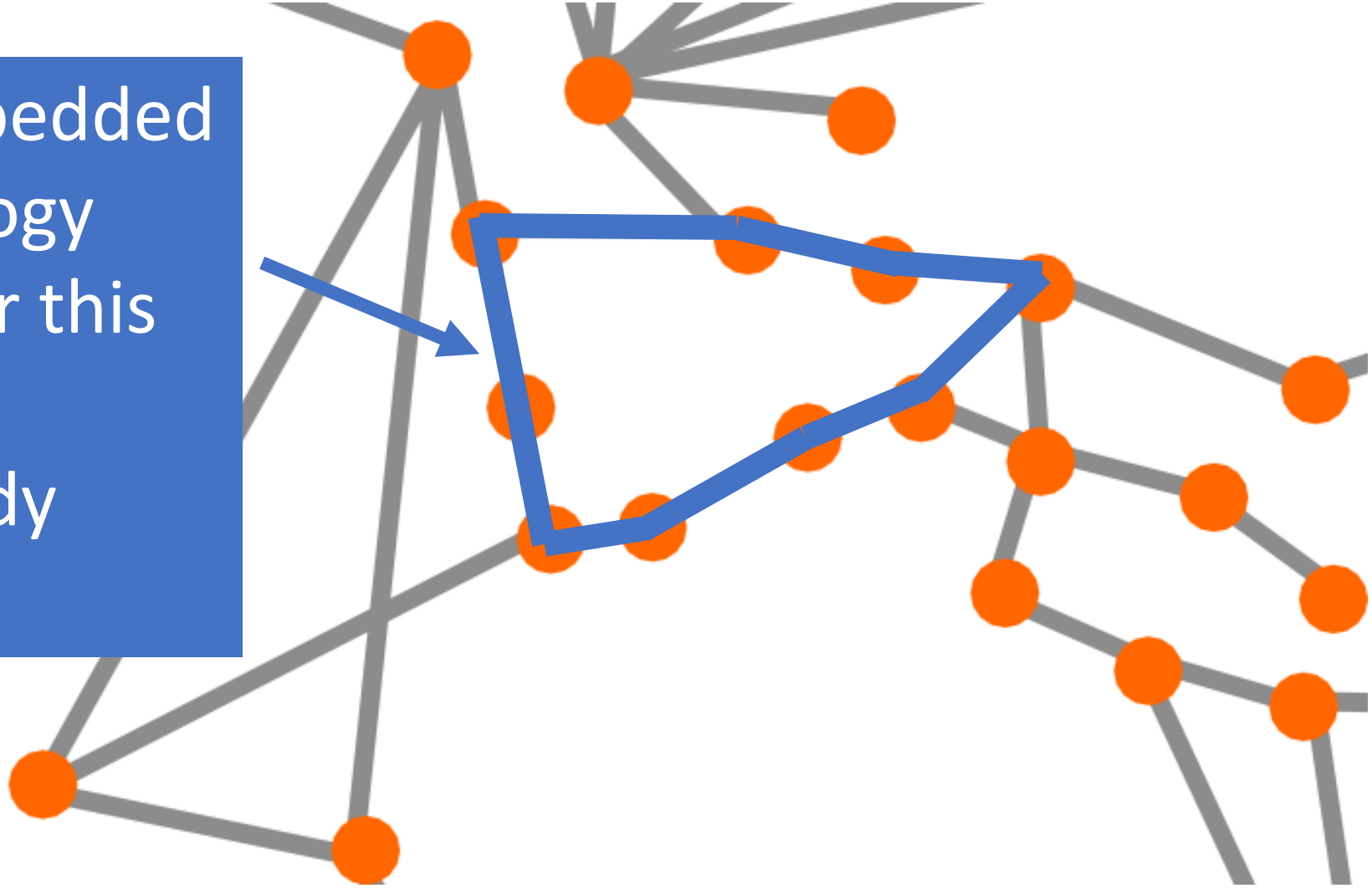
1. Allocate as much as possible on shortest path
2. Allocate to longer paths

Limitations of greedy routing

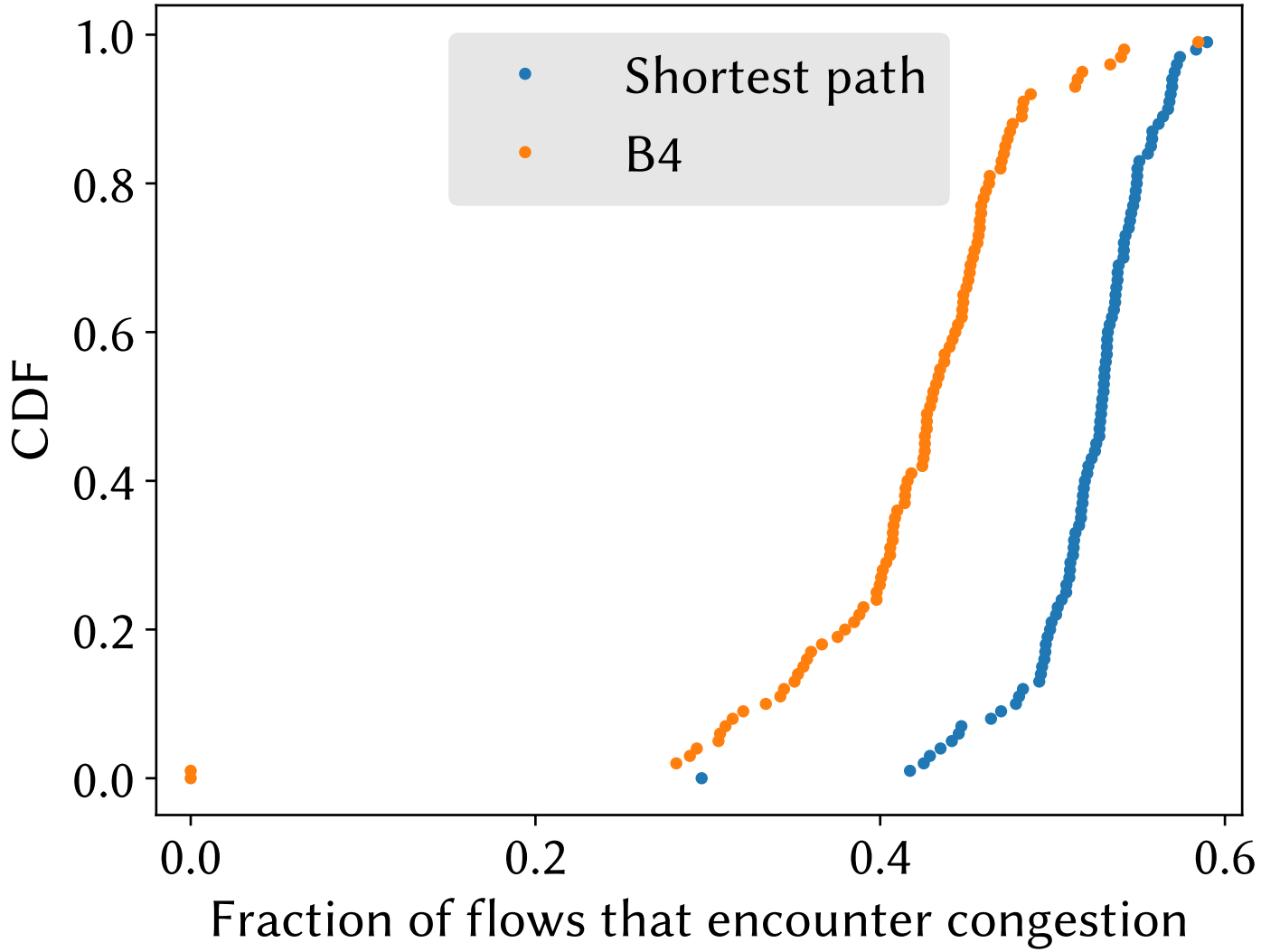


Limitations of greedy routing

Rings embedded in a topology can trigger this problem with greedy routing

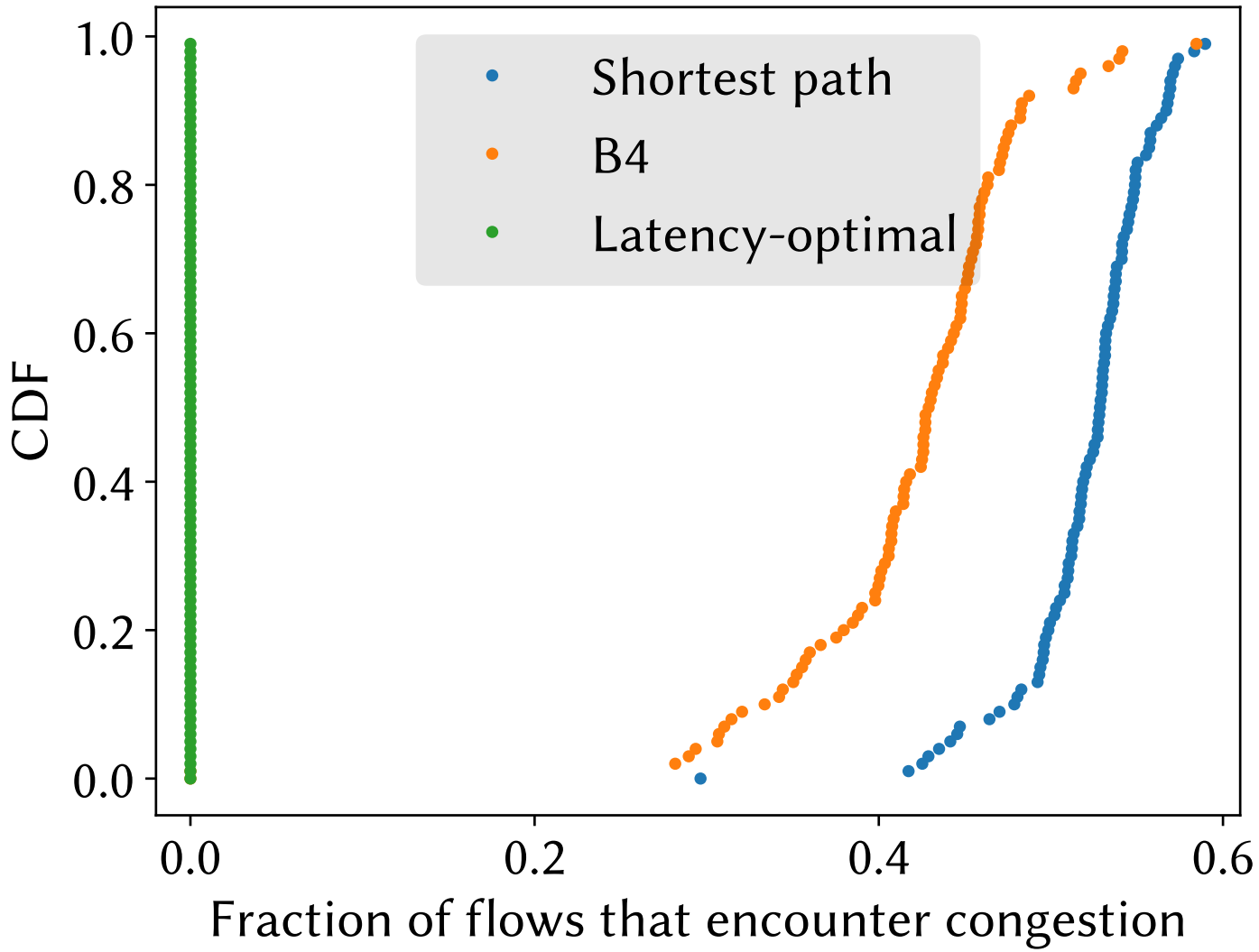


B4 for the win … sort of



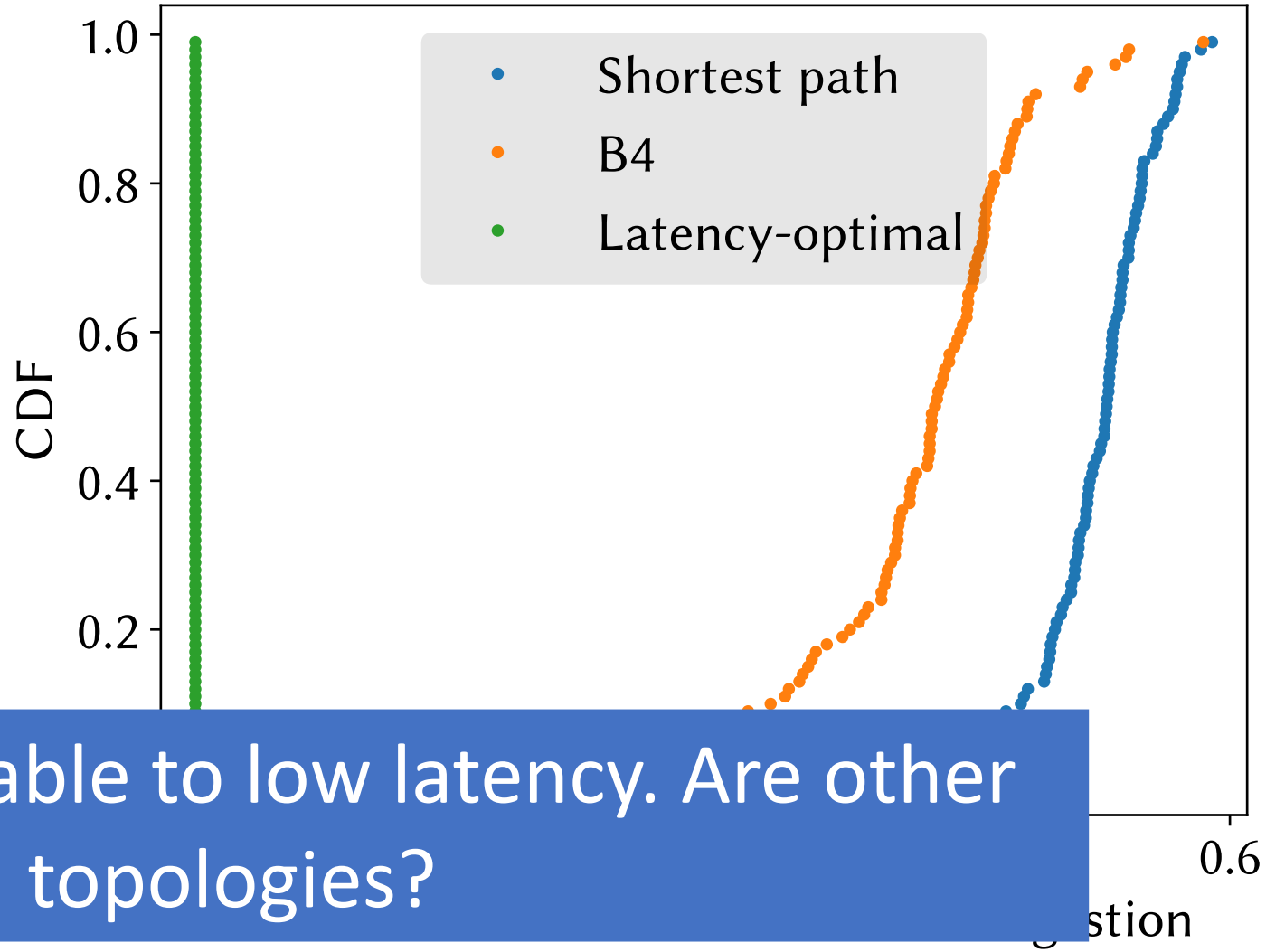
Can we do better?

Yes, a placement which *both* avoids congestion and minimizes propagation delay does exist!



Can we do better?

Yes, a placement which *both* avoids congestion and minimizes propagation delay does exist!



So GTS is amenable to low latency. Are other topologies?

How might we quantify a topology's potential for low latency under load?

- Want a metric to capture a topology's inherent potential for low latency
- Should be:
  - traffic matrix-agnostic
  - routing algorithm-agnostic

How might we quantify a topology's potential for low latency under load?

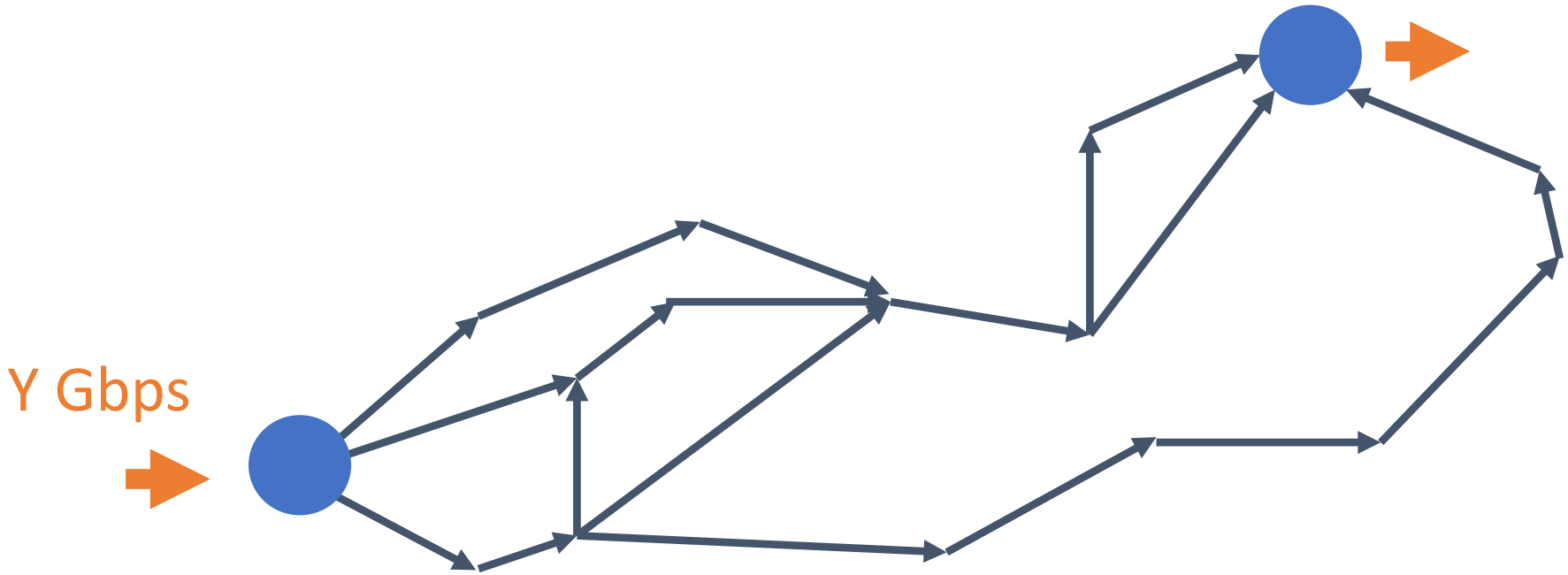
- Want a metric to capture a topology's inherent potential for low latency
- Should be:
  - traffic matrix-agnostic
  - routing algorithm-agnostic
- Want to capture two things:
  - topology's potential for routing around congestion hot spots
  - …without incurring long propagation delay

How might we quantify a topology's potential for low latency under load?

- Want a metric to capture a topology's inherent potential for low latency
- Should be:
  - traffic m
  - routing a
- Want to cap
  - topology's potential for routing around congestion hot spots
  - …without incurring long propagation delay

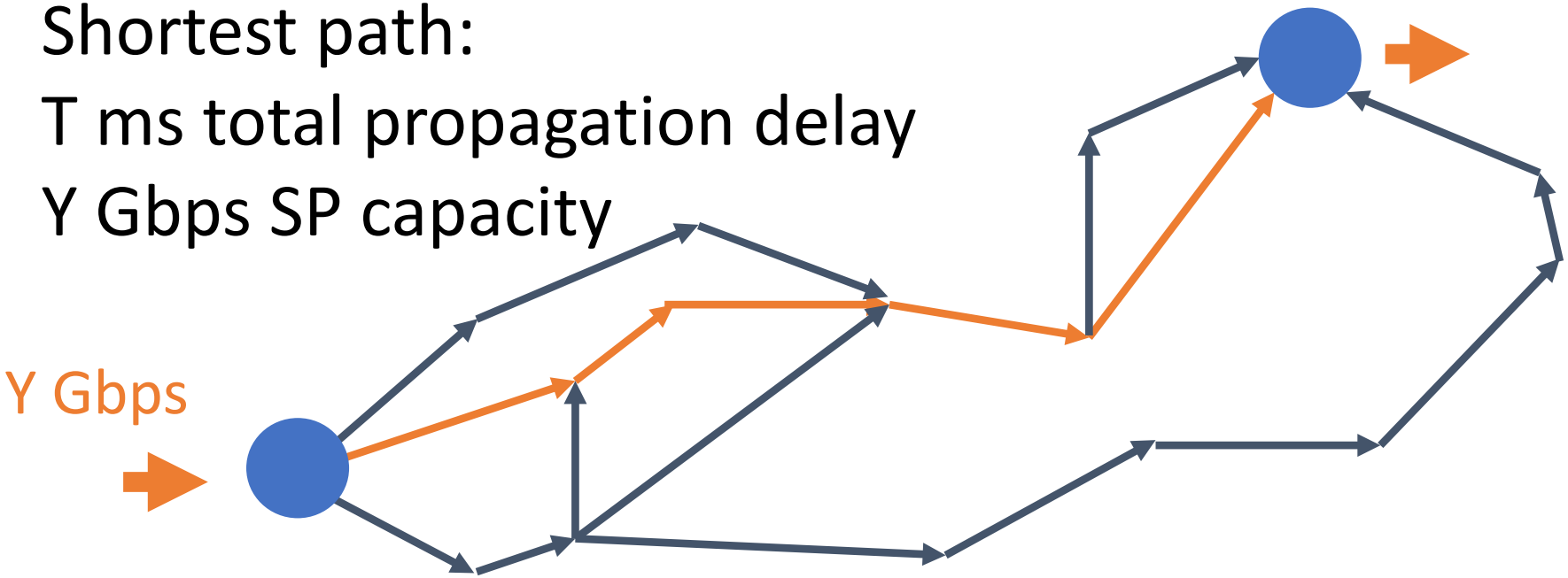
We want a metric that rewards alternate paths with short propagation delay

Alternate Path Availability (APA)



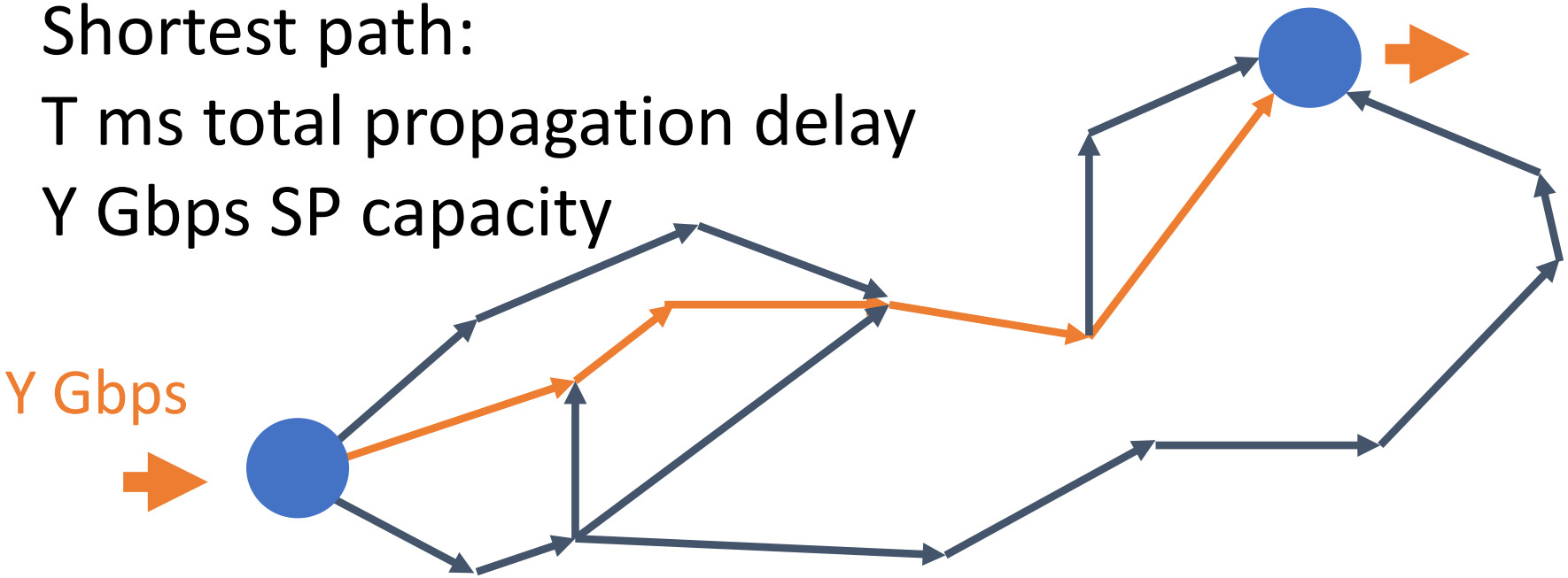
Alternate Path Availability (APA)

Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Alternate Path Availability (APA)

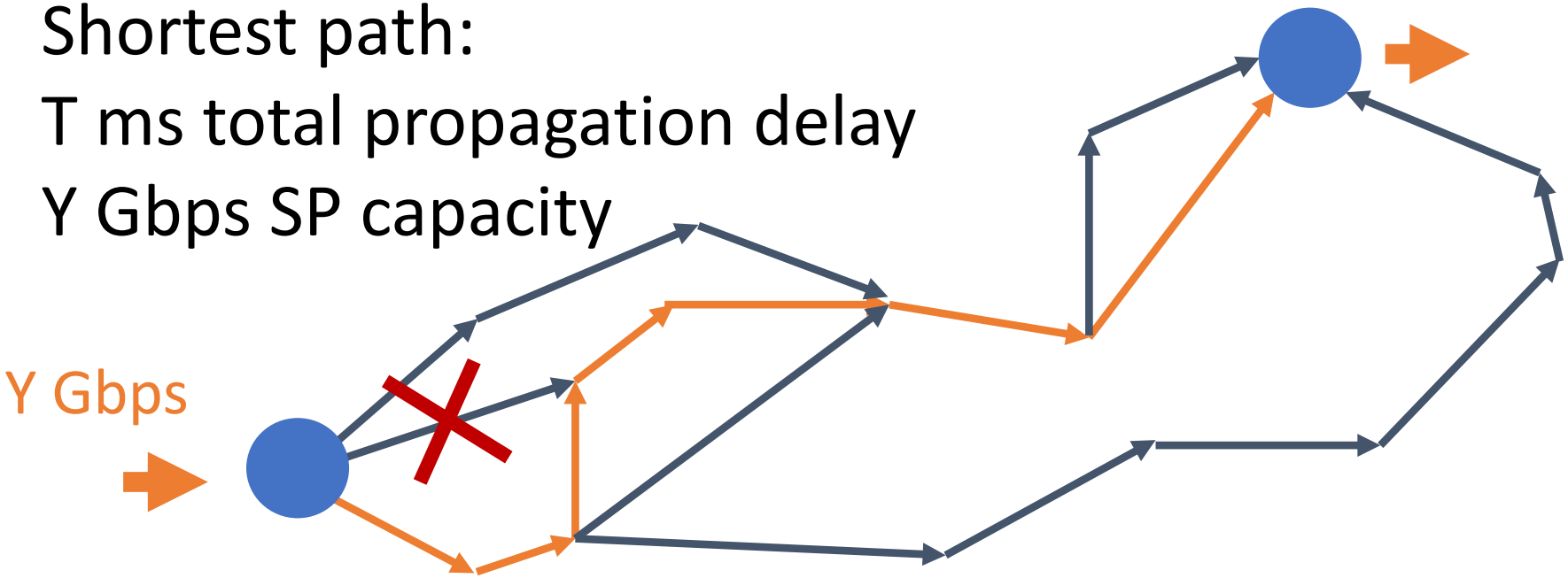
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

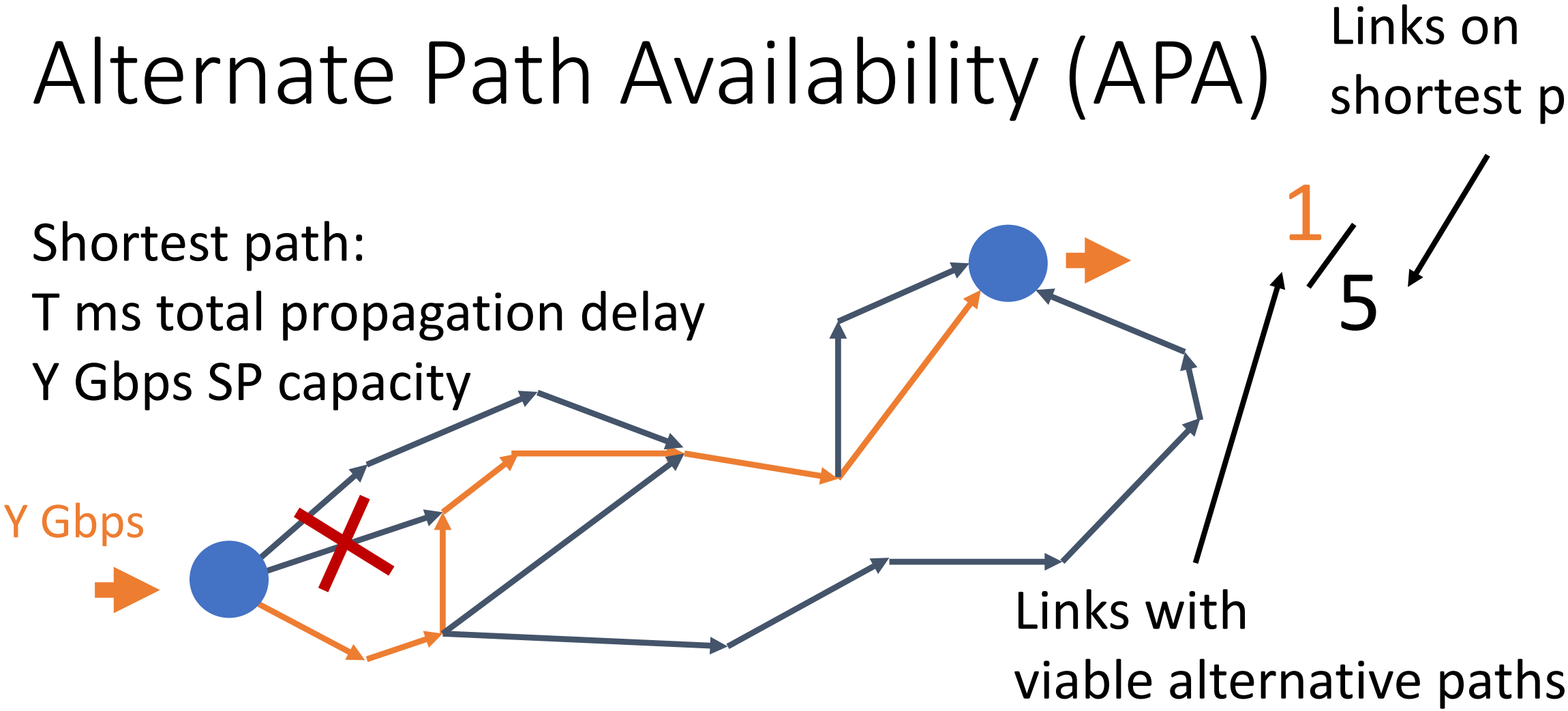
Alternate Path Availability (APA)

Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

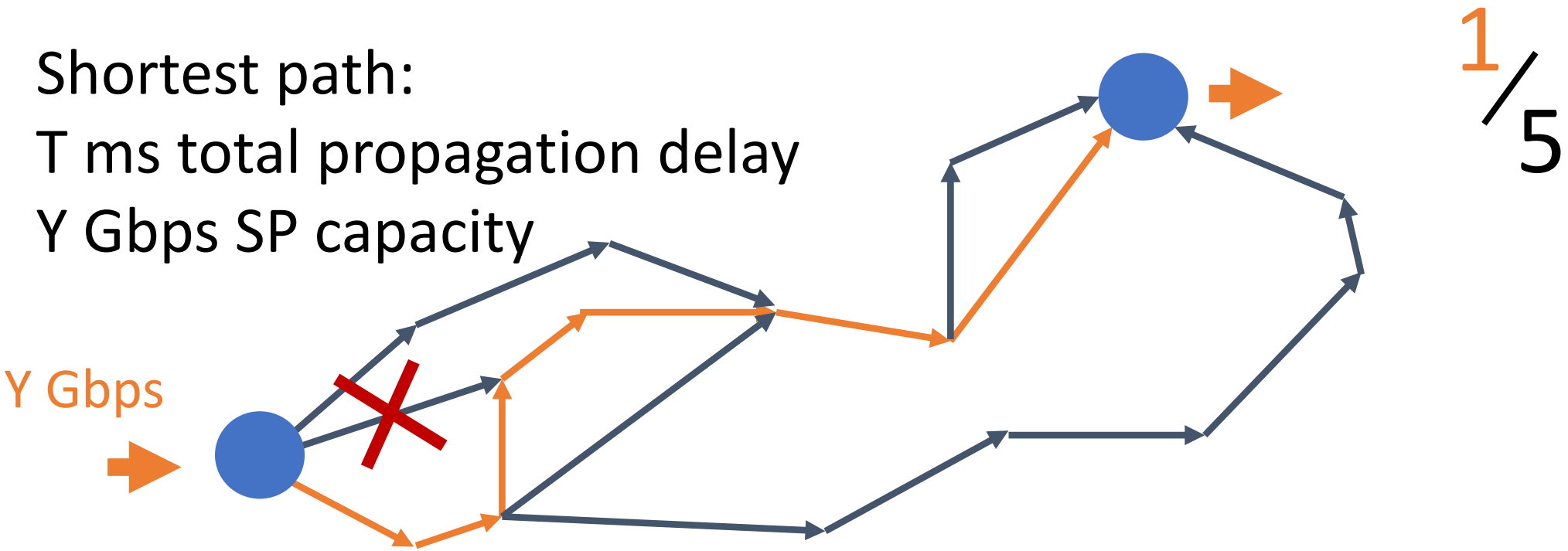
Alternate Path Availability (APA)



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

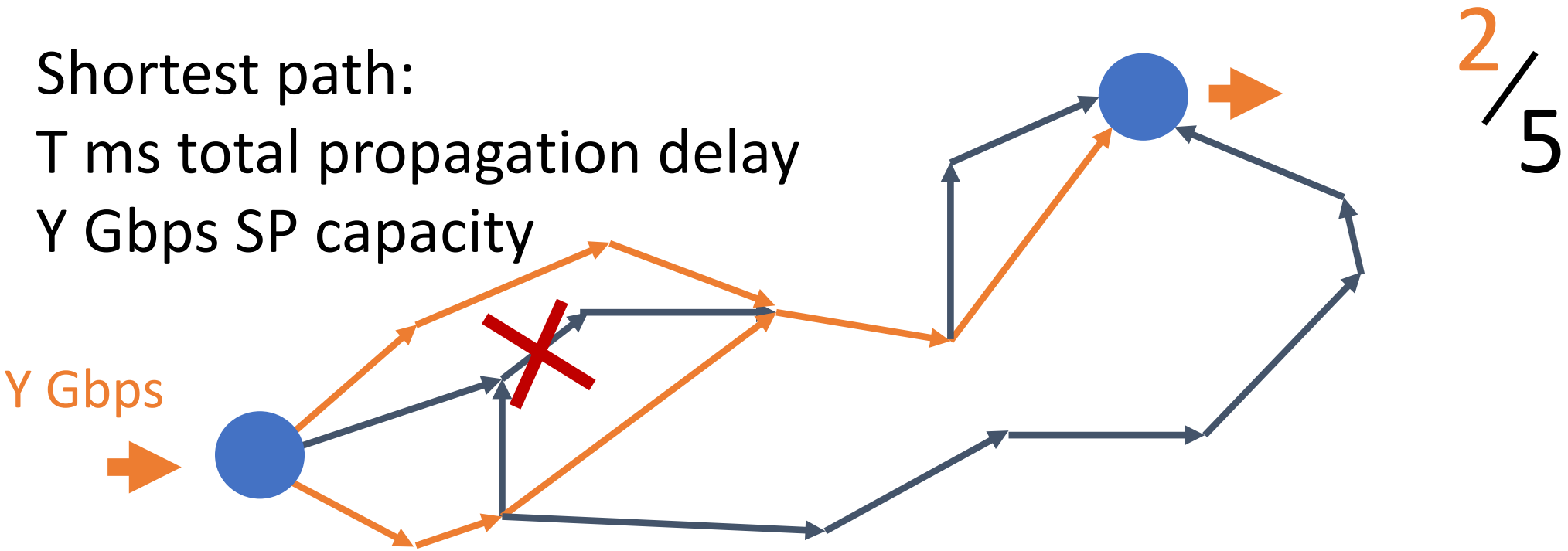
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

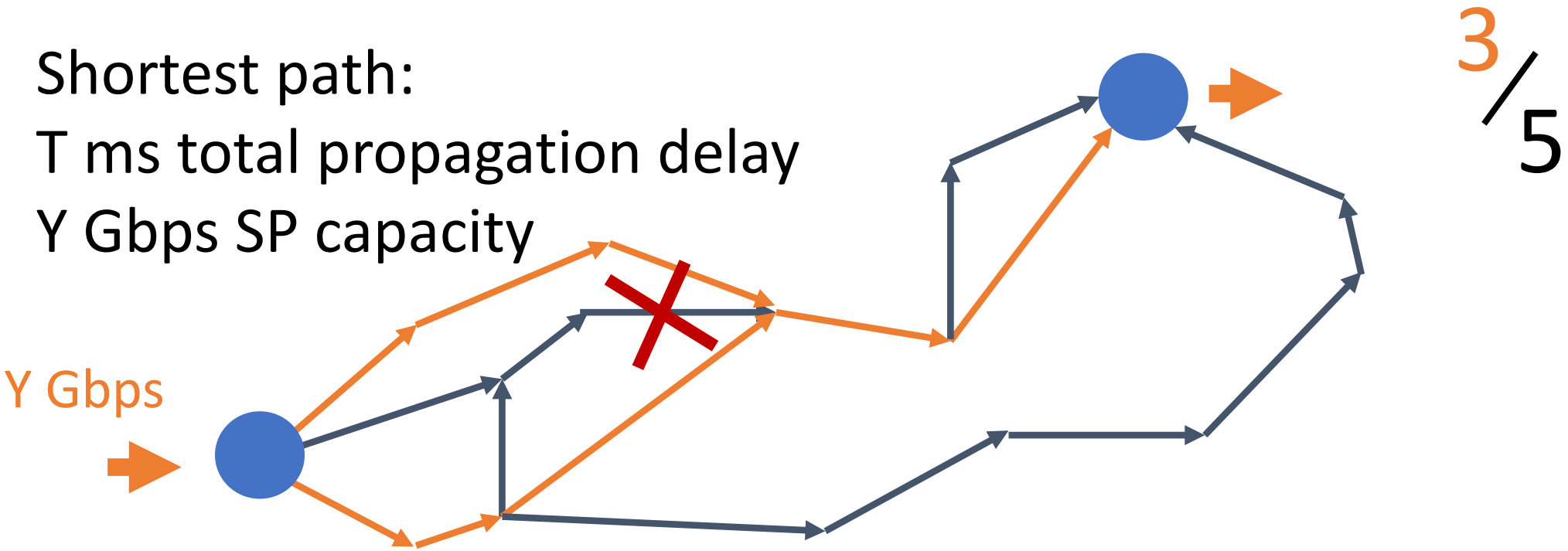
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

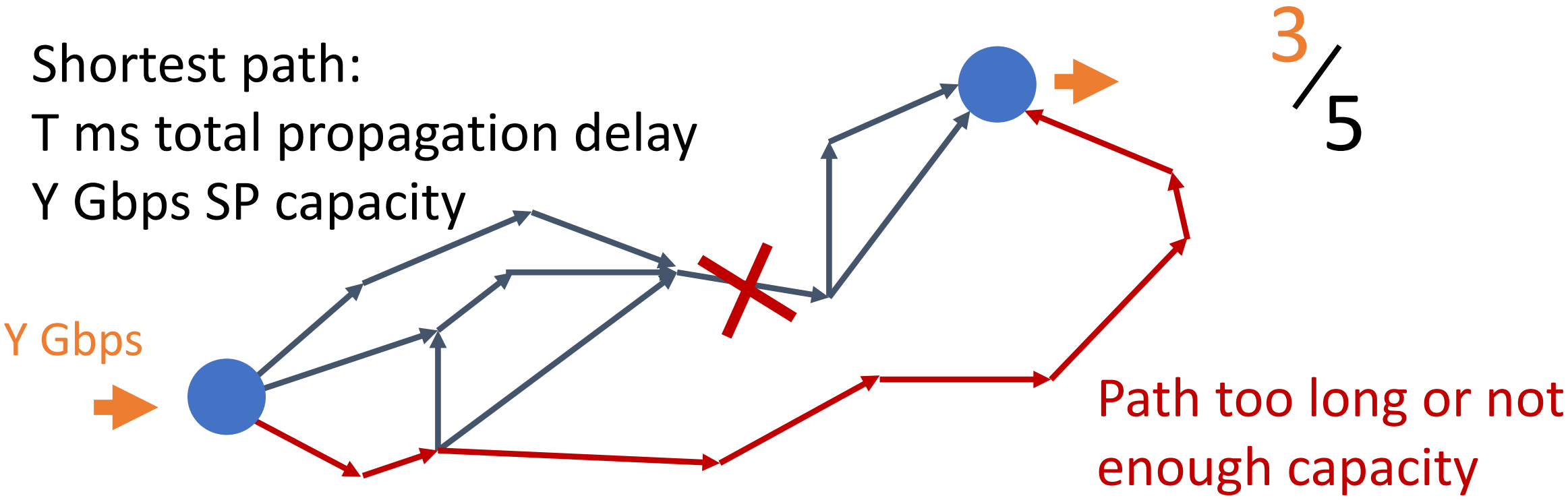
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

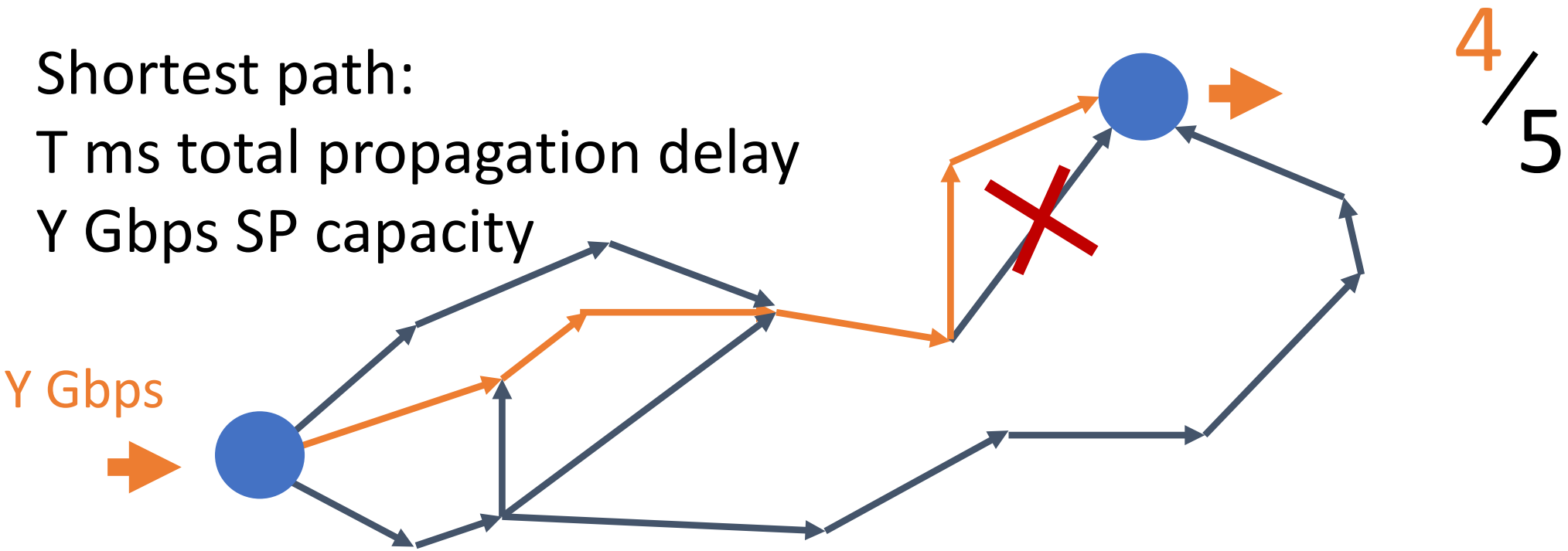
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

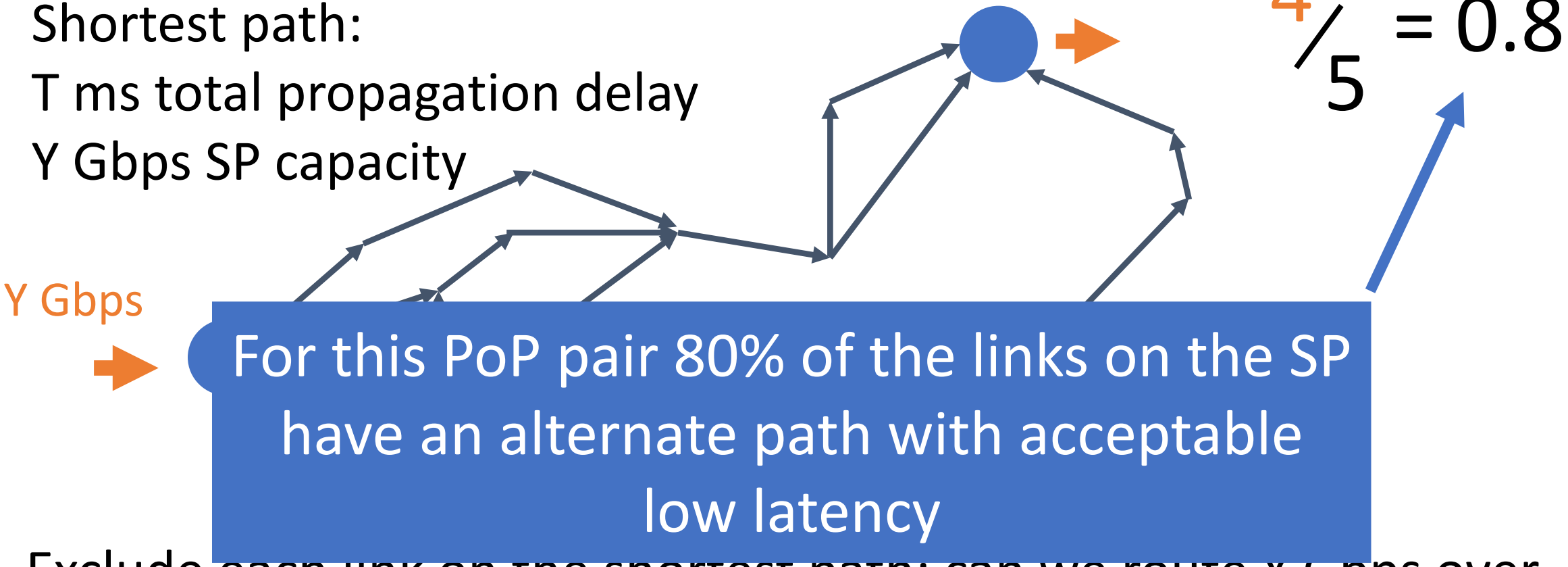
Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Alternate Path Availability (APA)

Shortest path:
T ms total propagation delay
Y Gbps SP capacity



Exclude each link on the shortest path; can we route Y Gbps over one or more alternative paths with delay < 1.4 T?

Low-latency path diversity (LLPD)

1. Compute APA for all PoP pairs

Low-latency path diversity (LLPD)

1. Compute APA for all PoP pairs

2. Compute LLPD = Fraction of PoP pairs with “good” path availability

Low-latency path diversity (LLPD)

1. Compute APA for all PoP pairs

2. Compute LLPD = Fraction of PoP pairs with "good" path availability

$$= \frac{\text{number of PoP pairs with APA} \geq 0.7}{\text{total number of PoP pairs}}$$

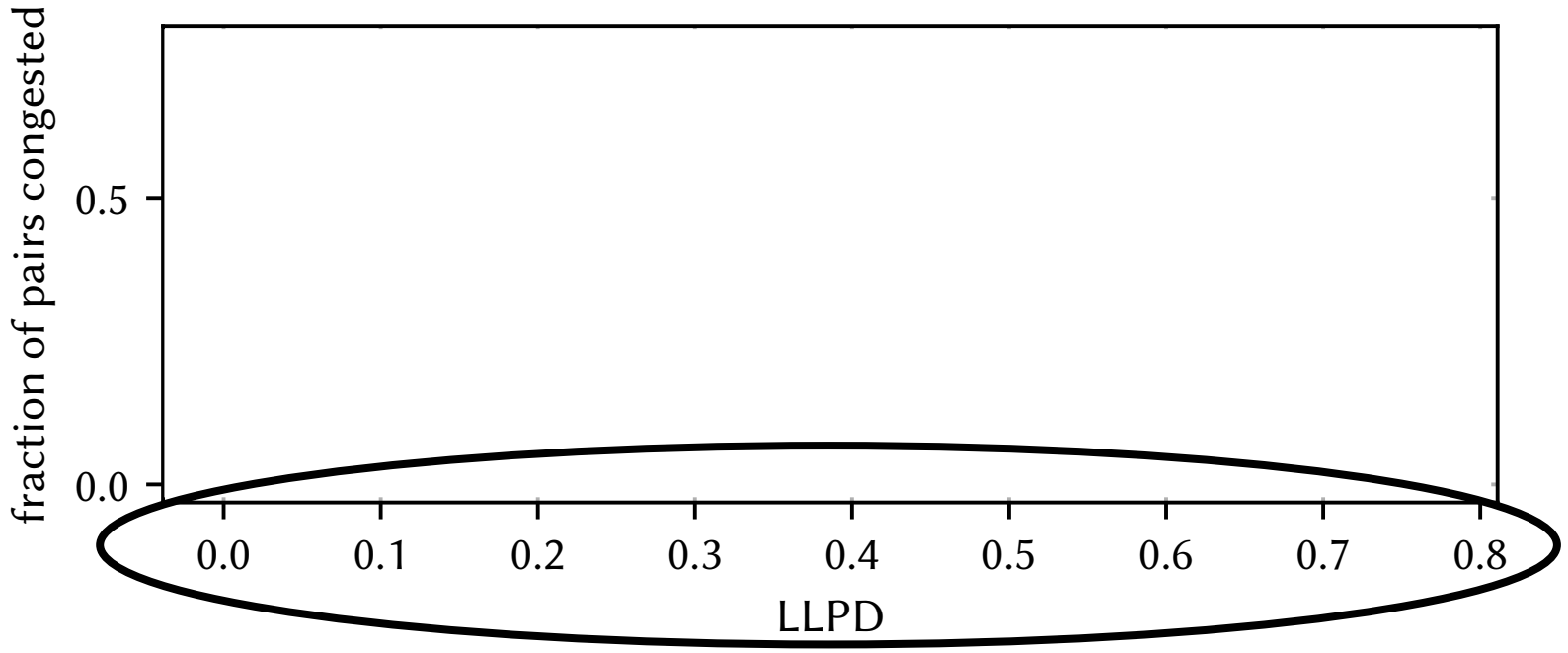
Low-latency path diversity (LLPD)

1. Compute APA for all PoP pairs

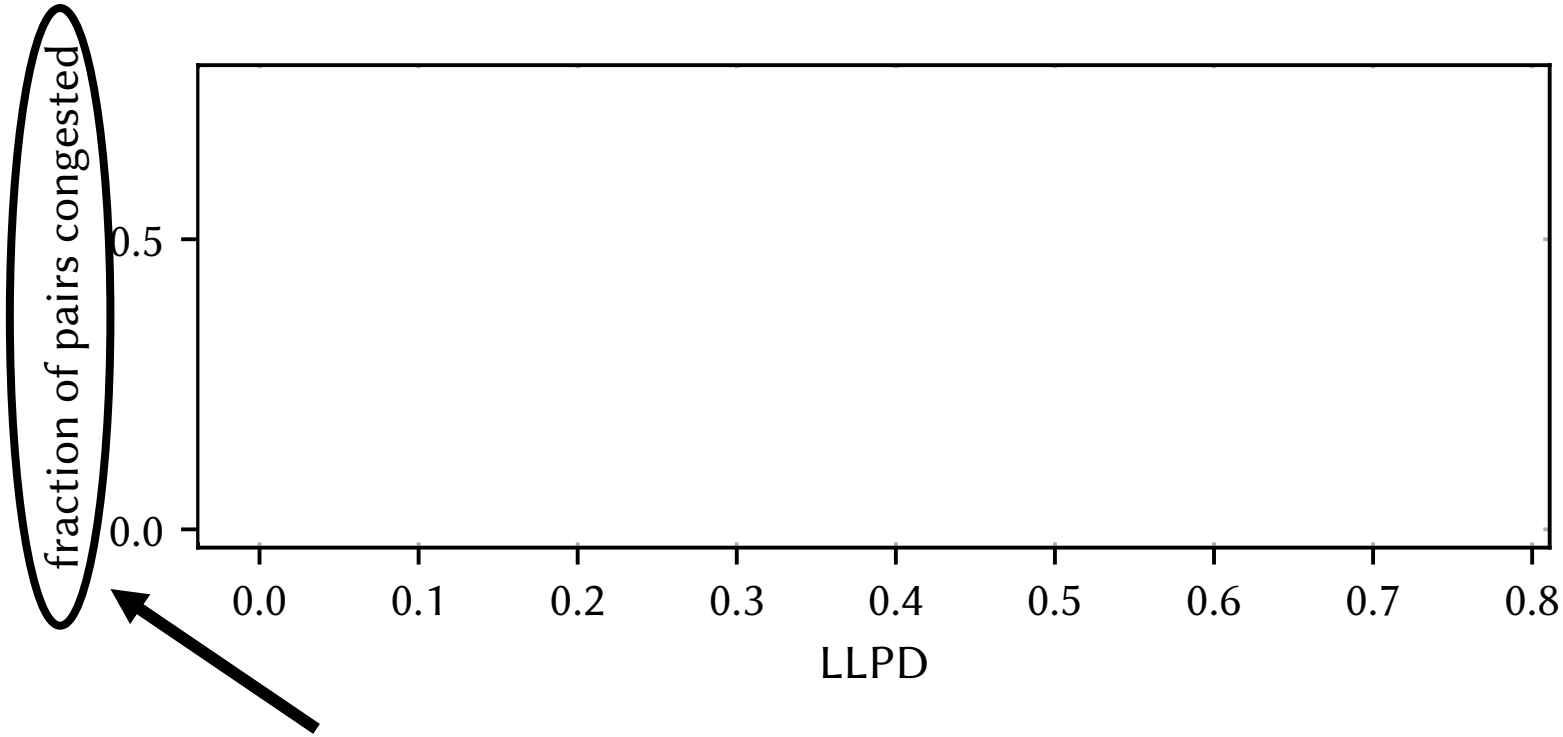
Empirically derived; metric not sensitive to picking different values

2. Compute LLPD = Fraction of PoP pairs with "good" path availability

$$= \frac{\text{number of PoP pairs with APA} \geq 0.7}{\text{total number of PoP pairs}}$$

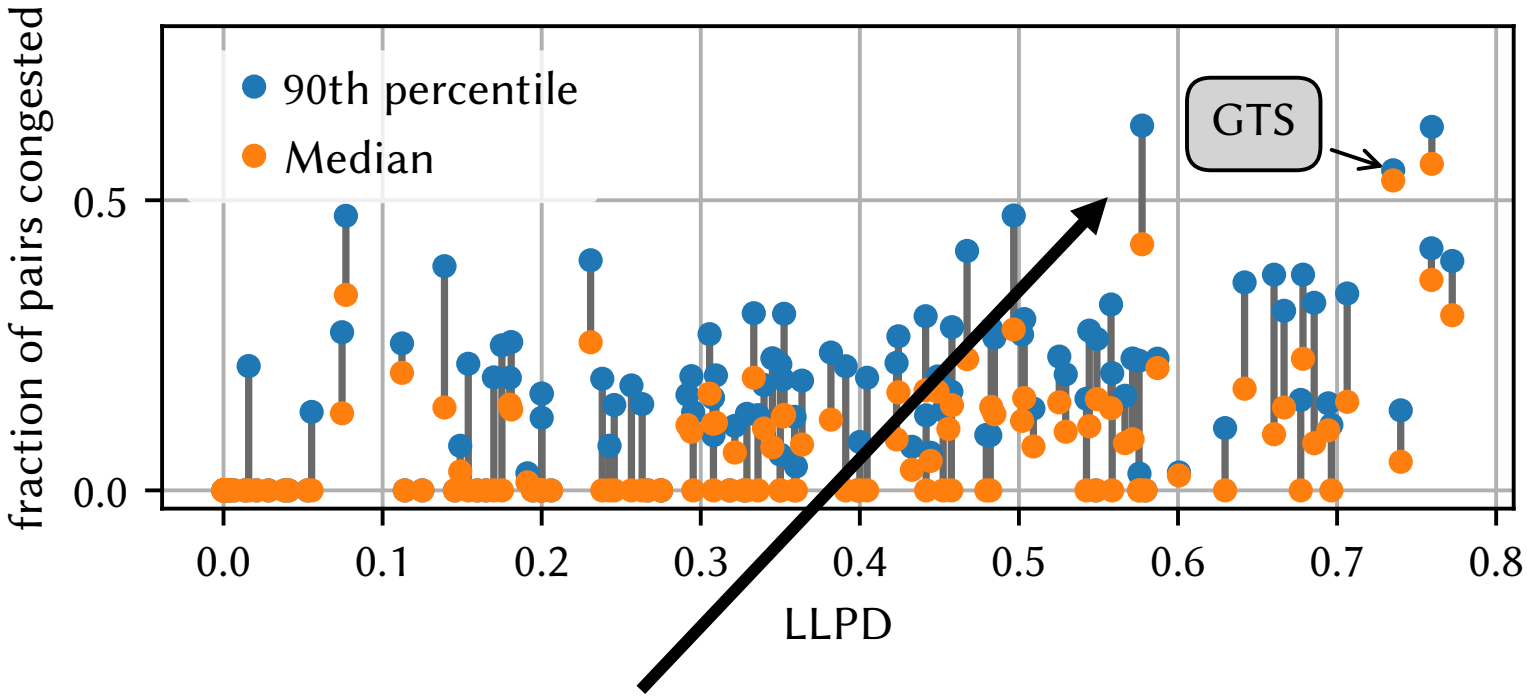


100+ real-world ISP topologies,
ranked by low-latency path diversity (LLPD)



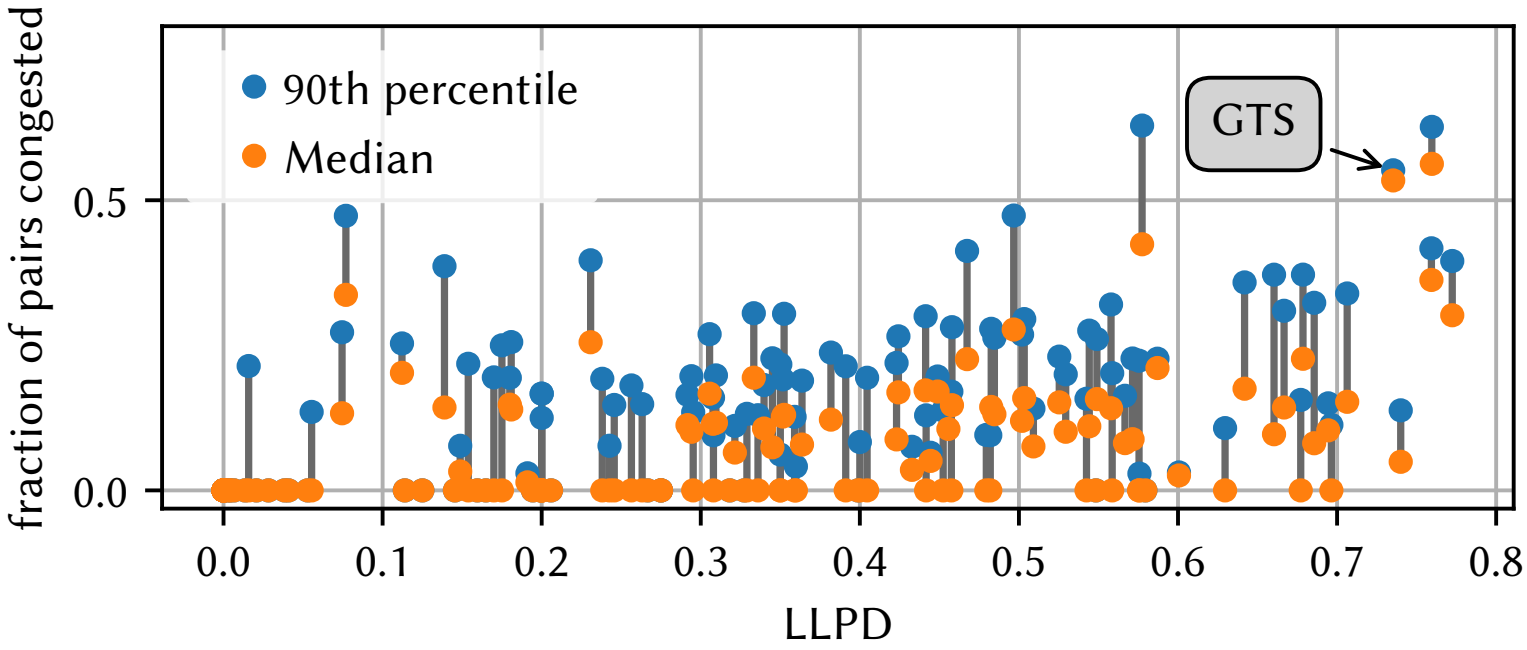
Generate TMs for each topology; plot fraction of (Src,Dst) PoP pairs in each TM that crosses at least one congested link

Shortest path routing congests links



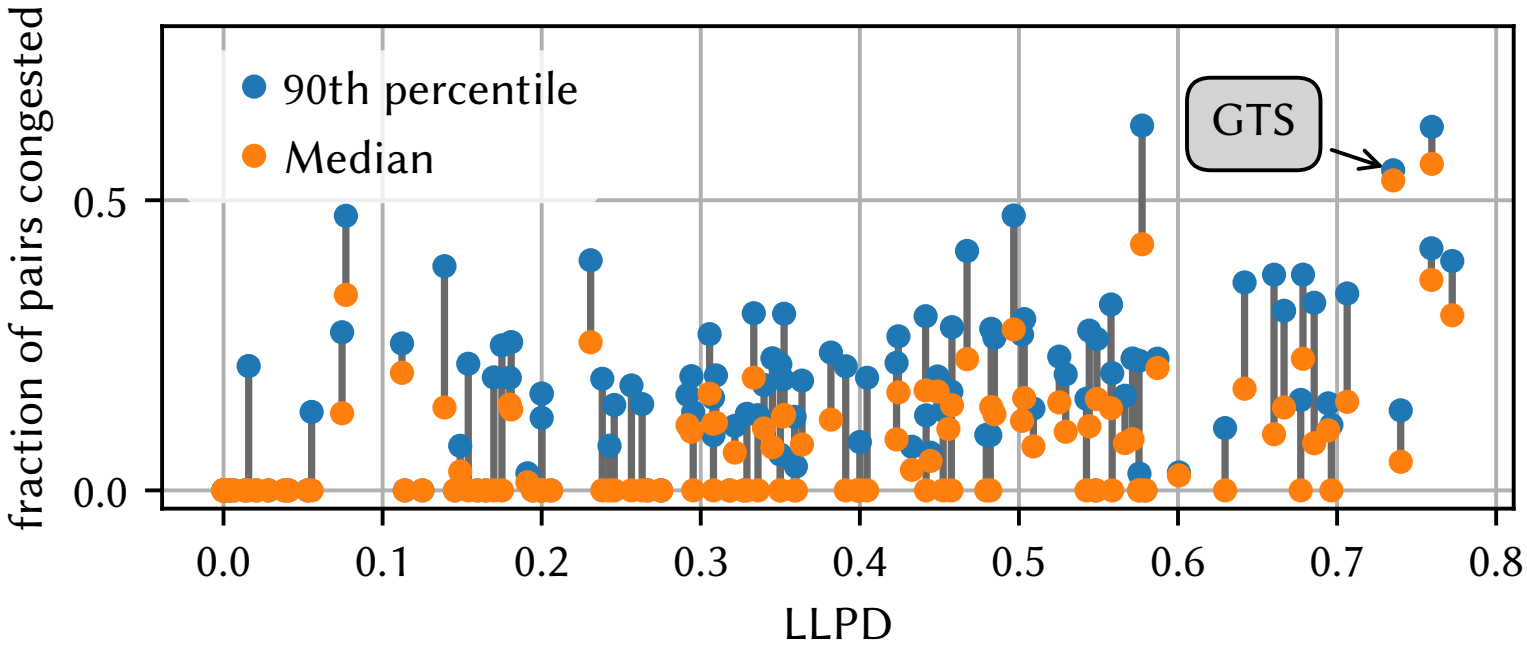
Two points per topology: median TM and 90th percentile TM; line shows spread of distribution

Shortest path routing congests links



Networks with high LLPD offer lots of alternative paths → shortest path routing experiences congestion

Shortest path routing congests links

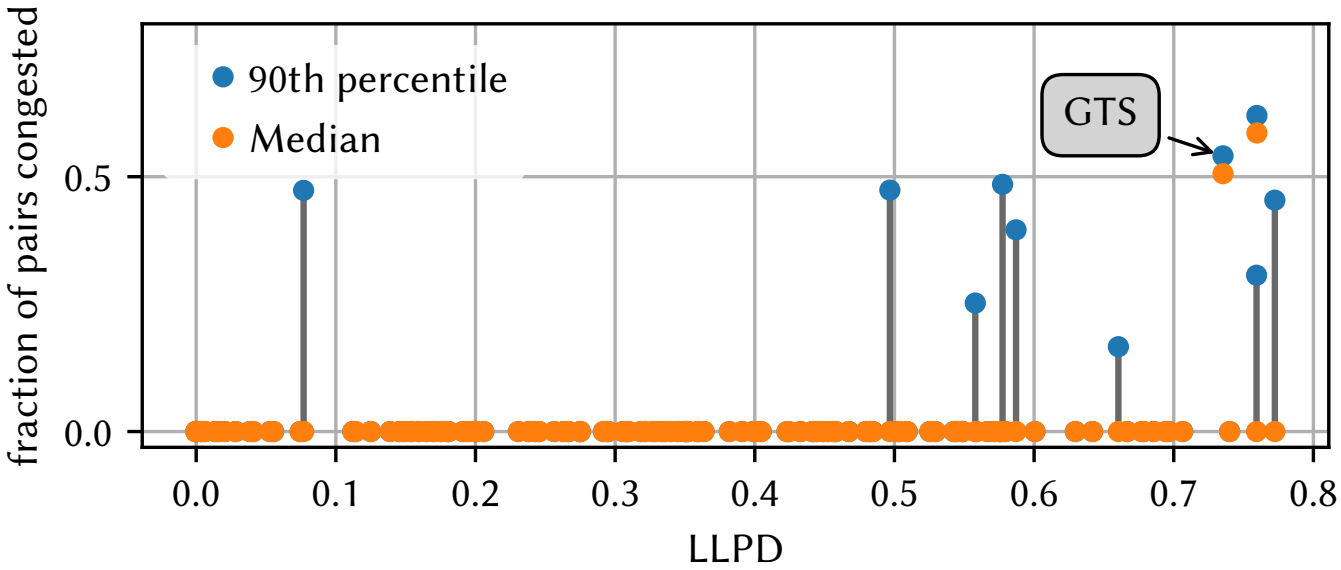


Networks w alternative paths → short ces congestion

No surprises here. What about B4?

B4 congests networks with high potential for low latency

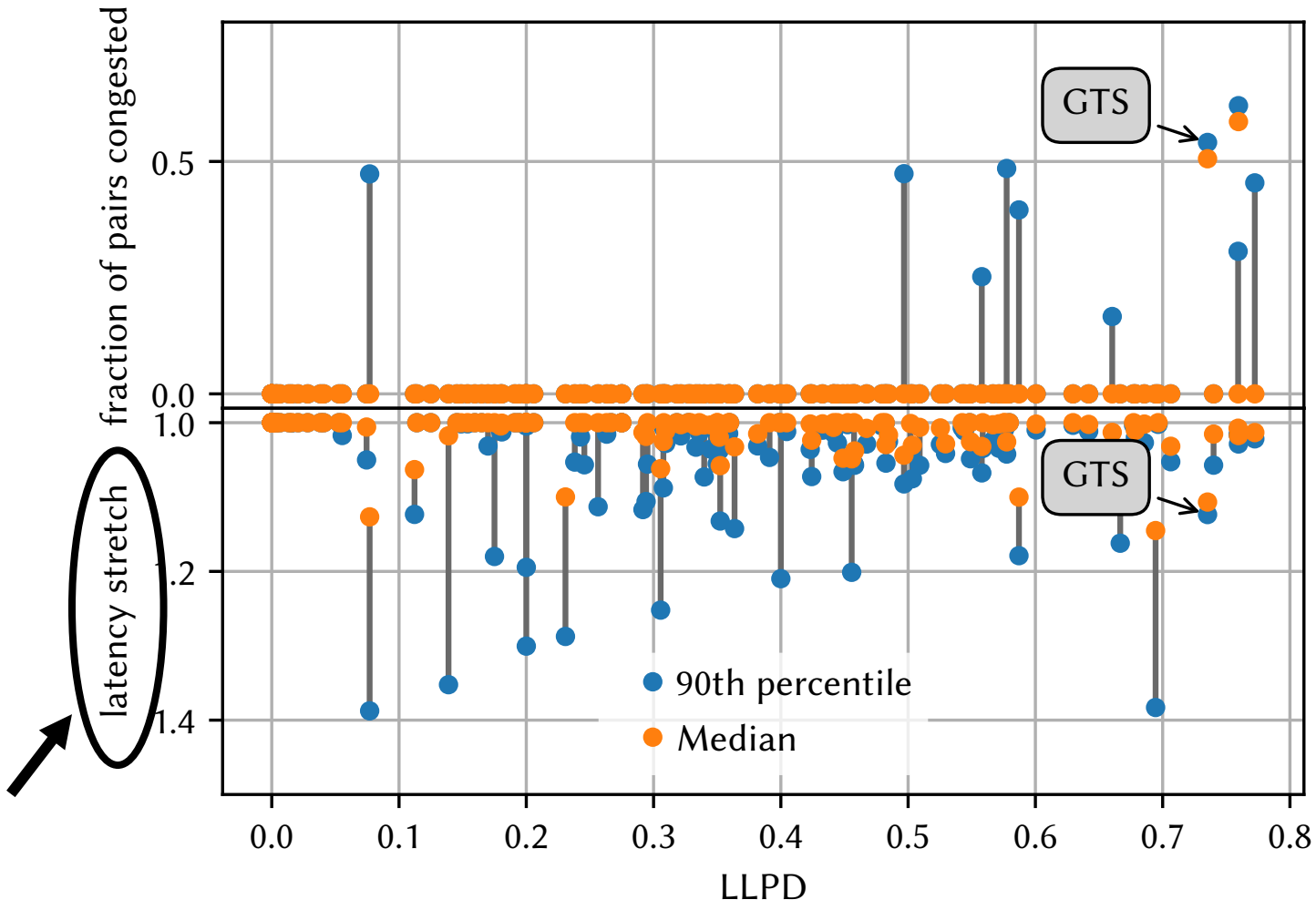
- Better at using alternative paths



B4 congests networks with high potential for low latency

- Better at using alternative paths

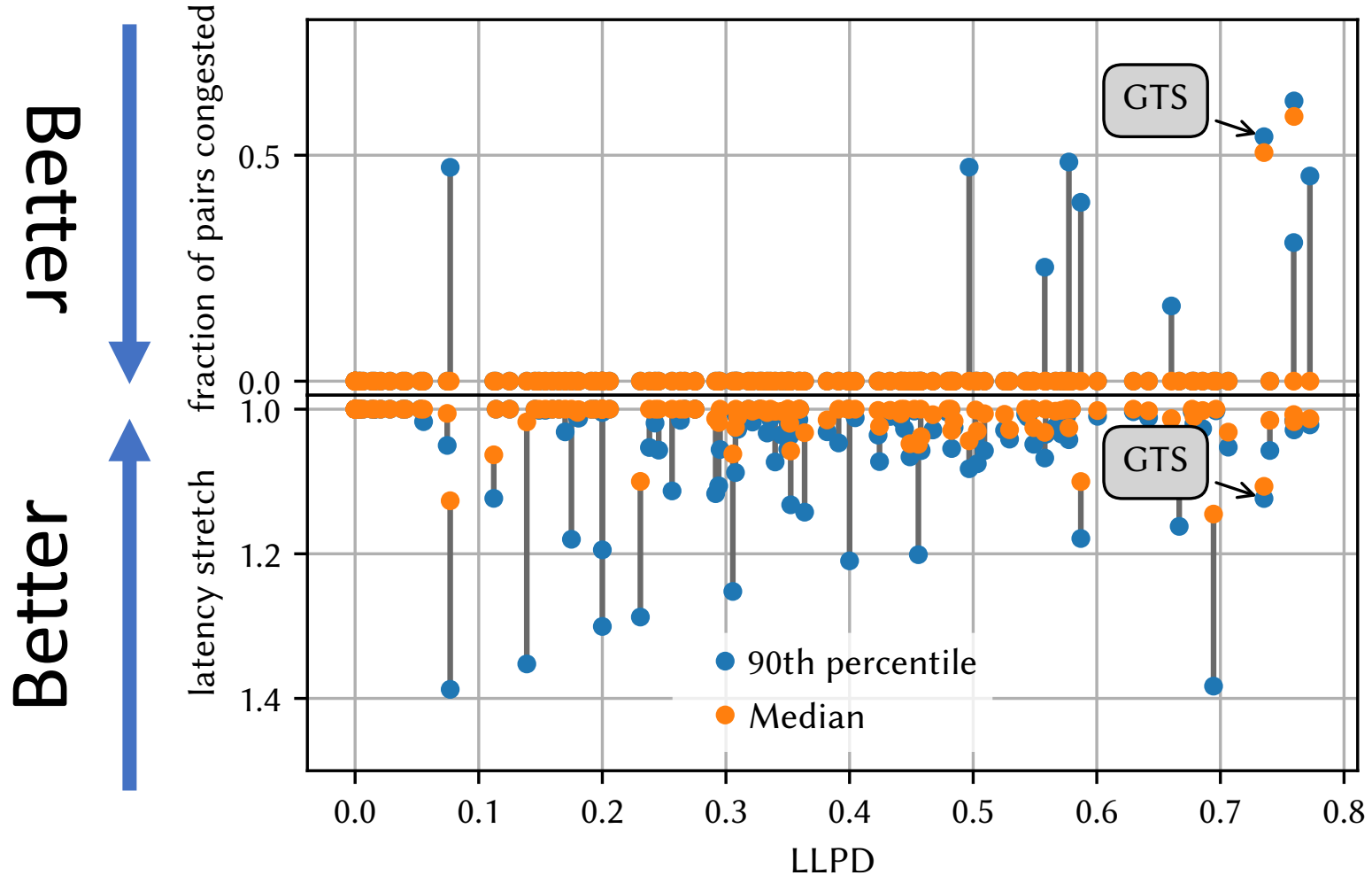
$$\frac{\text{total prop delay of all flows}}{\text{total prop delay if all flows routed on SP}}$$



B4 congests networks with high potential for low latency

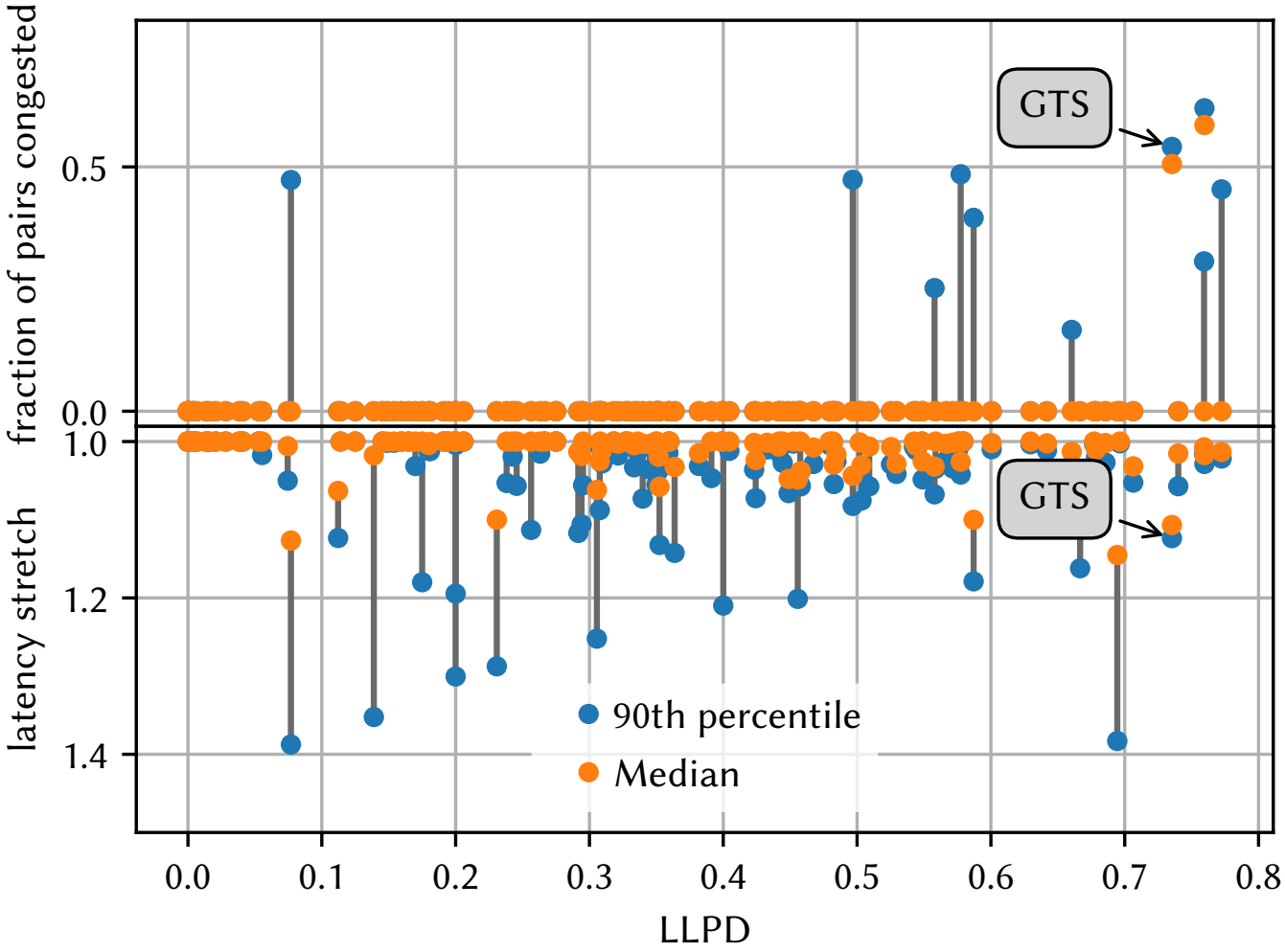
Congestion

Propagation delay



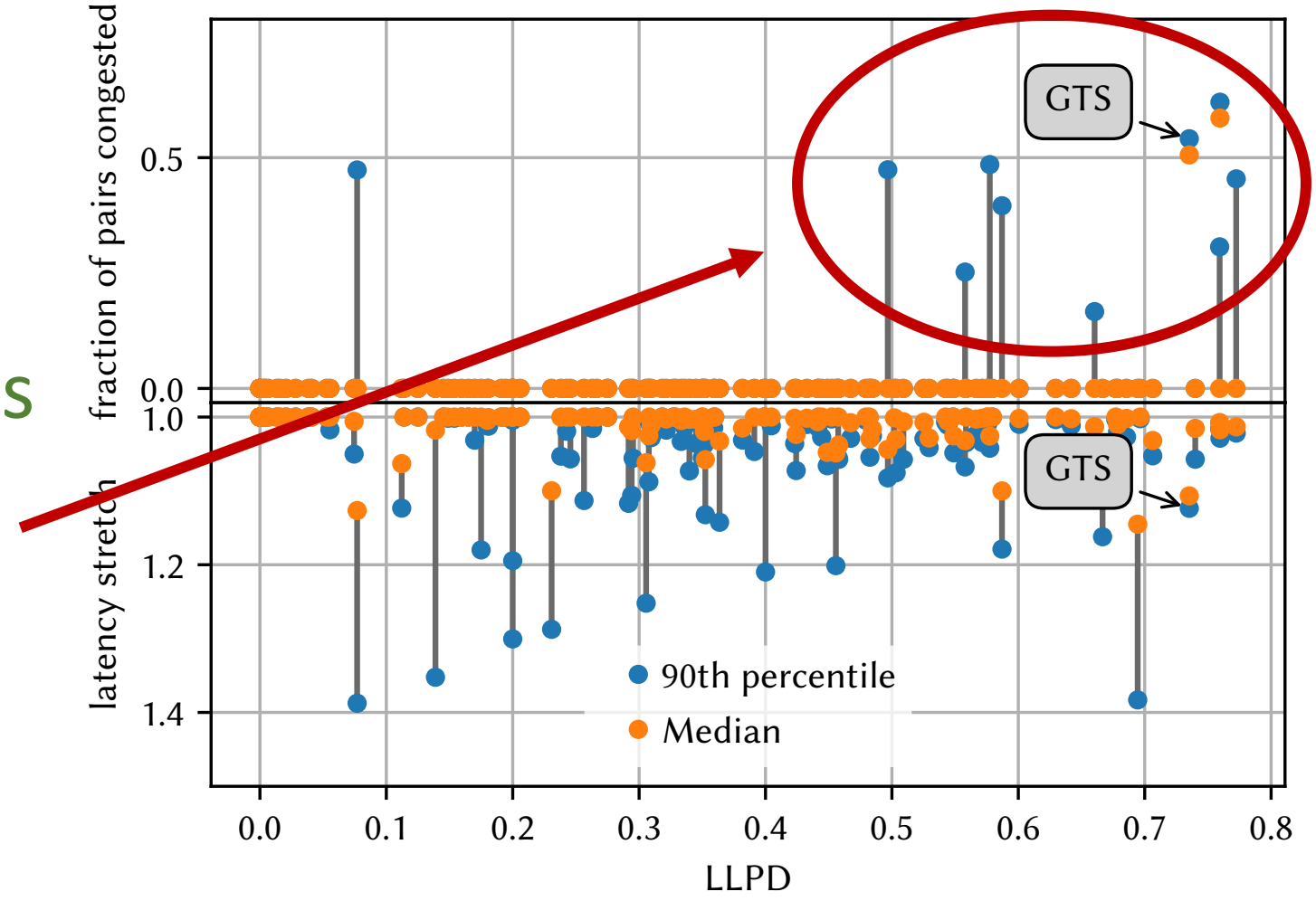
B4 congests networks with high potential for low latency

- Better at using alternative paths
- Some flows routed on non-shortest paths



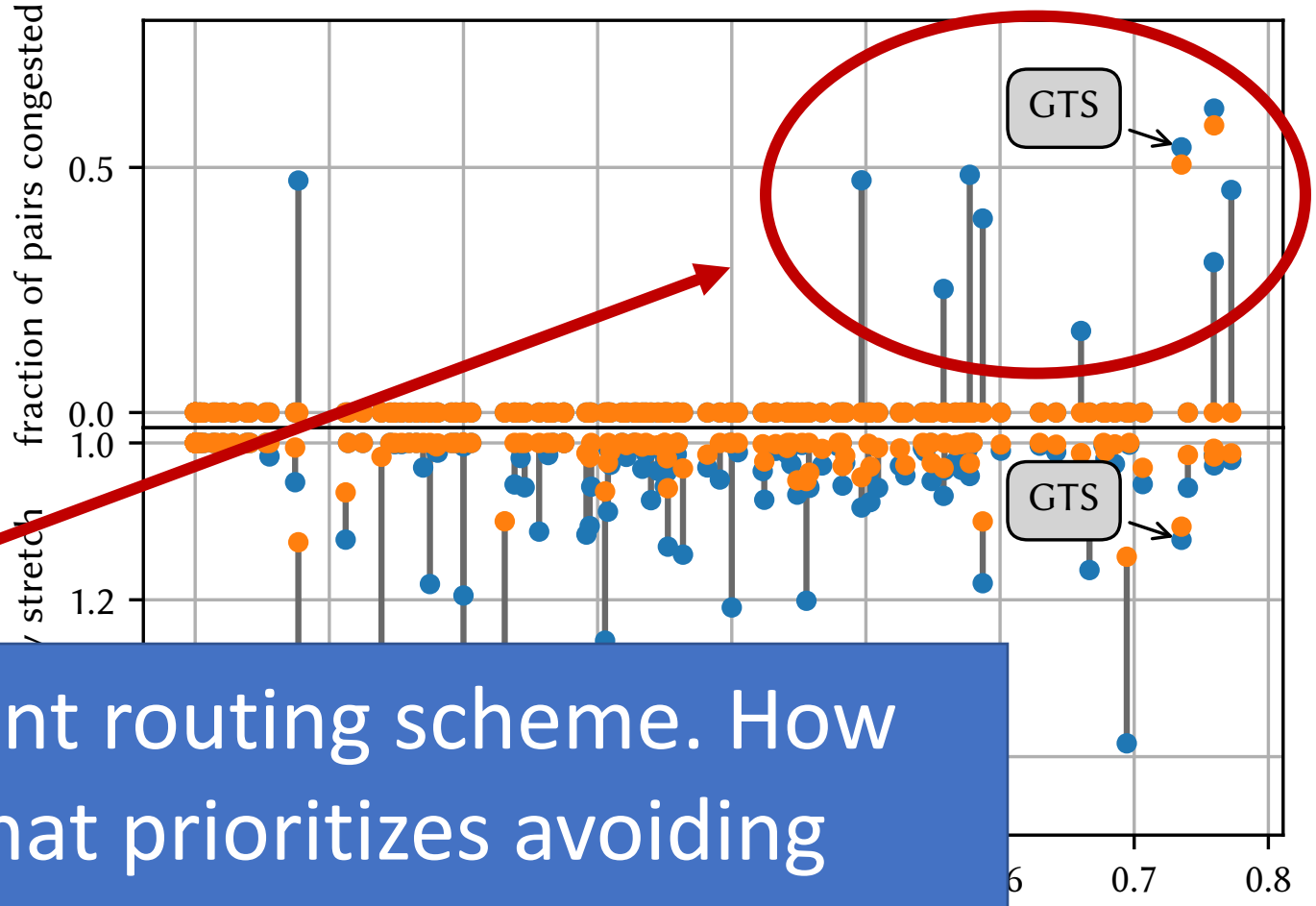
B4 congests networks with high potential for low latency

- Better at using alternative paths
- Some flows routed on non-shortest paths
- Still incurs congestion, and precisely on high-LLPD networks!



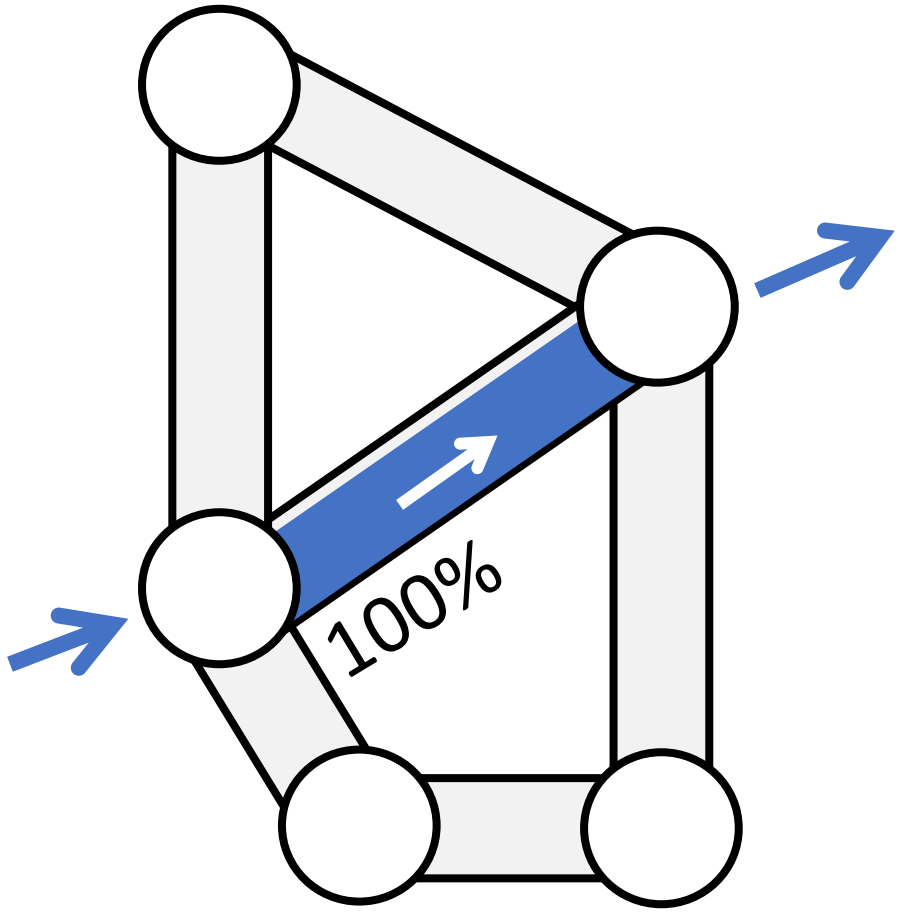
B4 congests networks with high potential for low latency

- Better at using alternative paths
- Some flows routed on non-shortest paths
- Still incurs congestion, and precis
high-L



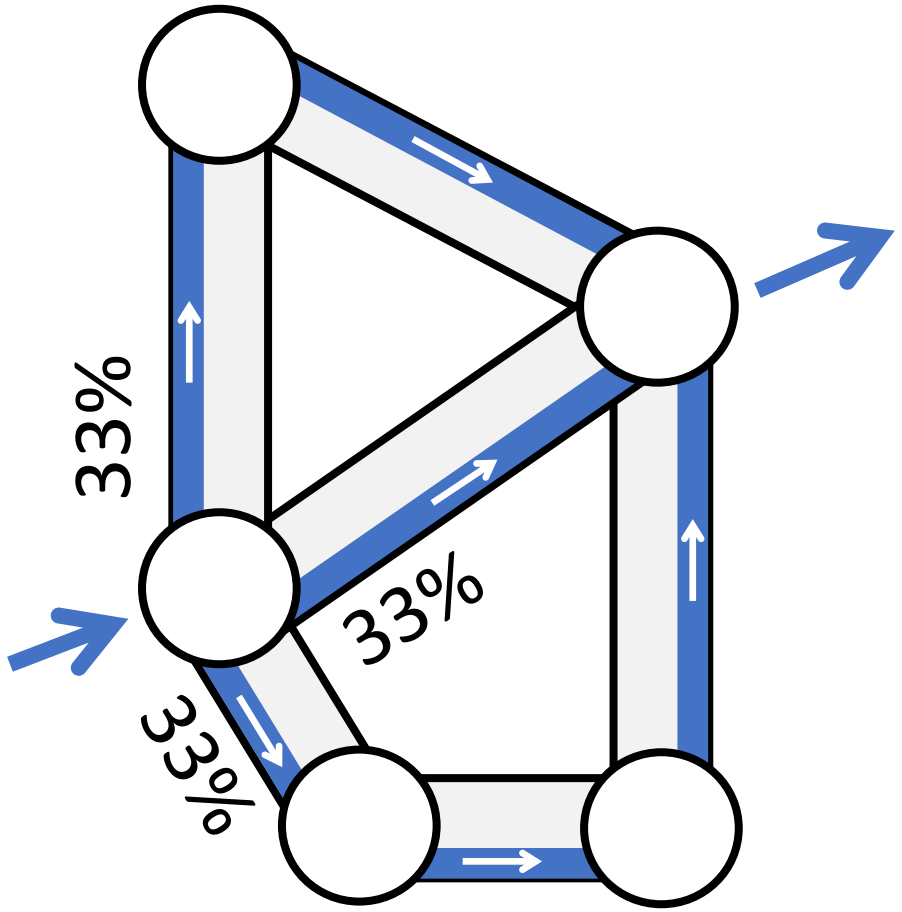
Need a different routing scheme. How about one that prioritizes avoiding congestion above all else?

Minimizing utilization avoids congestion



Minimizing utilization avoids congestion

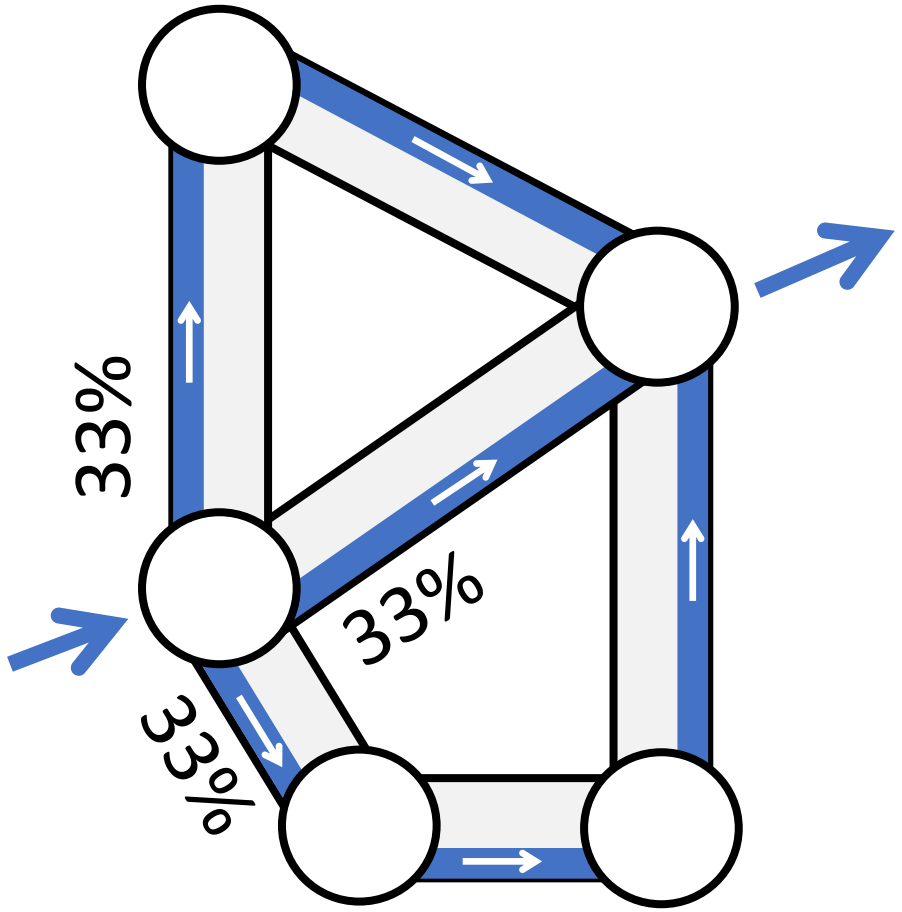
- Spread traffic out to leave spare capacity in case traffic levels increase
- A well-known technique called MinMax
- Does not care about propagation delay



Minimizing utilization avoids congestion

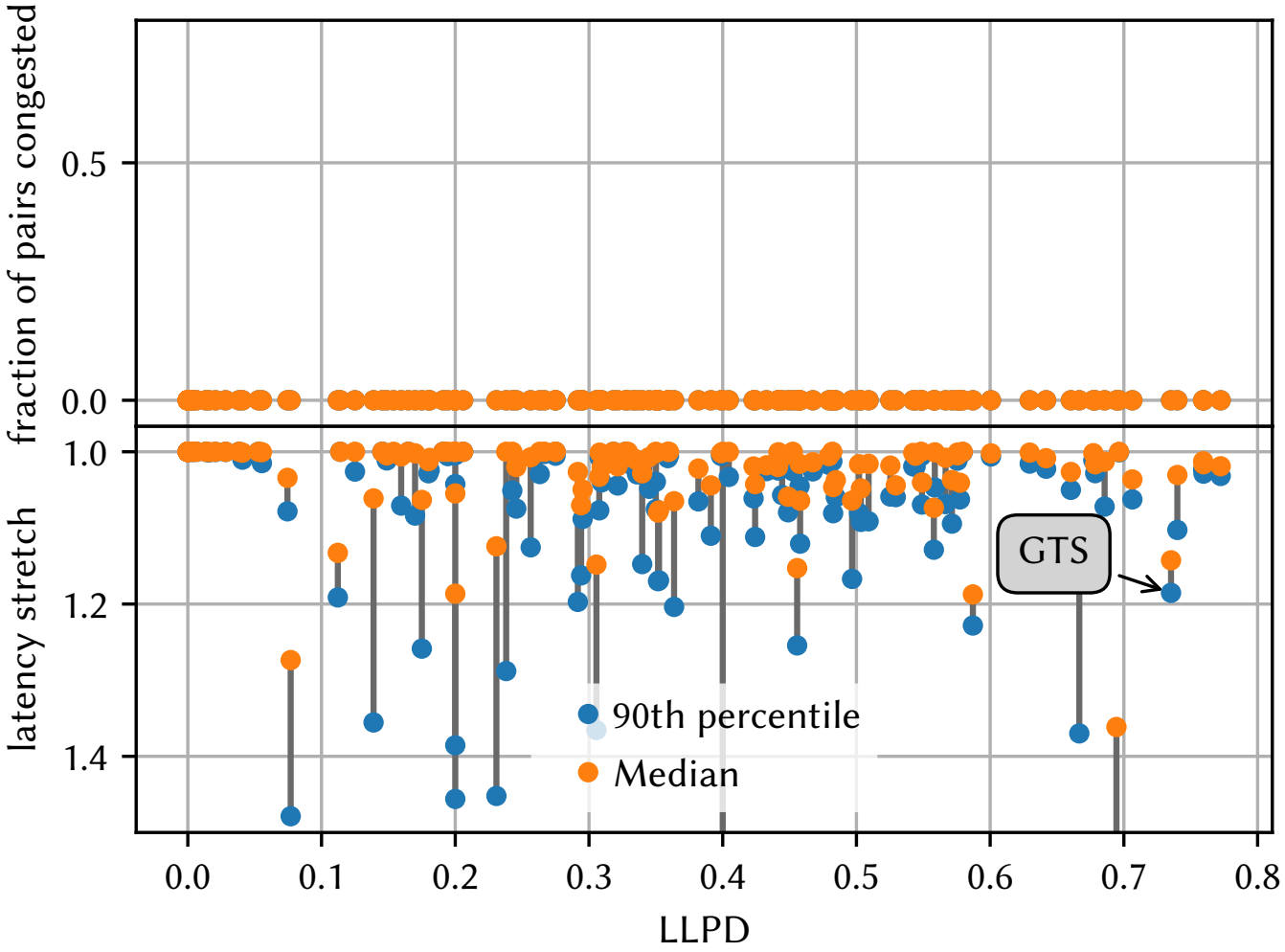
- Spread traffic out to leave spare capacity in case traffic levels increase
- A well-known technique called MinMax
- Does not care about pro

How does MinMax do?



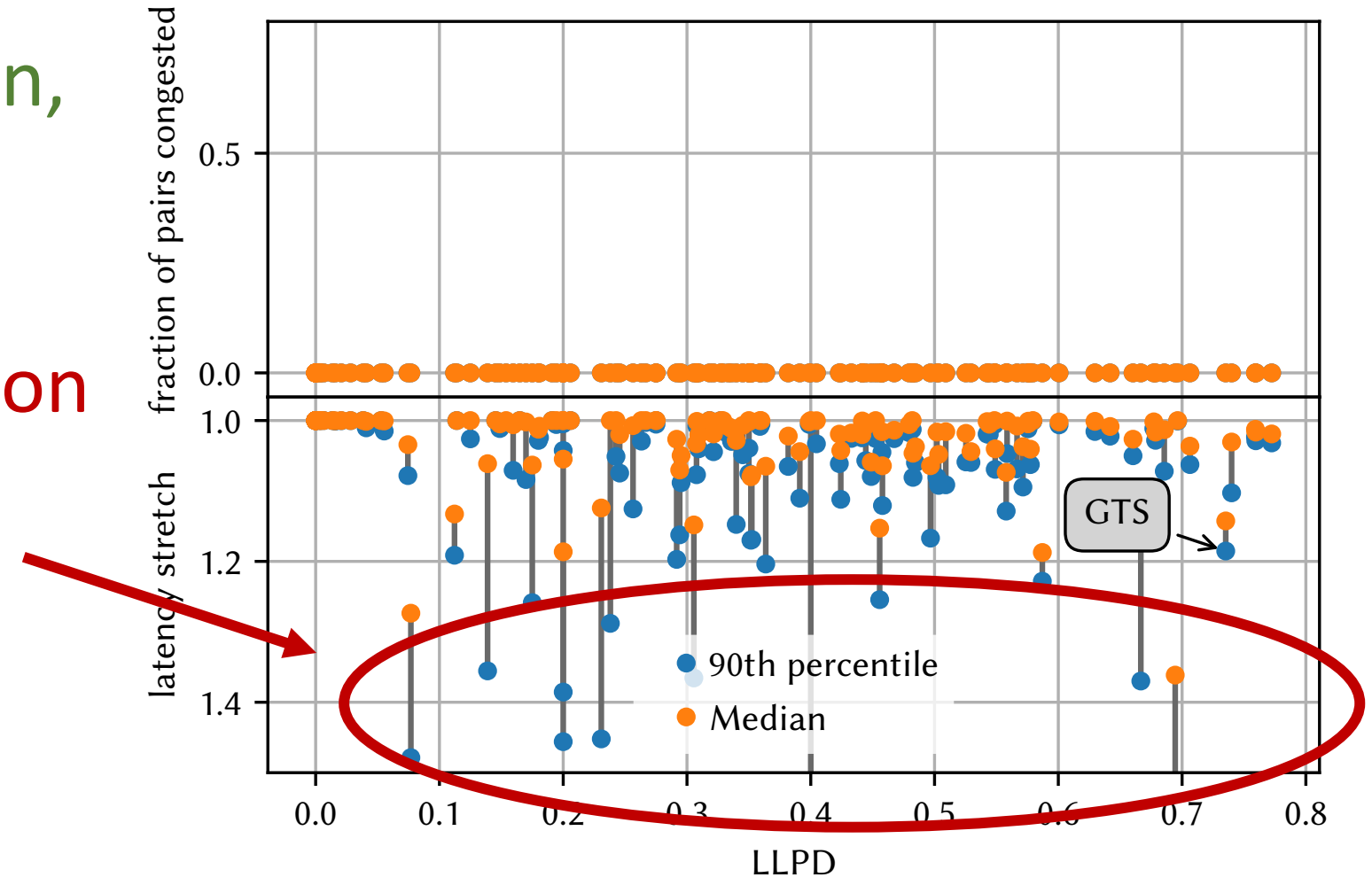
MinMax inflates propagation delay

- Minimizes utilization, designed to avoid congestion



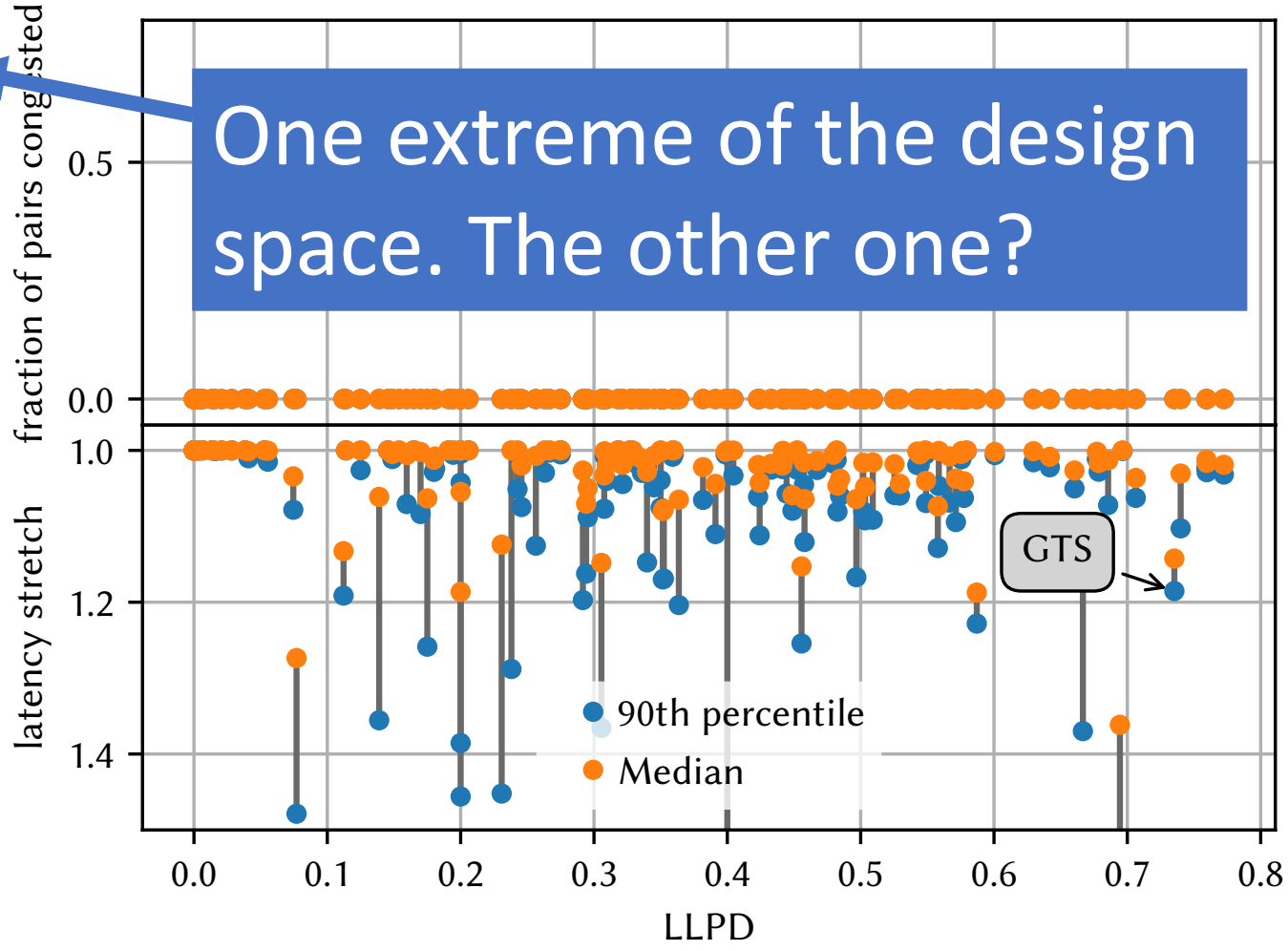
MinMax inflates propagation delay

- Minimizes utilization, designed to avoid congestion
- Routes some flows on paths with high propagation delay



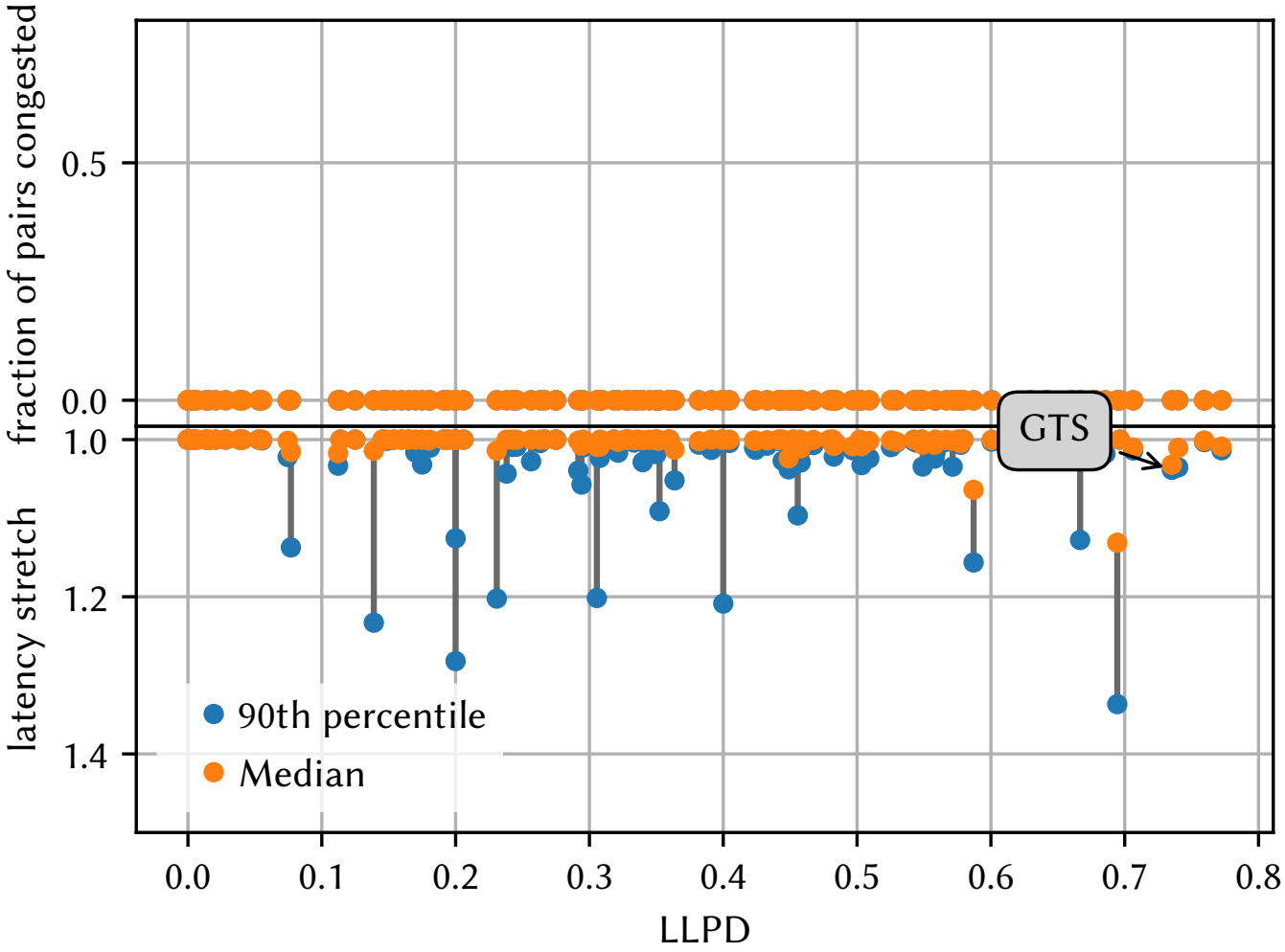
MinMax inflates propagation delay

- Minimizes utilization, designed to avoid congestion
- Routes some flows on paths with high propagation delay



Latency-optimal placement

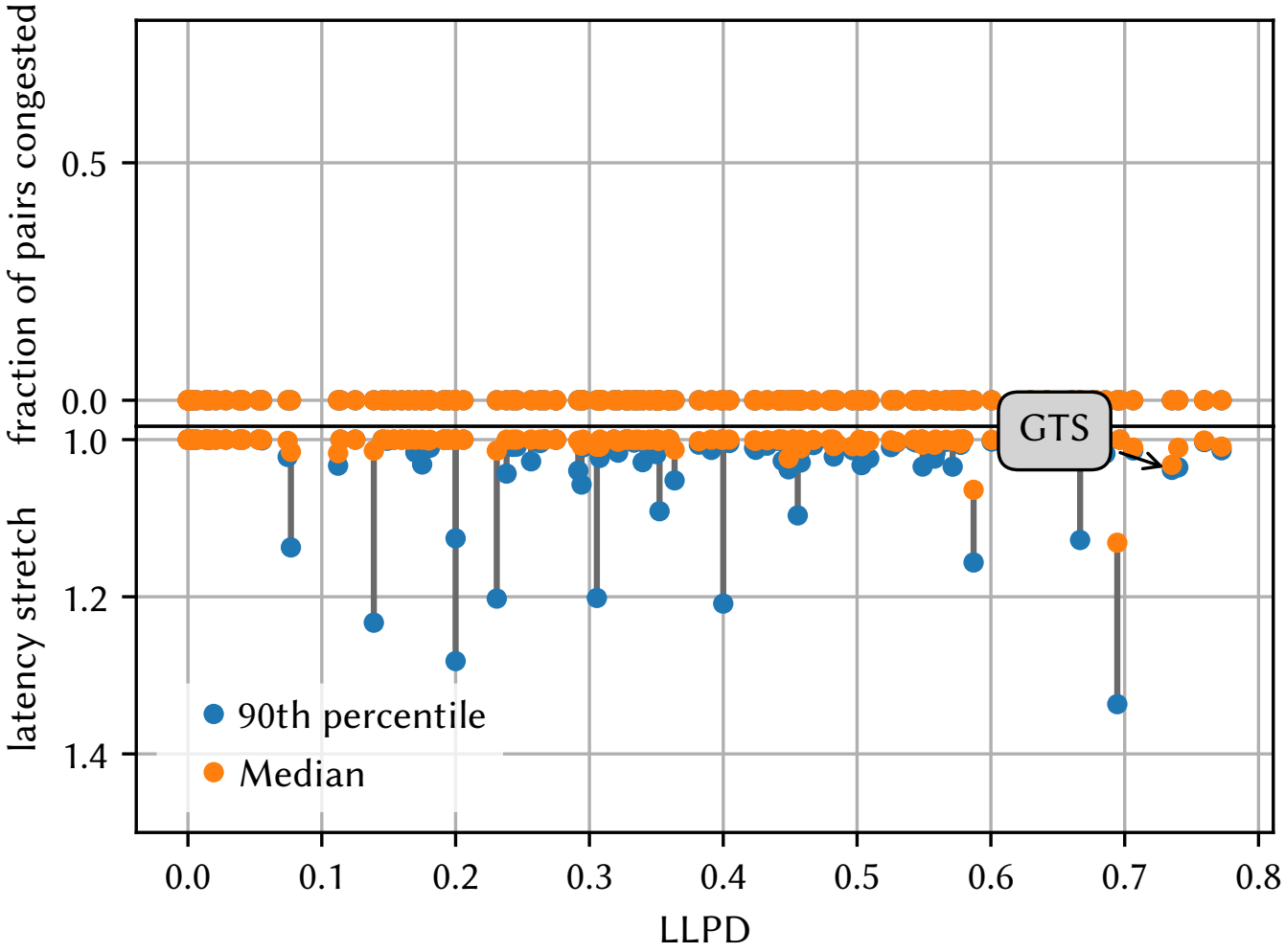
- Minimizes prop delay and avoids congestion
- Maximizes utilization of links on low-delay paths



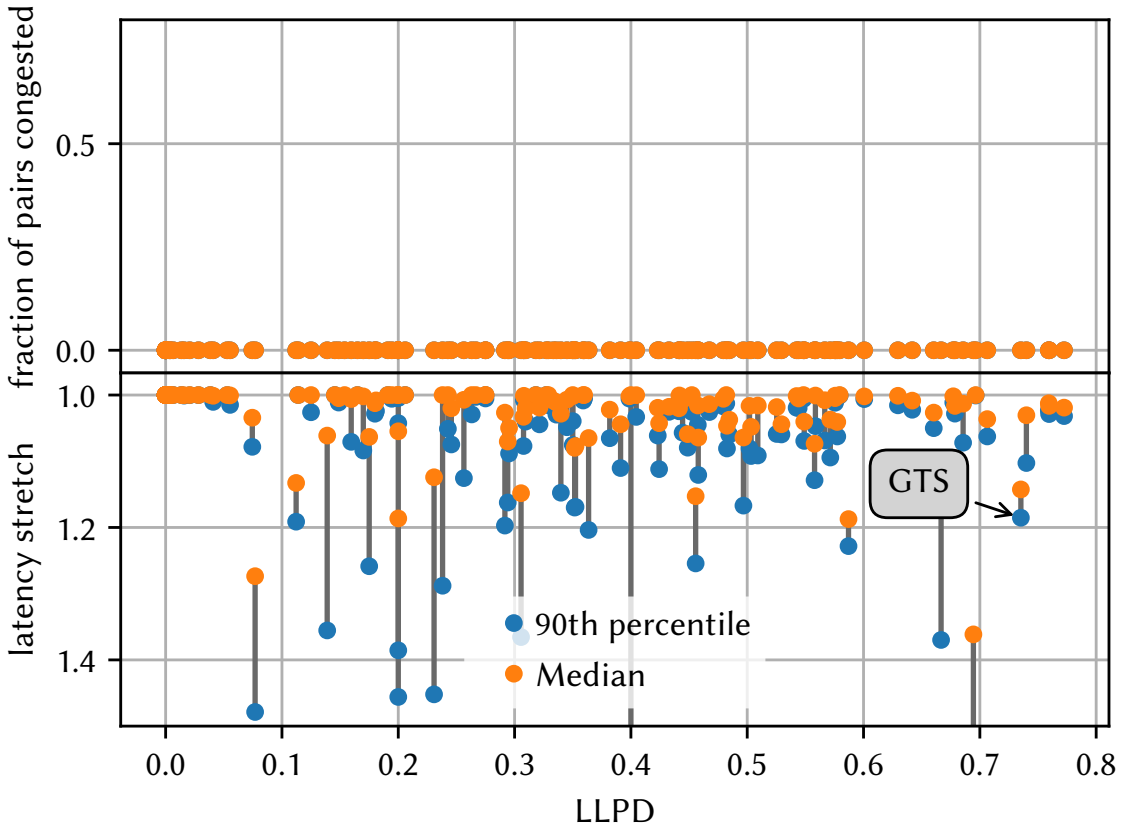
Latency-optimal placement

- Minimizes prop delay and avoids congestion
- Maximizes utilization of links on low-delay paths

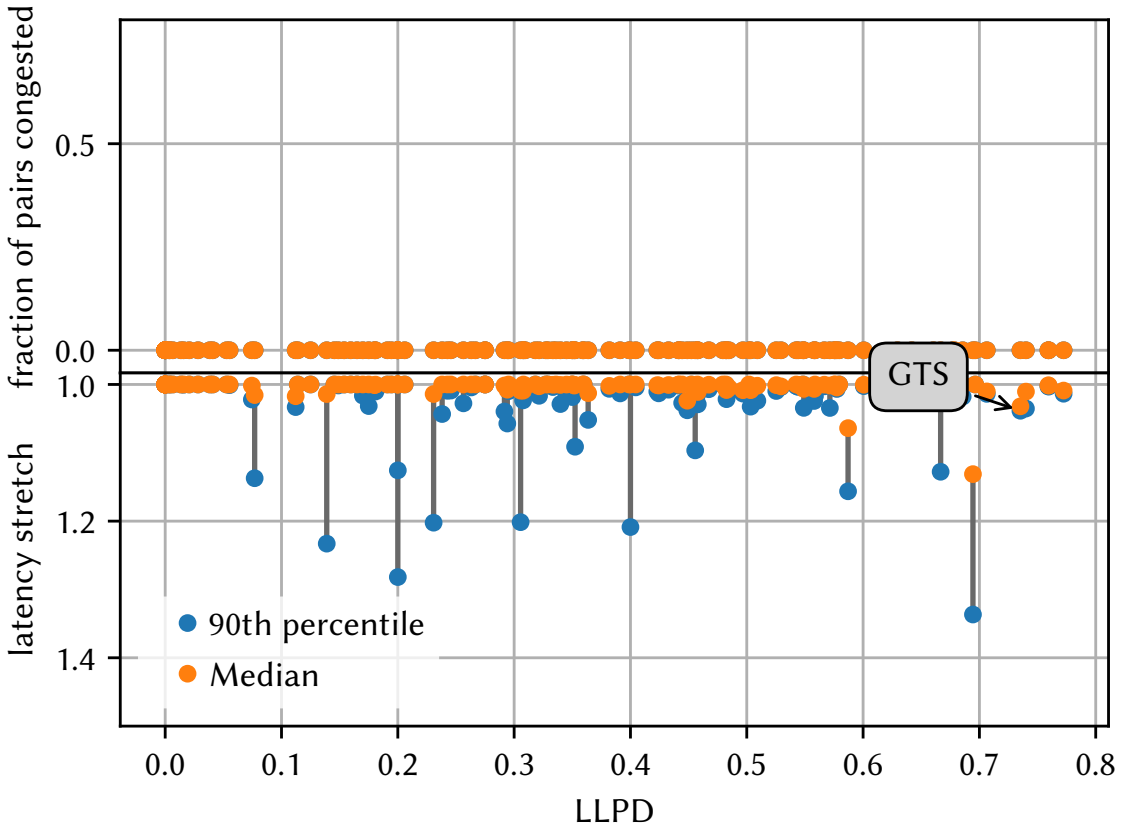
Assume it is possible to compute this at scale, more about that later…



Two extremes of congestion-free routing

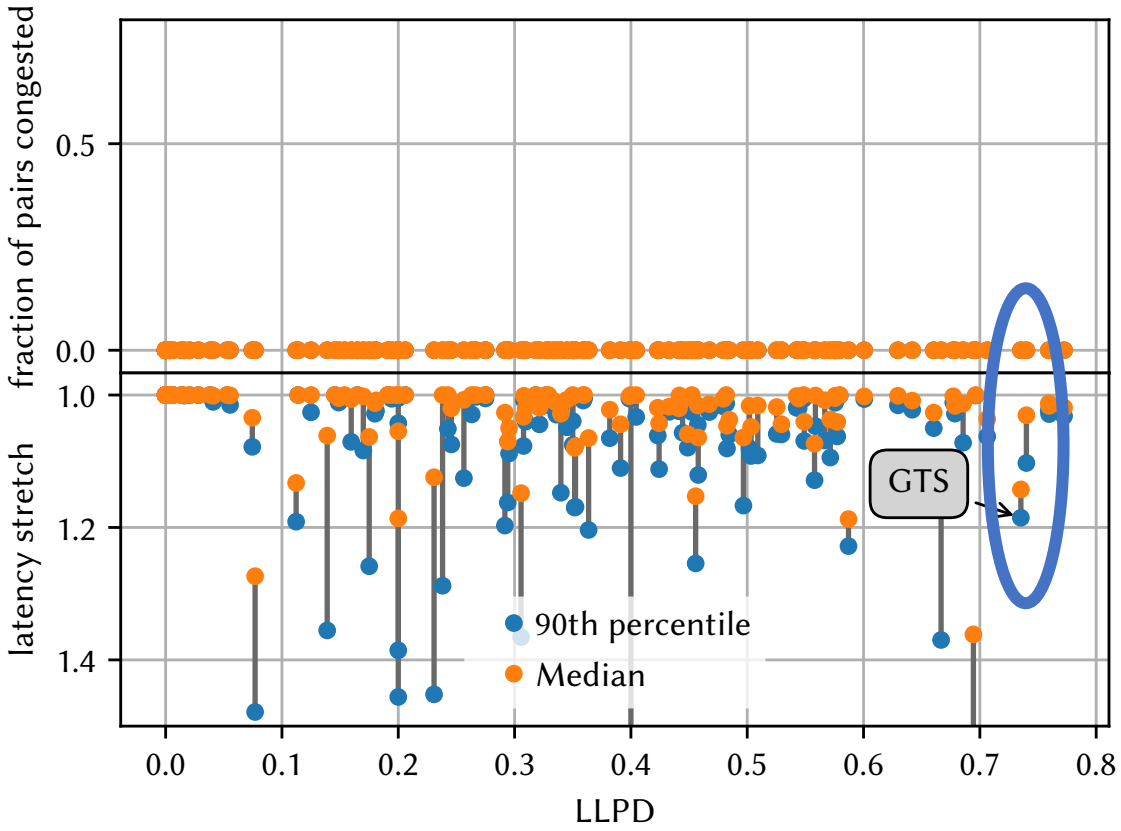


Minimize utilization (MinMax)

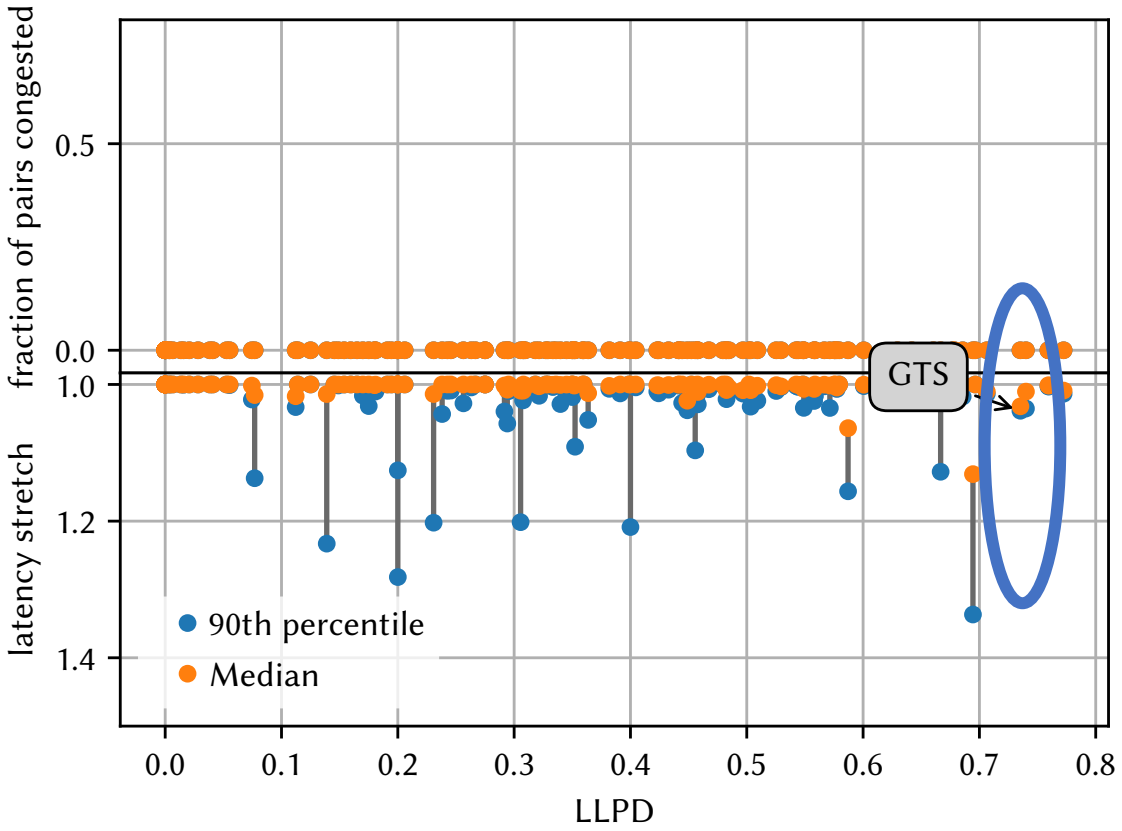


Minimize propagation delay and avoid congestion

Two extremes of congestion-free routing

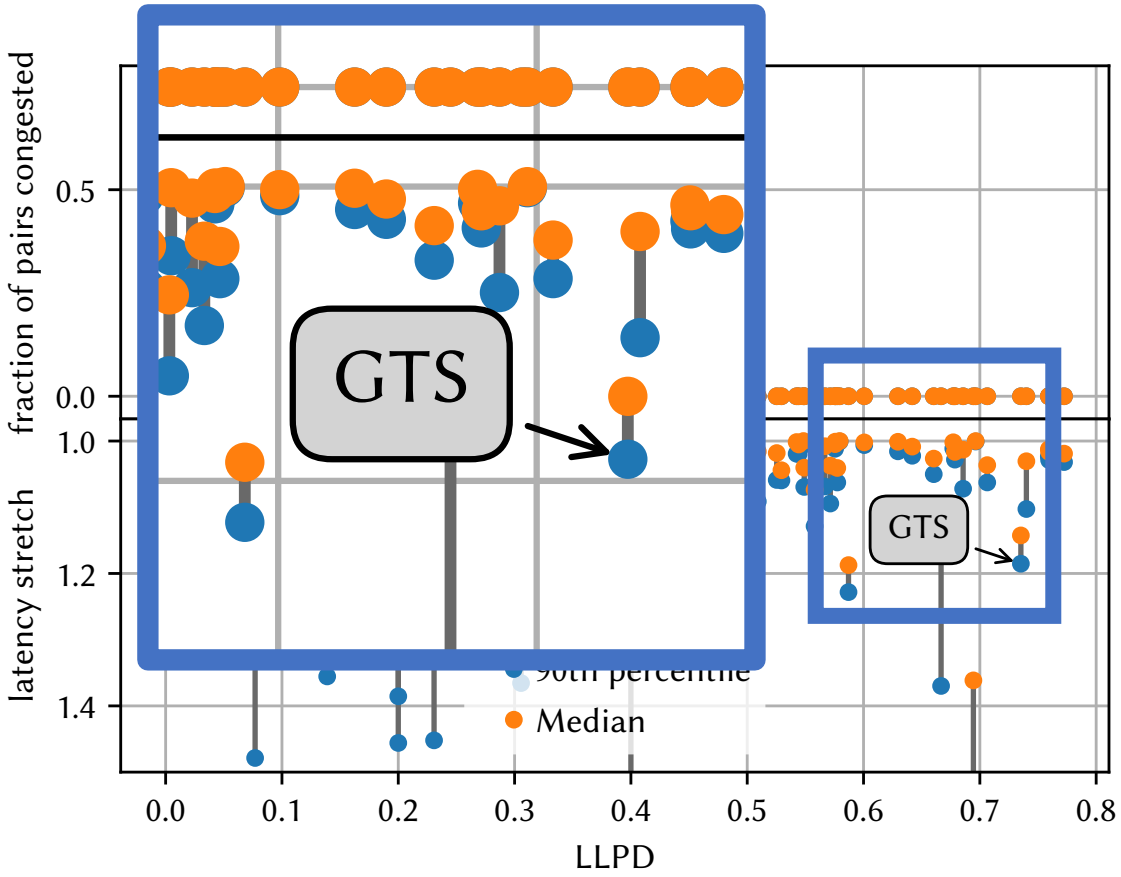


Minimize utilization (MinMax)

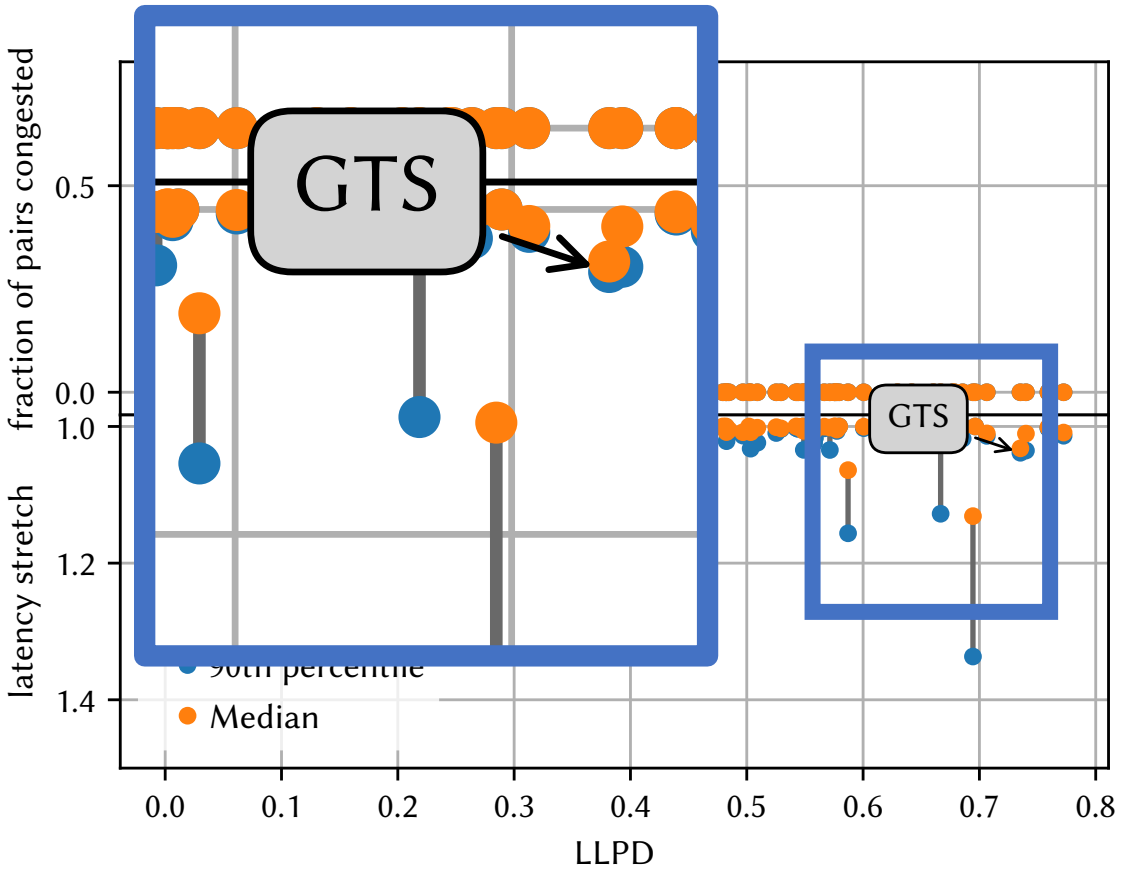


Minimize propagation delay and avoid congestion

Two extremes of congestion-free routing

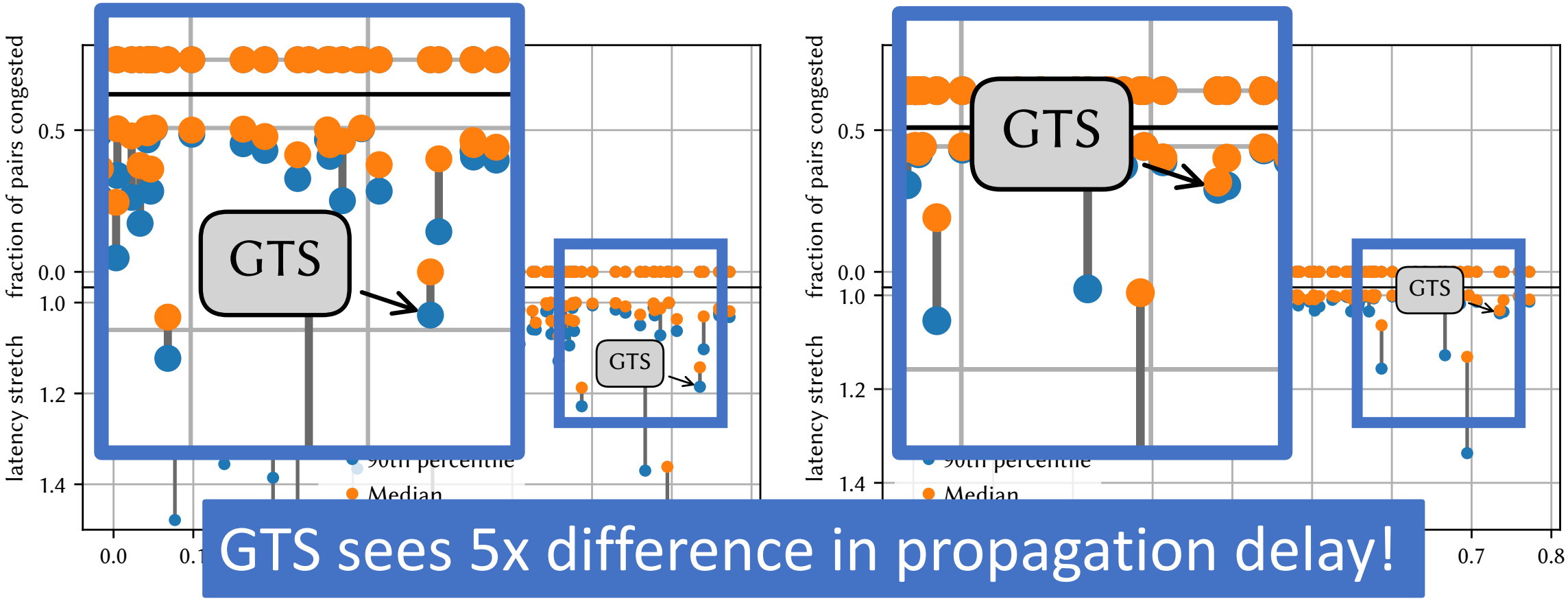


Minimize utilization (MinMax)



Minimize propagation delay and avoid congestion

Two extremes of congestion-free routing

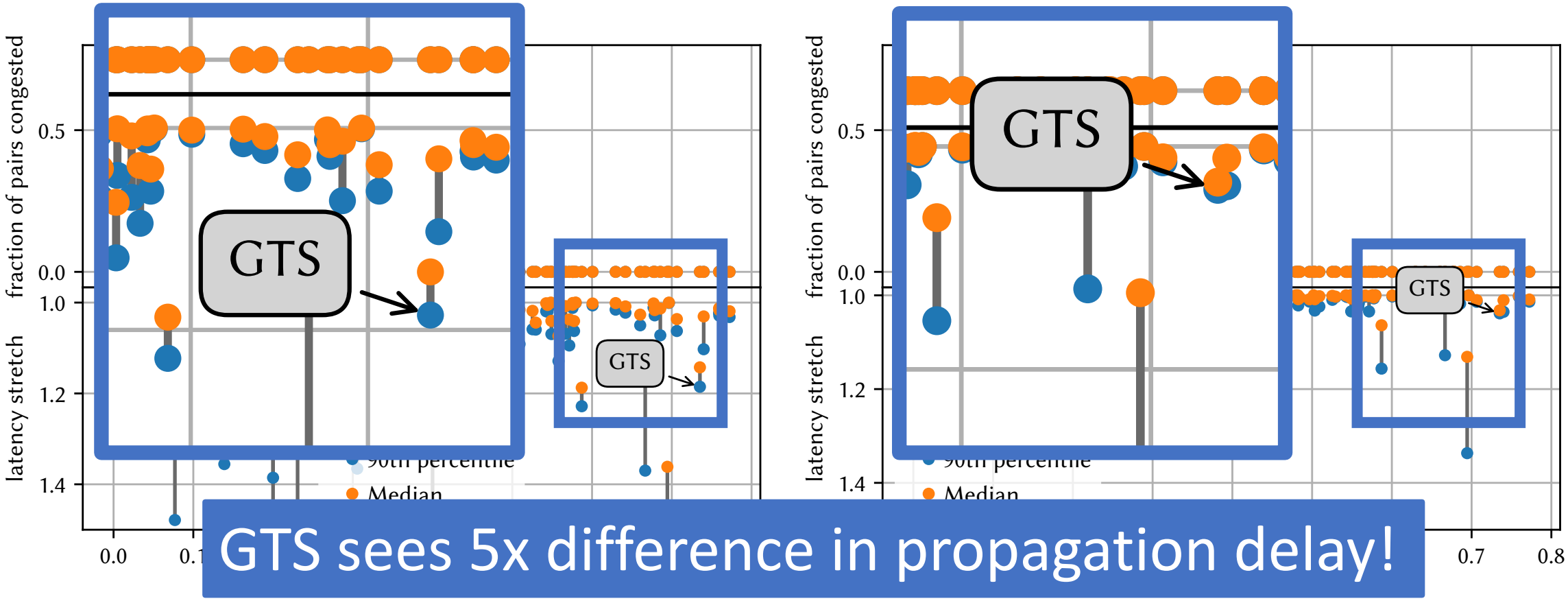


GTS sees 5x difference in propagation delay!

Minimize utilization (MinMax)

Minimize propagation delay and avoid congestion

Two extremes of congestion-free routing



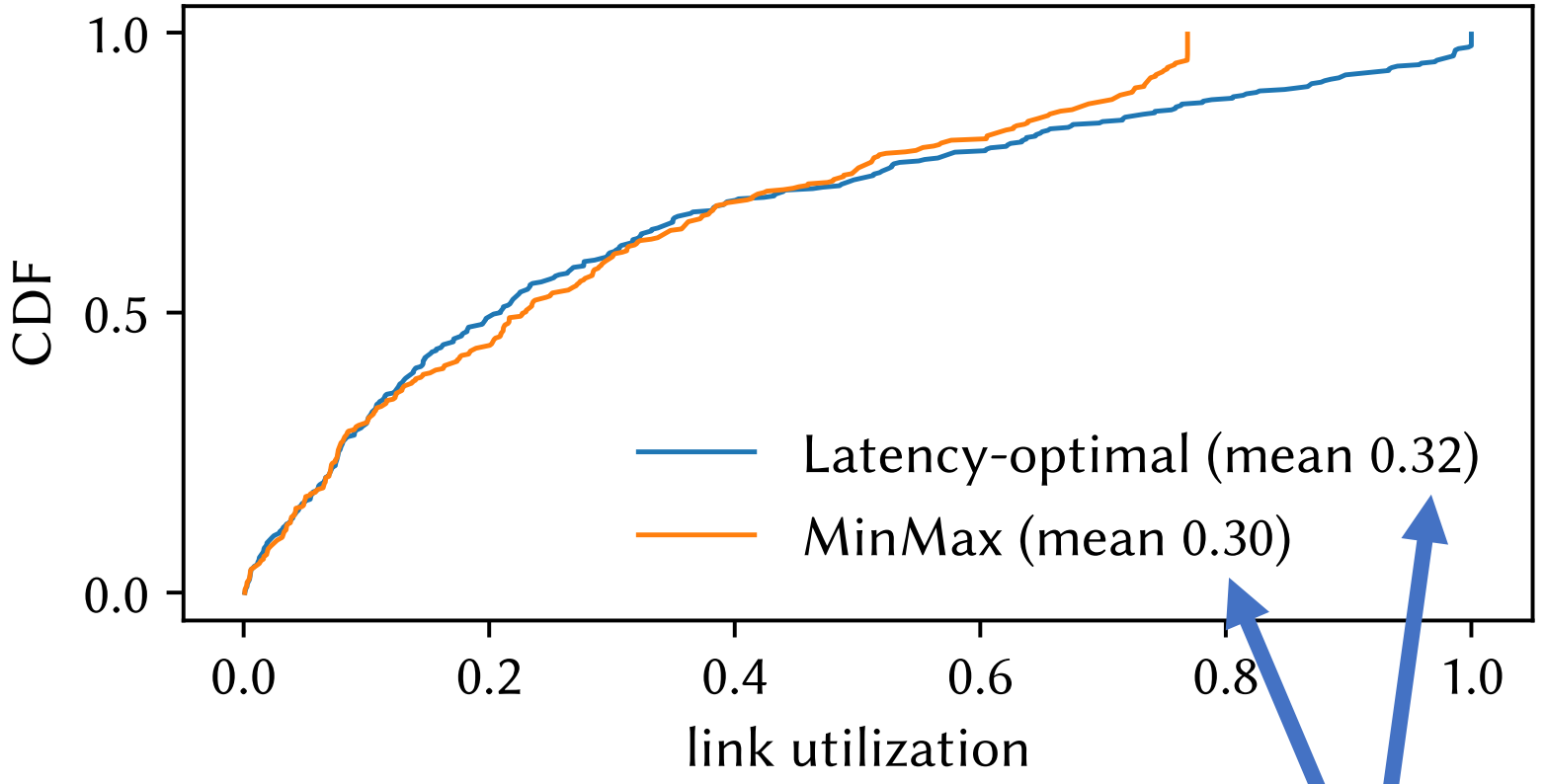
GTS sees 5x difference in propagation delay!

Minimize utilization (MinMax)

Minimize propagation delay and avoid congestion

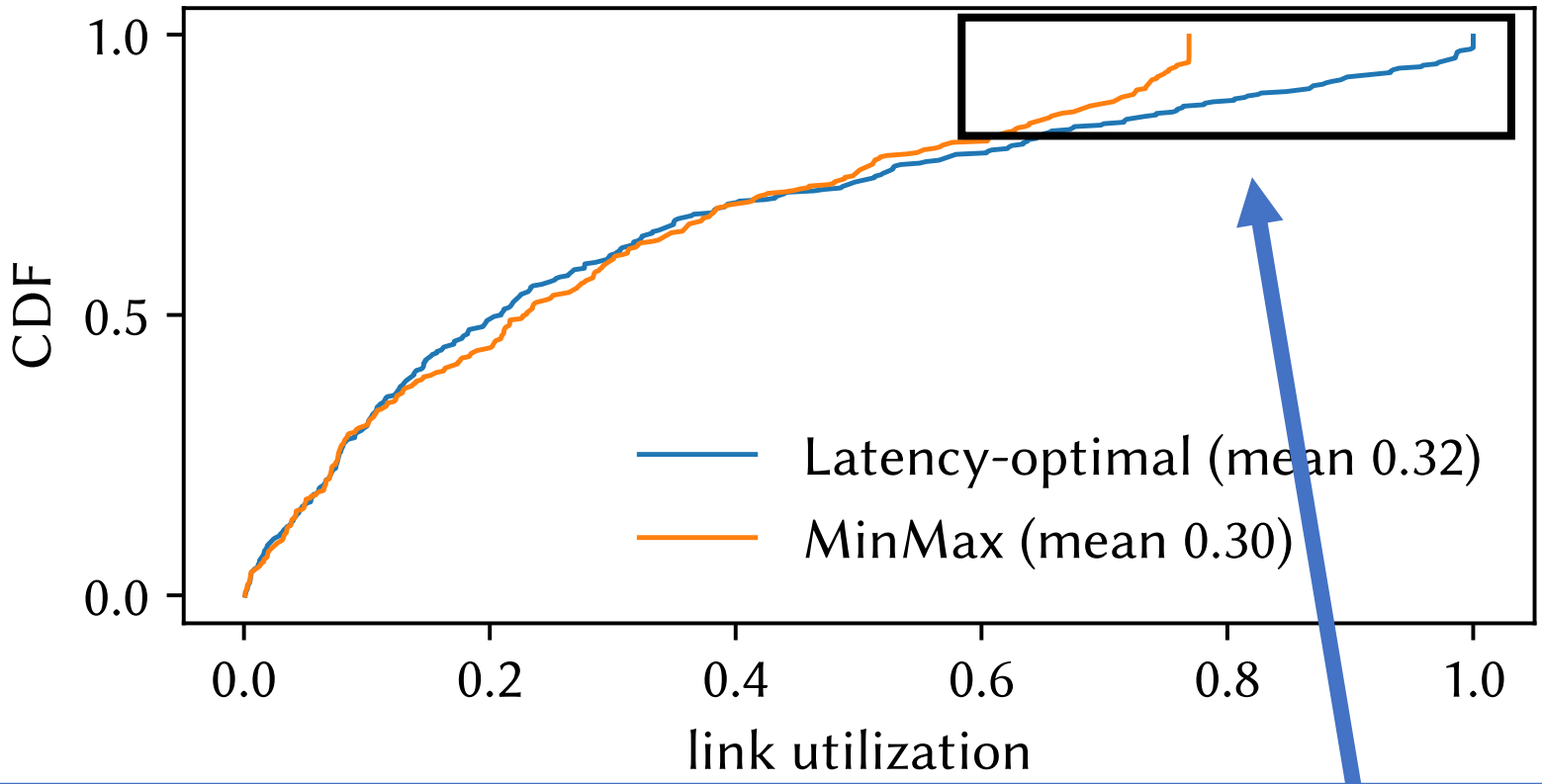
Let's focus on a single traffic matrix

Two extremes of congestion-free routing



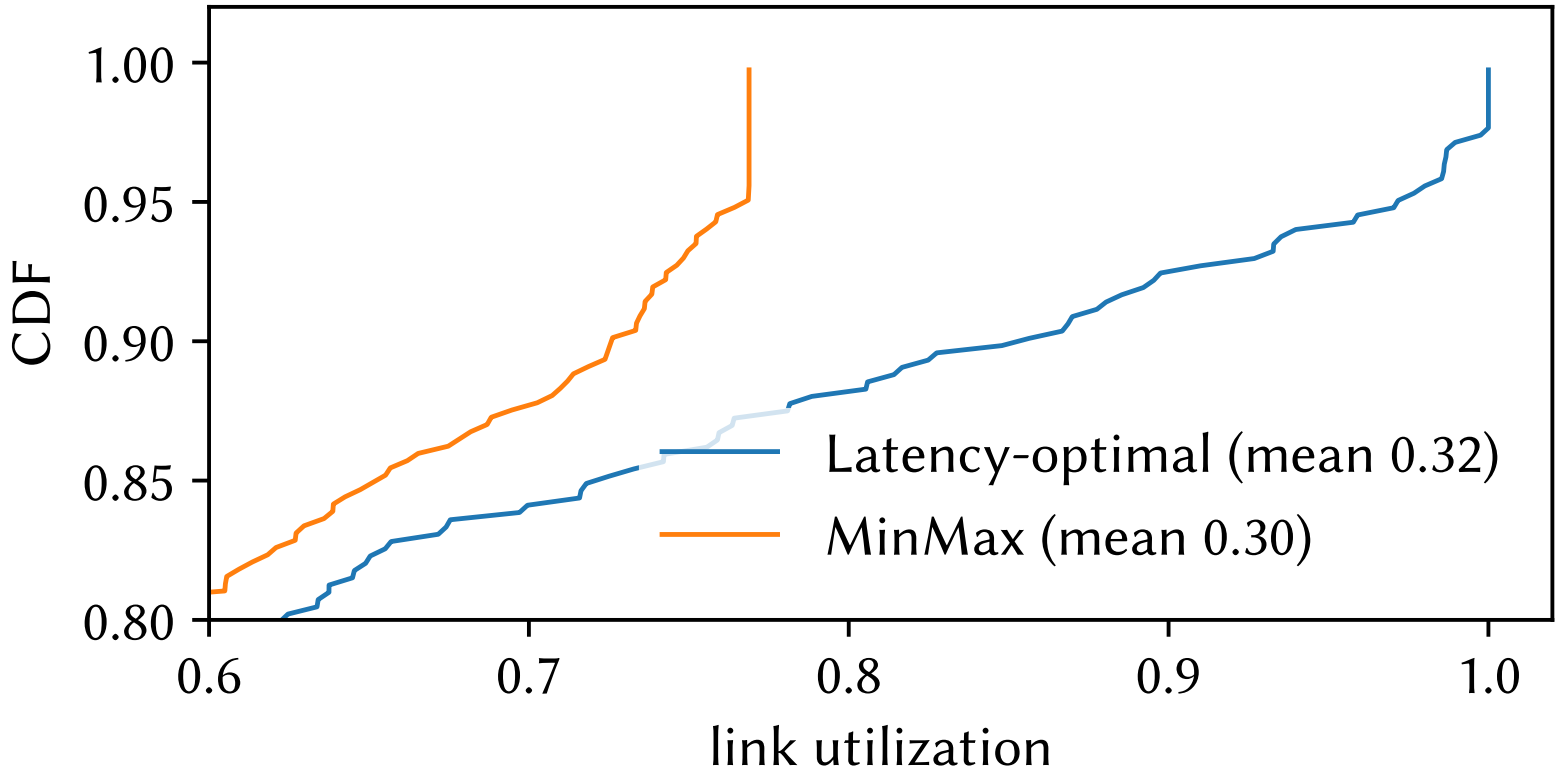
Significant delta in prop delay, but mean utilization the same

Two extremes of congestion-free routing

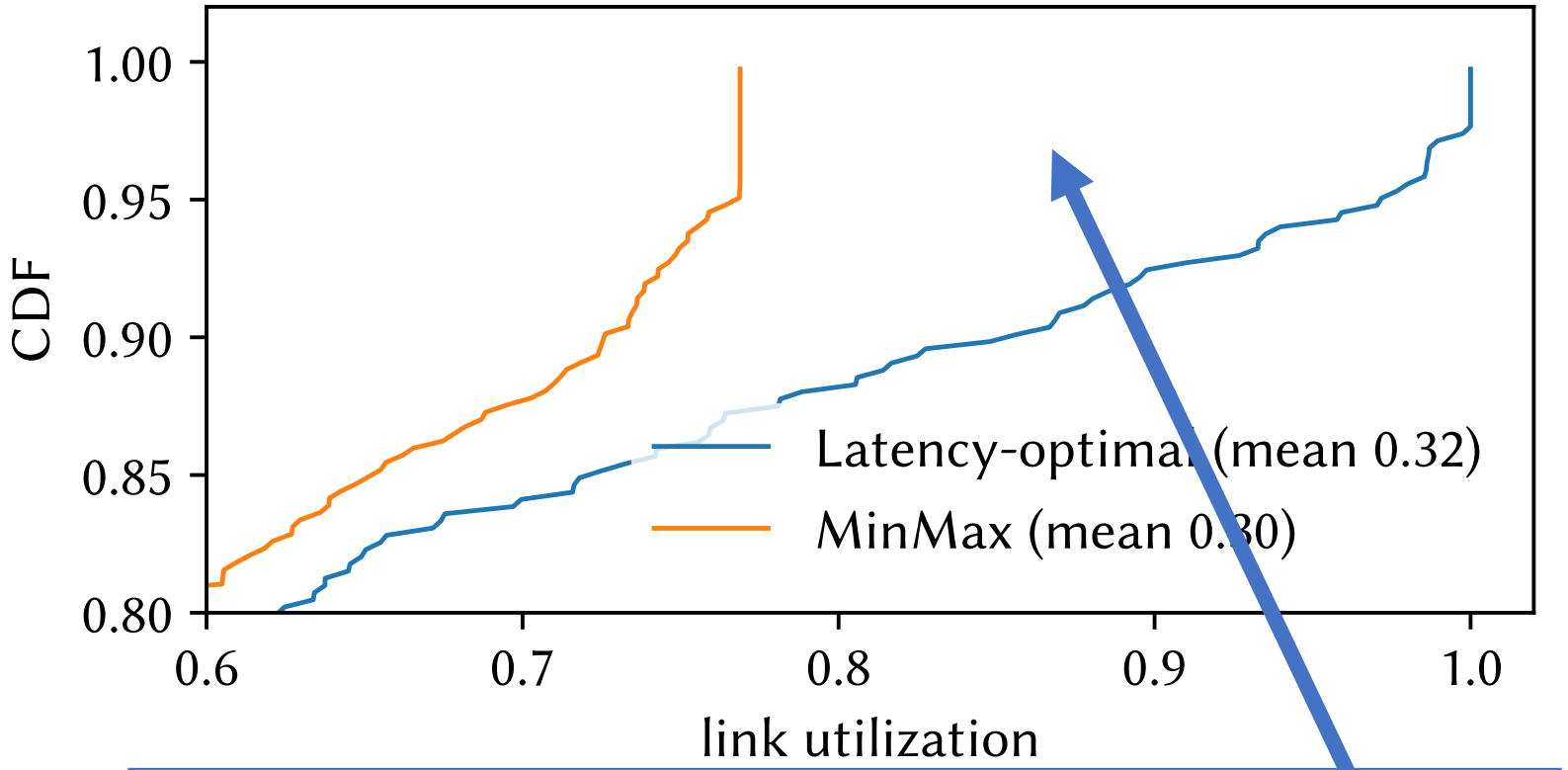


Links on short prop-delay paths “in demand” in latency-optimal placement

Two extremes of congestion-free routing

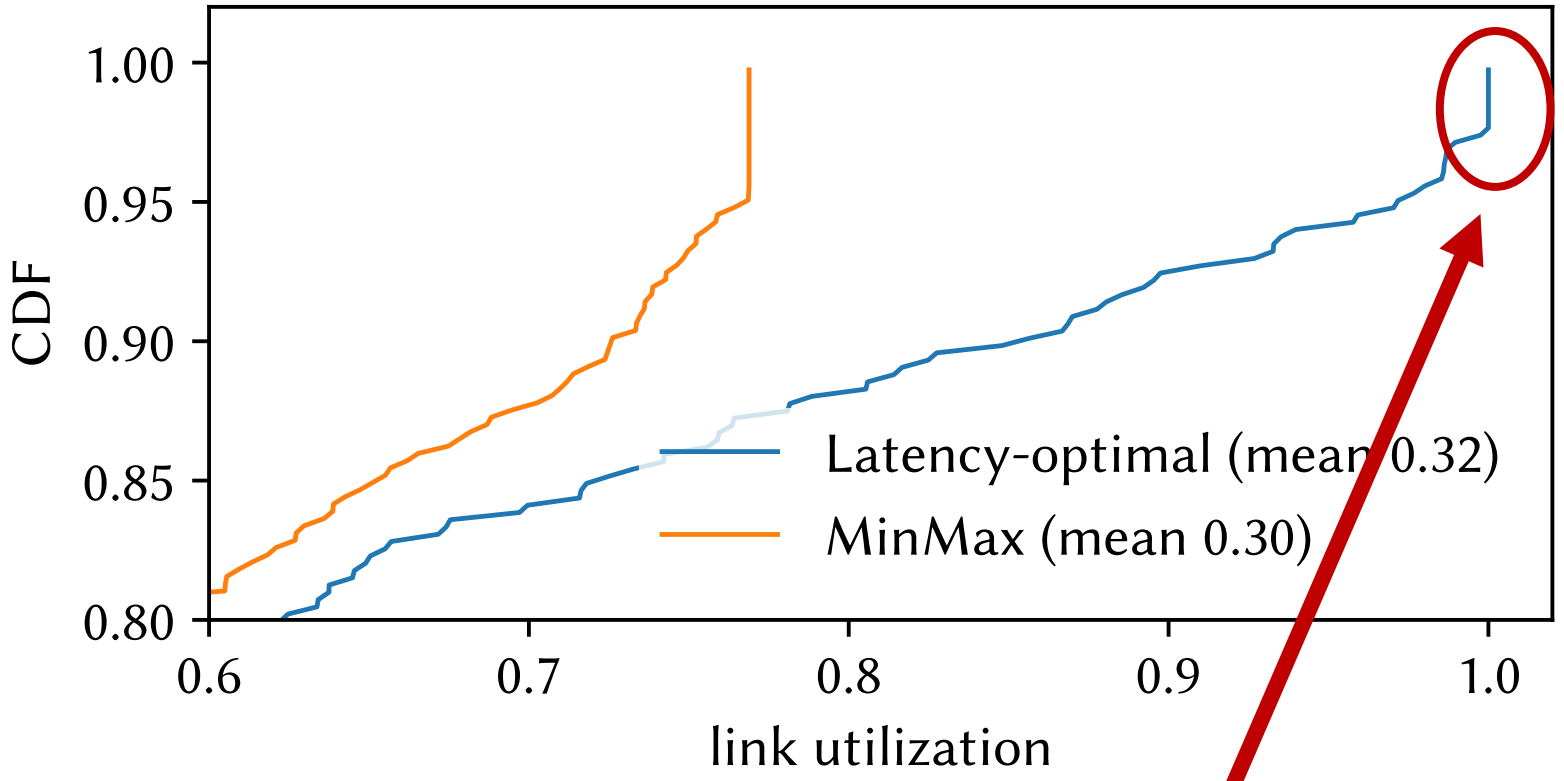


Two extremes of congestion-free routing



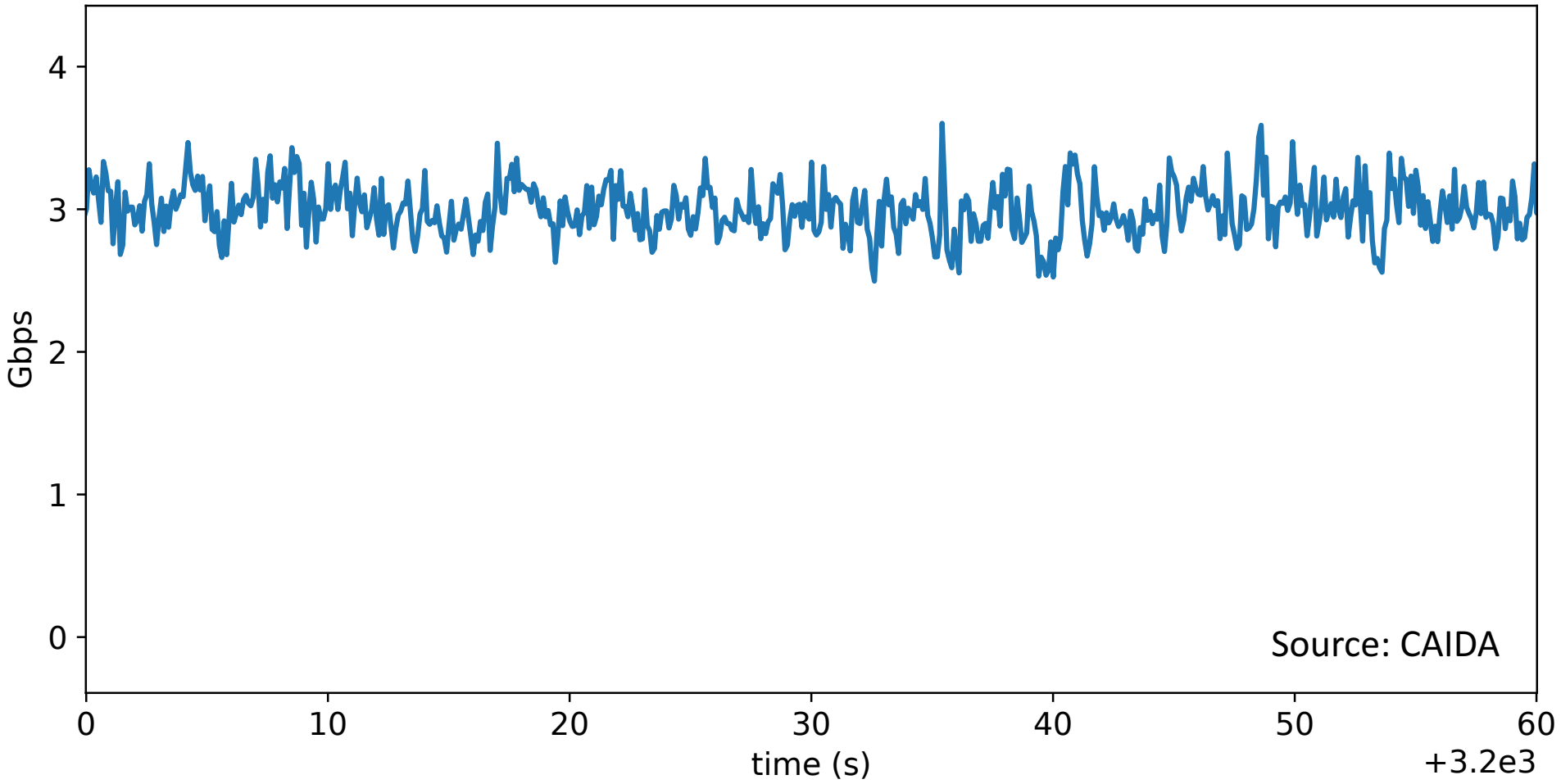
All possible congestion-free routing solutions lie in this range

Two extremes of congestion-free routing

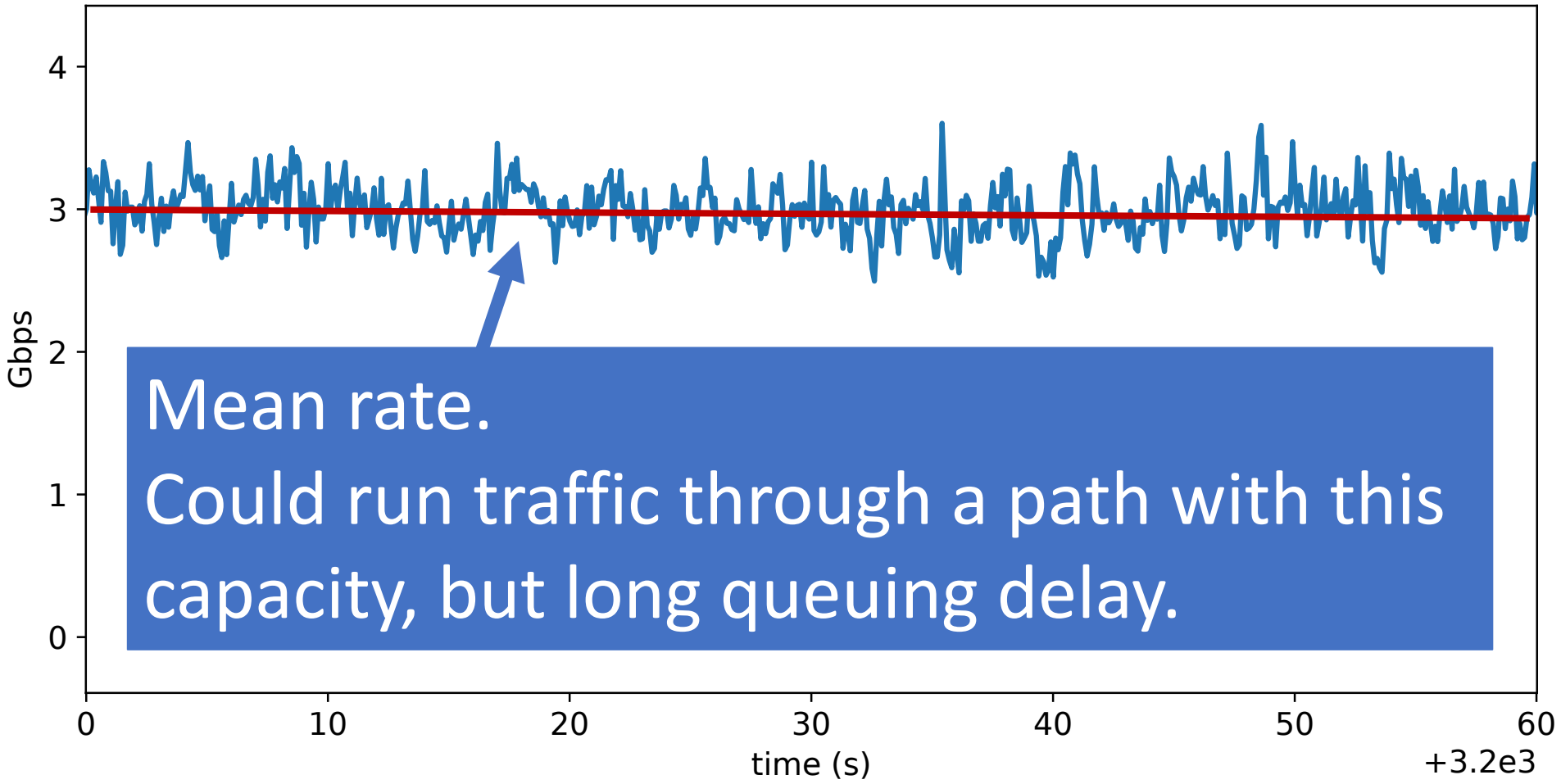


100% utilization? On an ISP?

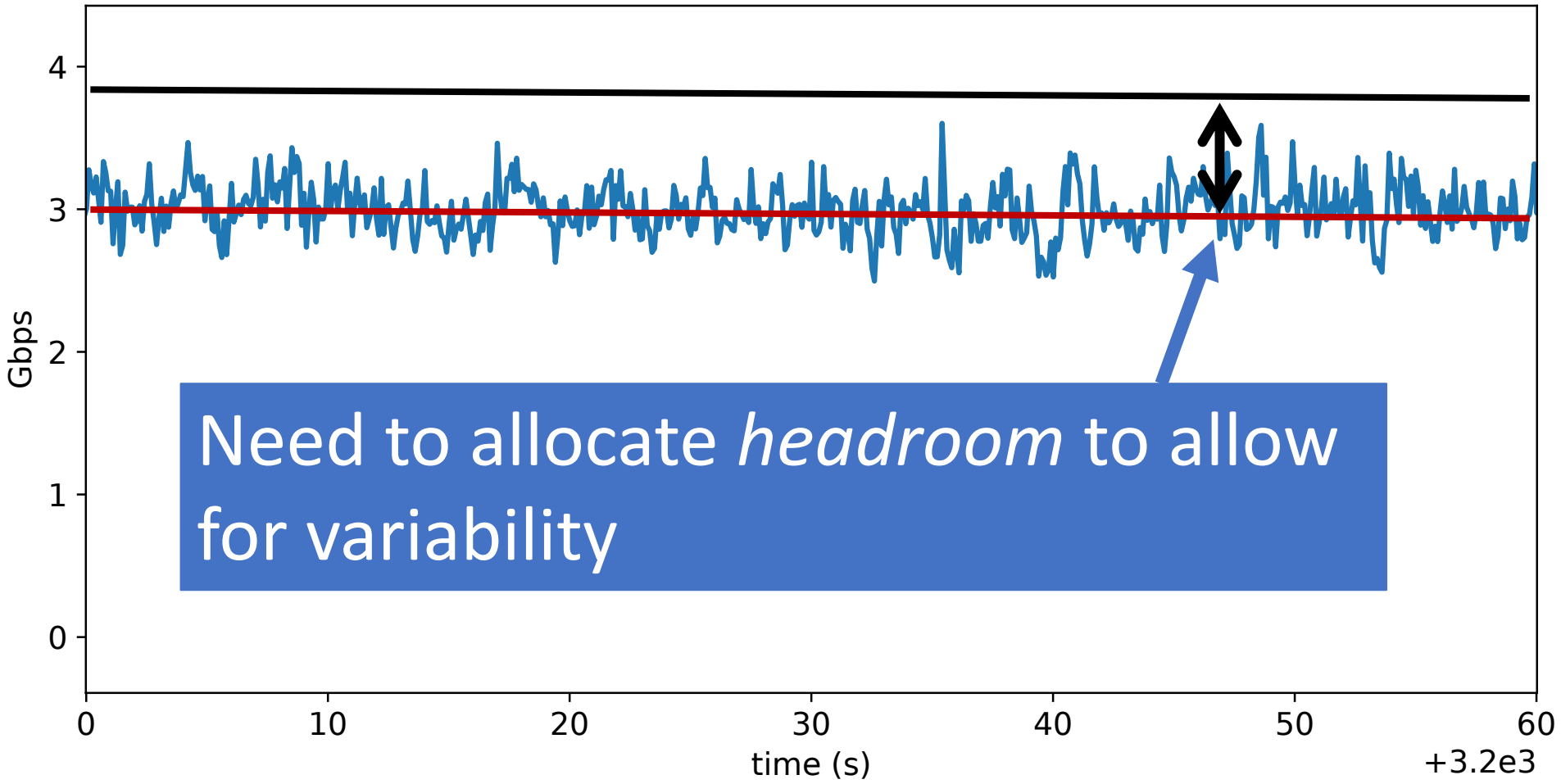
A minute from a core link



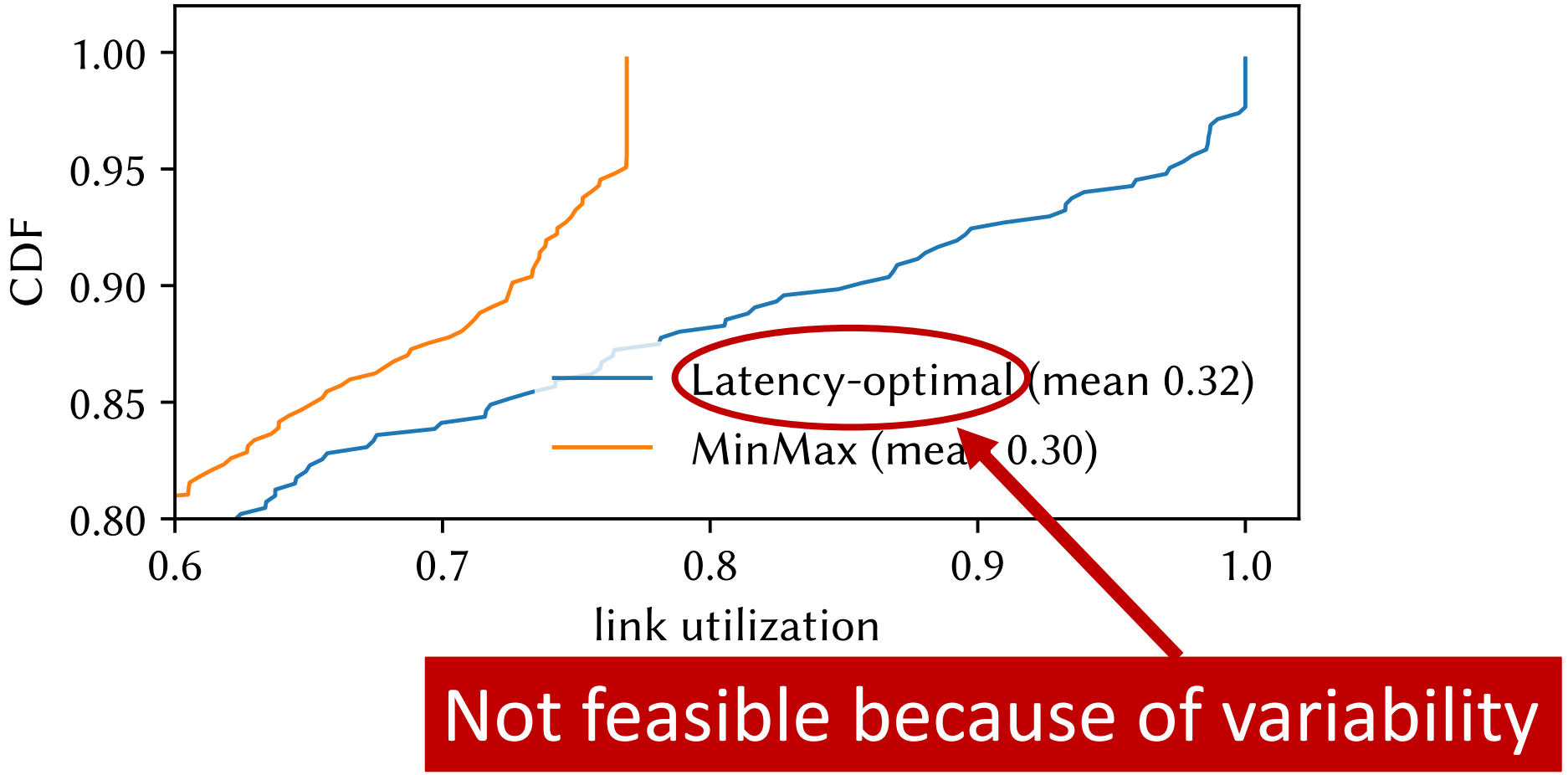
A minute from a core link



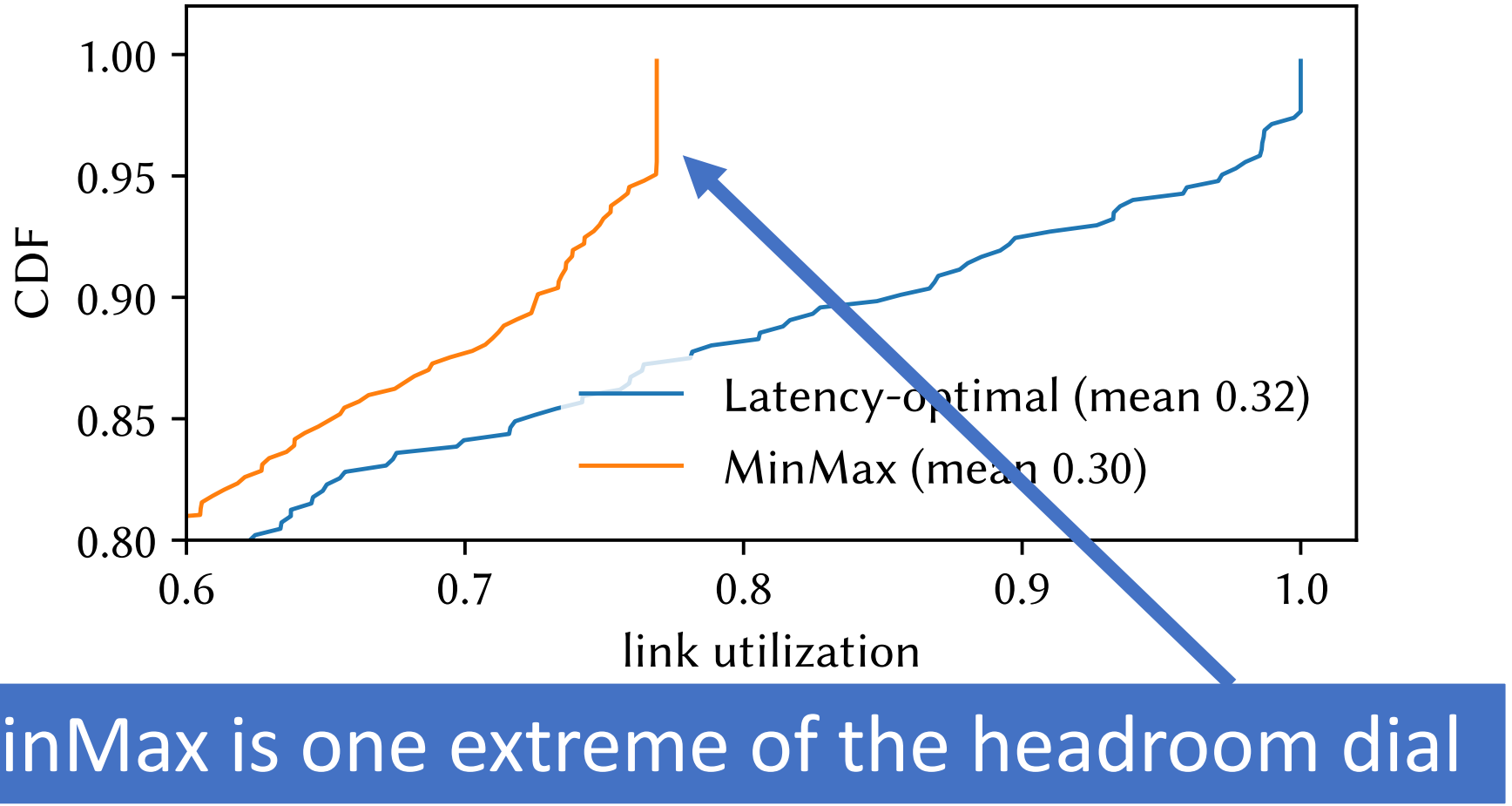
A minute from a core link



The headroom dial

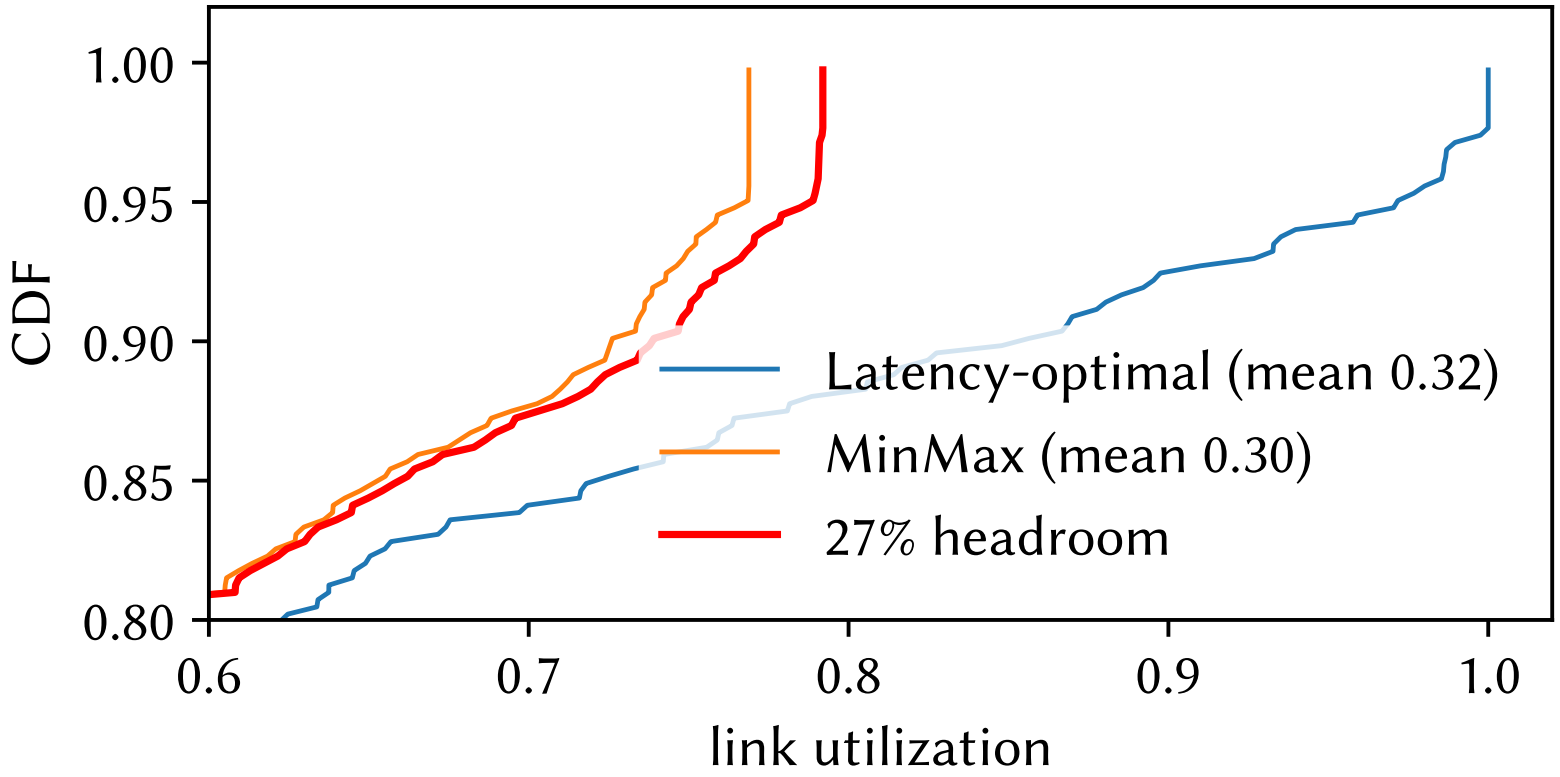


The headroom dial

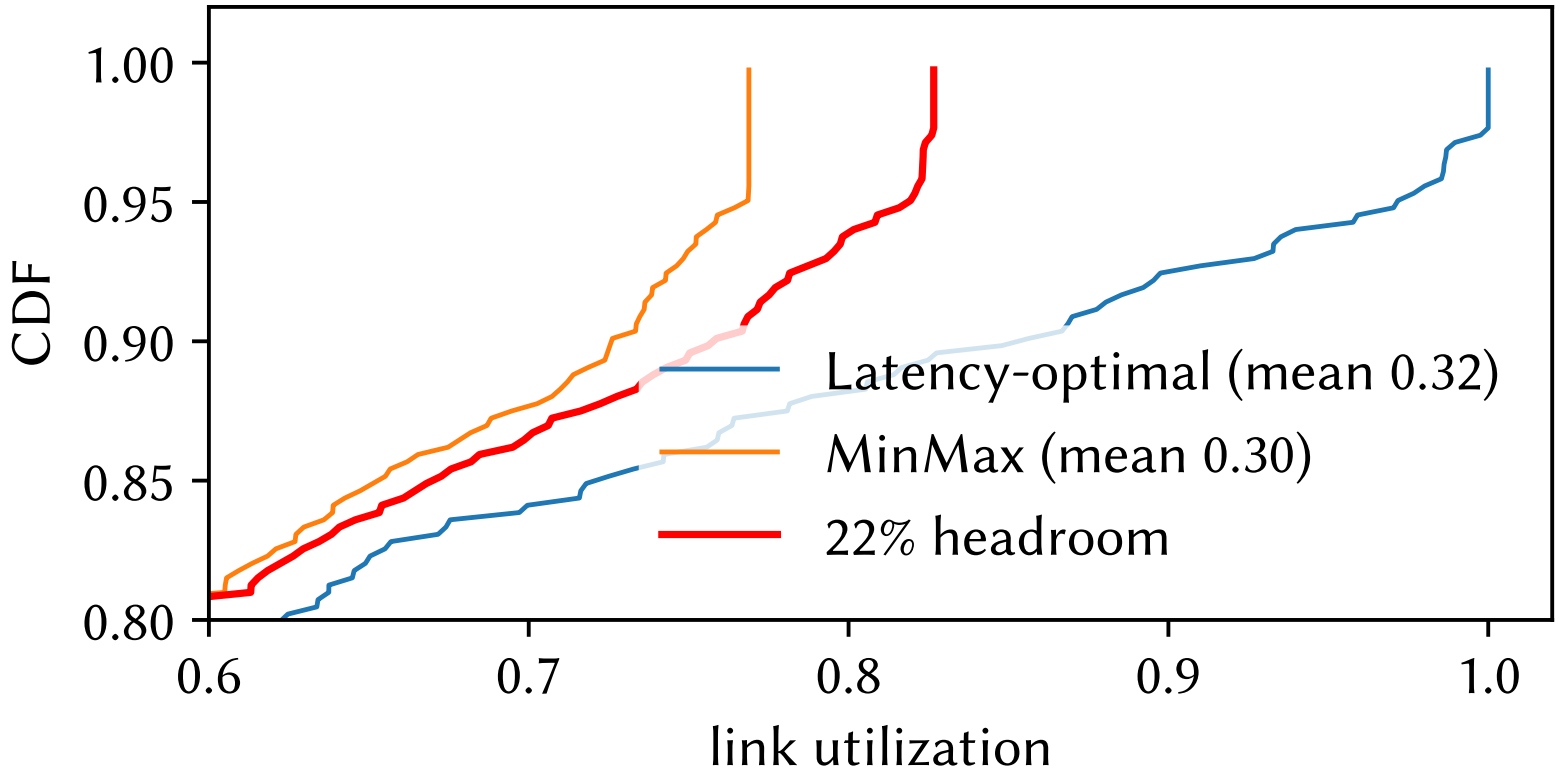


MinMax is one extreme of the headroom dial

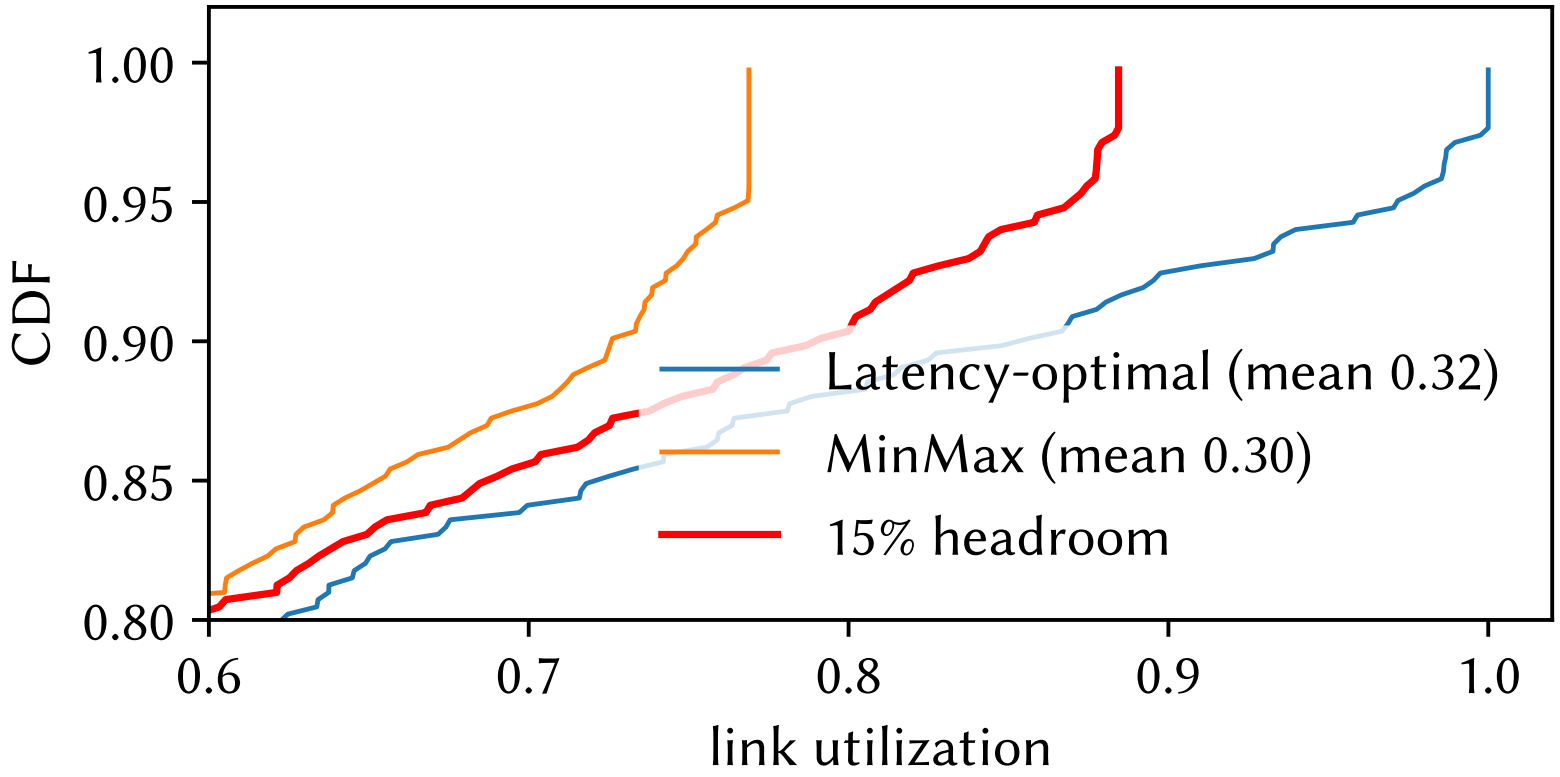
The headroom dial



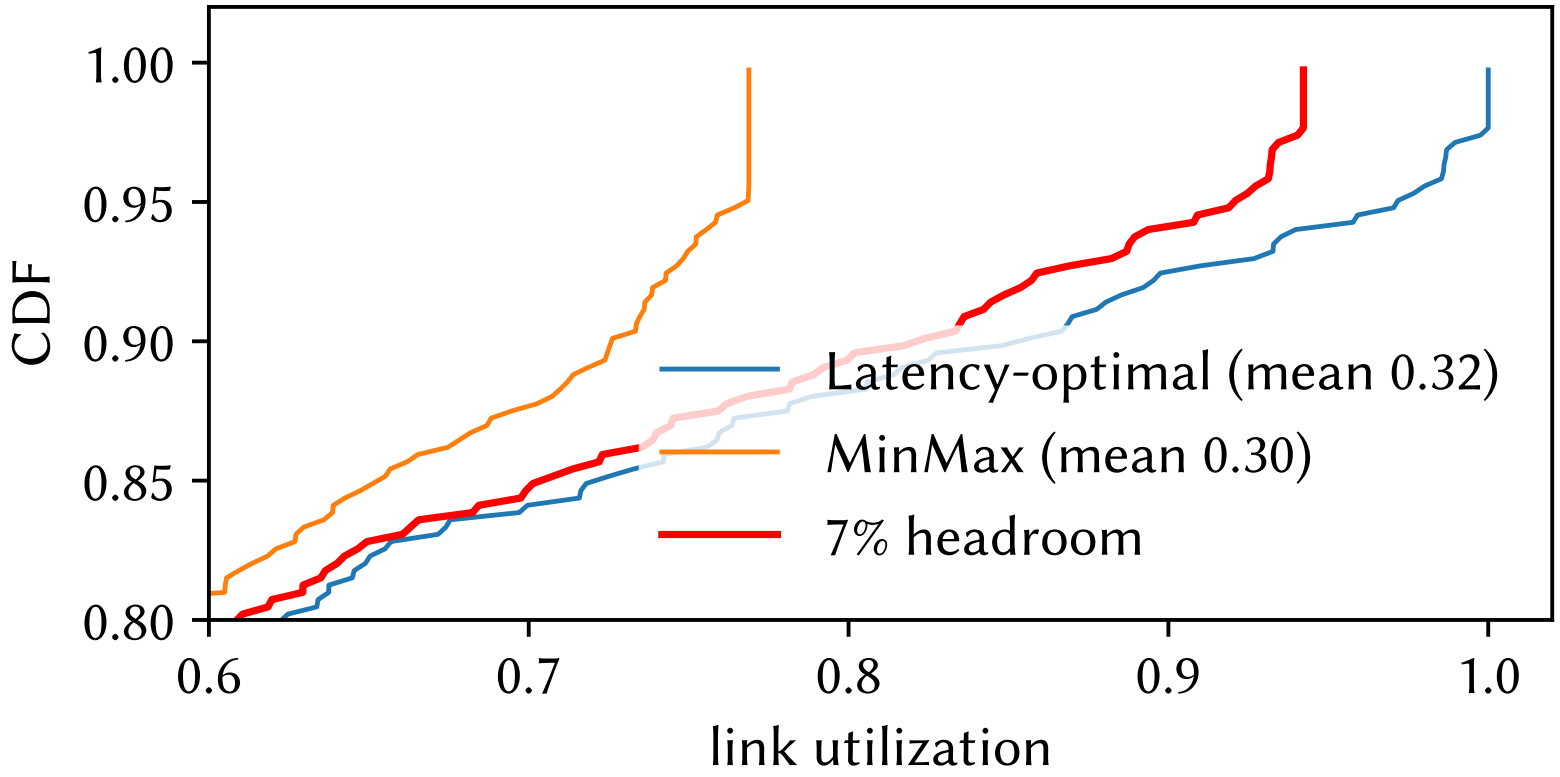
The headroom dial



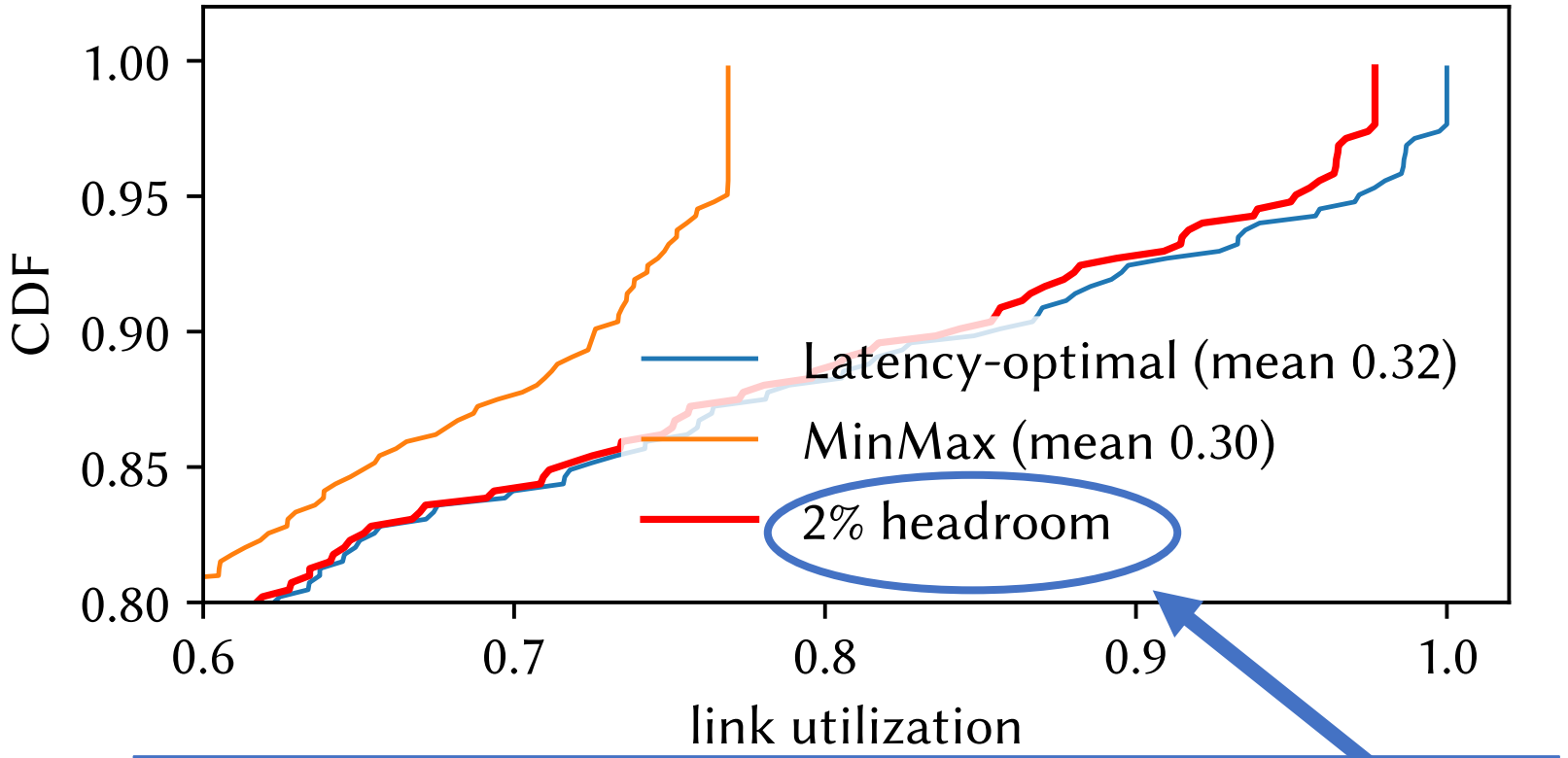
The headroom dial



The headroom dial



The headroom dial



Need to allow the minimal amount of headroom to cope with variability

Towards a low-latency routing system

Towards a low-latency routing system

Compute latency-optimal routing solution, sans headroom

- expressed the problem as one big linear program (largely straightforward)
- efficient *iterative* solution: add paths, solve, repeat …
- 400+ nodes, less than one second (vs. tens of minutes…)

Towards a low-latency routing system

Compute latency-optimal routing solution, sans headroom

- expressed the problem as one big linear program (largely straightforward)
- efficient *iterative* solution: add paths, solve, repeat …
- 400+ nodes, less than one second (vs. tens of minutes…)

Tune headroom dial to drive routing as close as possible to optimal solution while avoiding congestion

- predict how aggregates will statistically multiplex on a path by *convolving* their past demands

Towards a low-latency routing system

Compute latency-optimal routing solution, sans headroom

- expressed the problem as one big linear program (largely straightforward)
- efficient *iterative* solution: add paths, solve, repeat …
- 400+ nodes, l… …f minutes…)

More details in the paper!

Tune headroom dial to drive routing as close as possible to optimal solution while avoiding congestion

- predict how aggregates will statistically multiplex on a path by *convolving* their past demands

Are high-LLPD networks viable?

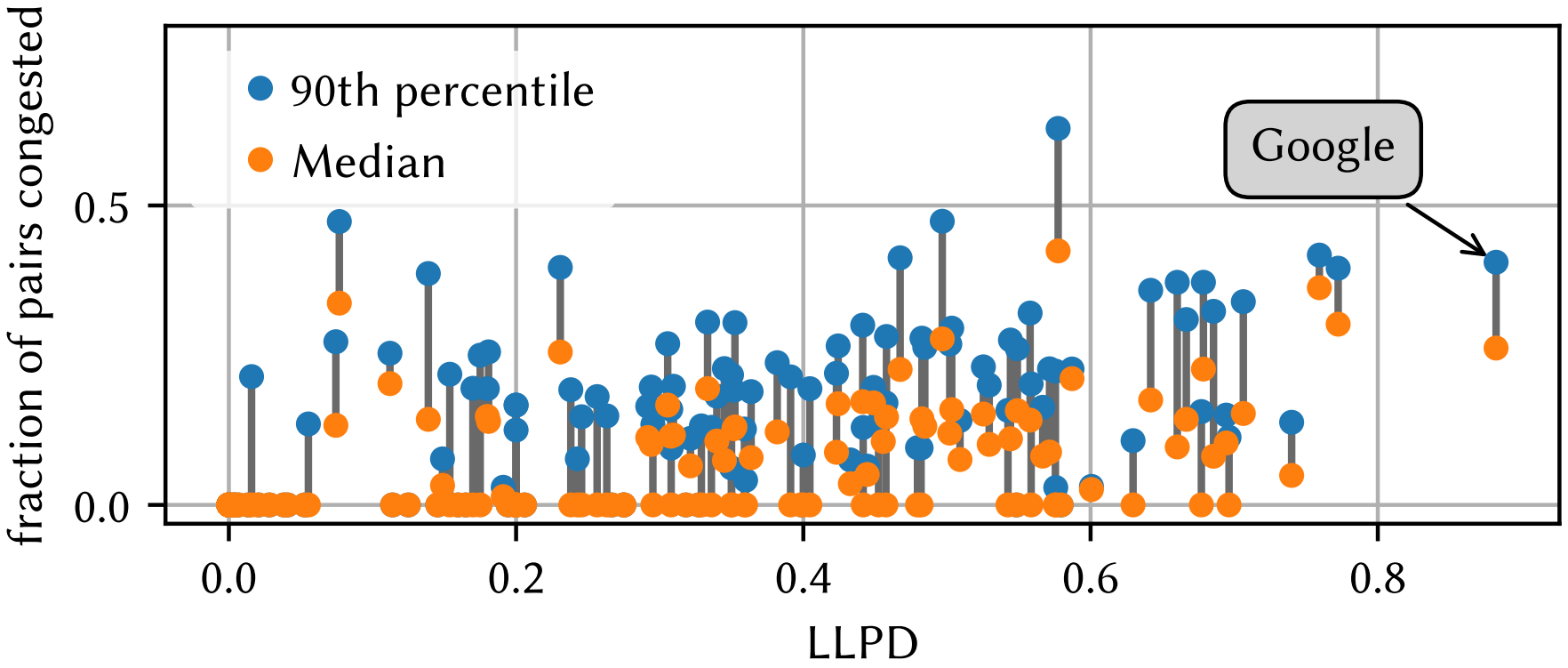
- A routing system may not be able to unlock the low-latency potential of a topology
- LLPD indicates that a topology has good potential for low latency

Are high-LLPD networks viable?

- A routing system may not be able to unlock the low-latency potential of a topology
- LLPD indicates that a topology has good potential for low latency
- But will anyone ever really build a modern WAN with high LLPD?

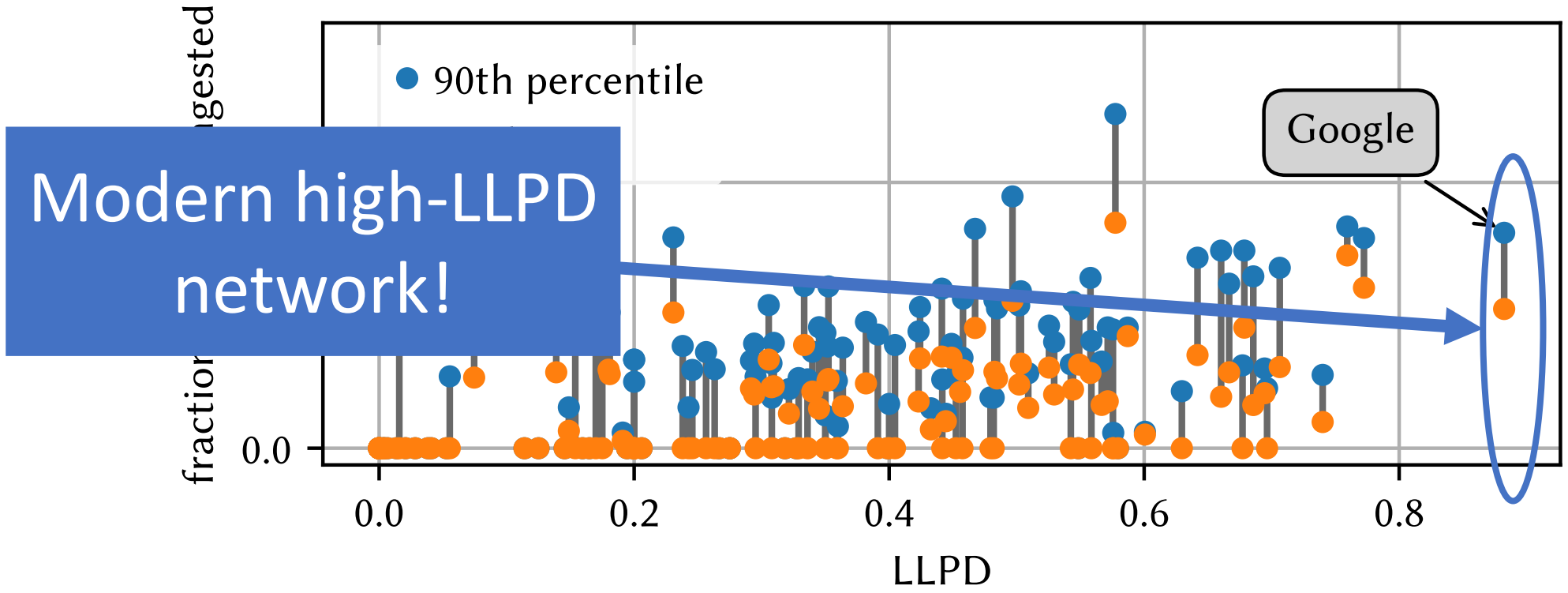
Are high-LLPD networks viable?

- Repeated SP experiment, but added Google's network



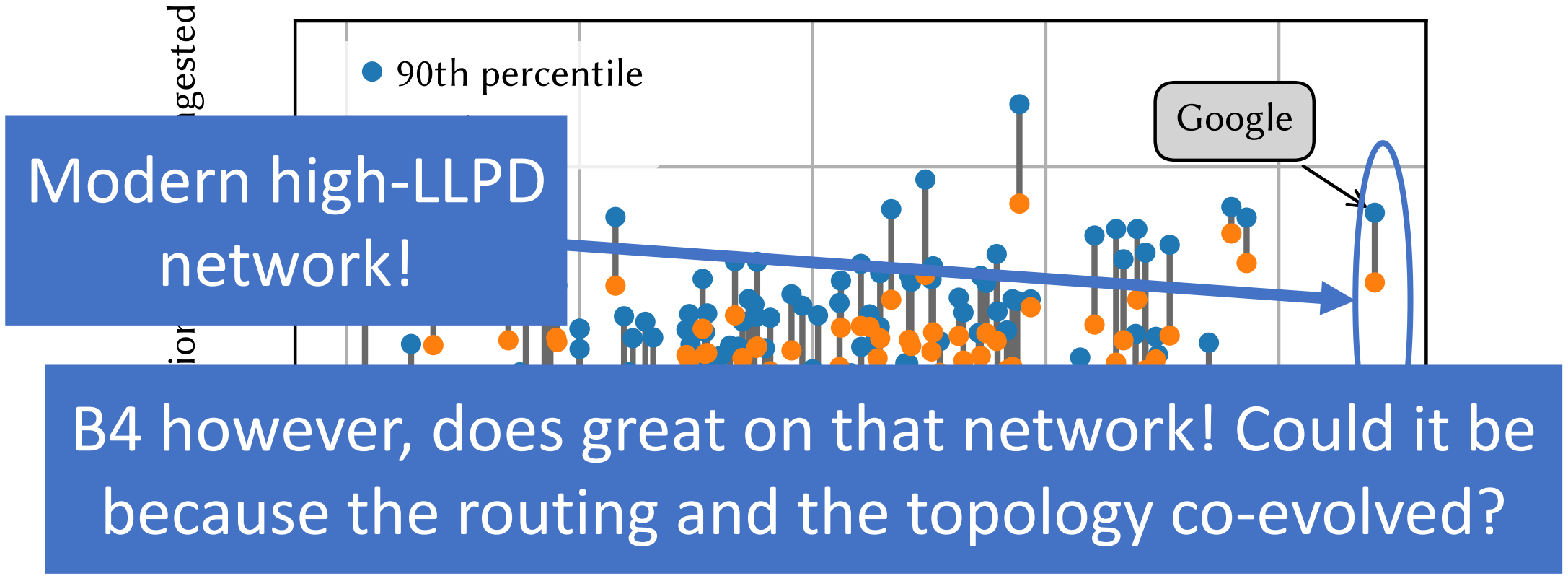
Are high-LLPD networks viable?

- Repeated SP experiment, but added Google's network



Are high-LLPD networks viable?

- Repeated SP experiment, but added Google's network



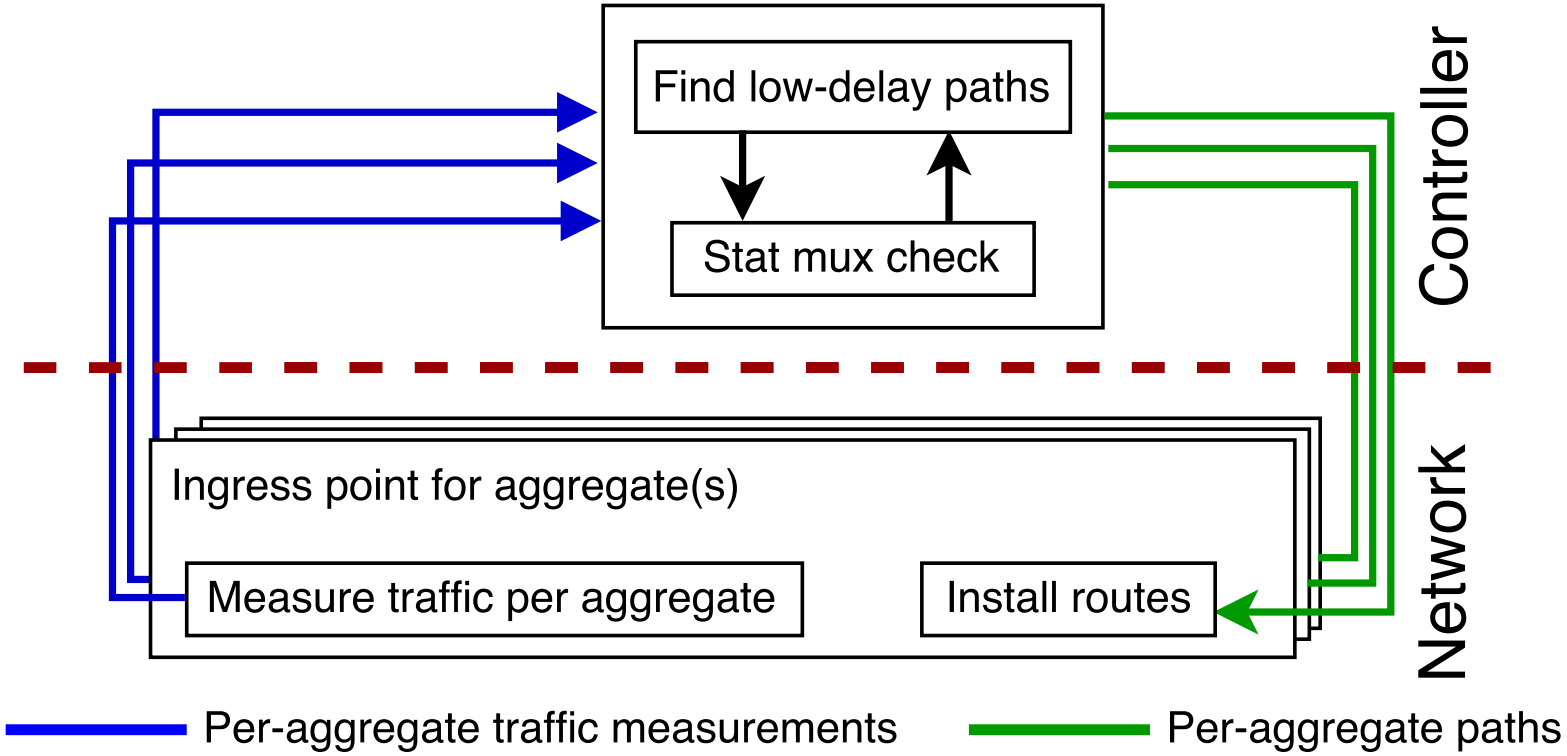
What topologies would people build if they knew the routing system would always extract the best from it?

Conclusions

- To achieve low latency:
  - topology must provide low-latency paths
  - the routing system must use them effectively
- State-of-the-art routing falters on high-LLPD topologies—precisely those with best potential for low latency
- Practical routing approach for high-LLPD topologies:
  - Efficient LP solution for optimal traffic placement
  - Tune headroom dial to avoid congestion (but as little toward MinMax as possible)

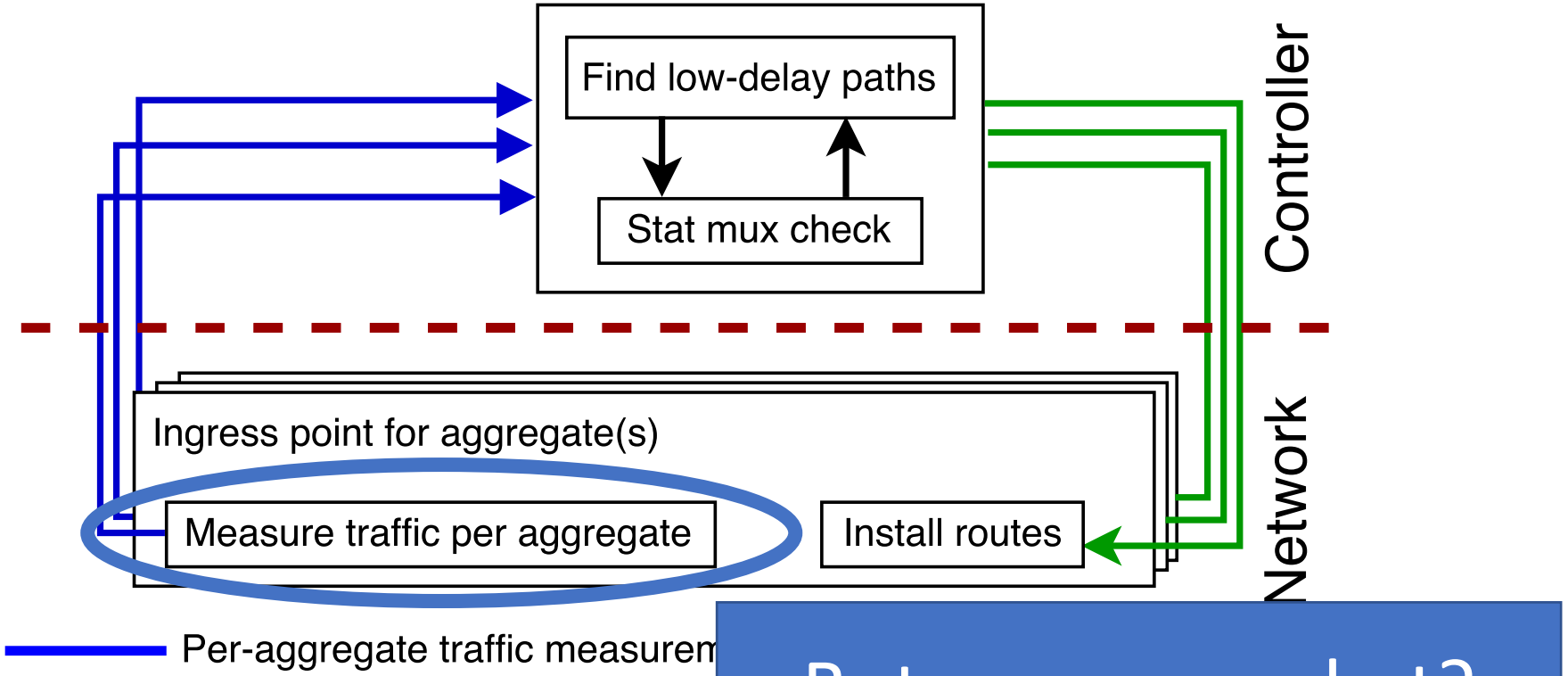
System Design

- Simple, centralized design



System Design

- Simple, centralized design



Measurements

- Only need measurements per aggregate, not per flow!
- Need to know enough to figure out both long and short-term variability for each aggregate
  - Sampling traffic level 10 times per second is enough to capture short-term variability due to TCP's congestion control…
  - …since RTTs in the ISP are long (order of 100ms)
  - Sampling 10 times per second well within reach of recent hardware [DevoFlow SIGCOMM 2011]

What about prioritization?

- If you can you should definitely prioritize delay-sensitive traffic
  - but identifying this traffic in the ISP setting may not be trivial, since no single operator controls all sources
  - also, what about bandwidth-hungry low-latency traffic (e.g., VR)