Unbiased Experiments in Congested Networks

Bruce Spang Veronica Hannan Shravya Kunamalla Te-Yuan Huang Nick McKeown Ramesh Johari

We use A/B tests to see if an algorithm works in practice

A Buffer-Based Approach to Rate Adaptation: Evidence from a Large Video Streaming Service

Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell*, Mark Watson*
Stanford University, Netflix*
{huangty,rjohari,nickm}@stanford.edu, {mtrunnell,watsonm}@netflix.com

Learning *in situ*: a randomized experiment in video streaming

Francis Y. Yan, Hudson Ayers, Chenzhi Zhu[†], Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, Keith Winstein

Stanford University, [†]Tsinghua University

Staying Alive: Connection Path Reselection at the Edge

Raul Landa, Lorenzo Saino, Lennert Buytenhek and João Taveira Araújo
Fastly

The QUIC Transport Protocol: Design and Internet-Scale Deployment

Adam Langley, Alistair Riddoch, Alyssa Wilk, Antonio Vicente, Charles Krasic, Dan Zhang, Fan Yang, Fedor Kouranov, Ian Swett, Janardhan Iyengar, Jeff Bailey, Jeremy Dorfman, Jim Roskind, Joanna Kulik, Patrik Westin, Raman Tenneti, Robbie Shade, Ryan Hamilton, Victor Vasiliev, Wan-Teh Chang, Zhongyi Shi *
Google
quic-sigcomm@google.com

Proportional Rate Reduction for TCP

Nandita Dukkipati, Matt Mathis, Yuchung Cheng, Monia Ghobadi
Google, Inc.
Mountain View
California, U.S.A
{nanditad, mattmathis, ycheng}@google.com, monia@cs.toronto.edu

BBR: Congestion-Based Congestion Control

Measuring bottleneck bandwidth and round-trip propagation time

Neal Cardwell, Yuchung Cheng, C. Stephen Gunn, Soheil Hassas Yeganeh, Van Jacobson

BBR v2: A Model-based Congestion Control Performance Optimizations

Neal Cardwell, Yuchung Cheng,
Soheil Hassas Yeganeh, Priyaranjan Jha, Yousuk Seung, Kevin Yang,
Ian Swett, Victor Vasiliev, Bin Wu, Luke Hsiao, Matt Mathis
Van Jacobson

Reducing Web Latency: the Virtue of Gentle Aggression

Tobias Flach*, Nandita Dukkipati[†], Andreas Terzis[†], Barath Raghavan[†], Neal Cardwell[†], Yuchung Cheng[†], Ankur Jain[†], Shuai Hao*, Ethan Katz-Bassett*, and Ramesh Govindan*

*Department of Computer Science, University of Southern California
[†]Google Inc.

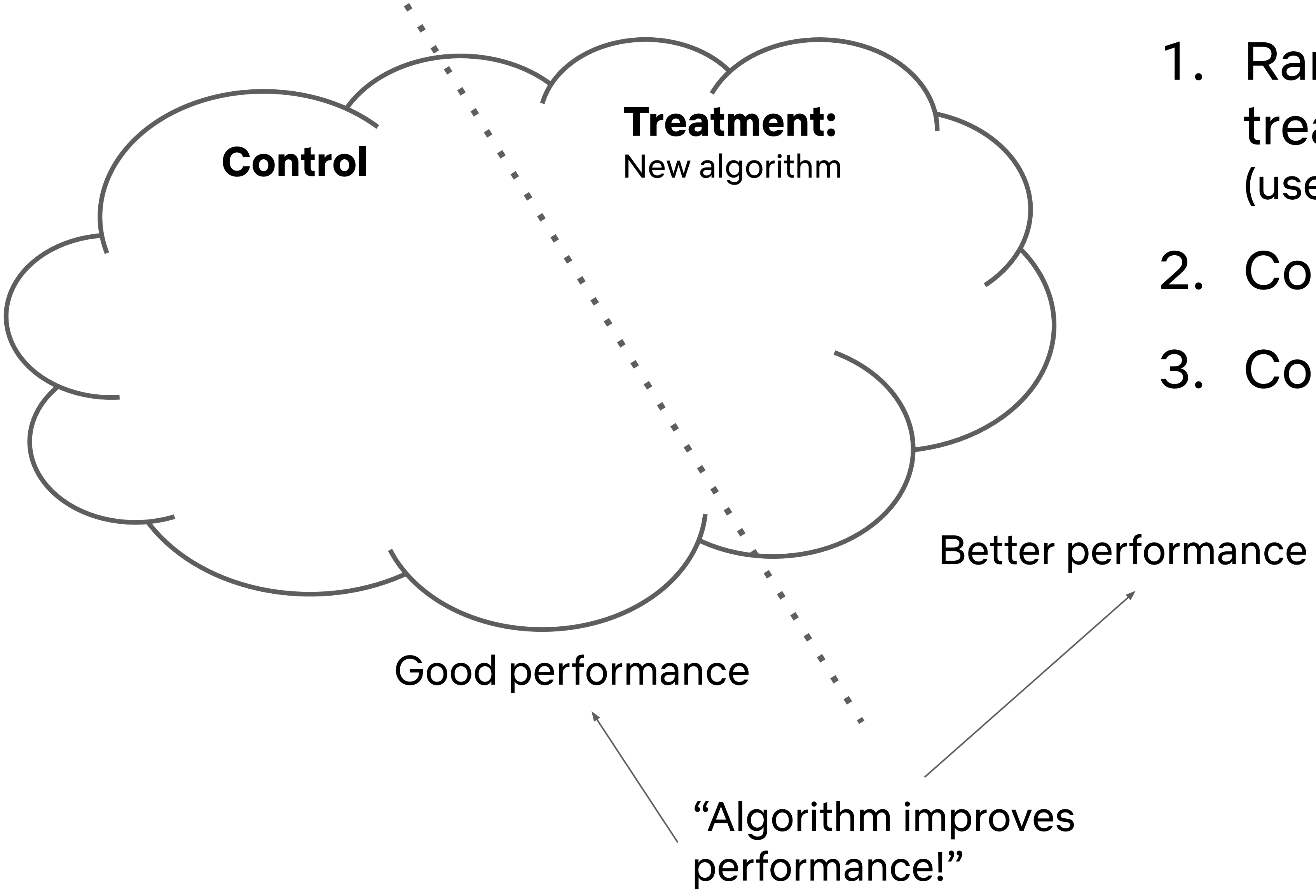
POSTED ON OCTOBER 21, 2020 TO ANDROID, DATA INFRASTRUCTURE, IOS, NETWORKING & TRAFFIC, WEB

How Facebook is bringing QUIC to billions

POSTED ON NOVEMBER 17, 2019 TO NETWORKING & TRAFFIC, VIDEO ENGINEERING

Evaluating COPA congestion control for improved video performance

What is an A/B test?



1. Randomly assign traffic to treatment/control (users, sessions, servers, etc…)
2. Collect data
3. Compare outcomes

A/B tests are used to generalize

We make decisions about deploying algorithms based on small A/B tests:

> "This algorithm improves performance by 10%"

This assumes that the outcome of one unit does not depend on other units

This is called interference

Examples of interference

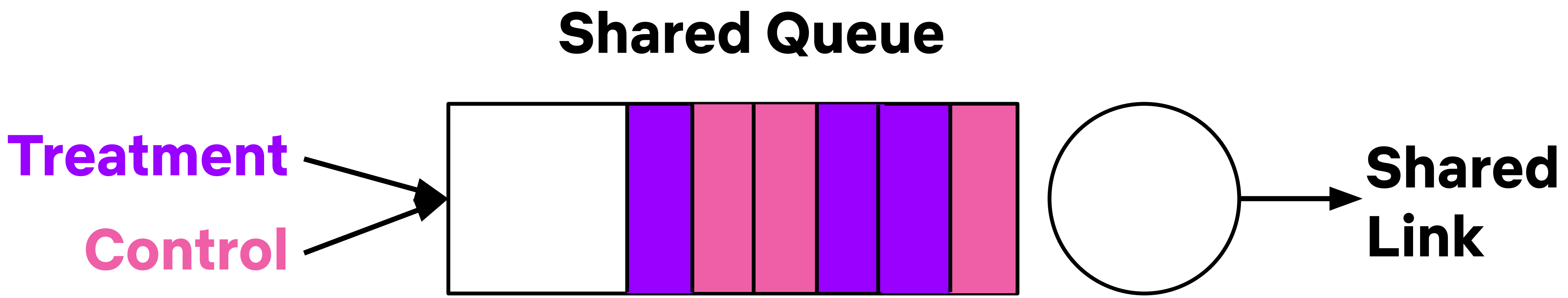
Lots of examples from causal inference

Social networks: a treatment that increases usage might also cause increased usage for friends in the control group.

Online auctions/markets: if treatment/control users bid against each other, making treated users more likely to win means that control users are more likely to lose.

And many more!

Interference exists in congested networks



Interference raises two questions

1. Does it matter?
2. What can we do about it?

Interference can make A/B tests extremely misleading

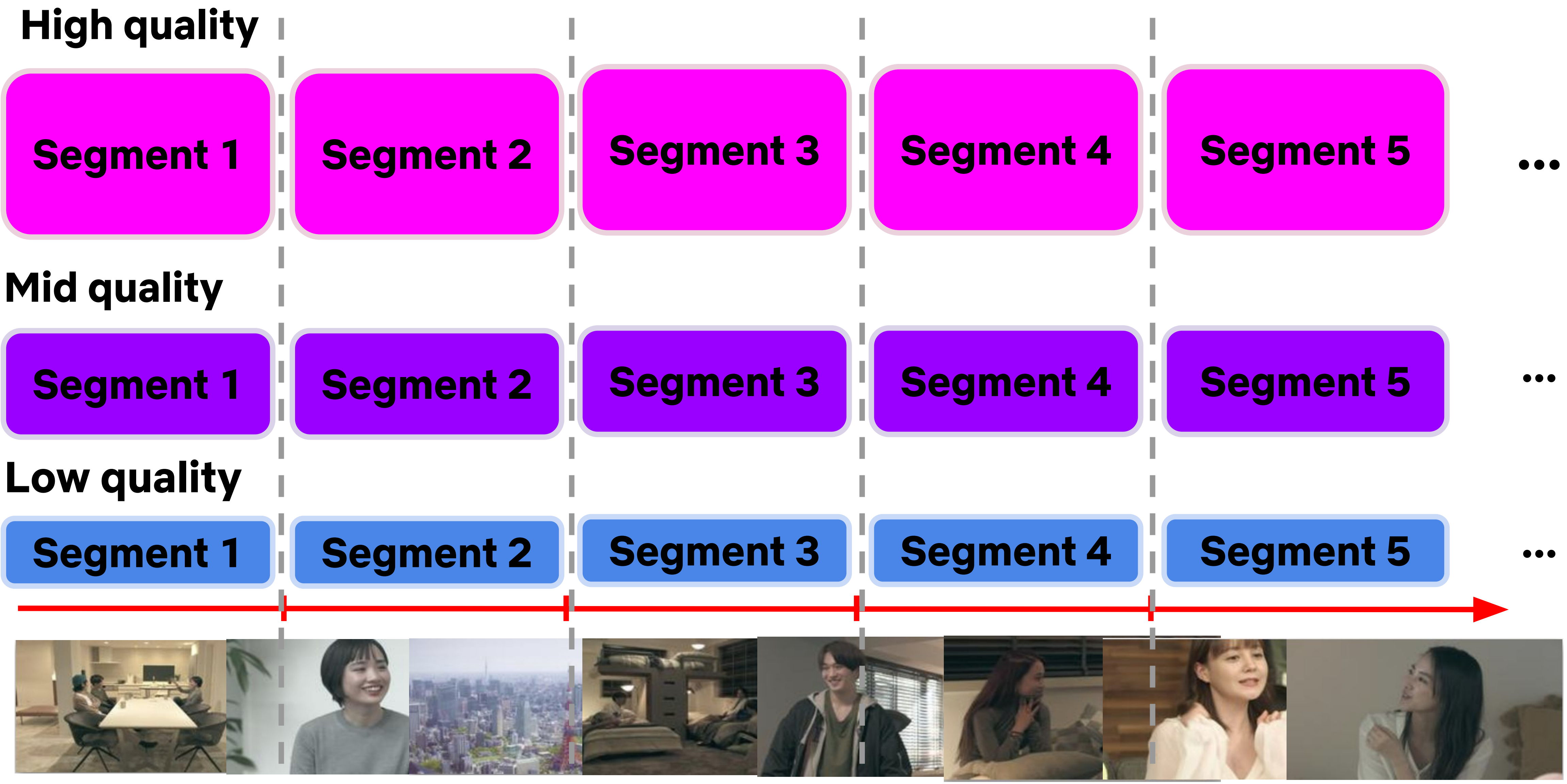
We ran an experiment which demonstrates this.

Treatment: capping bitrate to reduce traffic

In response to COVID-19, Netflix reduced traffic by 25% by capping bitrates.

Capping bitrates means that Netflix will not serve the highest quality versions of a video

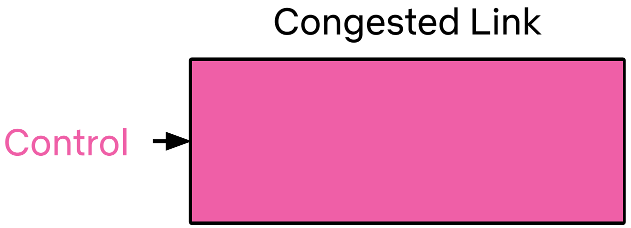
Videos are encoded at many different qualities



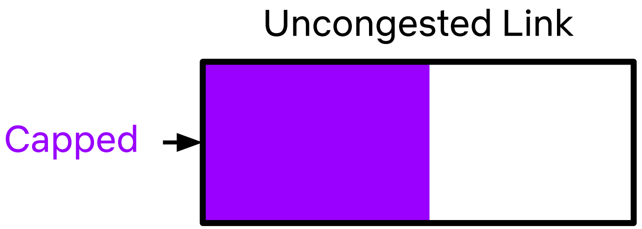
Bitrate capping limits video quality we can send

What could A/B tests look like with bitrate capping?

Originally:
Link is congested

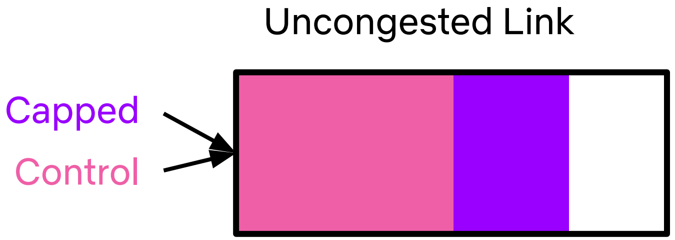


With Capping:
Link is not congested



Capping causes:
- Less bandwidth used
- Less congestion

One possibility:
Bitrate capping reduces congestion

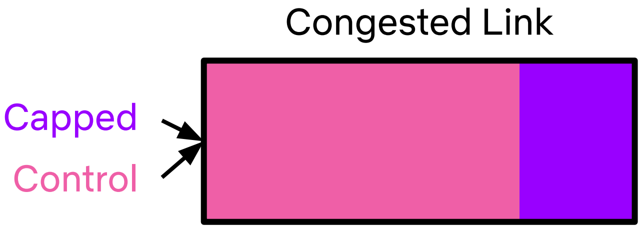


A/B test results:

✅ Capped uses less bandwidth

❌ Level of congestion is the same (no congestion)

Another possibility:
Control traffic increases, link stays congested



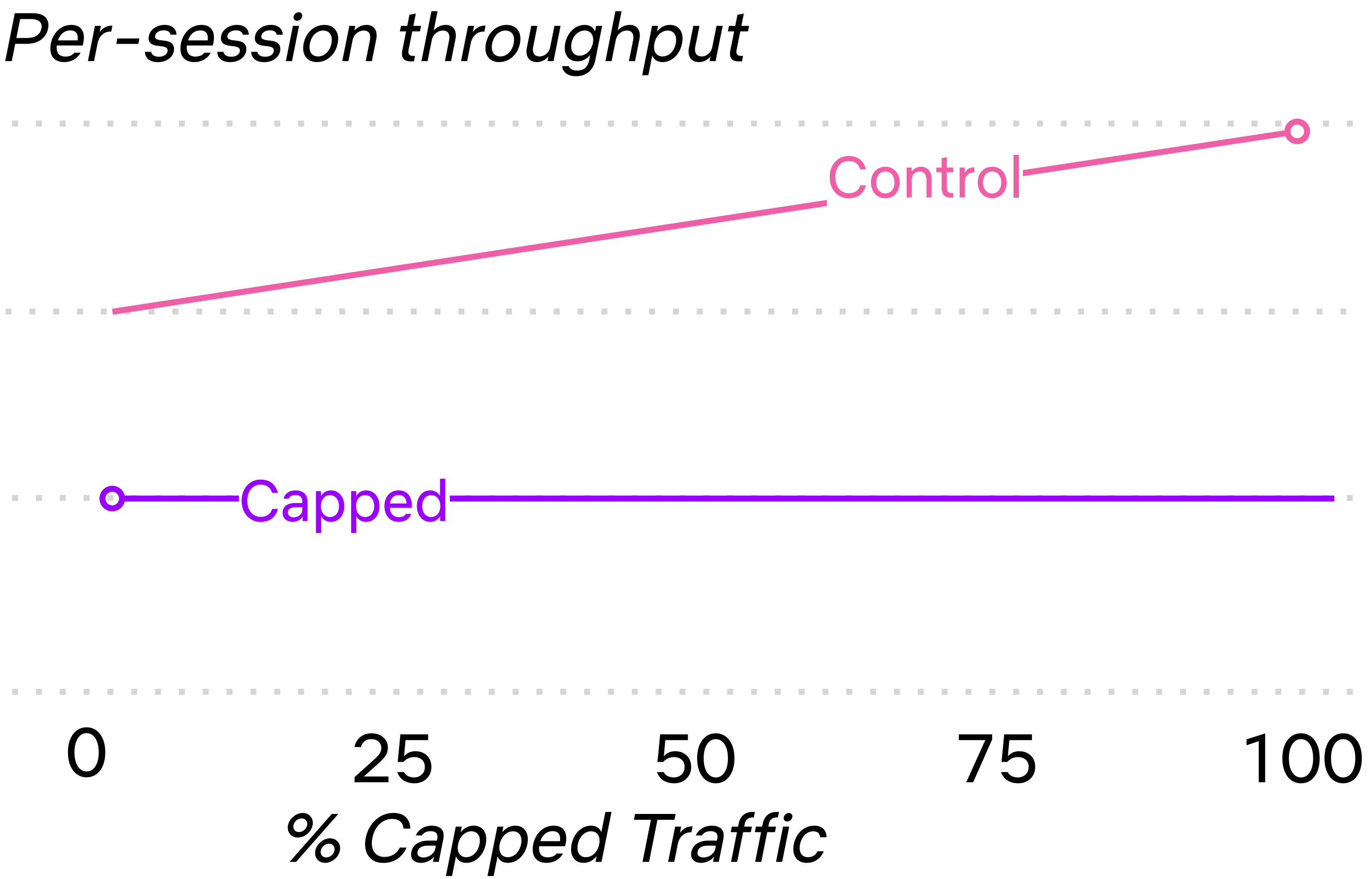
A/B test results:

Capped uses less bandwidth

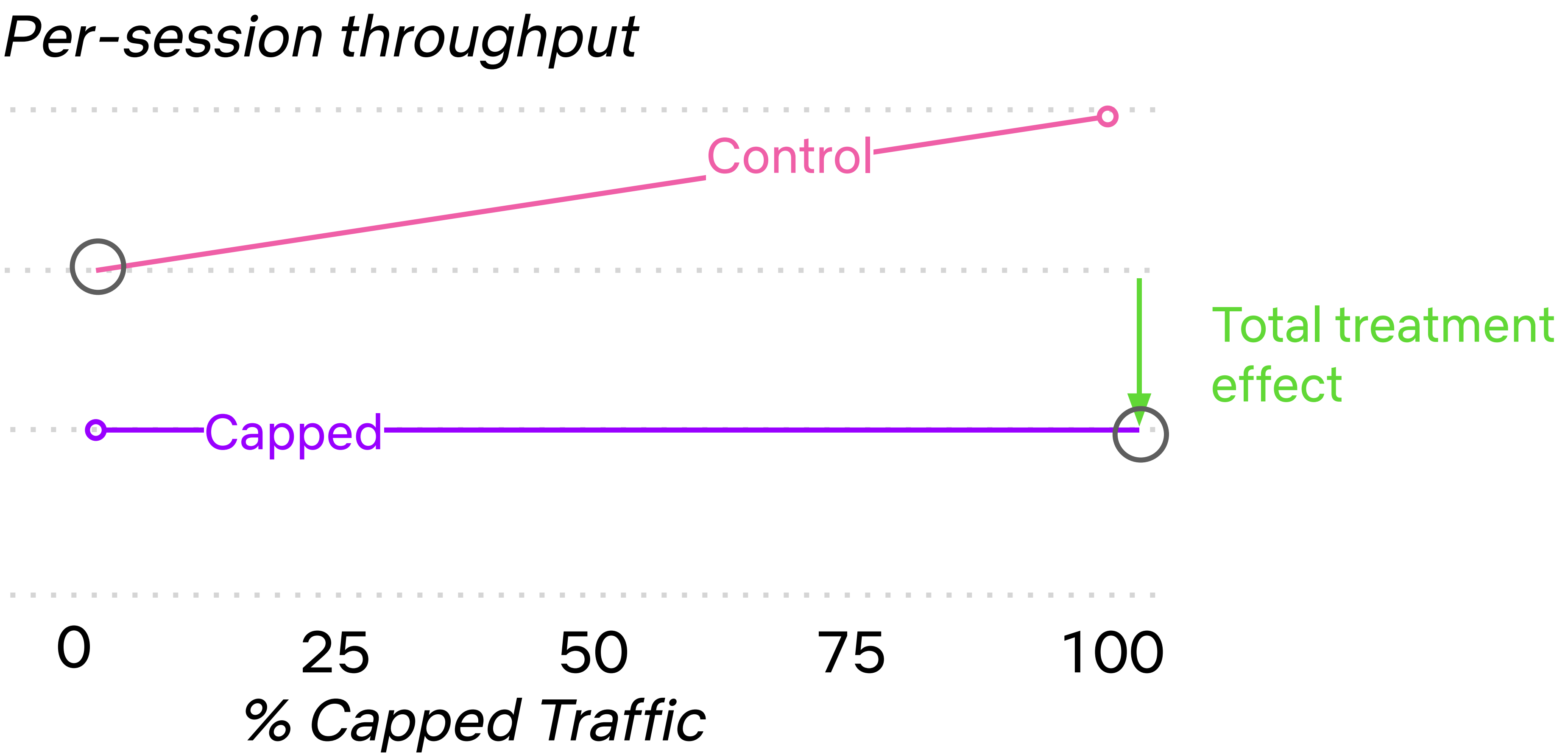
❌ Level of congestion is the same (some congestion)

A/B tests results do not reveal what happens when we cap traffic

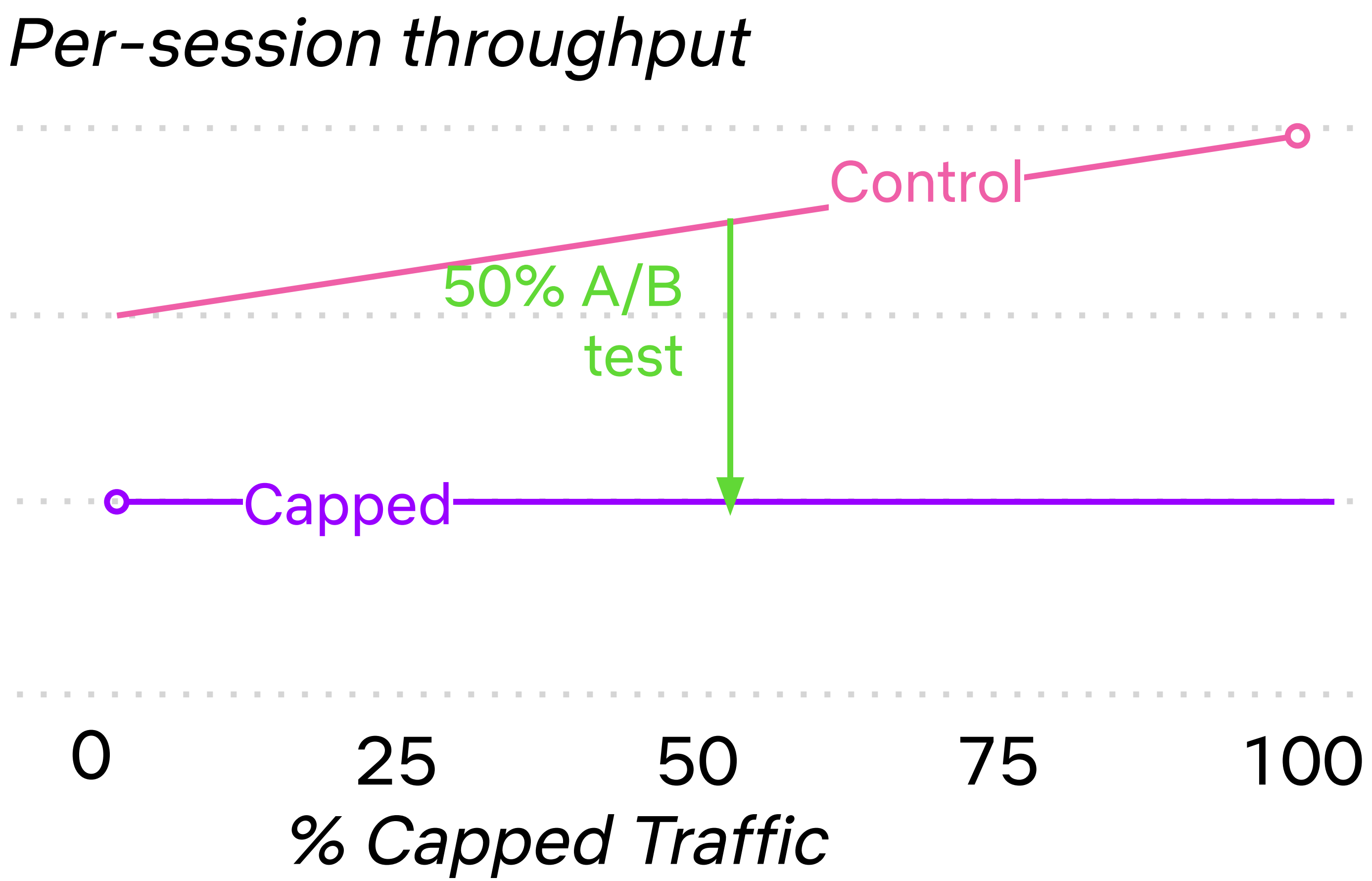
Imagine control throughput increases as traffic is capped



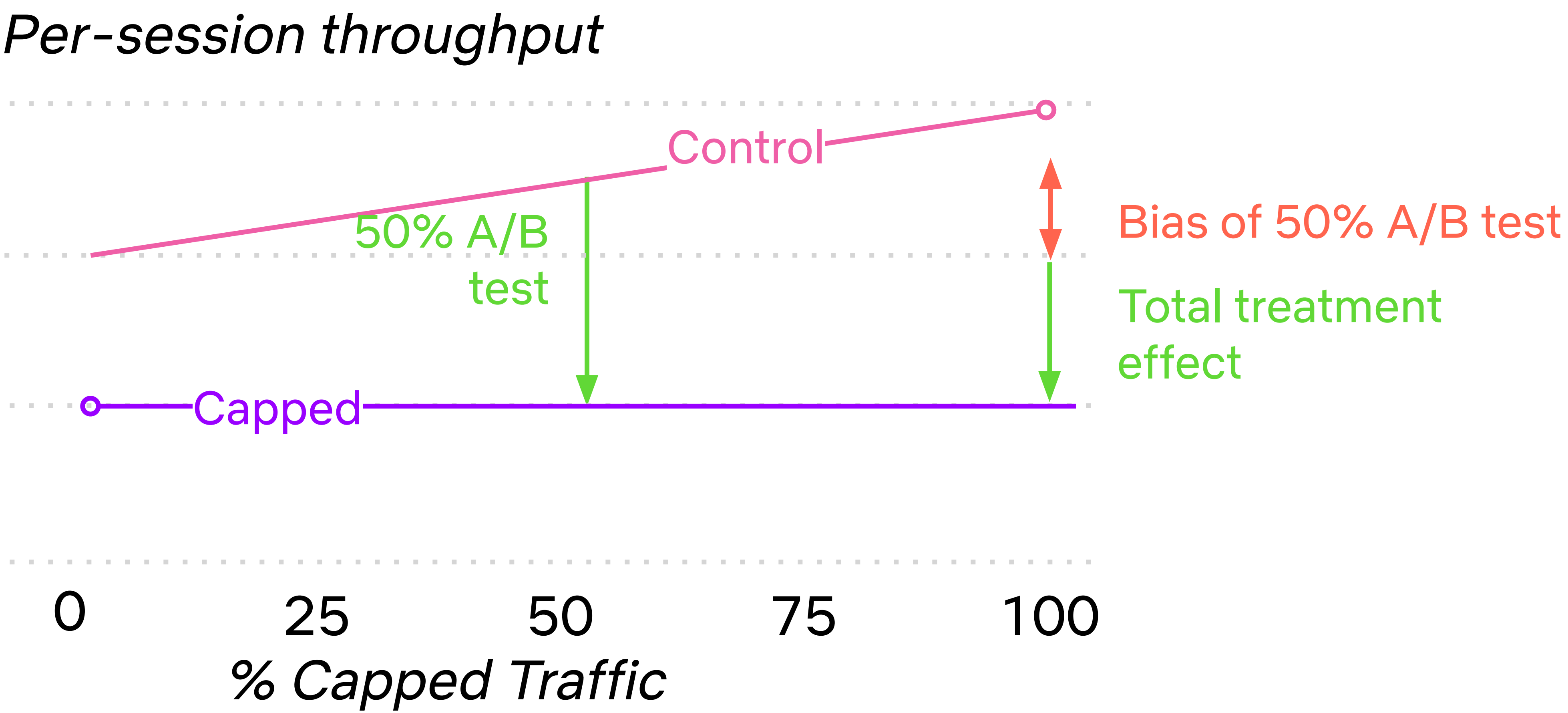
We want to measure the effect of capping



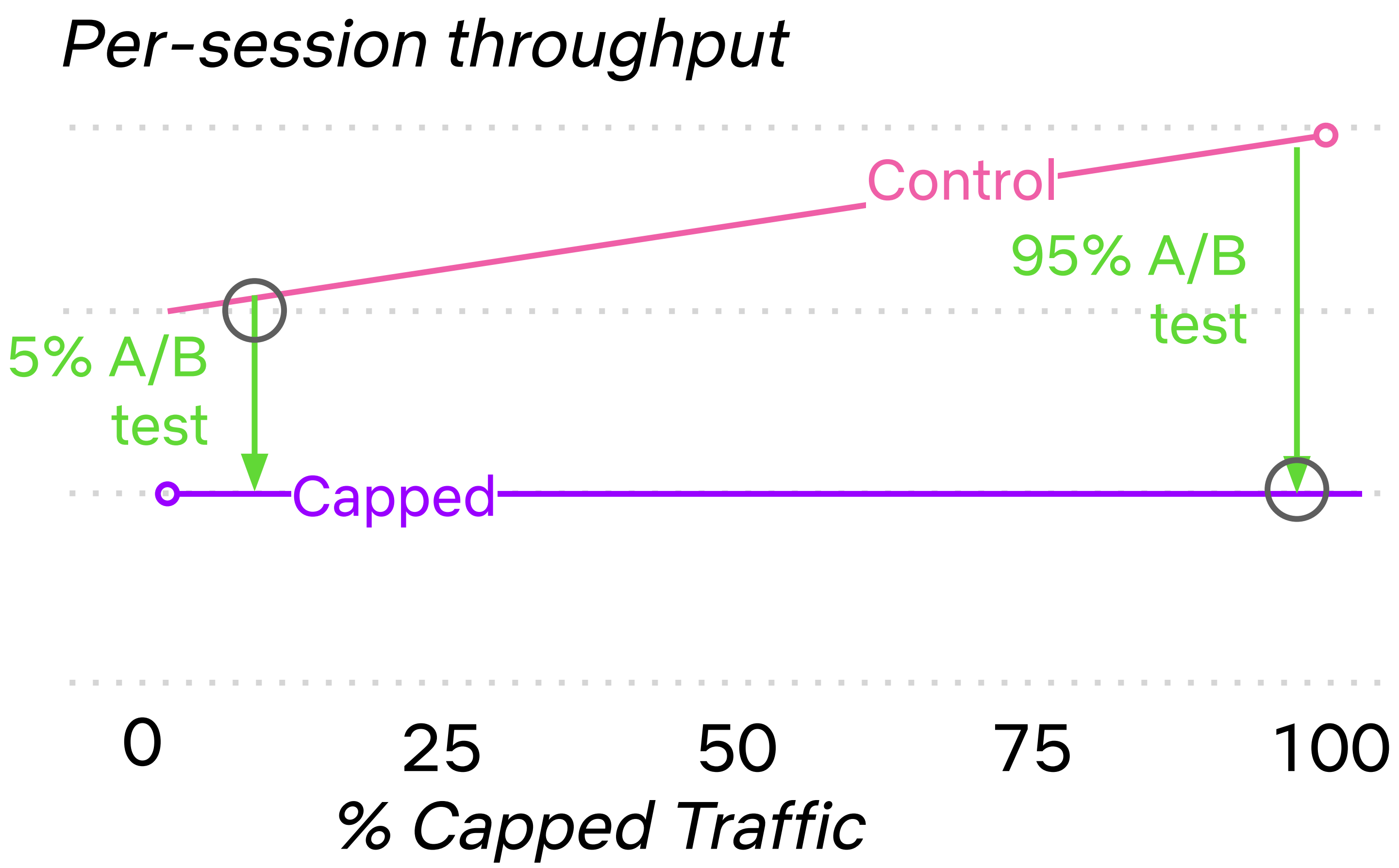
A/B tests look at one point on this graph



A/B tests give biased estimates of total treatment effects



With two measurements, we can measure capping effects and A/B test bias

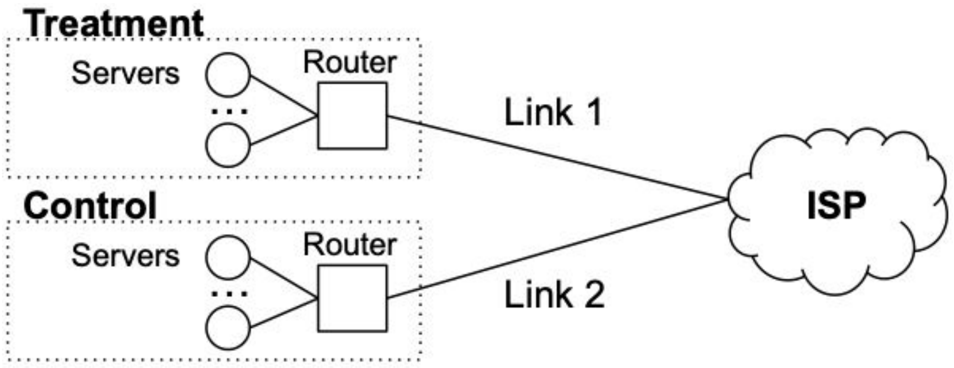


Comparing A/B tests with a pair of congested links

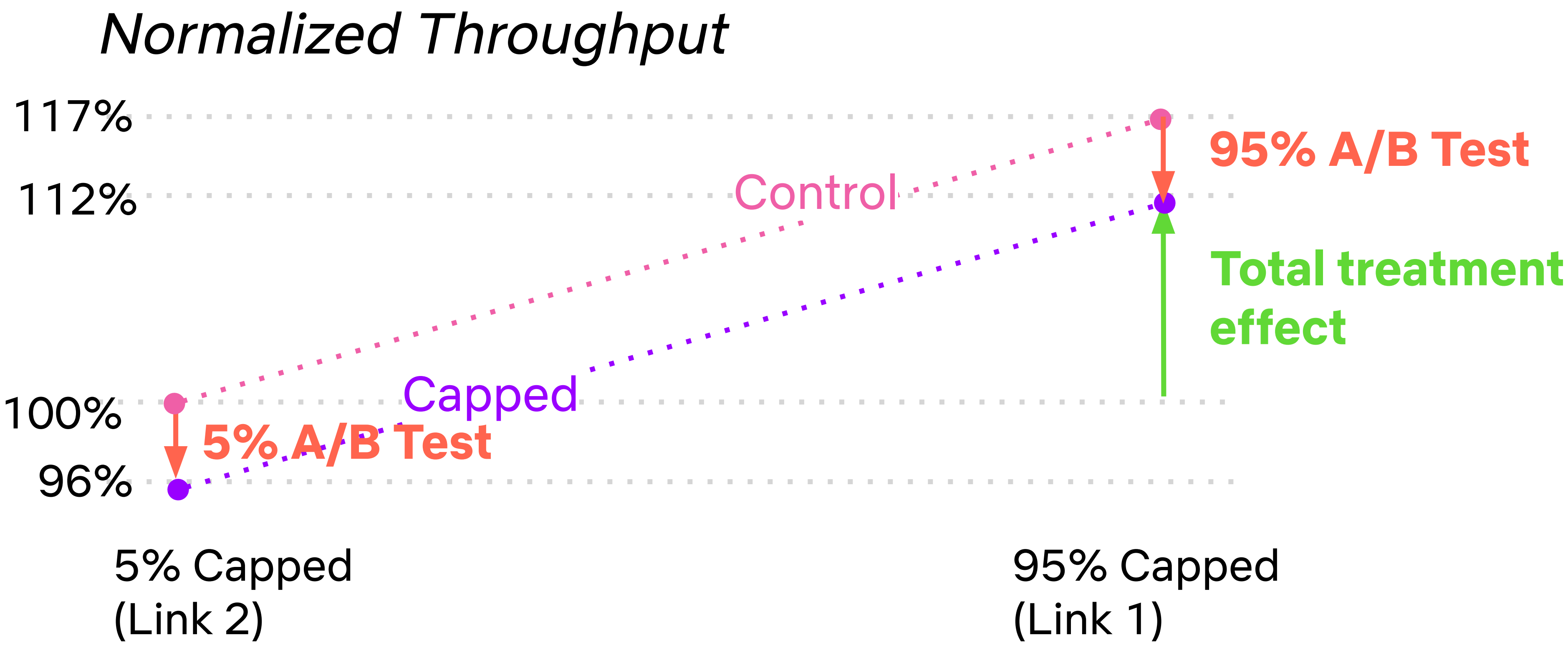
Found two reliably congested peering links with well-balanced traffic

Run two A/B tests on each link and compare:

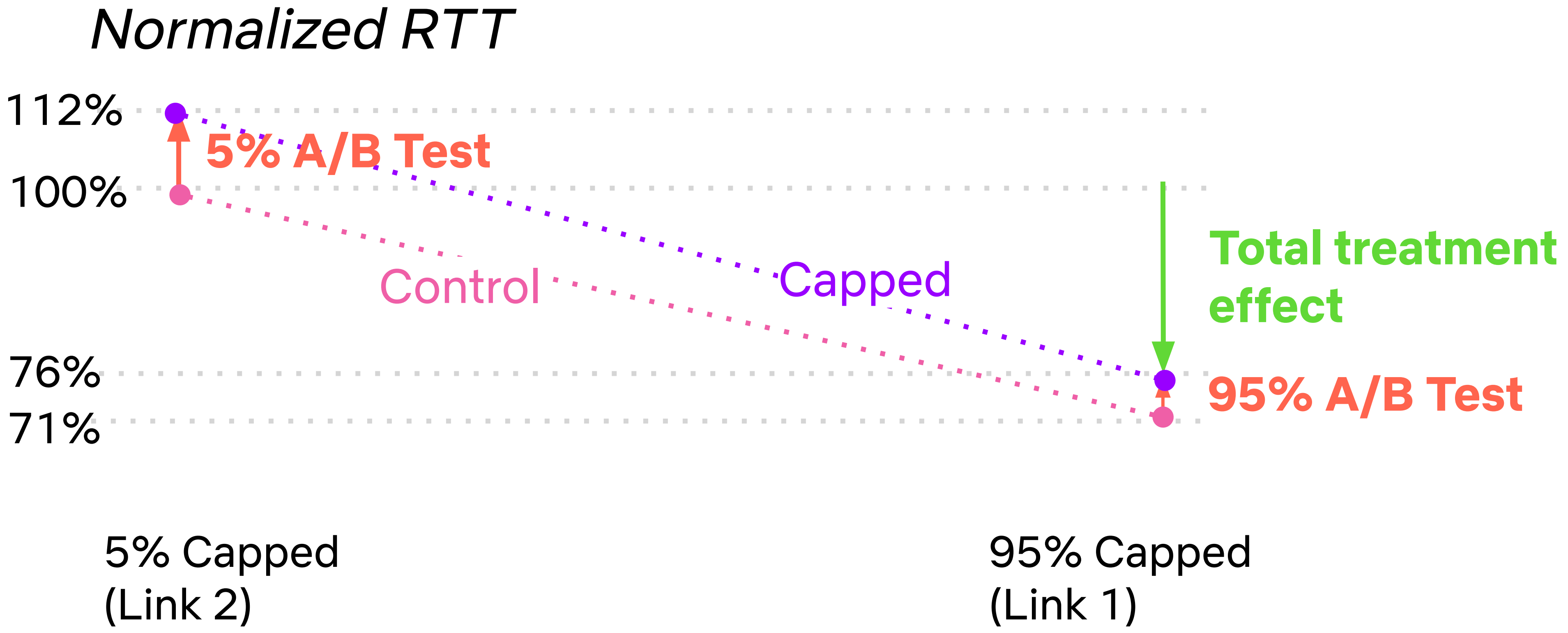
- **Link 1:** 95% capped, 5% uncapped
- **Link 2:** 5% capped, 95% uncapped



Capping improves throughput, despite A/B test results



A/B tests are also wrong about effects on RTT

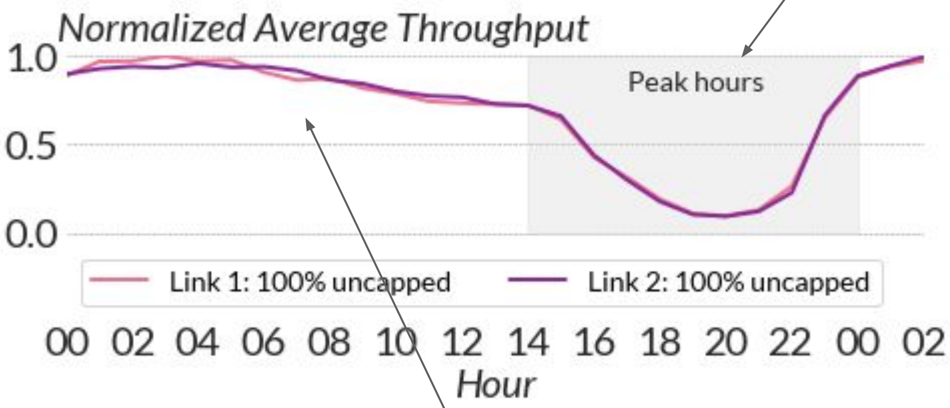


Per-session Throughput results

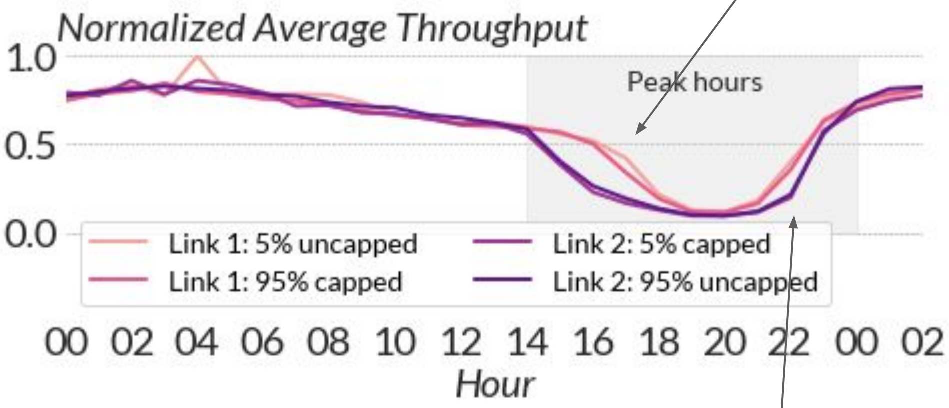
Throughput decreases during
congested peak hours

Capping delays the onset of
congestion for capped link

Before experiment



During experiment



Behavior is similar within a link

Two links have identical throughput

A/B tests do not reliably estimate TTE

Metric	Total Treatment Effect	A/B Test
Round Trip Time	25% better	5-15% worse
Throughput	12% better	5% worse
Play Delay	10% better	Did not change

and more in the paper…

A/B tests are biased when run in congested networks

This is concerning!

Risks of congestion interference

Common development process:

1. Come up with idea
2. Implement idea
3. A/B test idea
4. Iterate

   …

5. Deploy idea

Could give up too early on a good idea, or continue with an approach that doesn't work

Could deploy things that don't work as expected, leading to production issues or longer development time

We can run experiments that remove bias

Paired link experiment is just one example

In the paper we also discuss:

- Event studies
- Switchback experiments

Use event studies when deploying changes

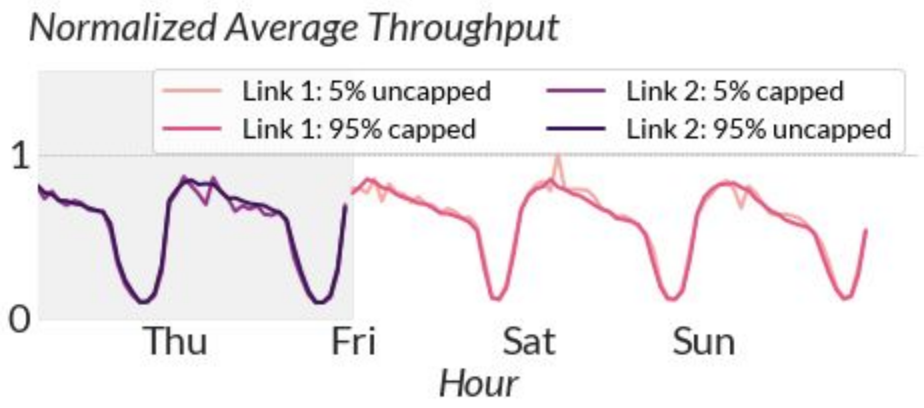
Switch most traffic to treatment and compare before/after

Pros:

- Estimates TTE
- Easy to do when deploying changes

Cons:

- Seasonality issues



Use switchbacks for more accurate measurements

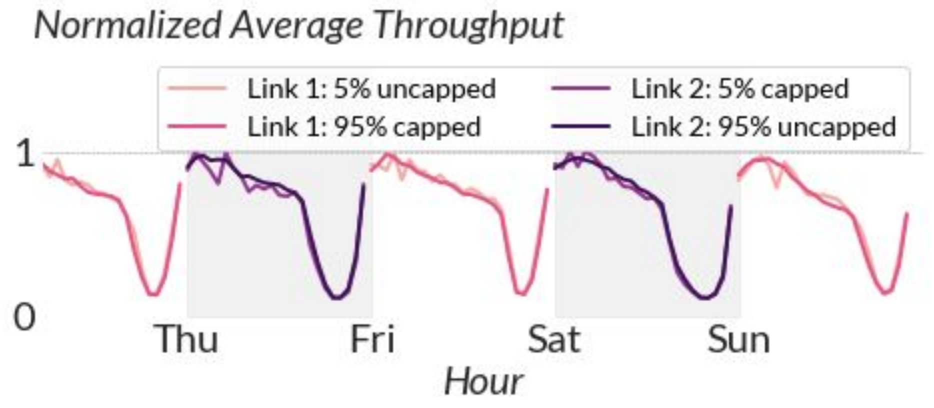
Switch back and forth between treatment/control

Pros:

- Estimates TTE
- More robust to seasonality

Cons:

- Carryover effects



Lots more to be done!

- Any A/B test using a congested network has the possibility of bias
- We encourage more measurement to tell if interference matters for your experiments.
- We would love to see total treatment effects measured for new algorithms
- Need for better experiment methodology for networks

Thank you!
Email: bspang@stanford.edu

ArXiv: