A New Way of Traffic Engineering in Named Data Networking

Klaus Schneider[1], Beichuan Zhang[1], Lotfi Benmohamed[2]

September 20, 2018

[1]The University of Arizona, [2]NIST

"A New Way of Traffic Engineering **using** NDN"

What is Traffic Engineering?

Any mechanism that manipulates the traffic flow (other than shortest path routing)

What is Traffic Engineering?

Any mechanism that manipulates the traffic flow (other than shortest path routing)

Aka “optimal” routing

Why do we need Traffic Engineering?

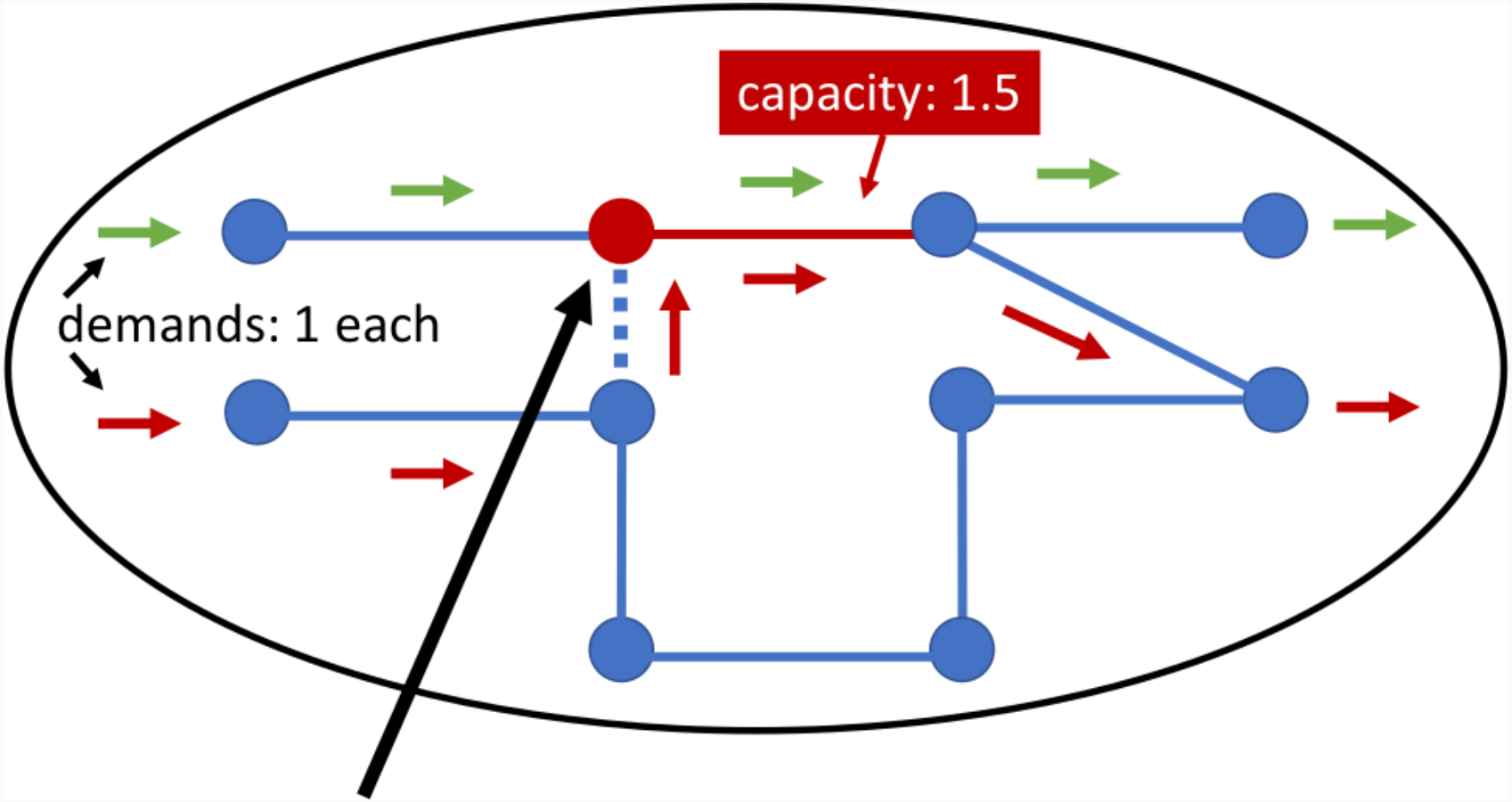


Figure from "On low-latency-capable topologies" [1]

Why do we need Traffic Engineering?

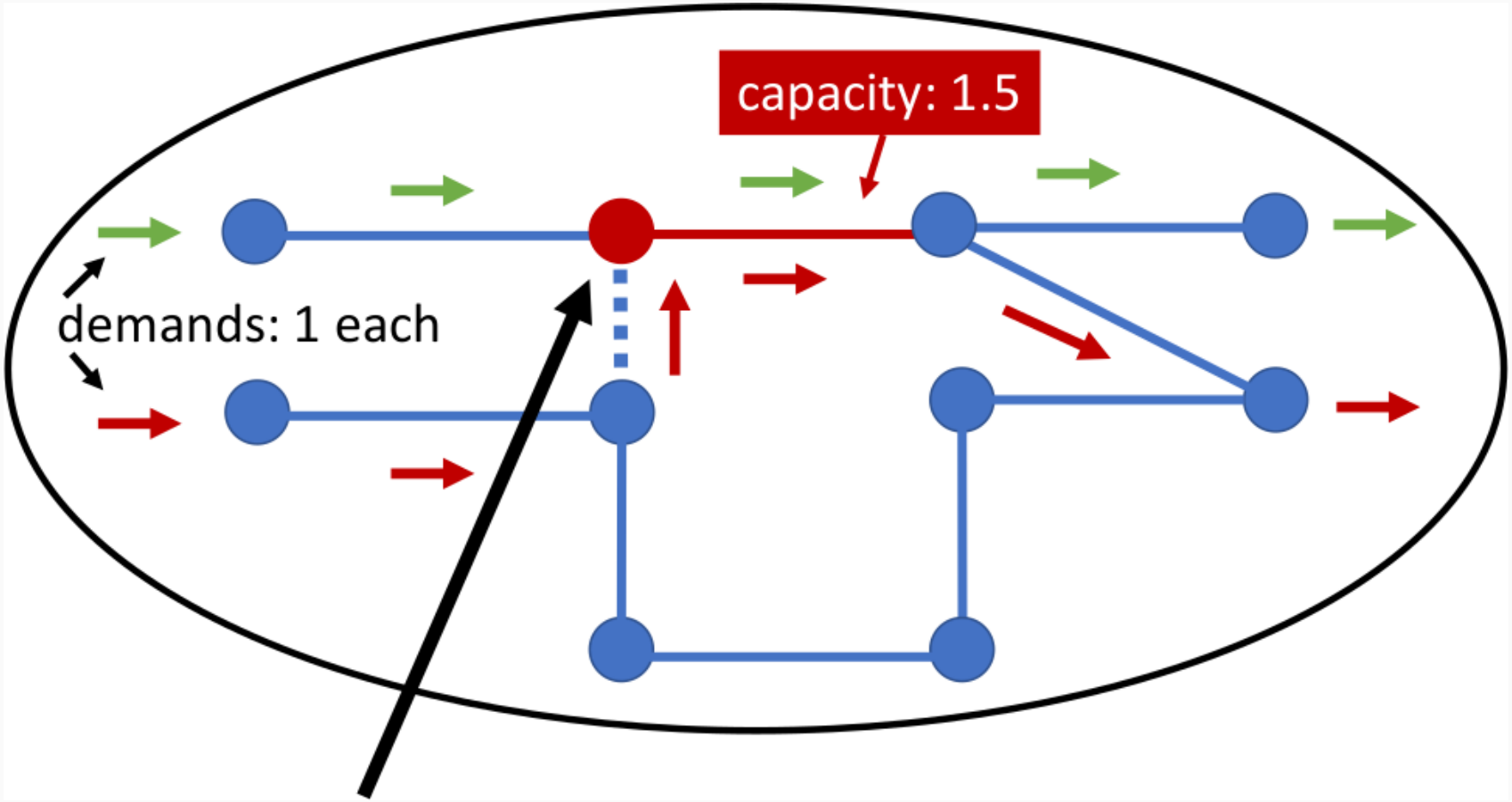


Figure from "On low-latency-capable topologies" [1]

SP Routing: Congestion + unused BW on non-SP

Why do we need Traffic Engineering?

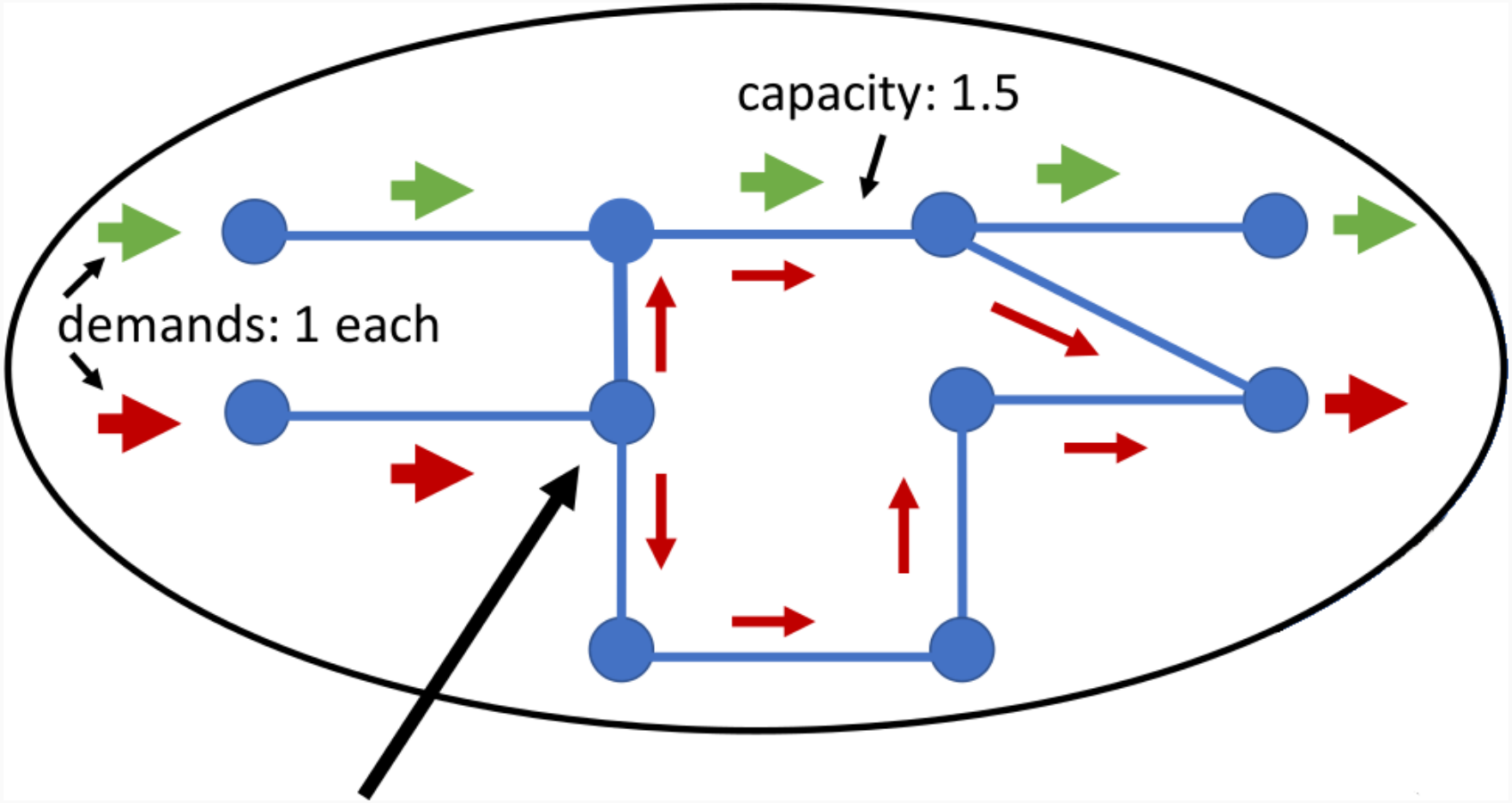


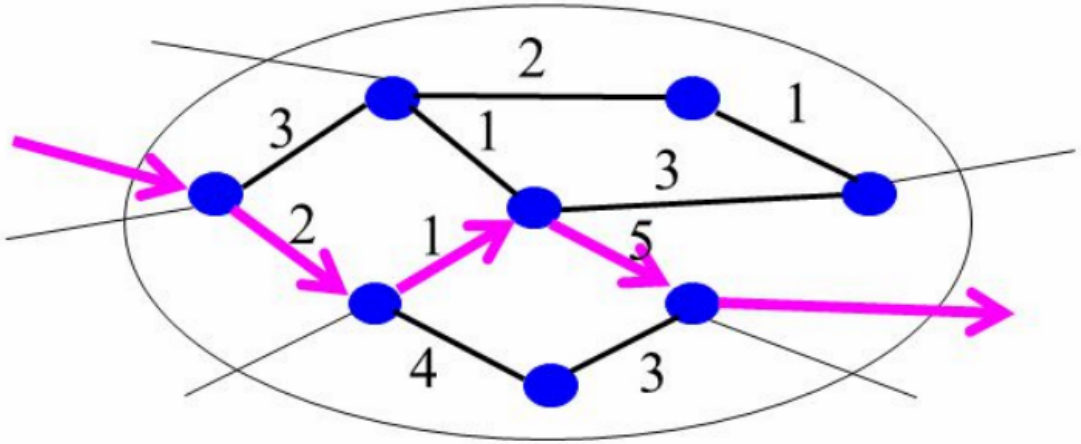
Figure from "On low-latency-capable topologies" [1]

$\Rightarrow$ **Split traffic to non-shortest path!**

The Current Way of Traffic Engineering

1. SP Routing + Link-weight Tuning

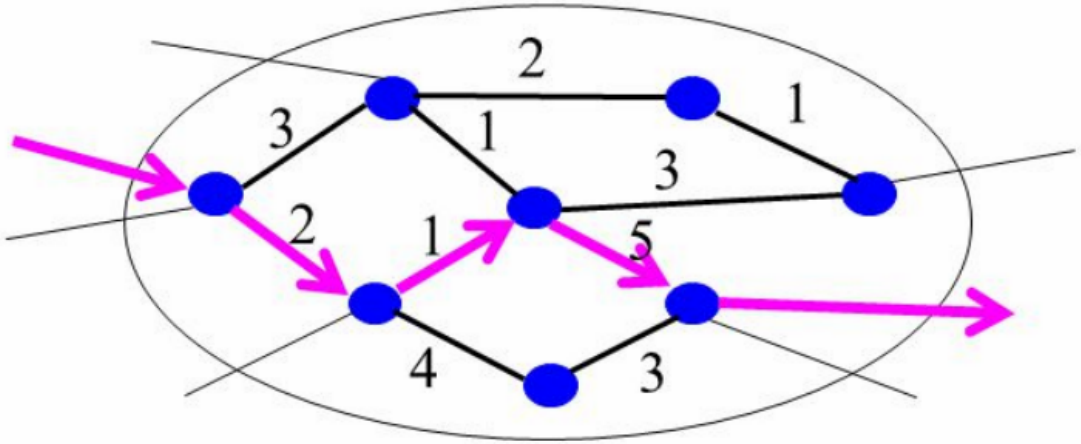
- Weights configured by the AS's network operator
  - Simple heuristics: link capacity or physical distance
  - Traffic engineering: tuning the link weights to the traffic



Jennifer Rexford – "MIRED: Managing IP Routing is Extremely Difficult"

1. SP Routing + Link-weight Tuning

- Weights configured by the AS's network operator
  - Simple heuristics: link capacity or physical distance
  - Traffic engineering: tuning the link weights to the traffic



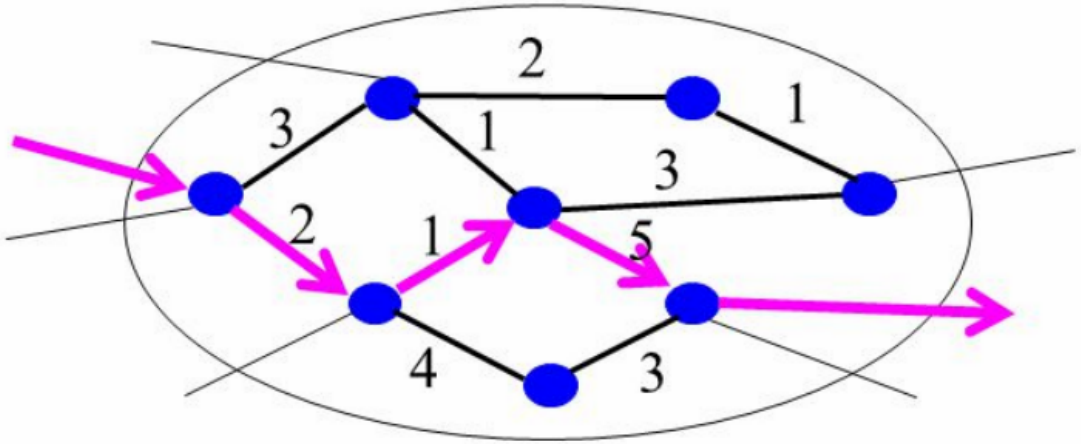
Jennifer Rexford – "MIRED: Managing IP Routing is Extremely Difficult"

Problems:

1. Quite imprecise tool

1. SP Routing + Link-weight Tuning

- Weights configured by the AS's network operator
  - Simple heuristics: link capacity or physical distance
  - Traffic engineering: tuning the link weights to the traffic



Jennifer Rexford – "MIRED: Managing IP Routing is Extremely Difficult"

Problems:

1. Quite imprecise tool
2. **Global Side-effects** (changing weights can cause cong. in other network areas)

2. MinMax Routing (e.g. MATE, TeXCP)

- Input: Traffic Matrix (flows between endpoints)
- Output: Routing that minimizes the Max. Link Utilization

2. MinMax Routing (e.g. MATE, TeXCP)

- Input: Traffic Matrix (flows between endpoints)
- Output: Routing that minimizes the Max. Link Utilization

$\Rightarrow$ Allows headroom to deal with traffic fluctuations

2. MinMax Routing (e.g. MATE, TeXCP)

- Input: Traffic Matrix (flows between endpoints)
- Output: Routing that minimizes the Max. Link Utilization

$\Rightarrow$ Allows headroom to deal with traffic fluctuations

Problems:

- MLU metric susceptible to outliers (small link capacity)

2. MinMax Routing (e.g. MATE, TeXCP)

- Input: Traffic Matrix (flows between endpoints)
- Output: Routing that minimizes the Max. Link Utilization

$\Rightarrow$ Allows headroom to deal with traffic fluctuations

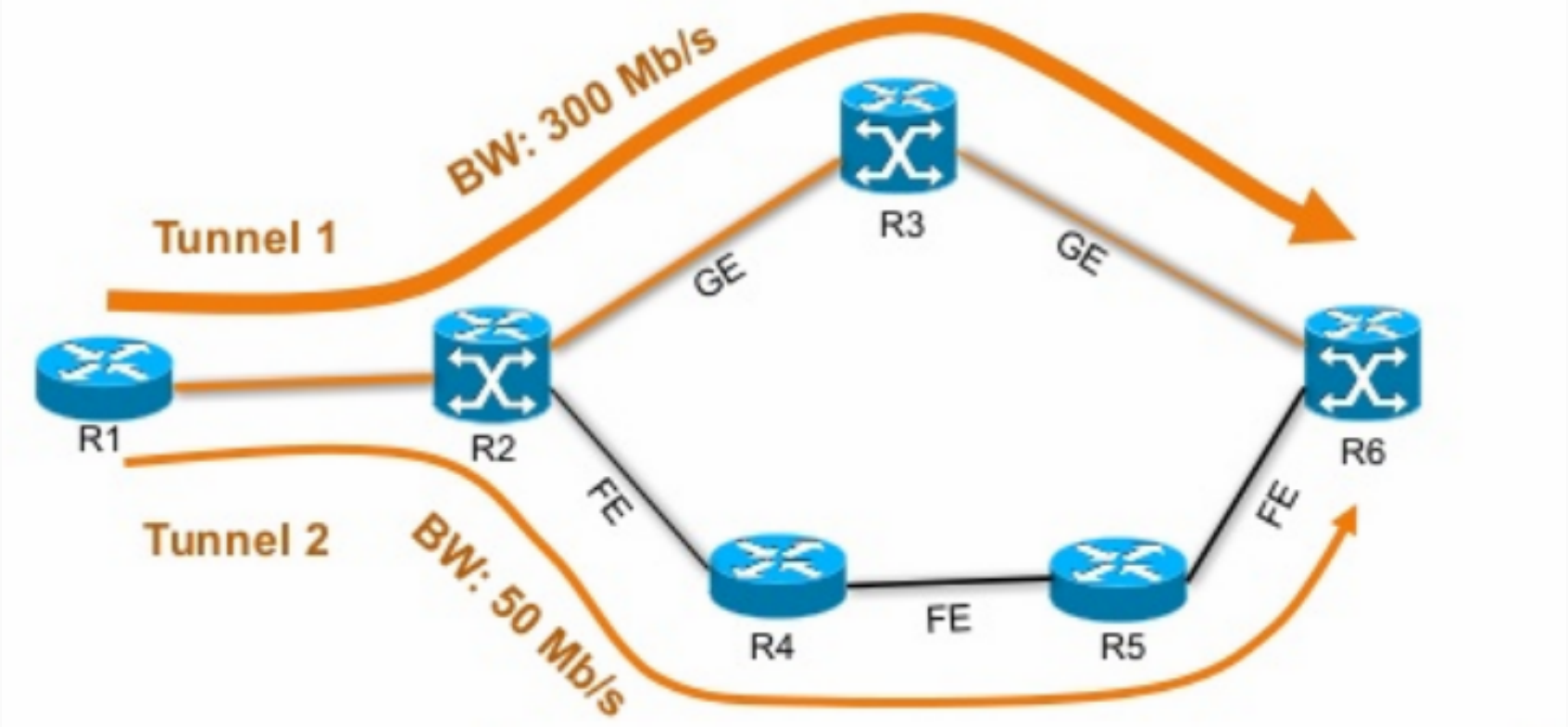
Problems:

- MLU metric susceptible to outliers (small link capacity)
- Doesn't consider link propagation delay!

$\Rightarrow$ Unnecessarily long paths

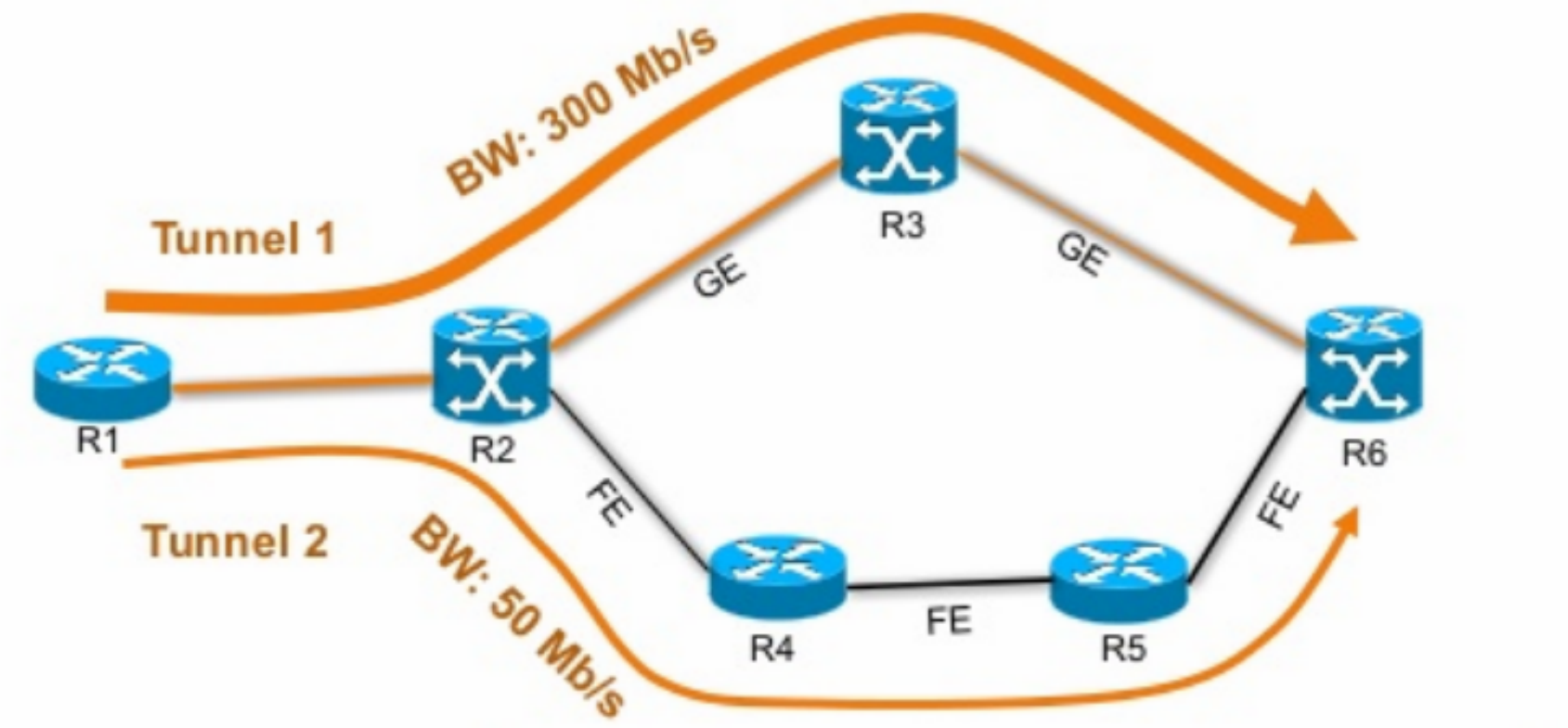
3. End-to-End Tunnels

MPLS, **RSVP-TE**, **Segment Routing**, **B4** [2], **SWAN** [3]



3. End-to-End Tunnels

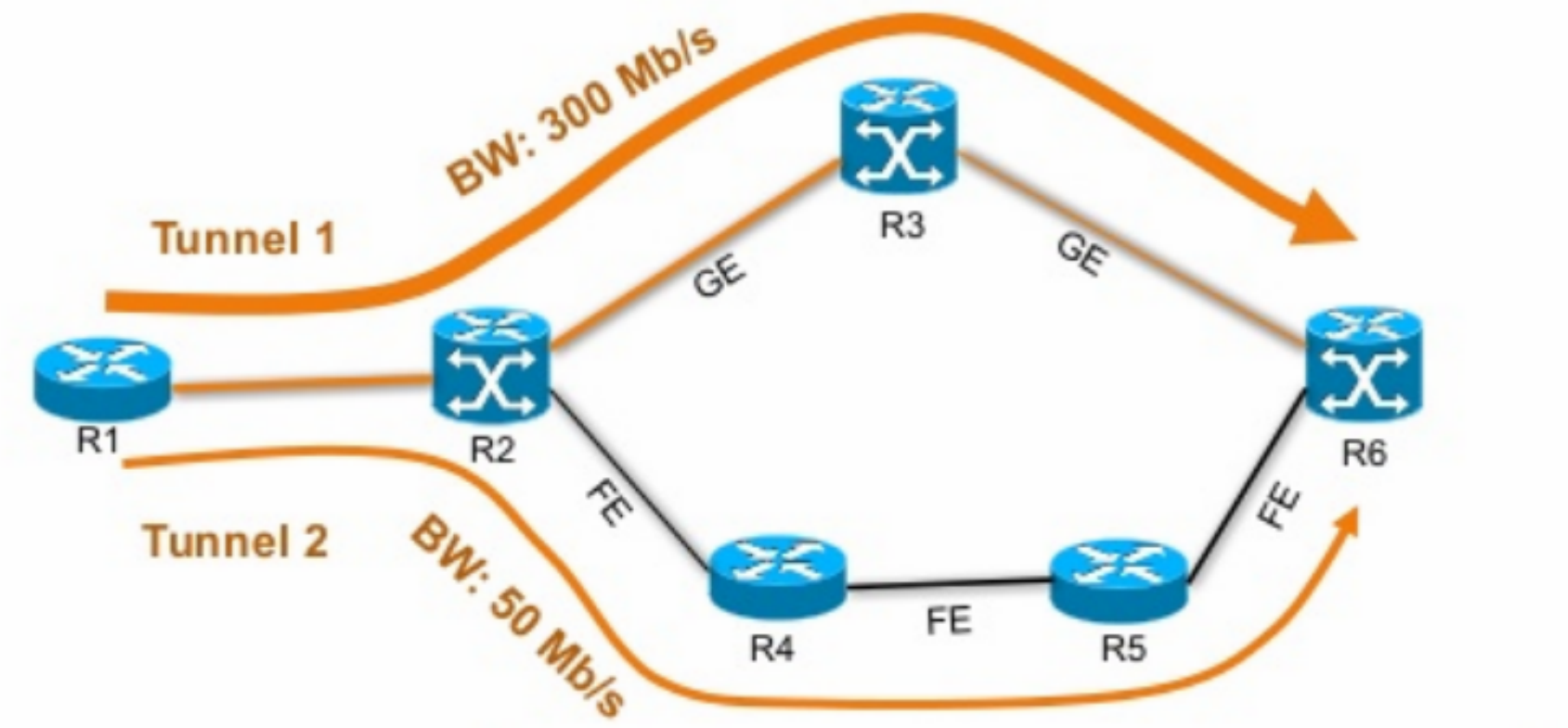
MPLS, **RSVP-TE**, **Segment Routing**, **B4** [2], **SWAN** [3]



1. New flow: Find path that satisfies BW req. (CSPF)

3. End-to-End Tunnels

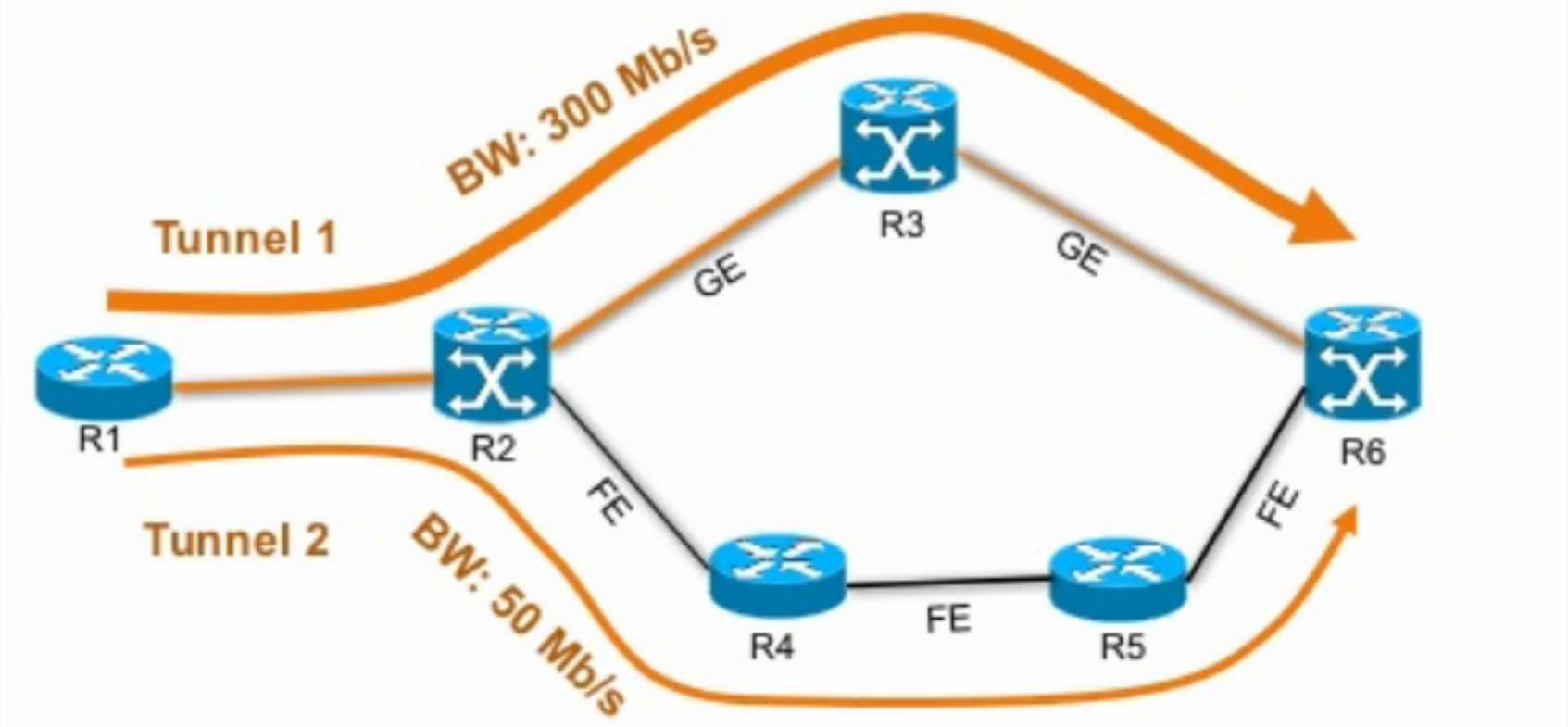
MPLS, **RSVP-TE**, **Segment Routing**, **B4** [2], **SWAN** [3]



1. New flow: Find path that satisfies BW req. (CSPF)
2. Reserve Bandwidth

3. End-to-End Tunnels

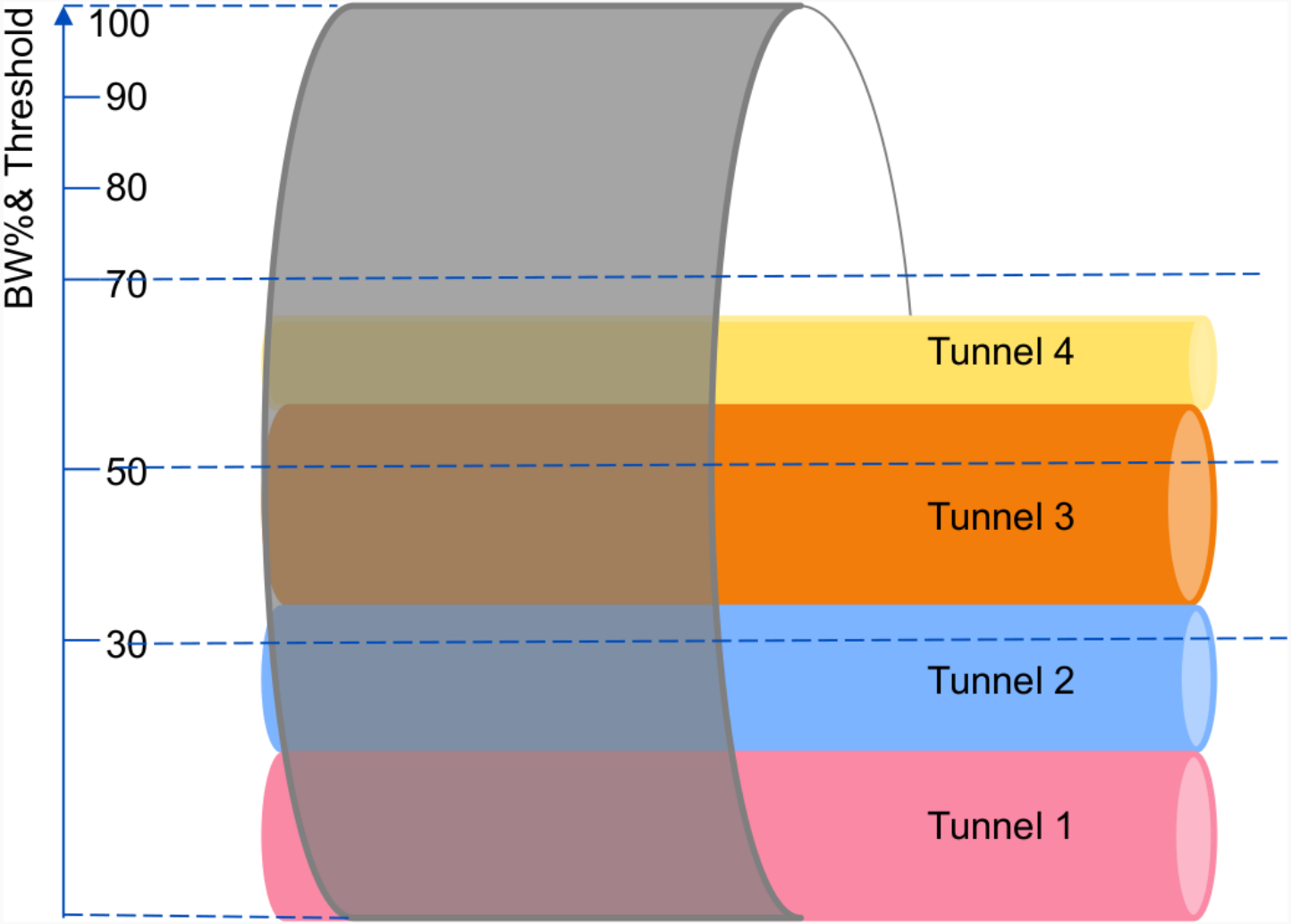
MPLS, **RSVP-TE**, **Segment Routing**, **B4** [2], **SWAN** [3]



1. New flow: Find path that satisfies BW req. (CSPF)
2. Reserve Bandwidth
3. Periodically re-evaluate BW assignments (AutoBW)

Problems with End-to-End Tunnels

1. **Granularity: Large tunnels don't fit into small pipes**



Problems with End-to-End Tunnels

2. **Manual setup of LSPs** (number & which ones)

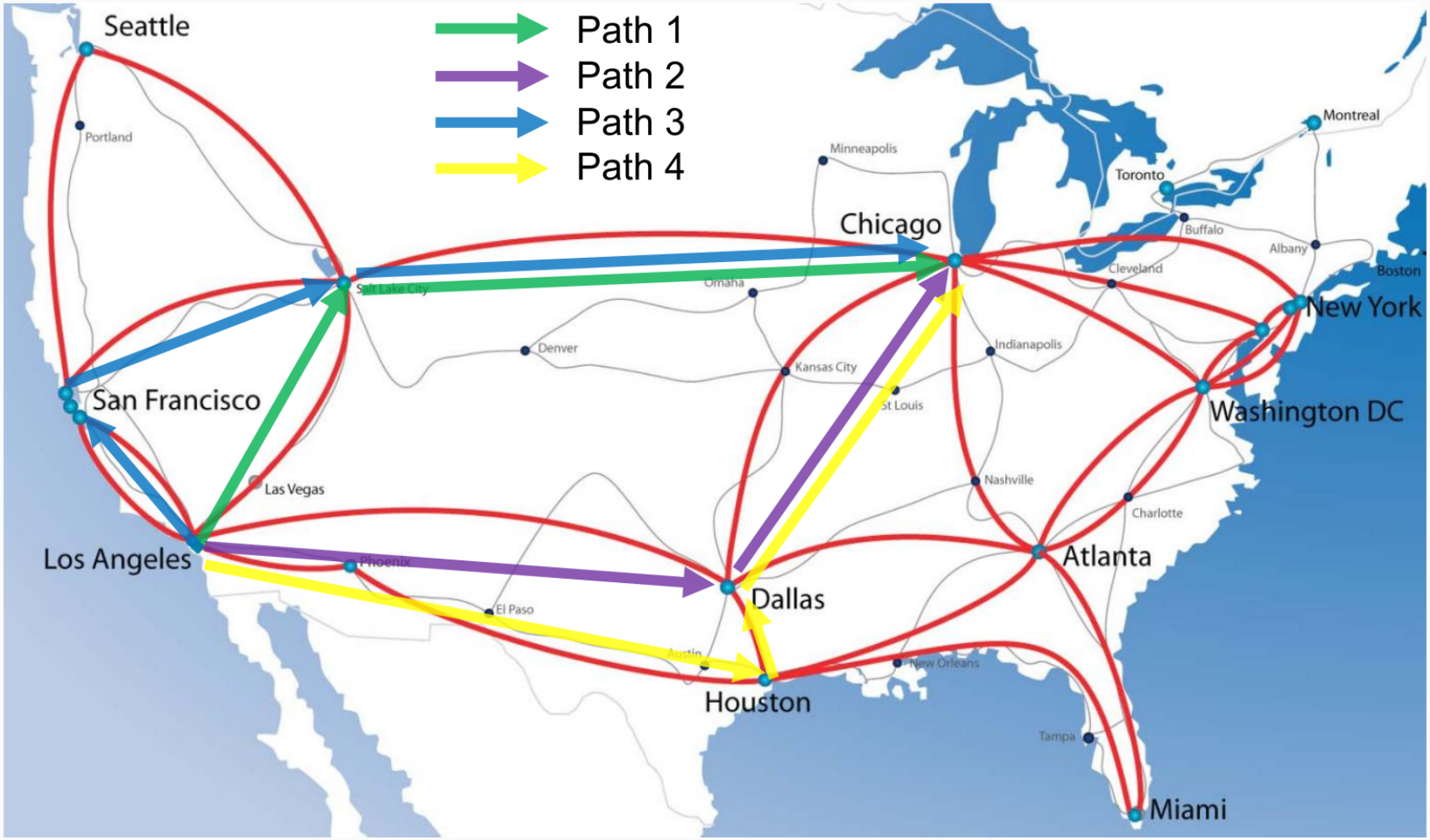


Figure from "MPLS RSVP-TE Auto-Bandwidth – Lessons Learned" [4]

Problems with End-to-End Tunnels

3. Fluctuating link utilization & slow adjustment

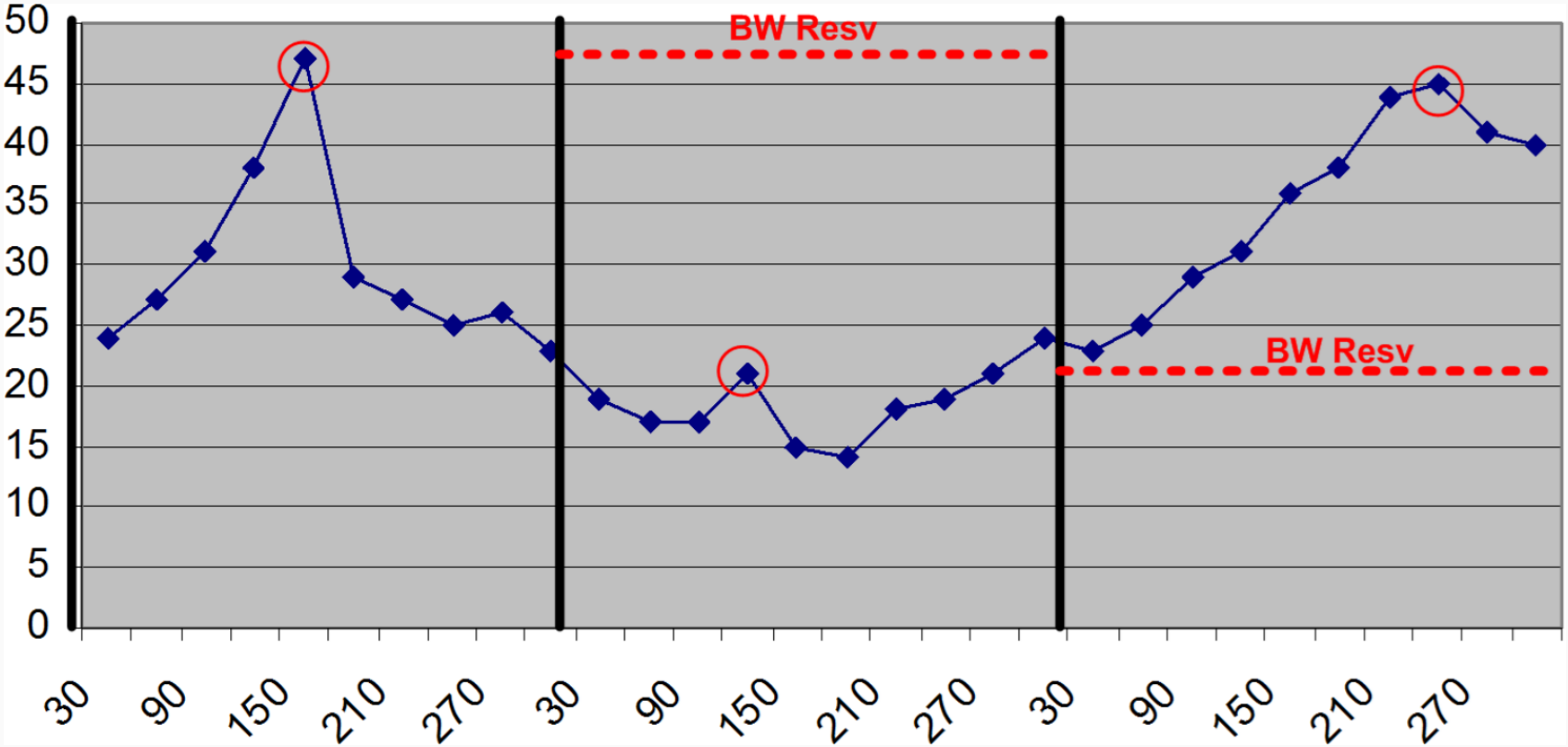


Figure from "MPLS RSVP-TE Auto-Bandwidth – Lessons Learned" [4]

$\Rightarrow$ **Risks underutilization or congestion!**

Problems with End-to-End Tunnels

4. AutoBW doesn't see actual congestion (packet loss)

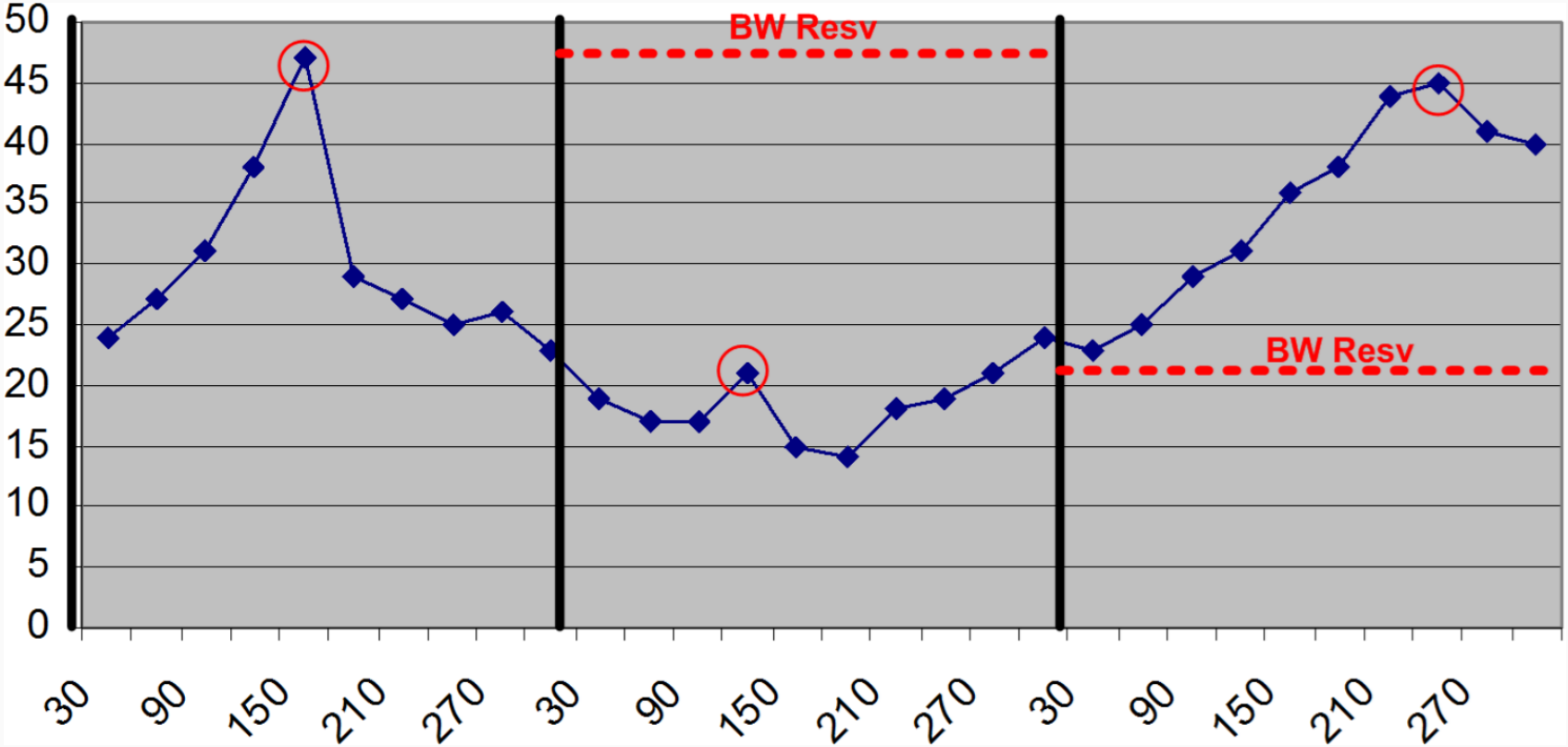


Figure from "MPLS RSVP-TE Auto-Bandwidth – Lessons Learned" [4]

$\Rightarrow$ Endpoints slow down without AutoBW noticing

Problems with End-to-End Tunnels

All in all: MPLS-TE quite complex approach

$\Rightarrow$ Lots of manual work; reliance on operator and/or proprietary software

+ Routing & Congestion Control are separated

A New Way of Traffic Engineering

A New Way of Traffic Engineering

Goals: Combine MP Routing & Congestion Control

1. **Simplify Configuration**
    - No manual configuration
    - No pre-established tunnels

A New Way of Traffic Engineering

Goals: Combine MP Routing & Congestion Control

1. **Simplify Configuration**
    - No manual configuration
    - No pre-established tunnels
2. **Improve Performance**
    - Use network resources more efficiently

A New Way of Traffic Engineering

Goals: Combine MP Routing & Congestion Control

1. **Simplify Configuration**
    - No manual configuration
    - No pre-established tunnels
2. **Improve Performance**
    - Use network resources more efficiently
    - React quickly to changing network conditions

A New Way of Traffic Engineering

Goals: Combine MP Routing & Congestion Control

1. **Simplify Configuration**
   - No manual configuration
   - No pre-established tunnels
2. **Improve Performance**
   - Use network resources more efficiently
   - React quickly to changing network conditions
   - Consider both path cost & congestion:

   $\Rightarrow$ Trade-off between shorter and less congested paths

A New Way of Traffic Engineering

Goals: Combine MP Routing & Congestion Control

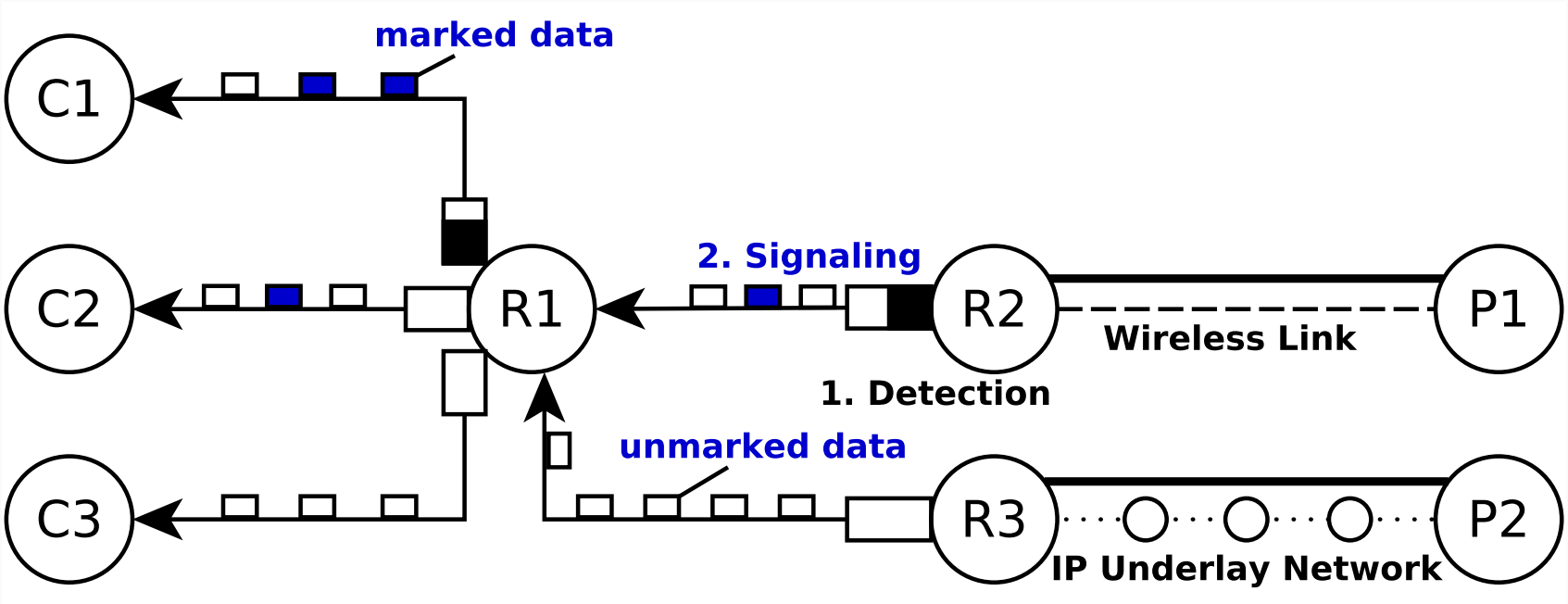
1. **Simplify Configuration**
   - No manual configuration
   - No pre-established tunnels
2. **Improve Performance**
   - Use network resources more efficiently
   - React quickly to changing network conditions
   - Consider both path cost & congestion:

   $\Rightarrow$ Trade-off between shorter and less congested paths

$+$ Support all NDN features (e.g. caching & multi-producer)

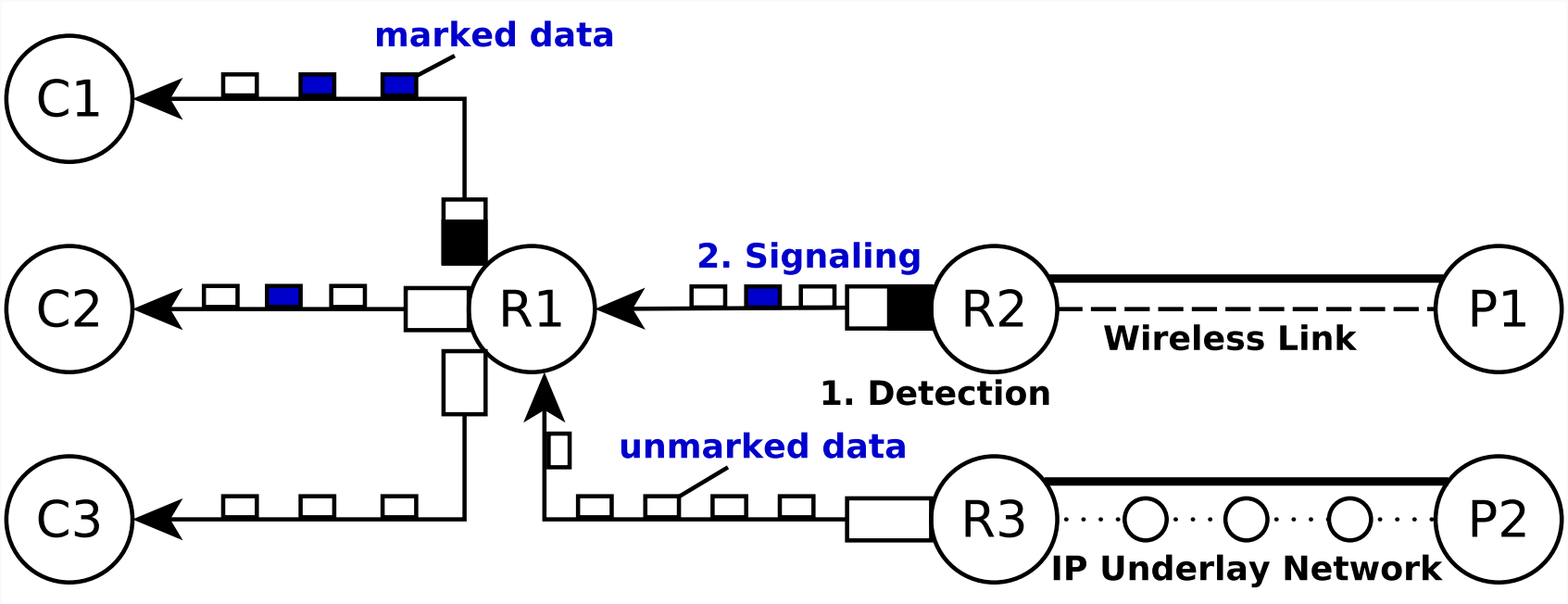
Design Principles: Explicit Congestion Notification

BW Estimation $\Rightarrow$ Congestion marks



Design Principles: Explicit Congestion Notification

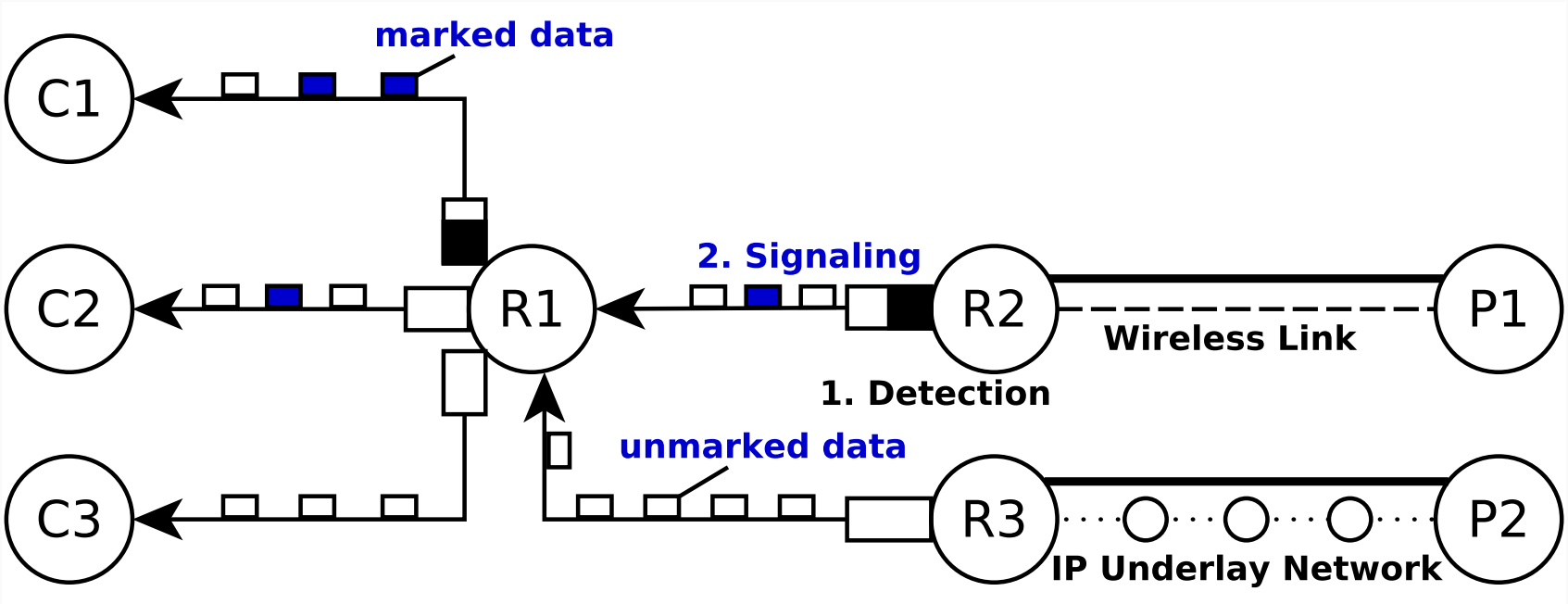
BW Estimation $\Rightarrow$ Congestion marks



- Works on **wireless links,** without BW estimation

Design Principles: Explicit Congestion Notification

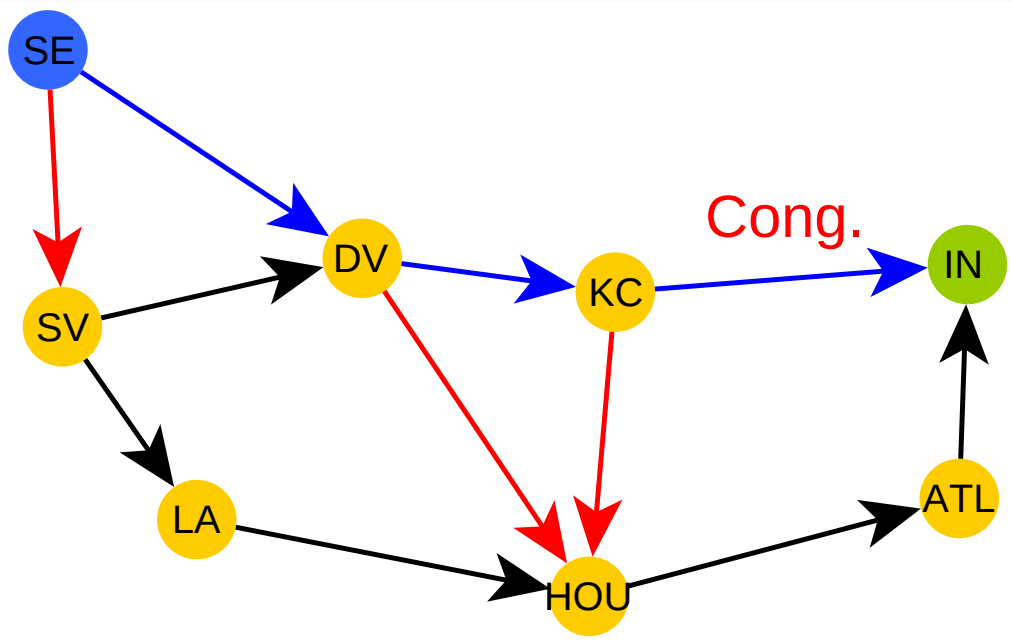
BW Estimation $\Rightarrow$ Congestion marks



- Works on **wireless links,** without BW estimation
- Possible to leave headroom, by signaling congestion early (e.g. virtual queue)

Design Principles: Independent HBH Decision

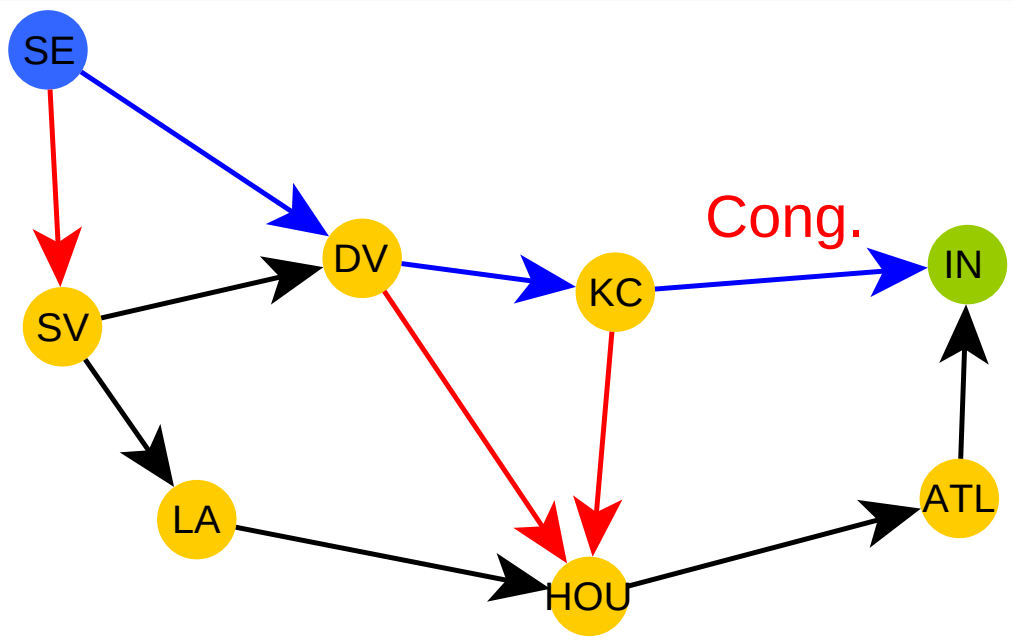
E2E Tunnels $\Rightarrow$ In-network nexthop choices



MP Routing: Routers have many NH that won't cause loops[5]

Design Principles: Independent HBH Decision

E2E Tunnels ⇒ In-network nexthop choices



MP Routing: Routers have many NH that won't cause loops[5]

⇒ Exponential # of possible paths, without any path establishment overhead

Design Principles: Use Path Cost & Congestion

Topology and path cost known through routing protocol.

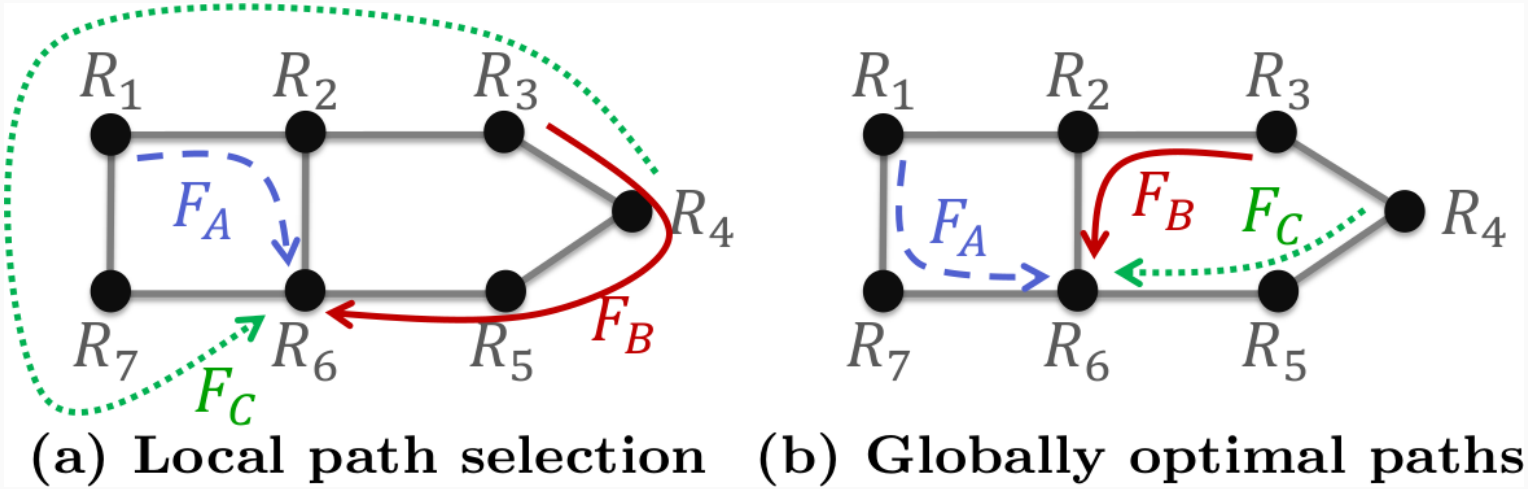
1. Start at shortest path
2. Split to longer paths when necessary (cong.)

Design Principles: Use Path Cost & Congestion

Topology and path cost known through routing protocol.

1. Start at shortest path
2. Split to longer paths when necessary (cong.)

Cost-awareness + fine-grained split avoids some MPLS issues:



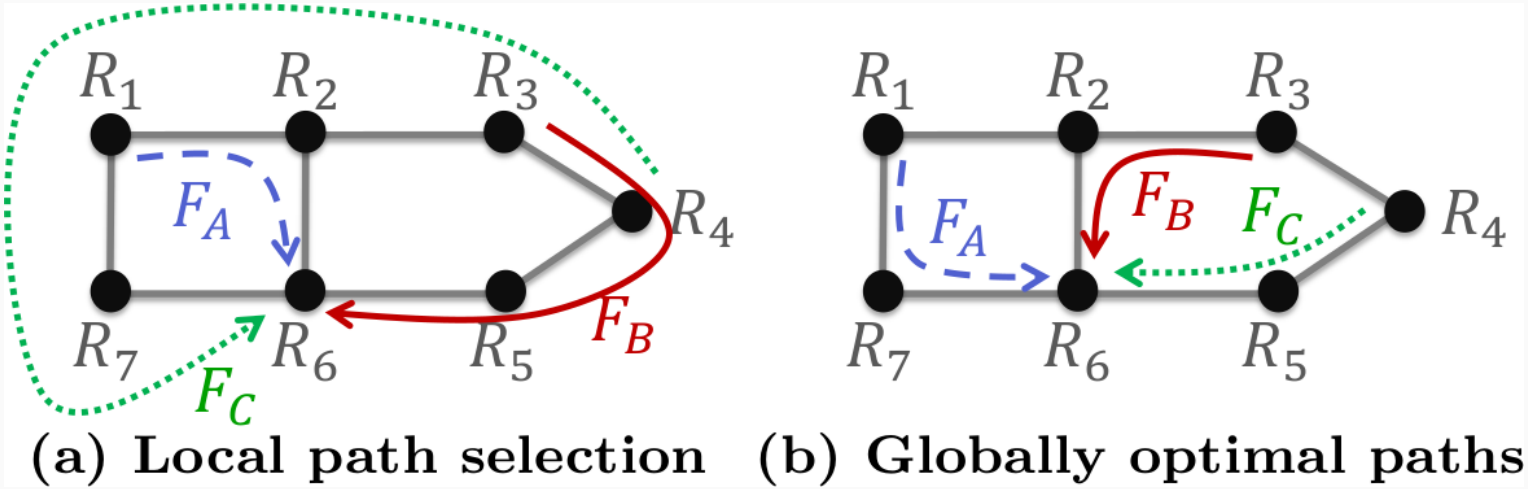
"Figure 2: Inefficient routing due to local allocation." from [3]

Design Principles: Use Path Cost & Congestion

Topology and path cost known through routing protocol.

1. Start at shortest path
2. Split to longer paths when necessary (cong.)

Cost-awareness + fine-grained split avoids some MPLS issues:

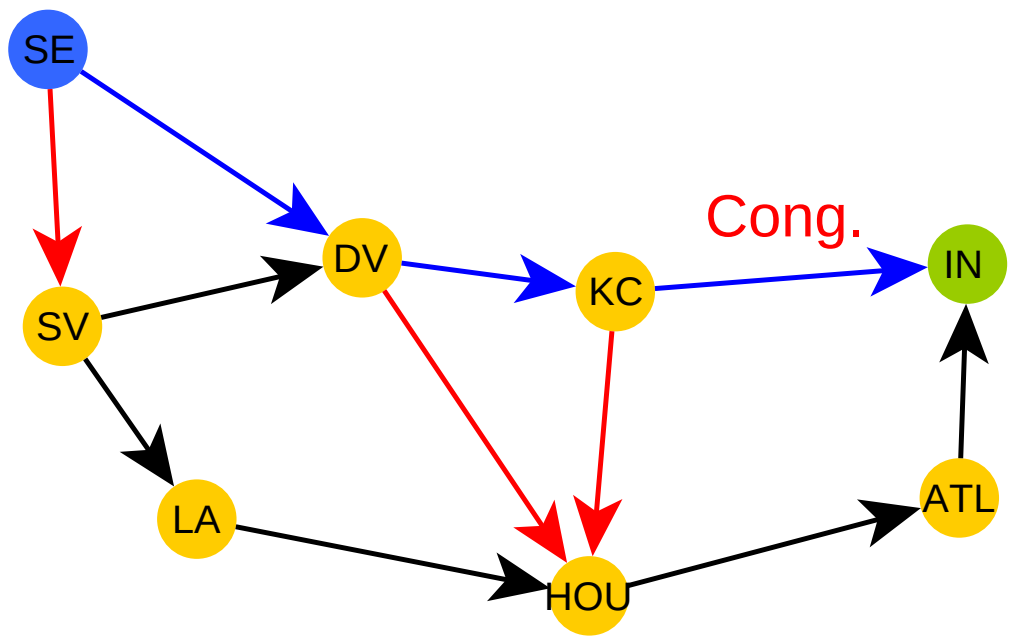


"Figure 2: Inefficient routing due to local allocation." from [3]

Problem is MPLS granularity, not local knowledge!

Design Principles: Coordinate who Splits

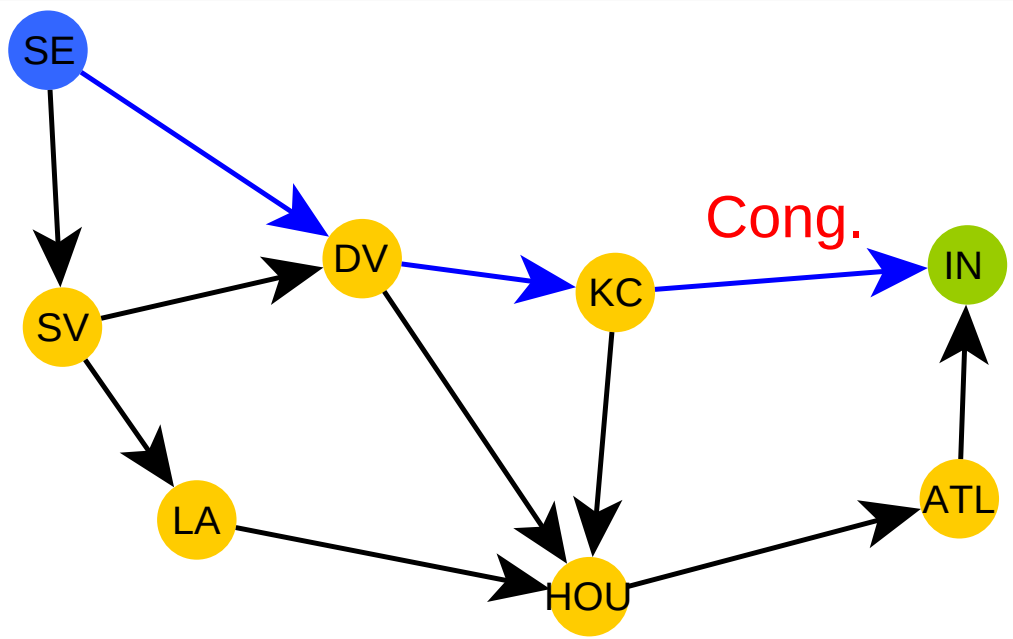
Split only at "best" location:



- Closest to congestion? (KC)
- Lowest Δ in path cost? (DV)

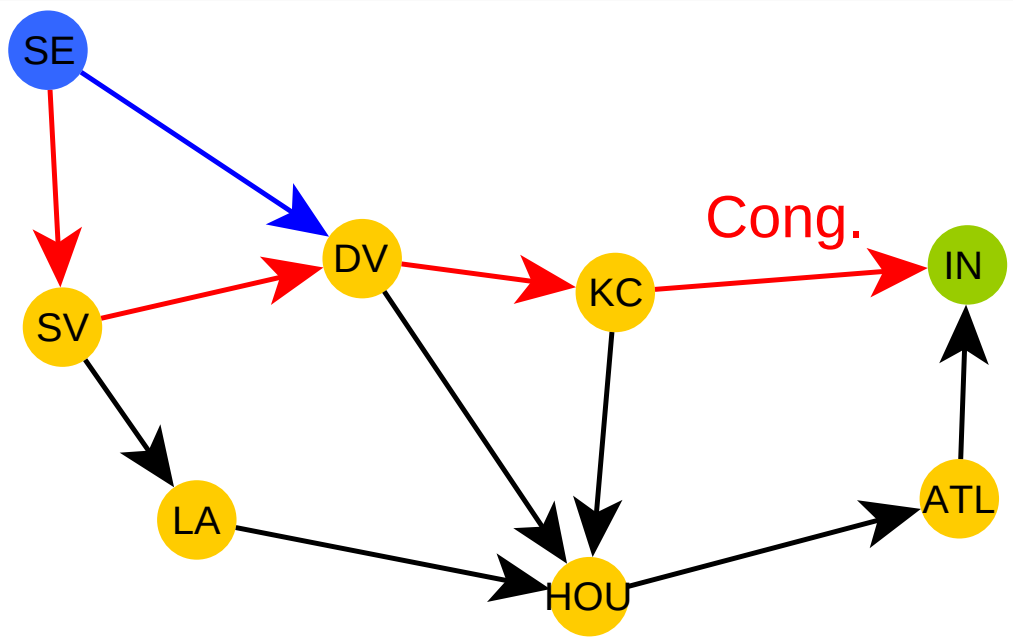
Design Principles: Probe New Paths Before Use

Avoid shared bottlenecks:



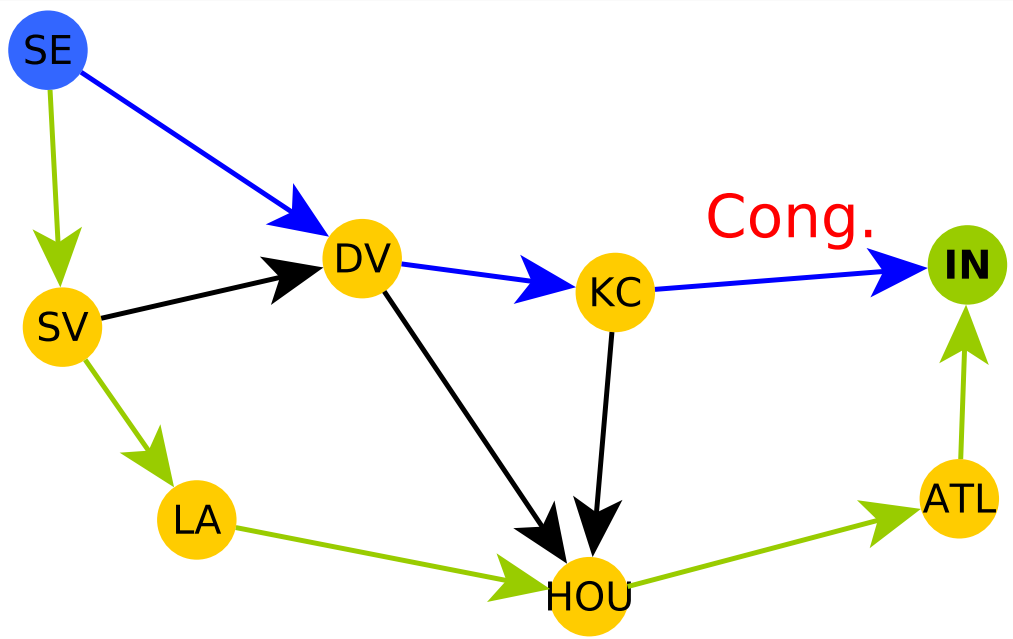
Design Principles: Probe New Paths Before Use

Avoid shared bottlenecks:



Design Principles: Probe New Paths Before Use

Avoid shared bottlenecks:



How does it work exactly?

How does it work exactly?

Paper & Code in roughly 3-6 months :)

Summary

Current Traffic Engineering has **drawbacks** e.g.:

- Granularity, global side-effects
- Ignores propagation delay
- Complex + manual operator work required

Summary

Current Traffic Engineering has **drawbacks** e.g.:

- Granularity, global side-effects
- Ignores propagation delay
- Complex + manual operator work required

$\Rightarrow$ **Use NDN forwarding plane to build better TE**

- Couple HBH Routing & Congestion Control
- Let's see if it works :)

The End

Thank you for your attention!

Klaus Schneider
klaus@cs.arizona.edu

References i

[1] Nikola Gvozdiev, Stefano Vissicchio, Brad Karp, and Mark Handley.
On low-latency-capable topologies, and their impact on the design of intra-domain routing.
In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 88–102. ACM, 2018.

[2] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, et al.
B4: Experience with a globally-deployed software defined wan.
In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 3–14. ACM, 2013.

[3] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer.
Achieving high utilization with software-driven wan.
In *ACM SIGCOMM Computer Communication Review*, volume 43, pages 15–26. ACM, 2013.

[4] Richard Steenbergen.
Mpls rsvp-te auto-bandwidth-lessons learned.
NANOG, 2013.

[5] Klaus Schneider and Beichuan Zhang.
How to establish loop-free multipath routes in named data networking.
Technical report, NDN Technical Report NDN-0044, 2017.

References ii

[6] Atul Khanna and John Zinky.
The revised arpanet routing metric.
ACM SIGCOMM Computer Communication Review, 19(4):45–56, 1989.