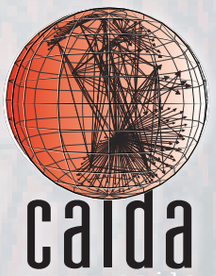
TCP CONGESTION SIGNATURES

Srikanth Sundaresan (Facebook)
Amogh Dhamdhere (CAIDA/UCSD)
kc Claffy (CAIDA/UCSD)
Mark Allman (ICSI)



Typical Speed Tests Don't Tell Us Much

Typical Speed Tests Don't Tell Us Much

Typical Speed Tests Don't Tell Us Much

Typical Speed Tests Don't Tell Us Much

- Upload and download throughput measurements: no information beyond that

Typical Speed Tests Don't Tell Us Much

What type of congestion did the TCP flow experience?

caida

Two Potential Sources of Congestion in the End-to-end Path

Two Potential Sources of Congestion in the End-to-end Path

- Self-induced congestion
  - Clear path, the flow is able to saturate the bottleneck link
  - eg: last-mile access link

Two Potential Sources of Congestion in the End-to-end Path

- Self-induced congestion
  - Clear path, the flow is able to saturate the bottleneck link
  - eg: last-mile access link
- External congestion
  - Flow starts on an already congested path
  - eg: congested interconnect

Two Potential Sources of Congestion in the End-to-end Path

- Self-induced congestion
  - Clear path, the flow is able to saturate the bottleneck link
  - eg: last-mile access link
- External congestion
  - Flow starts on an already congested path
  - eg: congested interconnect

Distinguishing the two cases has implications for users / ISPs / regulators

Does Throughput Indicate Type of Congestion?

- Cannot distinguish using just throughput numbers
  - Access plan rates vary widely, and are typically not available to content / speed test providers
  - eg: Speed test reports 5 Mbps – is that the access link rate (DSL), or a congested path?

Does Throughput Indicate Type of Congestion?

- Cannot distinguish using just throughput numbers
  - Access plan rates vary widely, and are typically not available to content / speed test providers
  - eg: Speed test reports 5 Mbps – is that the access link rate (DSL), or a congested path?

We can use the dynamics of TCP's startup phase, i.e., Congestion Signatures

TCP's RTT Congestion Signatures

TCP's RTT Congestion Signatures

- Flows experiencing self-induced congestion fill up an empty buffer during slow start
  - Hence increase the TCP flow RTT

TCP's RTT Congestion Signatures

- Flows experiencing self-induced congestion fill up an empty buffer during slow start
  - Hence increase the TCP flow RTT
- Externally congested flows encounter an already full buffer
  - Less potential for RTT increases

TCP's RTT Congestion Signatures

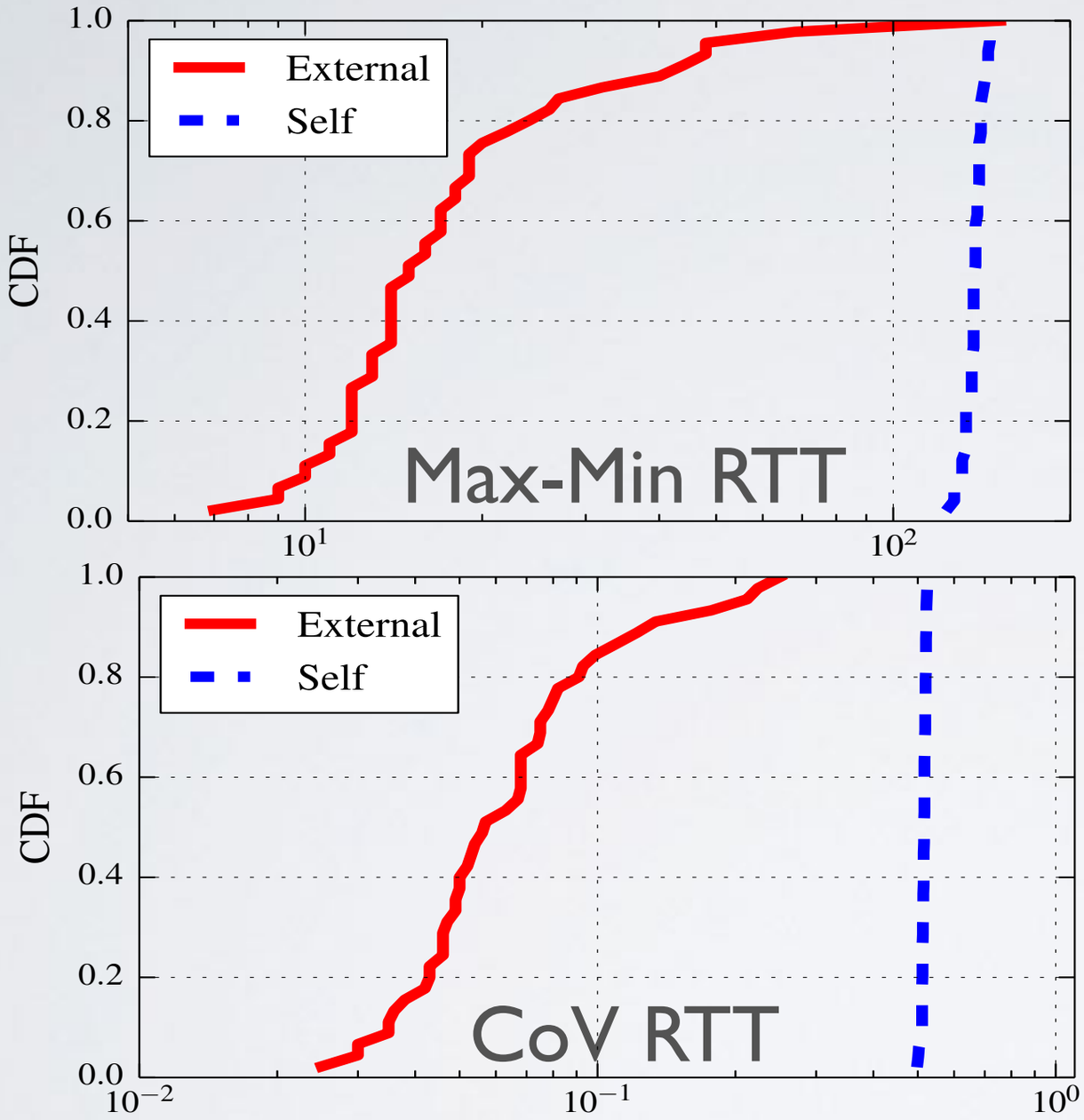
- Flows experiencing self-induced congestion fill up an empty buffer during slow start
  - Hence increase the TCP flow RTT
- Externally congested flows encounter an already full buffer
  - Less potential for RTT increases
- Self-induced congestion therefore has higher RTT variance compared to external congestion

TCP's RTT Congestion Signatures

- Flows experiencing self-induced congestion fill up an empty buffer during slow start
  - Hence increase the TCP flow RTT
- Externally congested flows encounter an already full buffer
  - Less potential for RTT increases
- Self-induced congestion therefore has higher RTT variance compared to external congestion

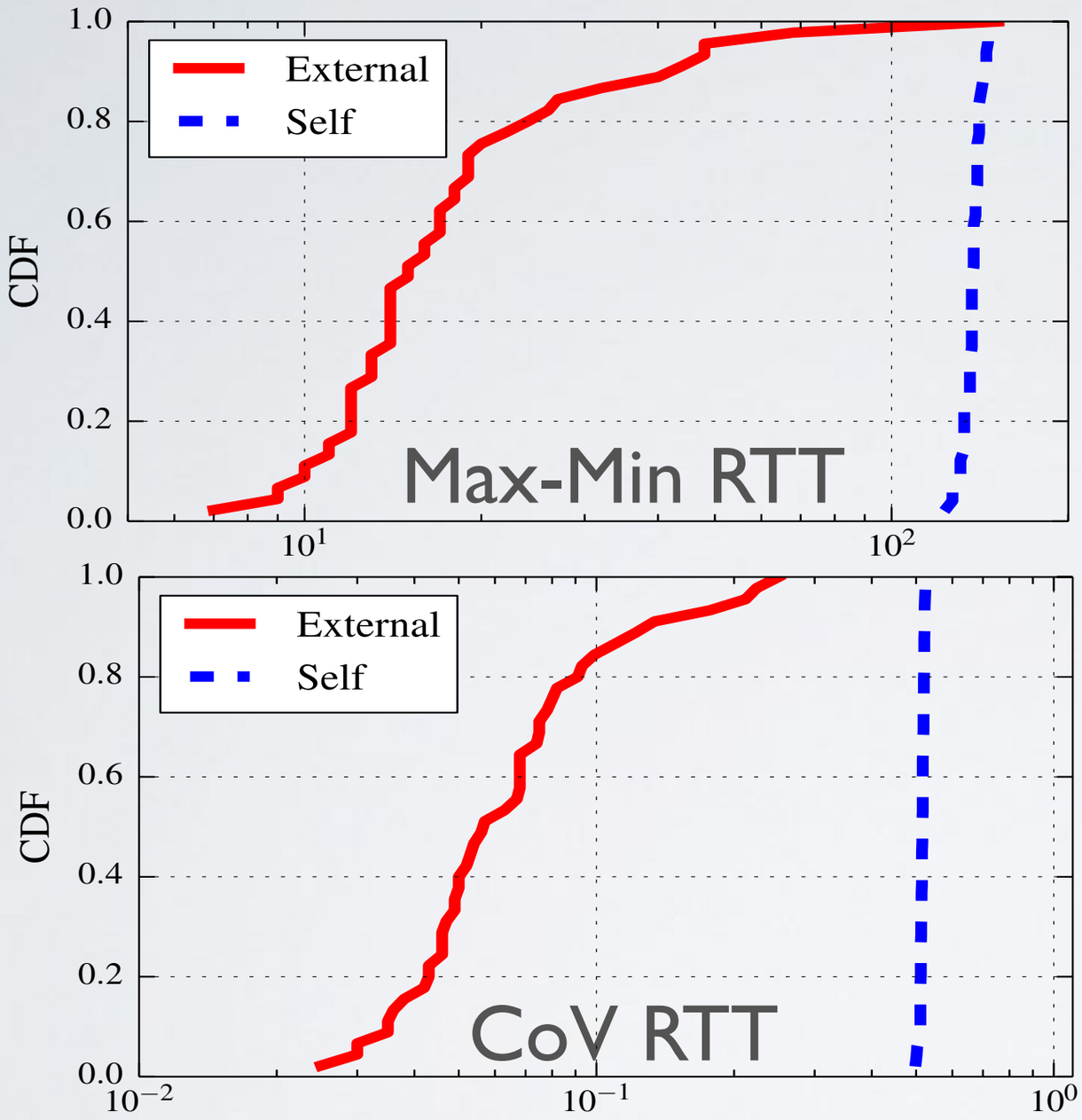
We can quantify this using Max-Min and CoV of RTT

Example Controlled Experiment



- 20 Mbps "access" link with 100 ms buffer
- 1 Gbps "interconnect" link with 50 ms buffer
- Self-induced congestion flows have higher values for both metrics and are clearly distinguishable

Example Controlled Experiment



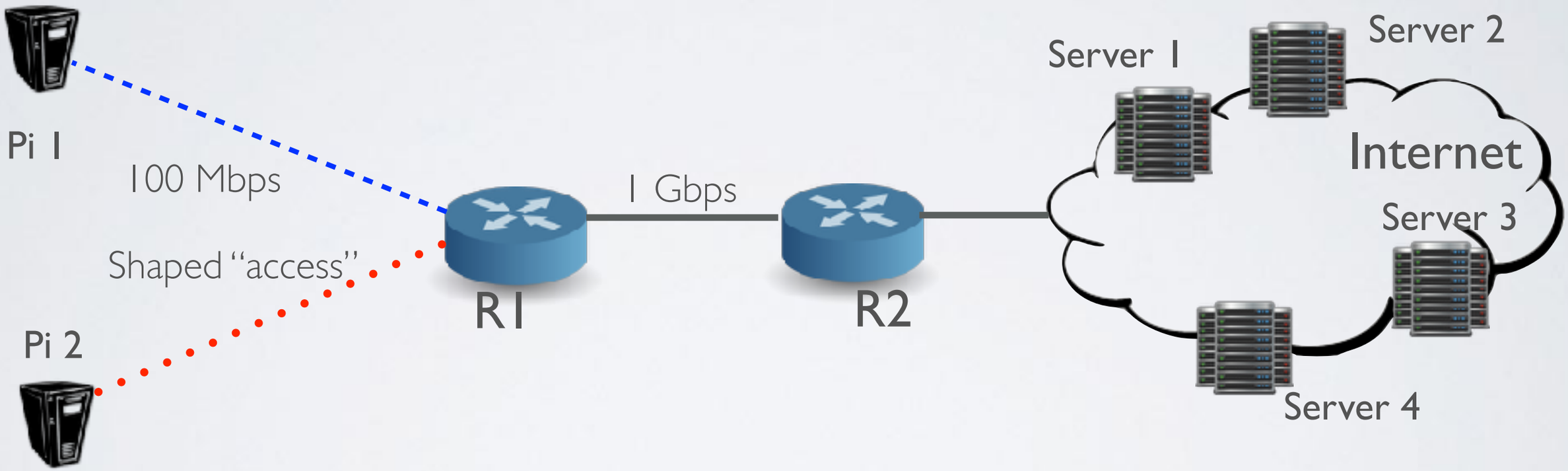
- 20 Mbps "access" link with 100 ms buffer
- 1 Gbps "interconnect" link with 50 ms buffer
- Self-induced congestion flows have higher values for both metrics and are clearly distinguishable

The two types of congestion exhibit widely contrasting behaviors

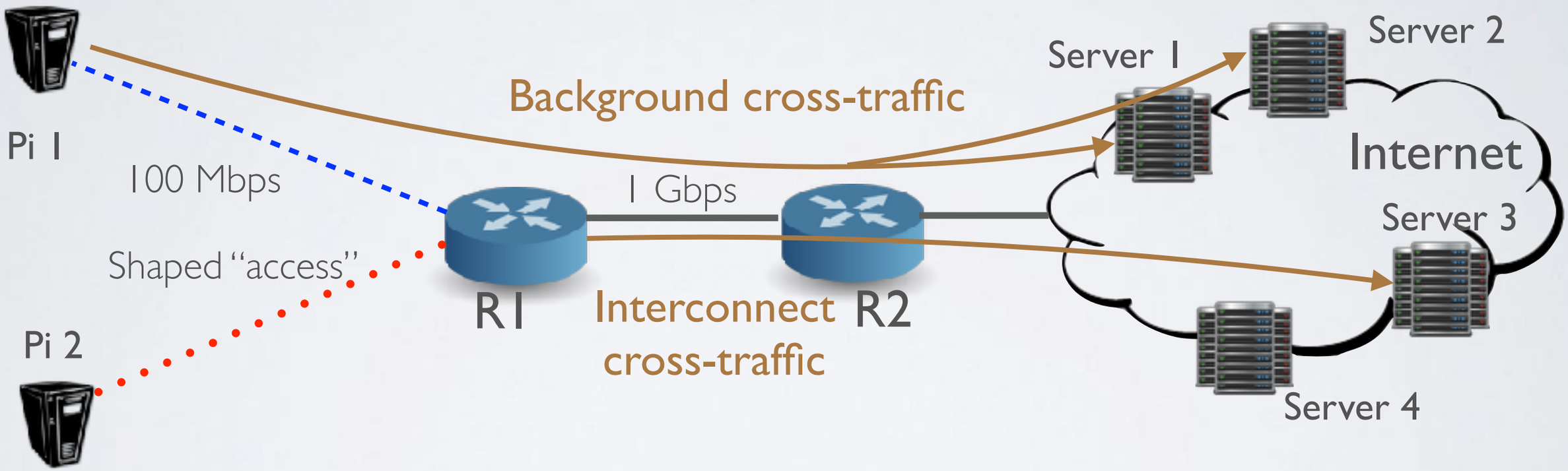
Model

- Max-min and CoV of RTT derived from RTT samples during slow start
- We feed the two metrics into a simple Decision Tree
  - We control the depth of the tree to a low value to minimize complexity
- We build the decision tree classifier using controlled experiments and apply it to real-world data

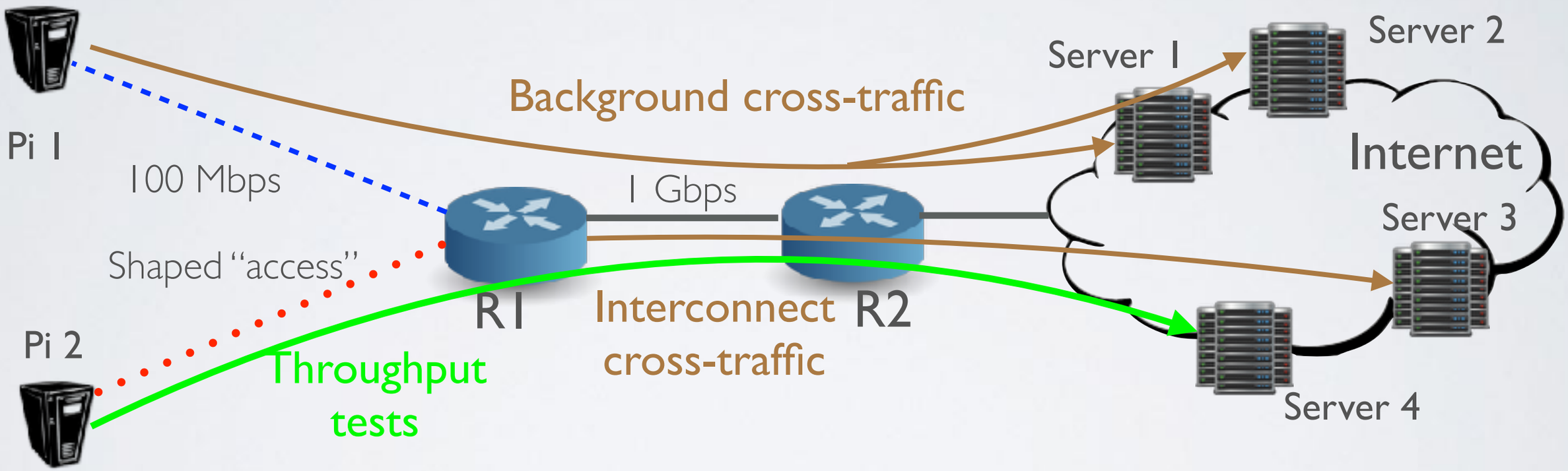
Validating the Method: Step 1- Controlled Experiments



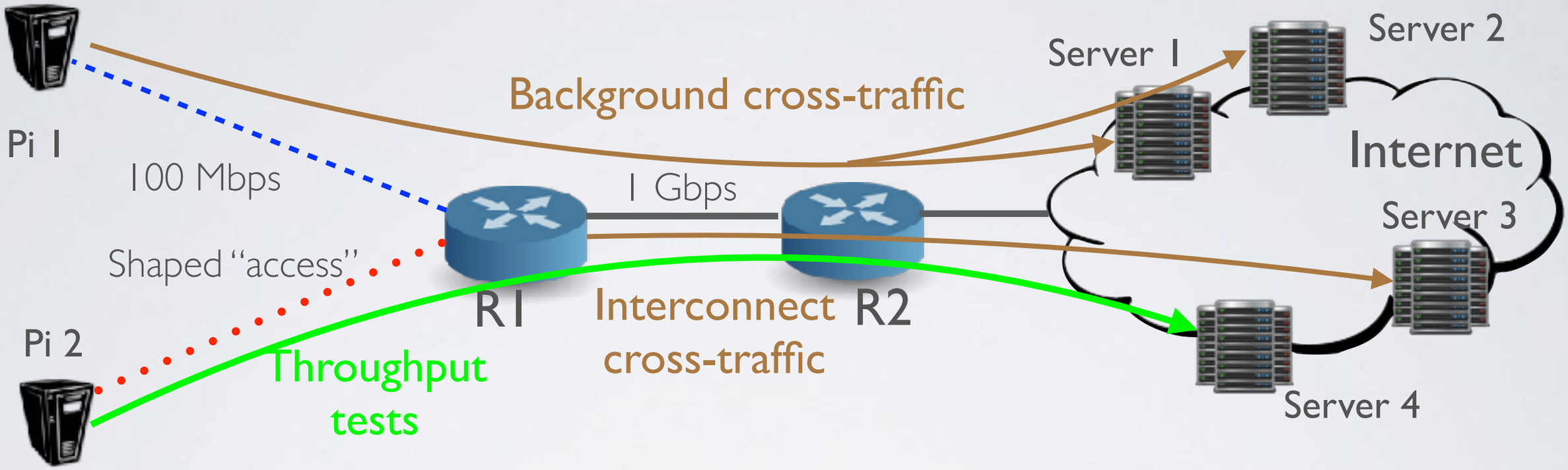
Validating the Method: Step 1- Controlled Experiments



Validating the Method: Step 1- Controlled Experiments

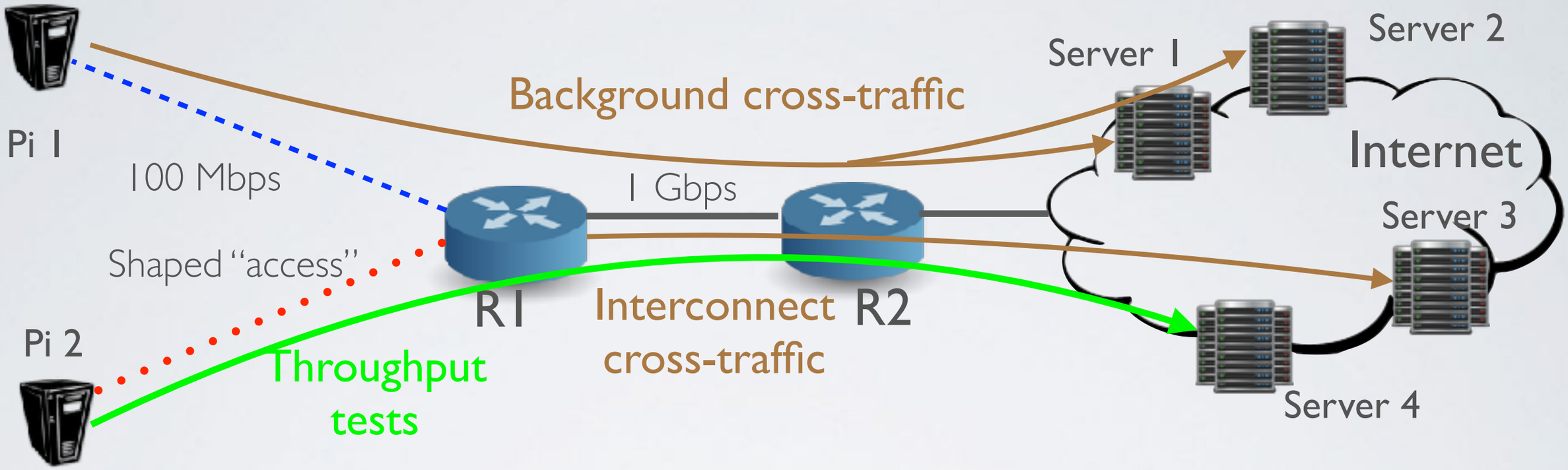


Validating the Method: Step 1- Controlled Experiments



- Emulated "access" link + "core" link
  - Wide range of access link throughputs, buffer sizes, loss rates, cross-traffic (background and congestion-inducing)
  - Can accurately label flows in training data as "self" or "externally" congested

Validating the Method: Step 1- Controlled Experiments



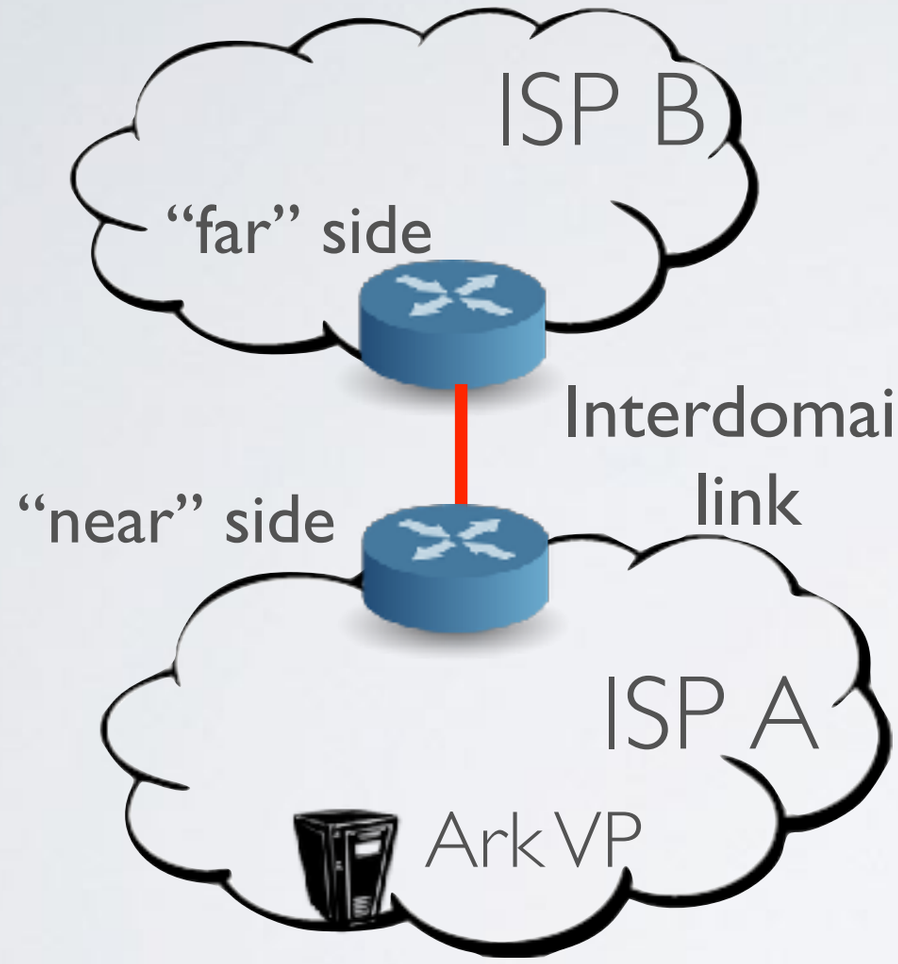
High accuracy: precision and recall > 90% in most settings

Validating the Method: Step 2

- From Ark VP in ISP A identified congested link with ISP B using TSLP*

*Dhamdhere et al. "Inferring Persistent Interdomain Congestion", SIGCOMM 2018

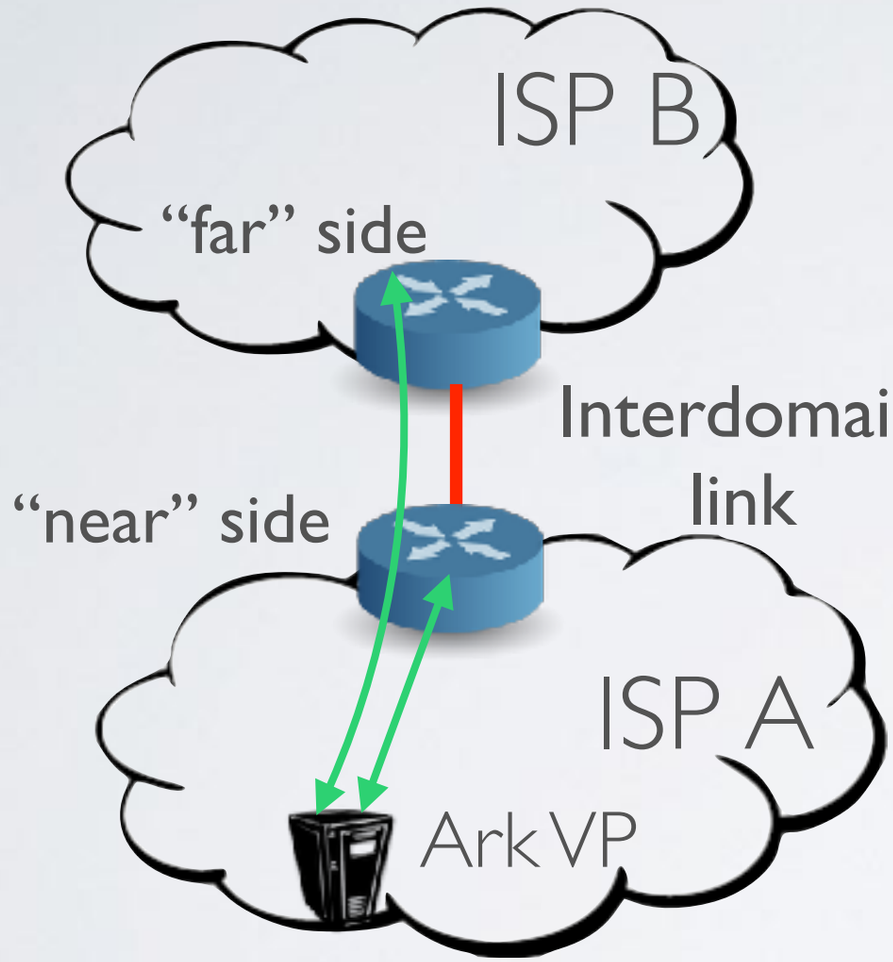
Validating the Method: Step 2



- From Ark VP in ISP A identified congested link with ISP B using TSLP*

*Dhamdhere et al. "Inferring Persistent Interdomain Congestion", SIGCOMM 2018

Validating the Method: Step 2



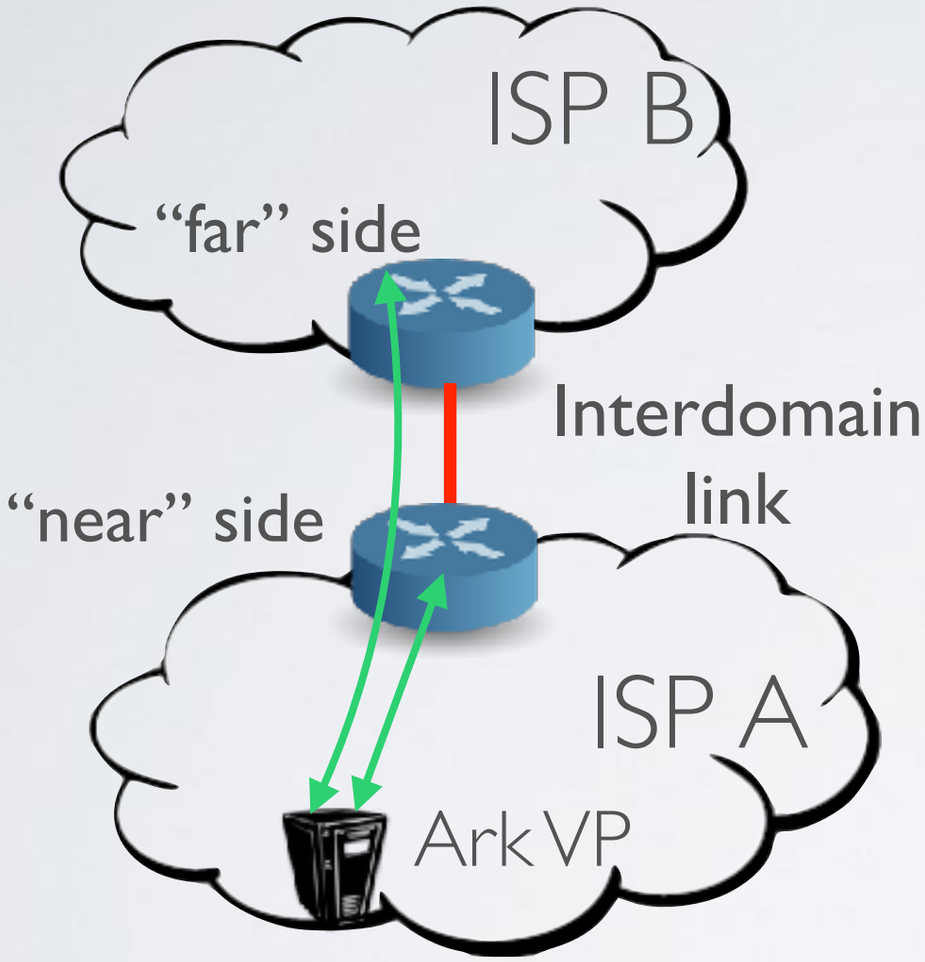
Latency measurements to "near" and "far" side of interdomain link over time

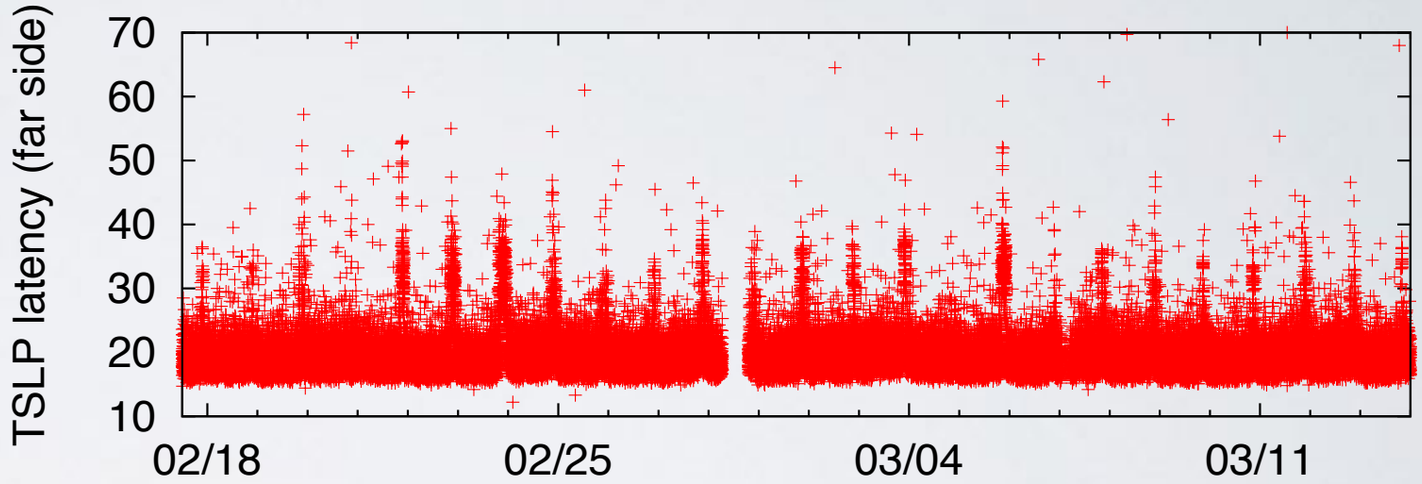
- From Ark VP in ISP A identified congested link with ISP B using TSLP*

*Dhamdhere et al. "Inferring Persistent Interdomain Congestion", SIGCOMM 2018

caida

Validating the Method: Step 2

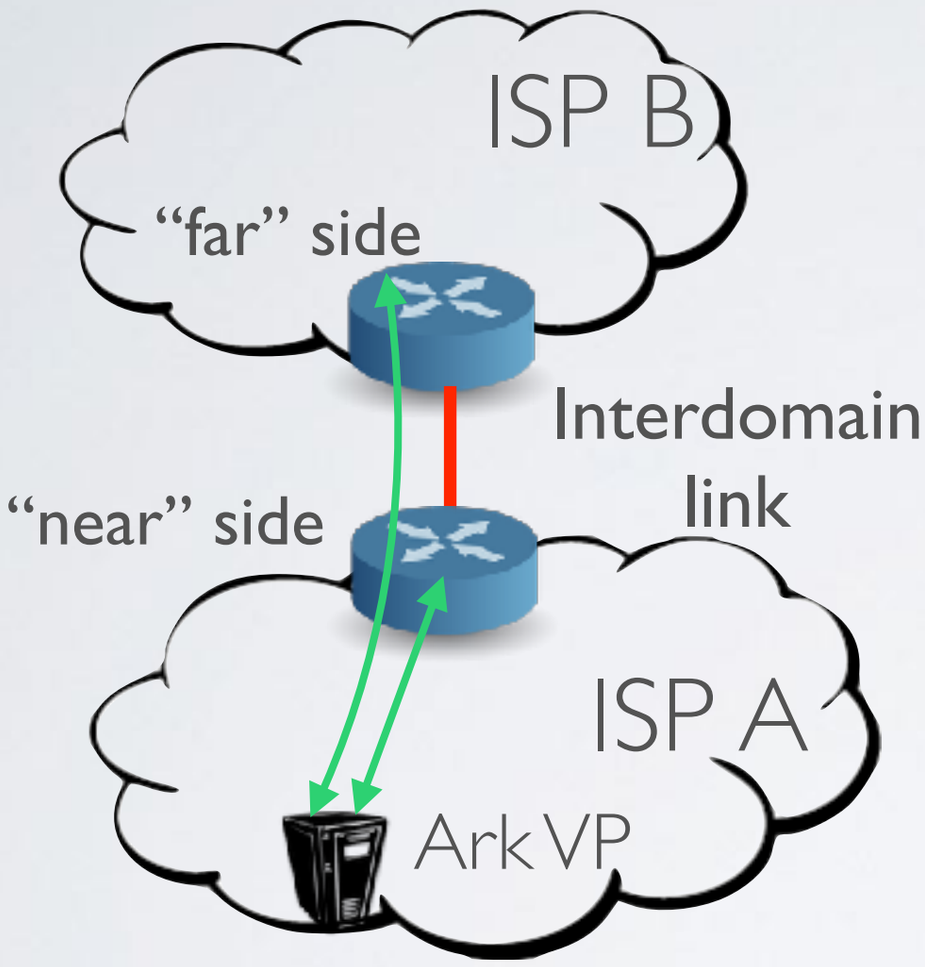


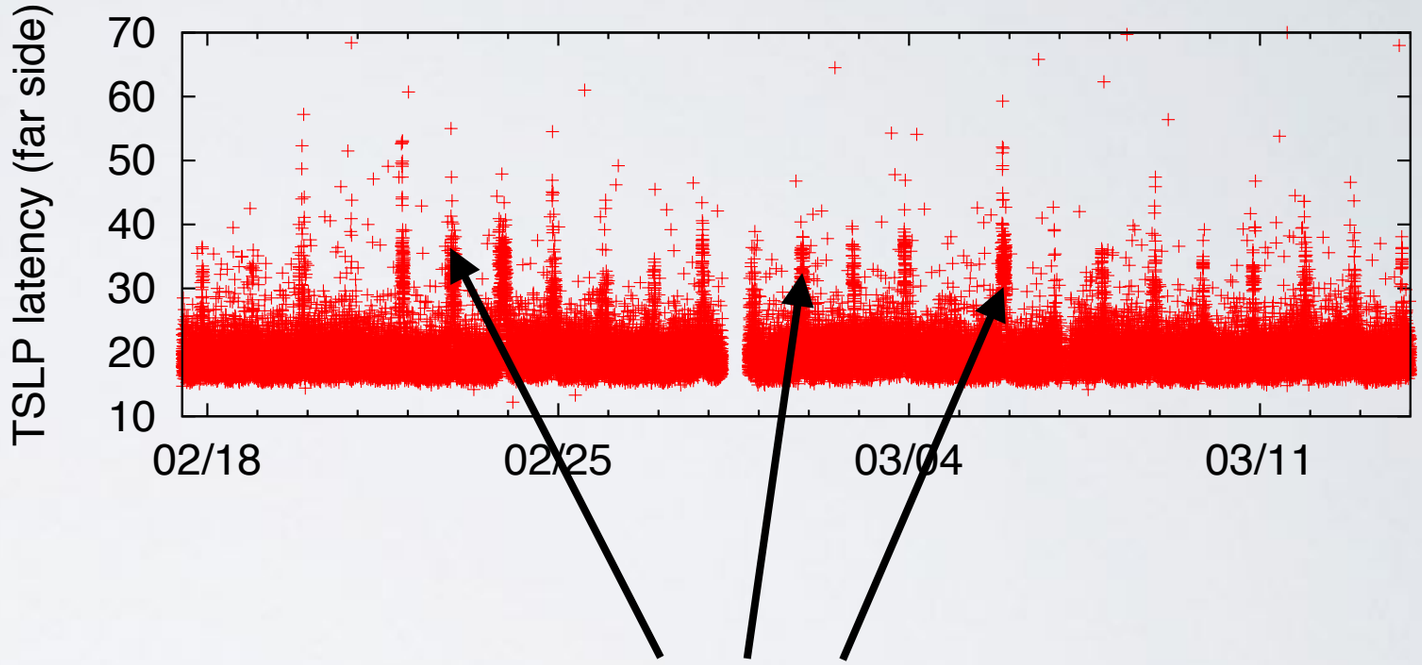


- From Ark VP in ISP A identified congested link with ISP B using TSLP*

*Dhamdhere et al. "Inferring Persistent Interdomain Congestion", SIGCOMM 2018

Validating the Method: Step 2



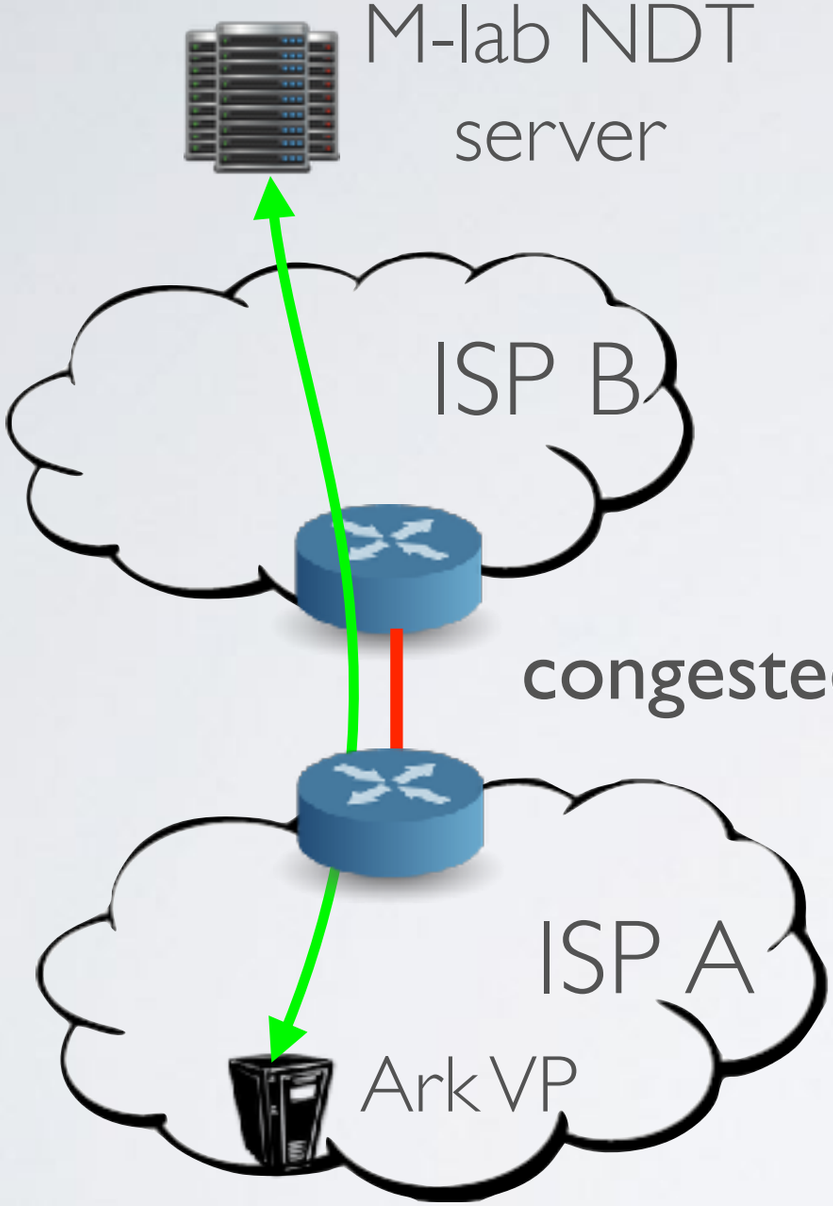


Diurnal latency elevation indicates congestion

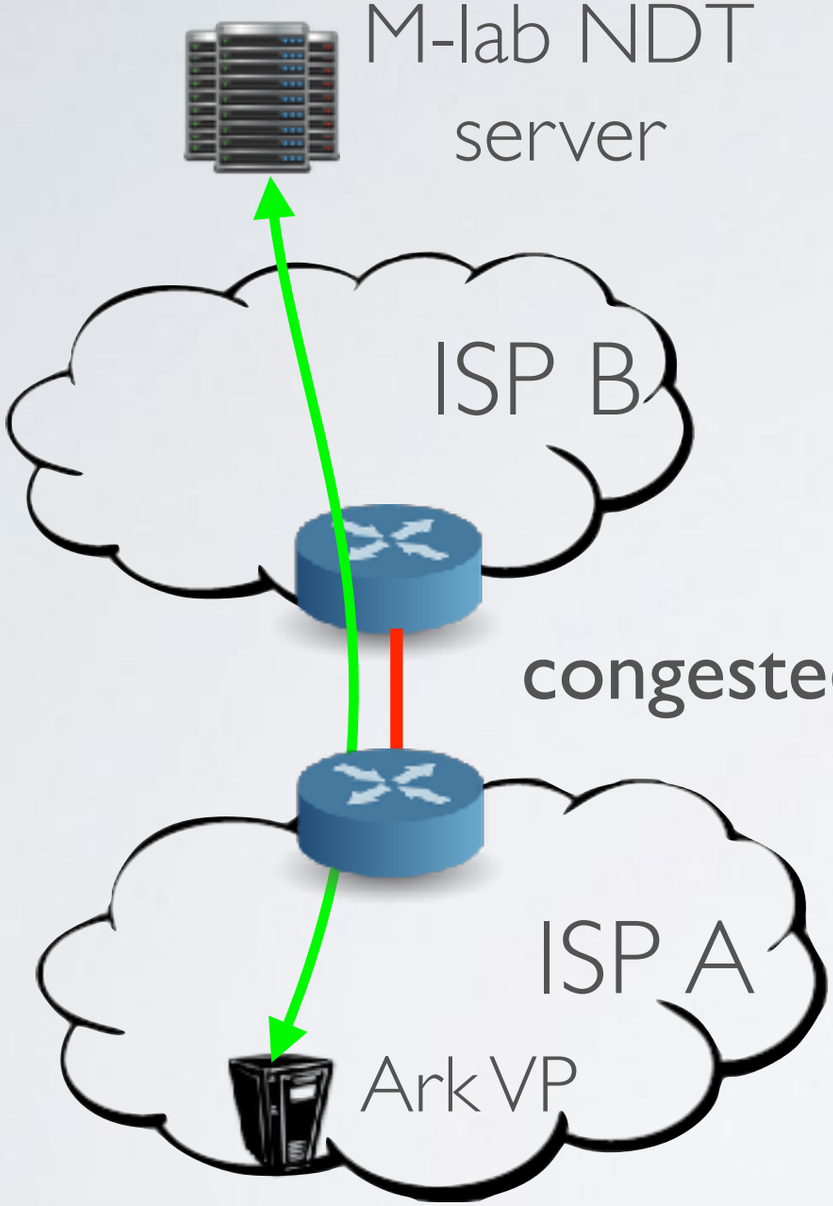
- From Ark VP in ISP A identified congested link with ISP B using TSLP*

*Dhamdhere et al. "Inferring Persistent Interdomain Congestion", SIGCOMM 2018

Validating the Method: Step 2

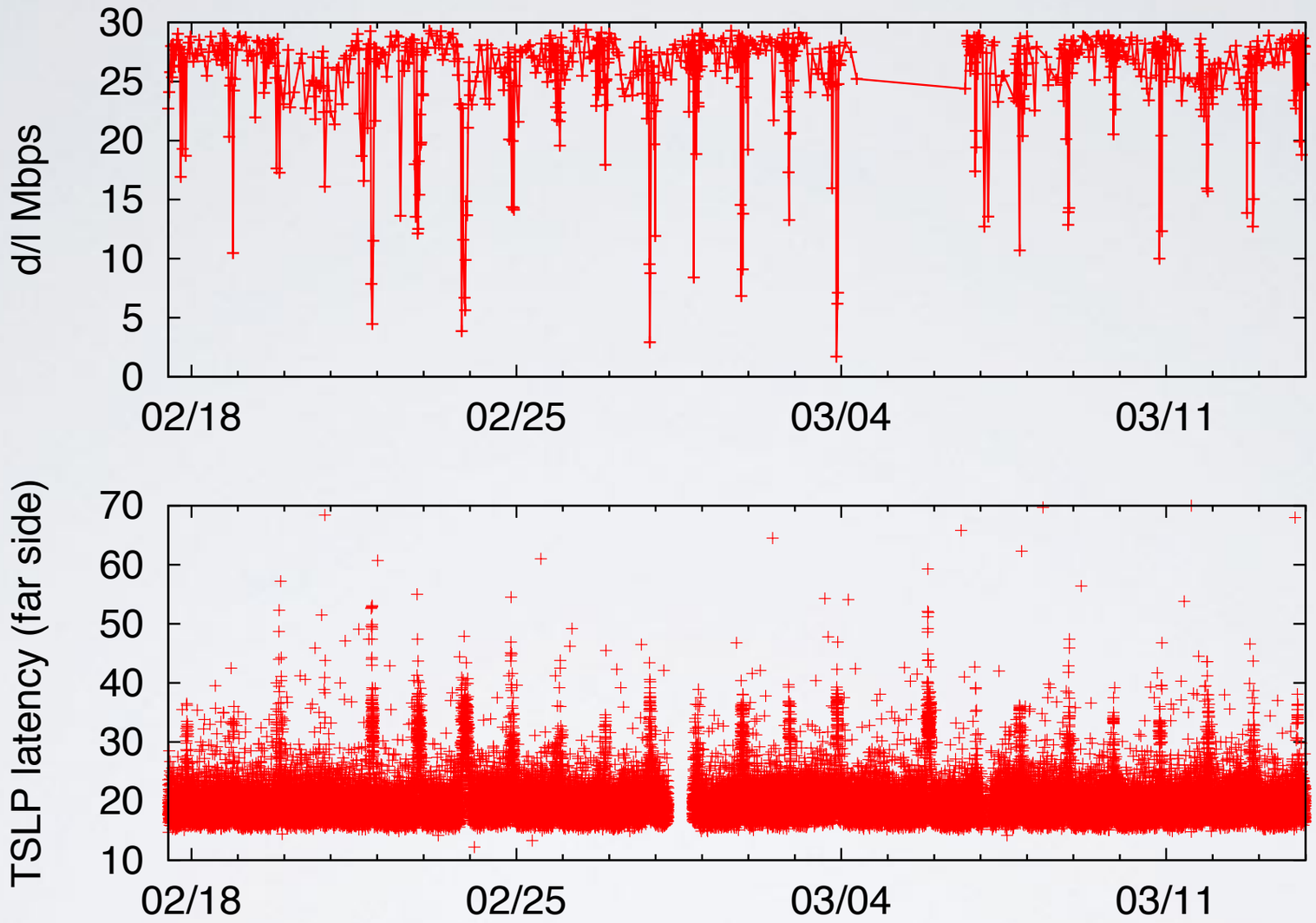


Validating the Method: Step 2



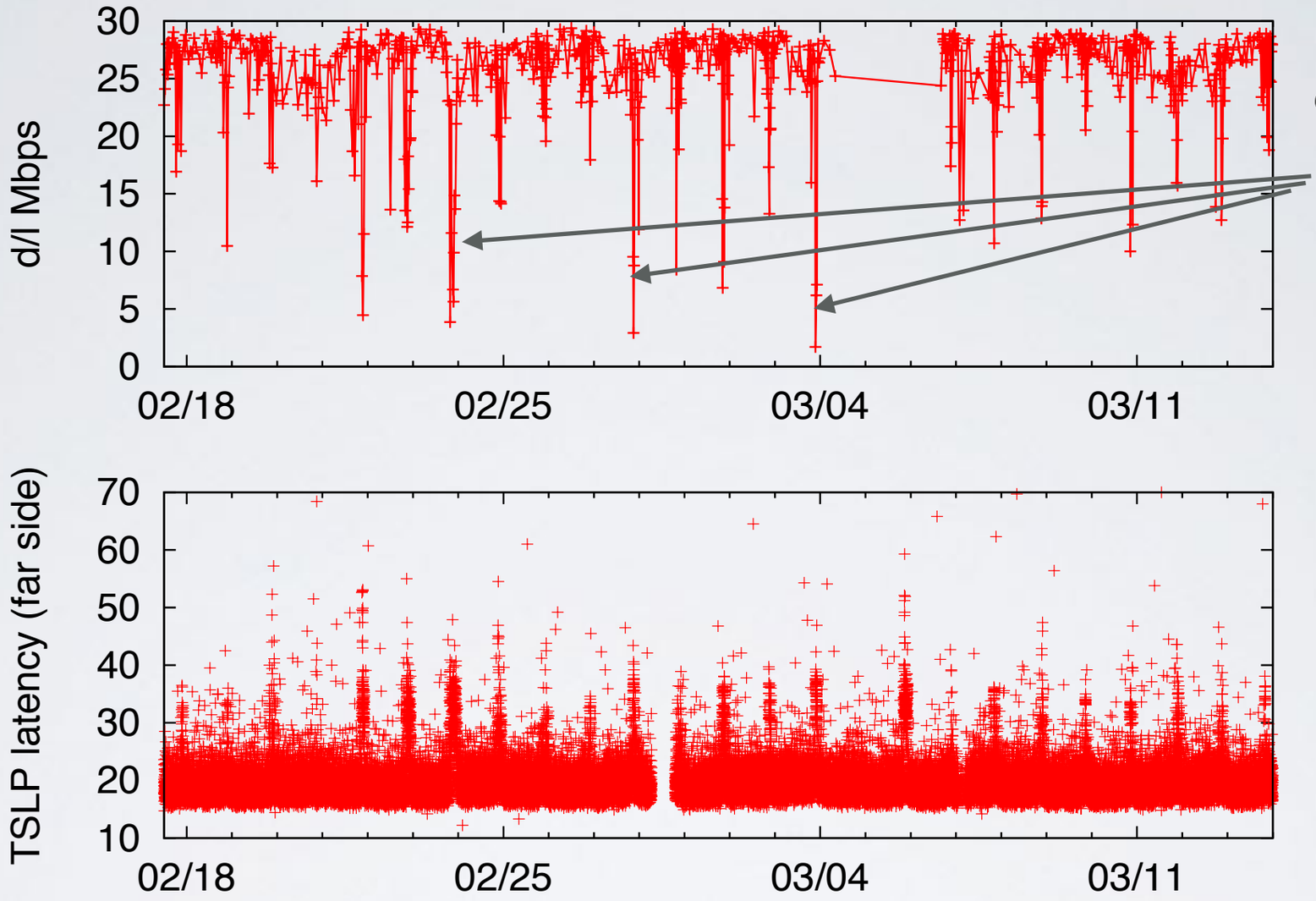
Throughput measurements from Ark VP to M-lab NDT server traversing congested interdomain link

Validation of the Method: Step 2



Strong correlation between throughput and TSLP latency: flows during elevated TSLP latency labeled as "externally" congested

Validation of the Method: Step 2



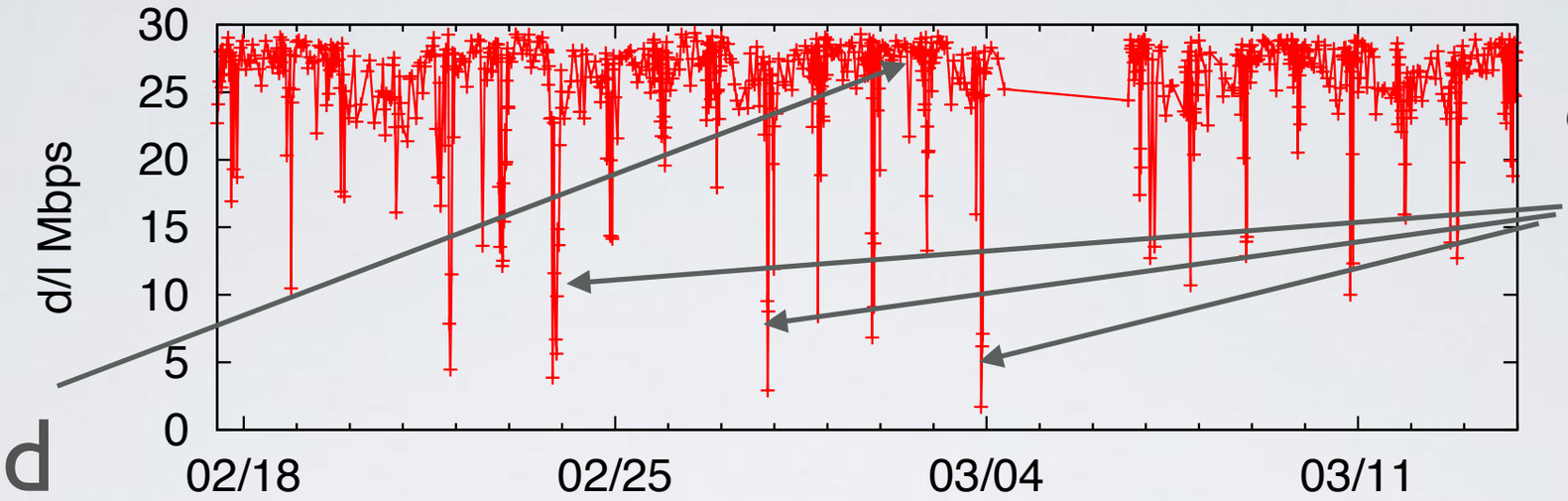
"Externally" congested

Strong correlation between throughput and TSLP latency: flows during elevated TSLP latency labeled as "externally" congested

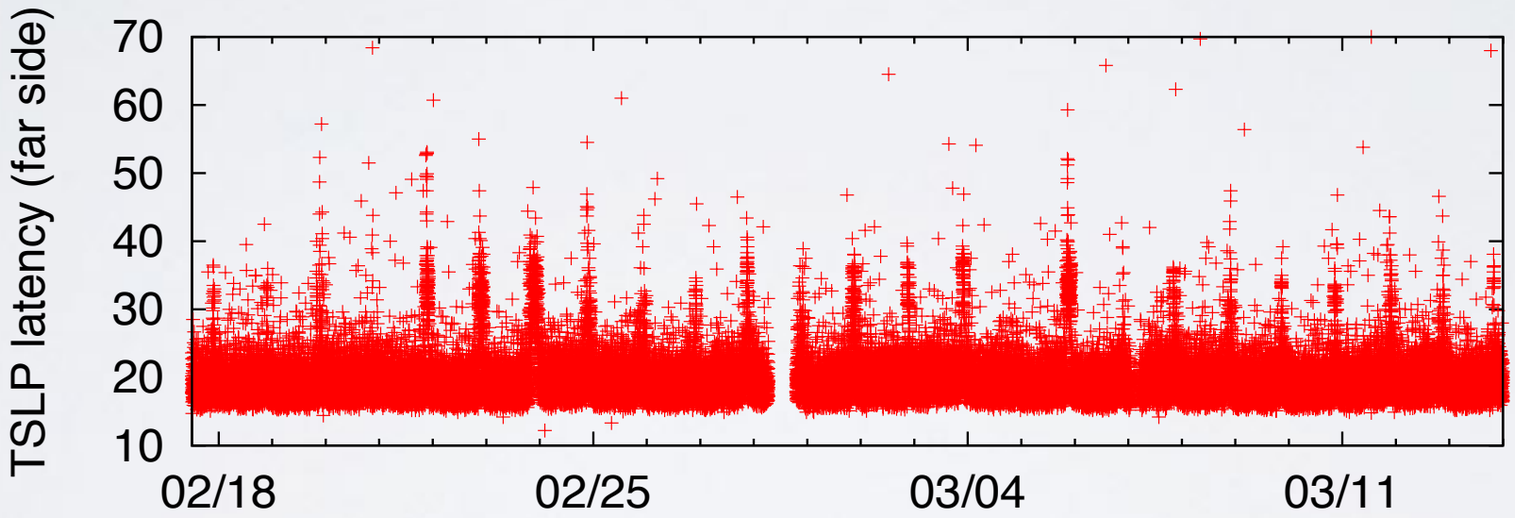
caida

Validation of the Method: Step 2

"self" congested



"Externally" congested

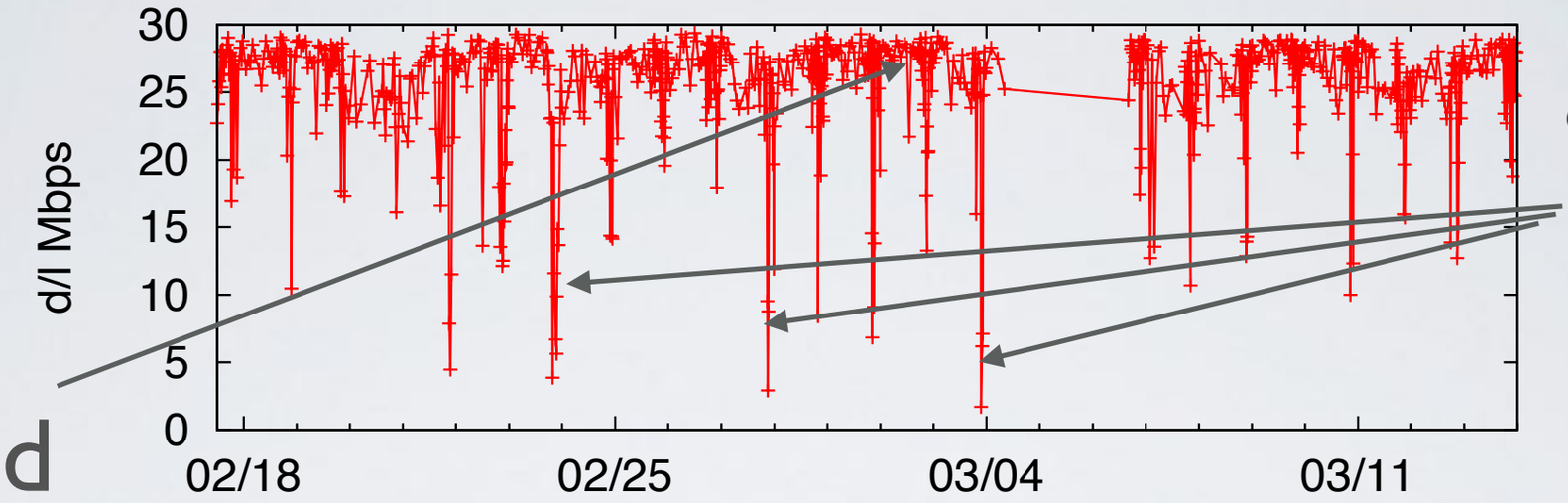


Strong correlation between throughput and TSLP latency: flows during elevated TSLP latency labeled as "externally" congested

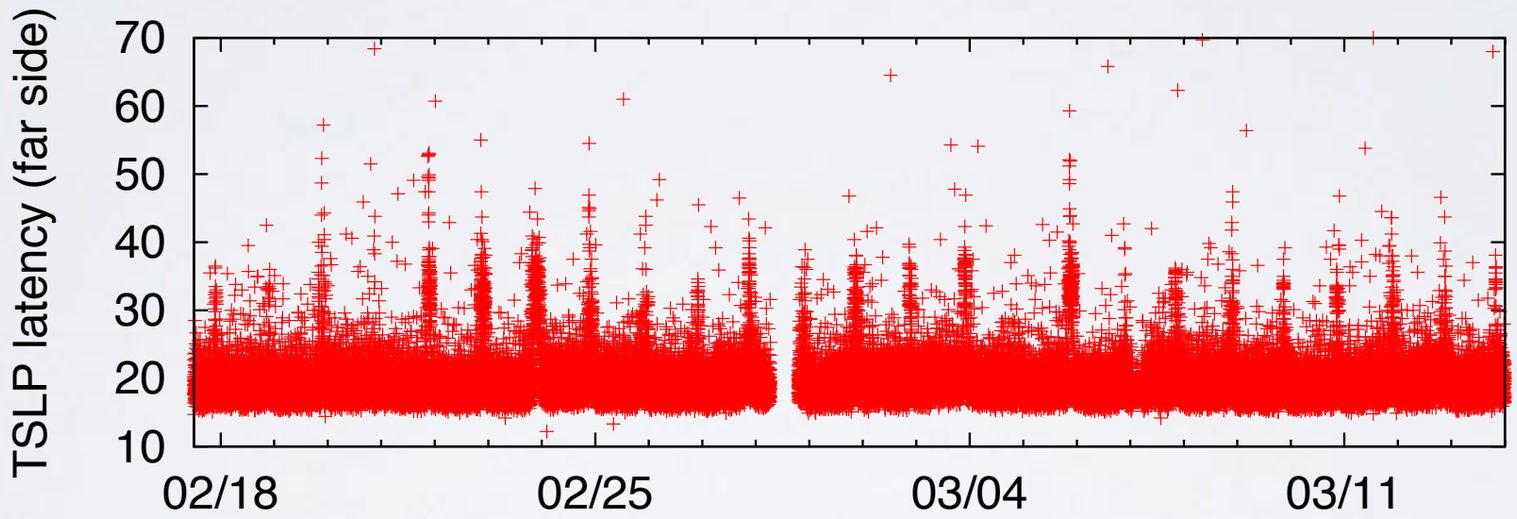
caida

Validation of the Method: Step 2

"self" congested



"Externally" congested

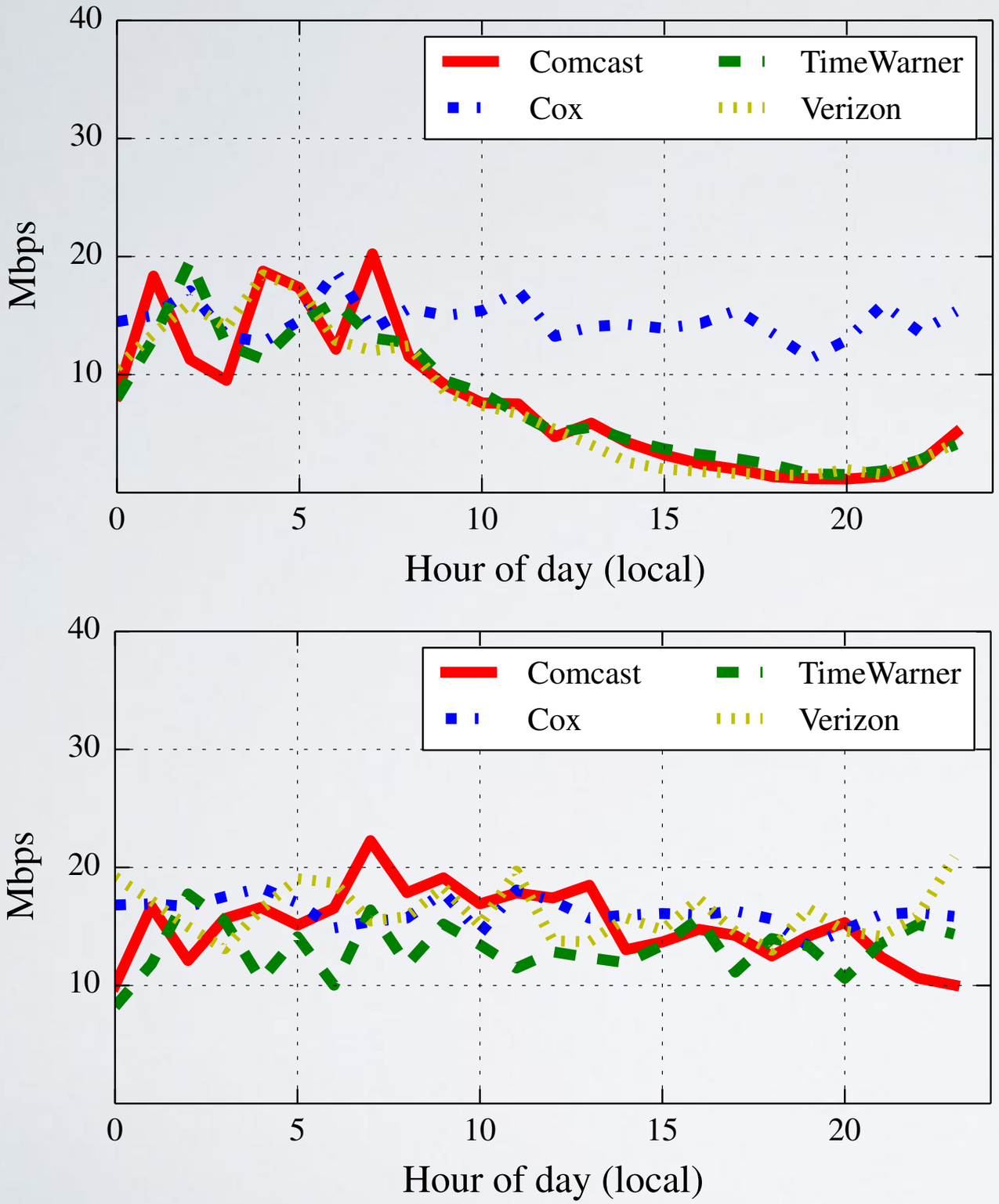


75%+ accuracy in detecting external congestion,
100% accuracy for self-induced congestion

Validation of the Method: Step 3

- M-lab's NDT test data for real-world validation
- Cogent interconnect issue in late 2013/early 2014
  - NDT tests to Cogent M-lab servers from several major U.S. ISPs saw significantly lower throughput during peak hours: Comcast, TWC, Verizon
  - Cox was notably not affected
  - Underlying cause was congested interconnects

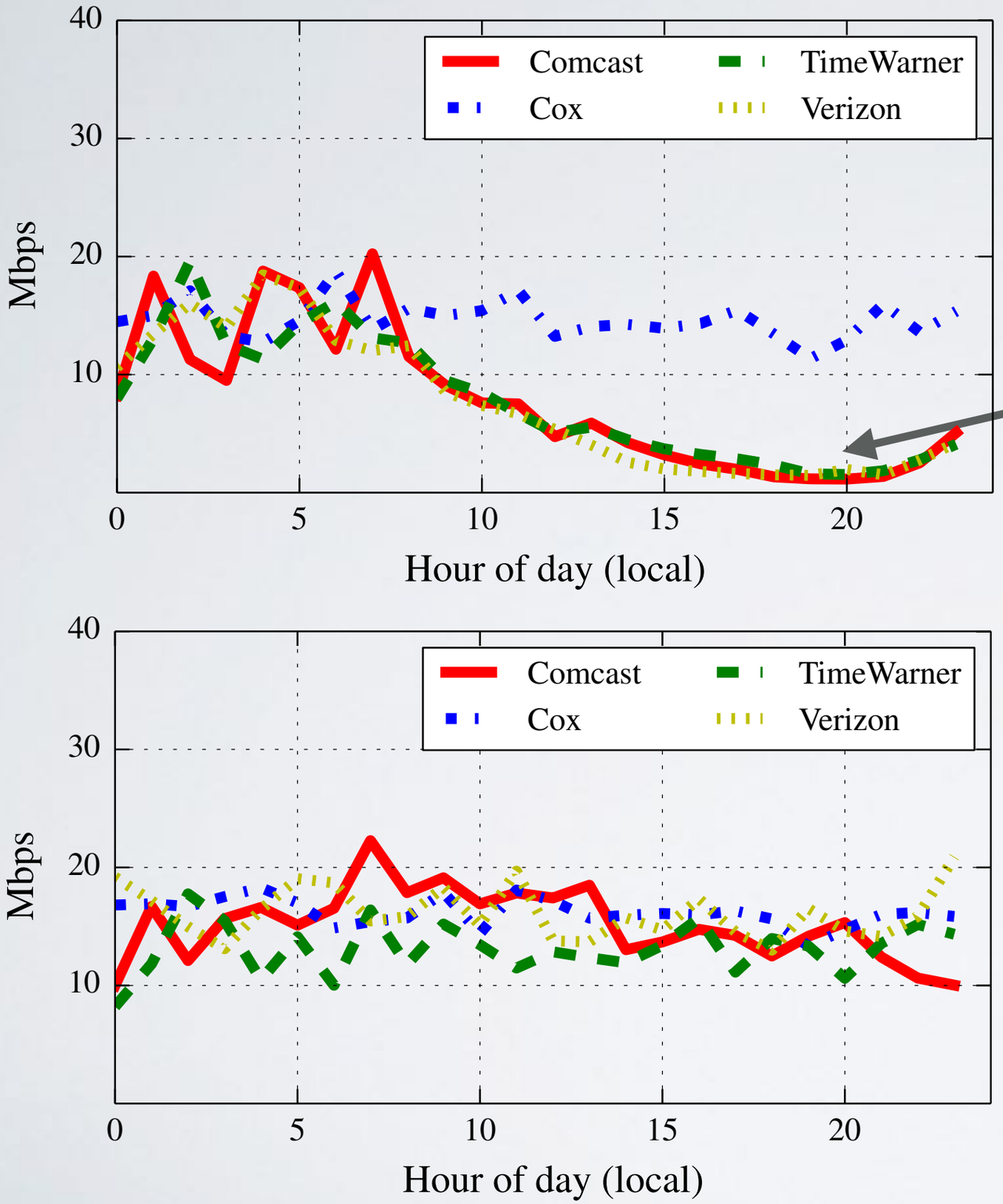
Using the M-lab Data



January 2014

April 2014

Using the M-lab Data

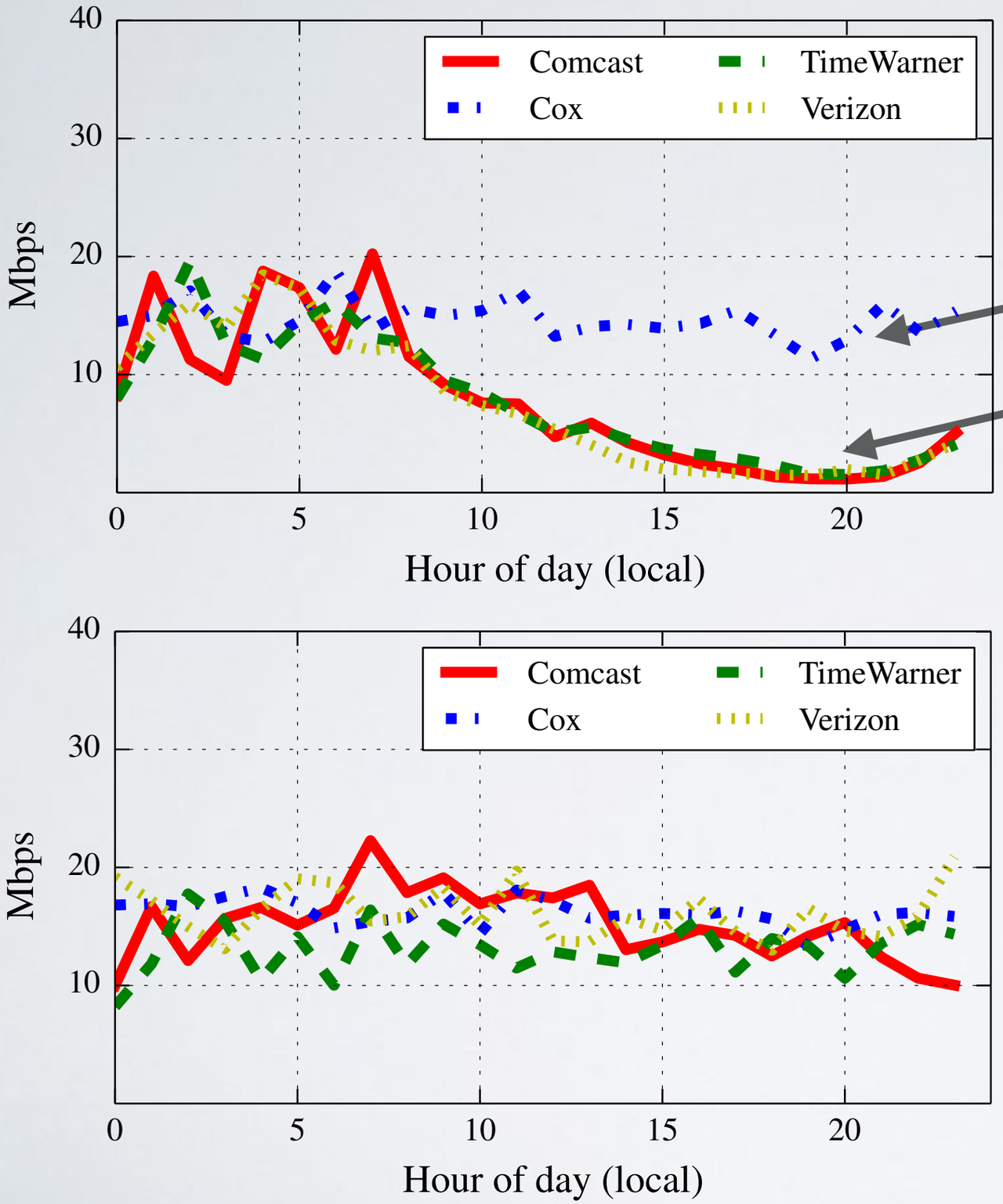


January 2014

Drop in peak-hour throughput for for Comcast, TWC, Verizon

April 2014

Using the M-lab Data



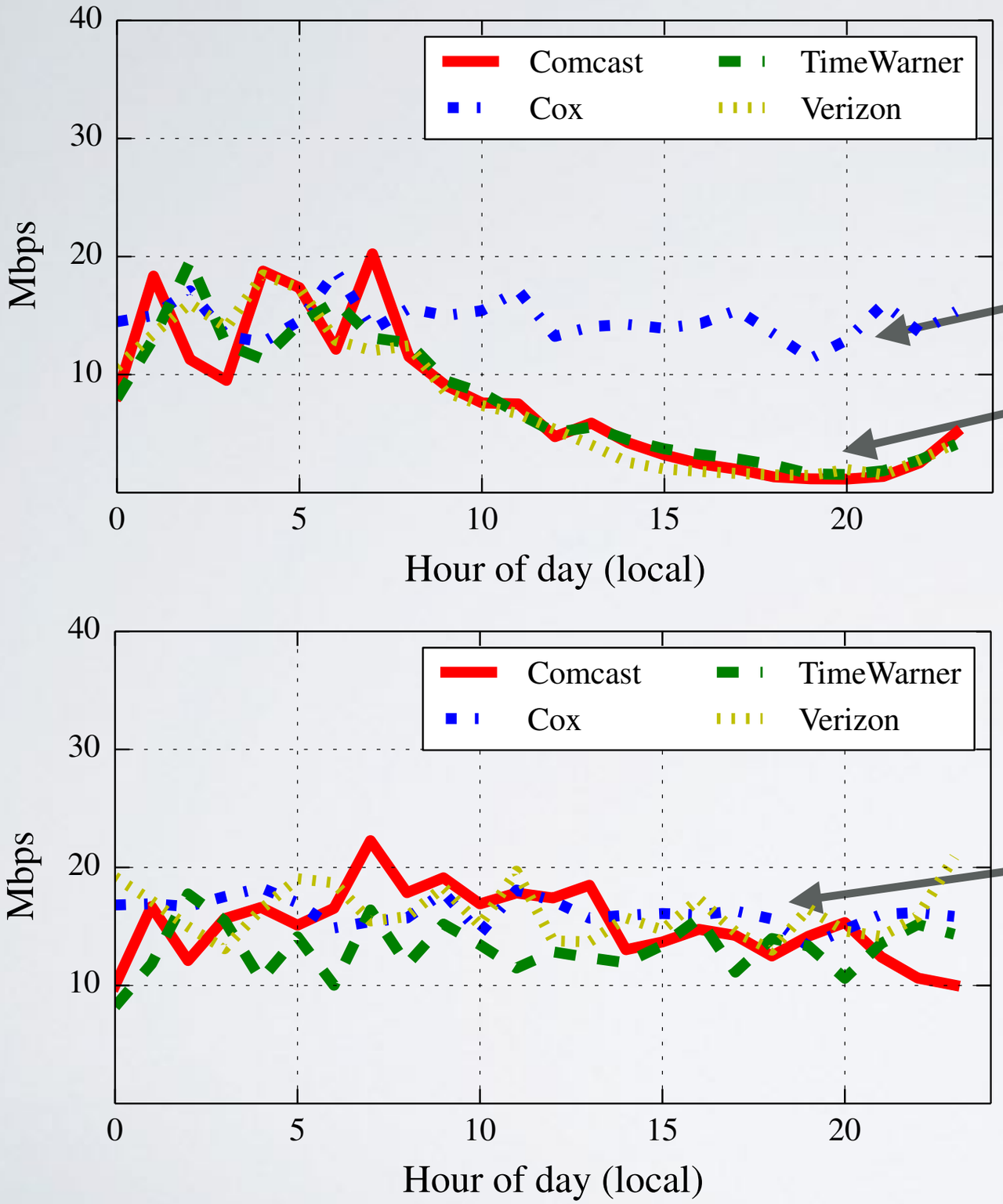
January 2014

Cox not affected

Drop in peak-hour throughput for for Comcast, TWC, Verizon

April 2014

Using the M-lab Data



January 2014

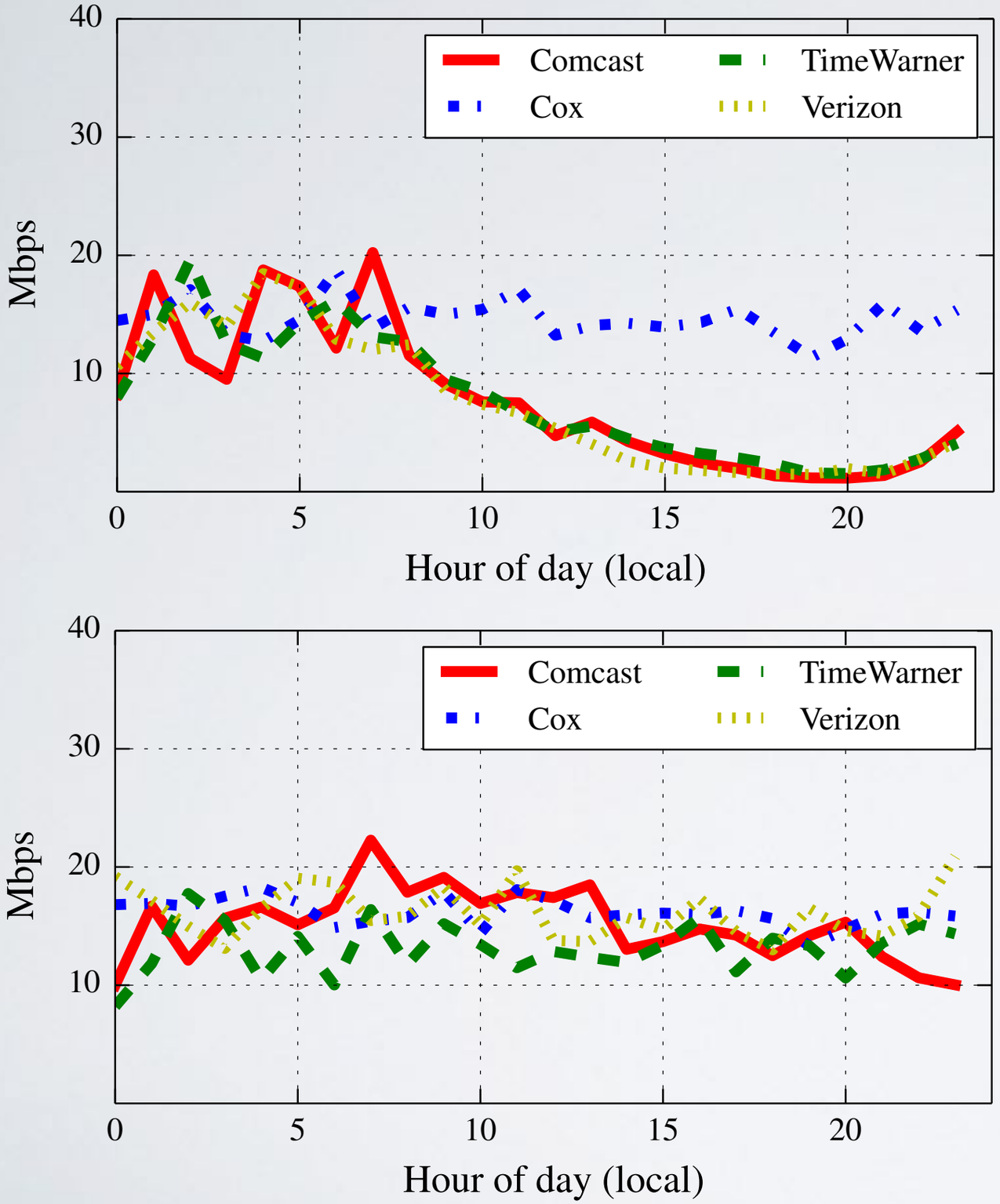
Cox not affected

Drop in peak-hour throughput for for Comcast, TWC, Verizon

April 2014

Interconnection dispute resolved; no diurnal effect

Using the M-lab Data



Peak hour tests in Jan/Feb 2014 are likely "externally" congested

Off-peak tests in Mar/Apr 2014 are likely "self" congested

Noisy Data

Noisy Data

- Difficult to infer interdomain congestion using throughput*

*Sundaresan et al. "Challenges in Inferring Interdomain Congestion using Throughput Measurements", IMC 2017

Noisy Data

- Difficult to infer interdomain congestion using throughput*
- All tests labeled "external" may not have traversed congested interconnects

*Sundaresan et al. "Challenges in Inferring Interdomain Congestion using Throughput Measurements", IMC 2017

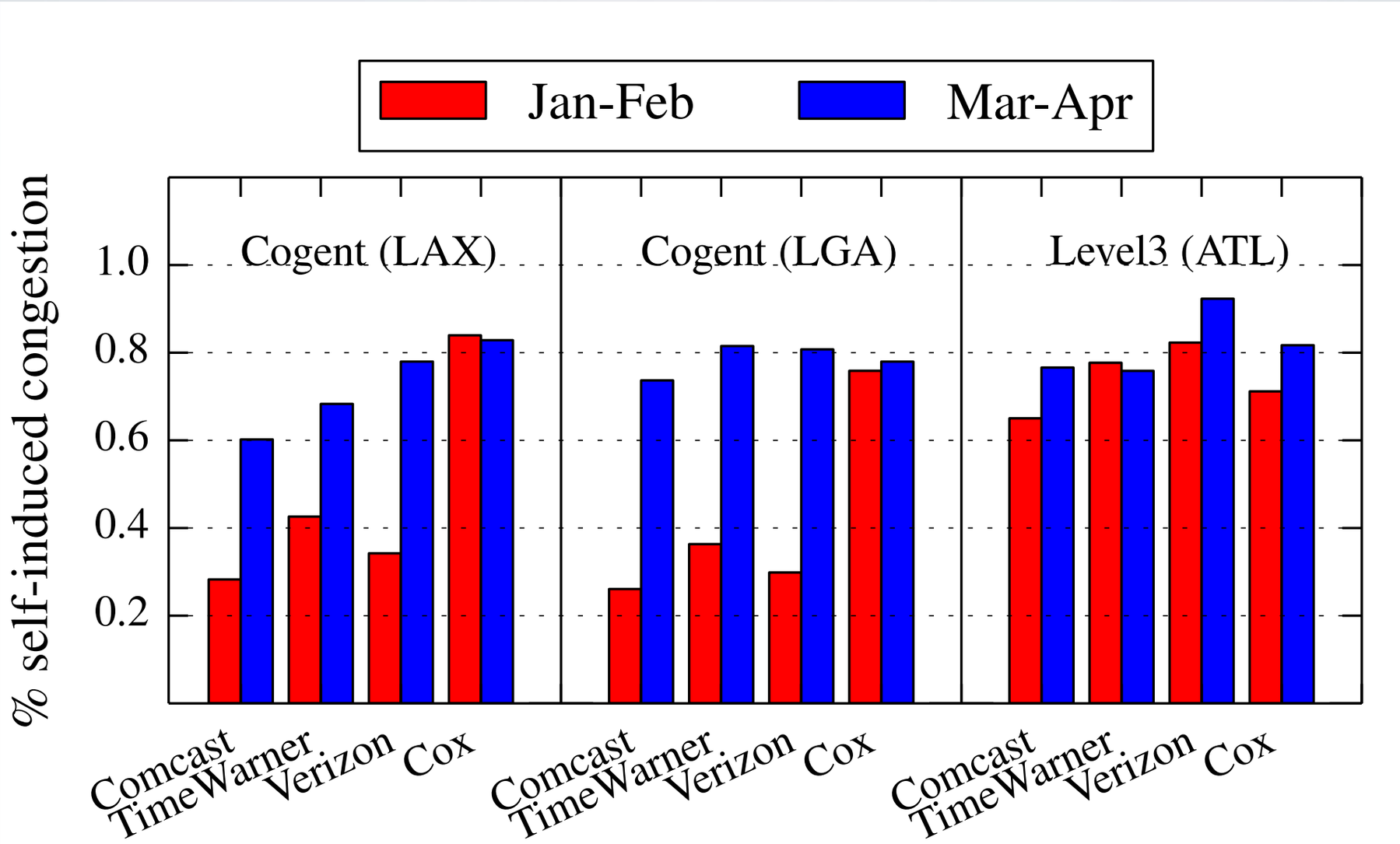
Noisy Data

- Difficult to infer interdomain congestion using throughput*
- All tests labeled "external" may not have traversed congested interconnects
- We do not expect to identify all peak hour tests as externally congested, and vice versa
  - Looking for qualitative differences

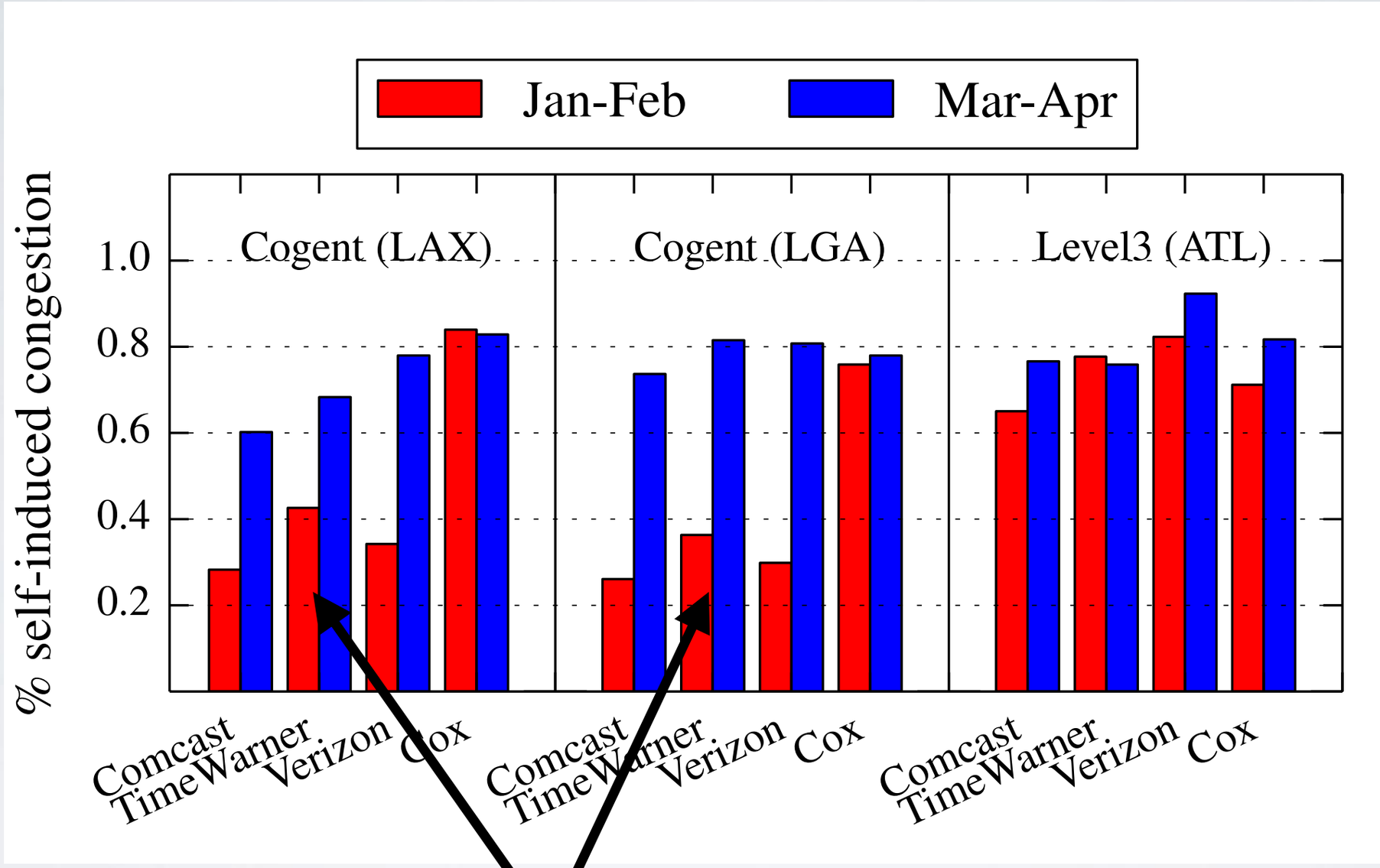
*Sundaresan et al. "Challenges in Inferring Interdomain Congestion using Throughput Measurements", IMC 2017

caida

Applying the Model to M-lab data



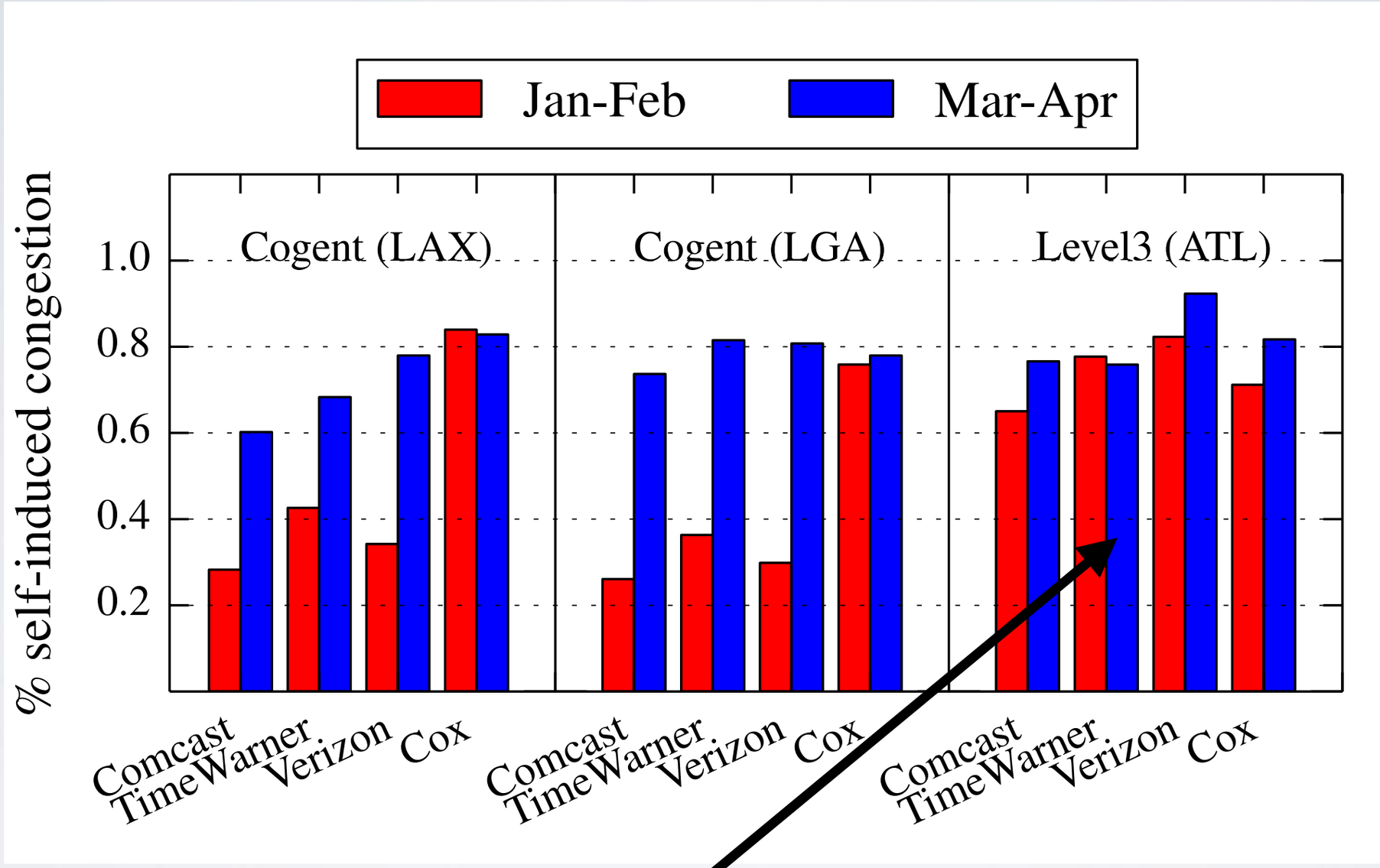
Applying the Model to M-lab data



Much lower incidences of self-induced congestion for Cogent in Jan/Feb 2014 as compared to Mar/Apr

caida

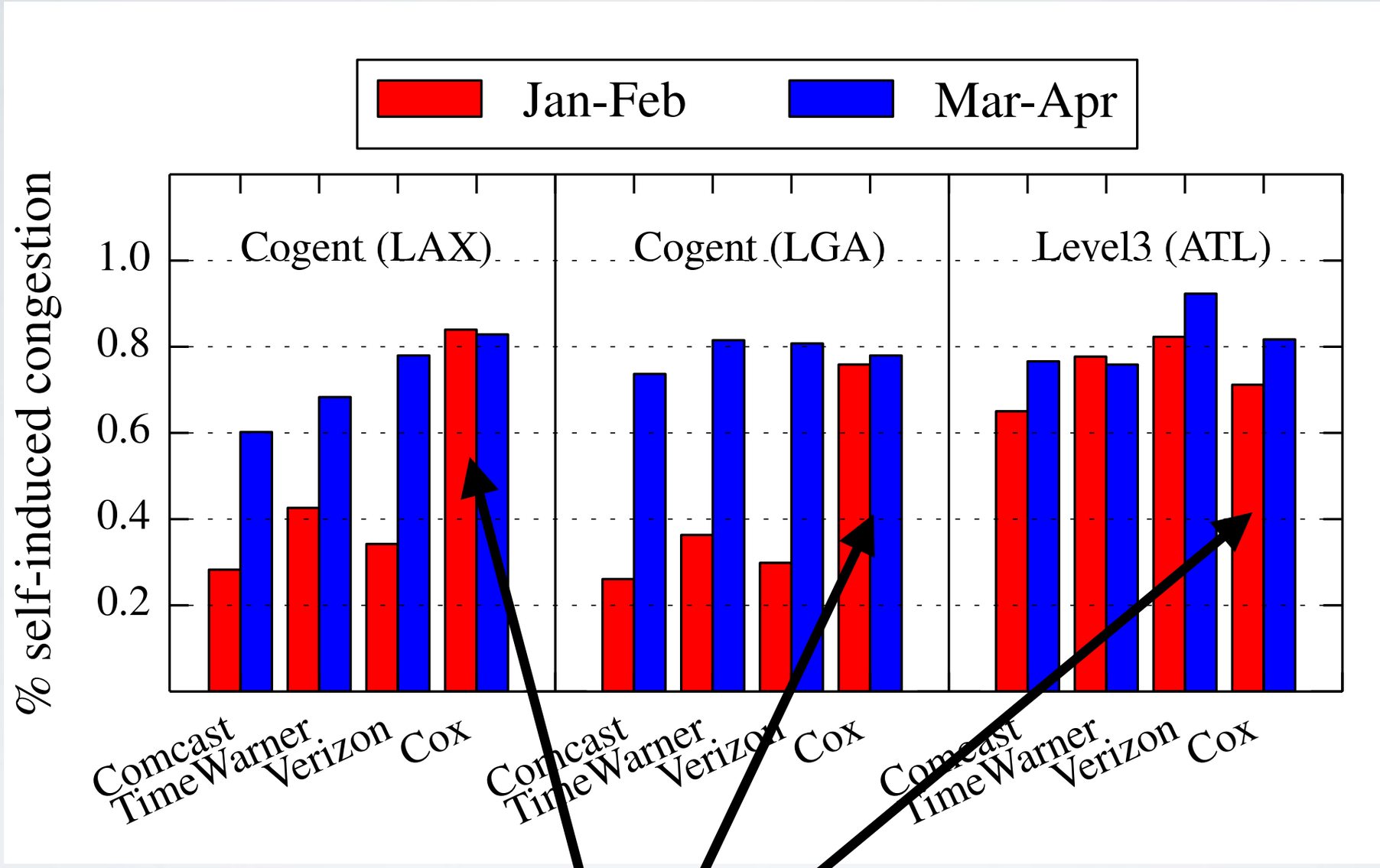
Applying the Model to M-lab data



Level3 does not show significant differences, was not affected by interconnection disputes

caida

Applying the Model to M-lab data

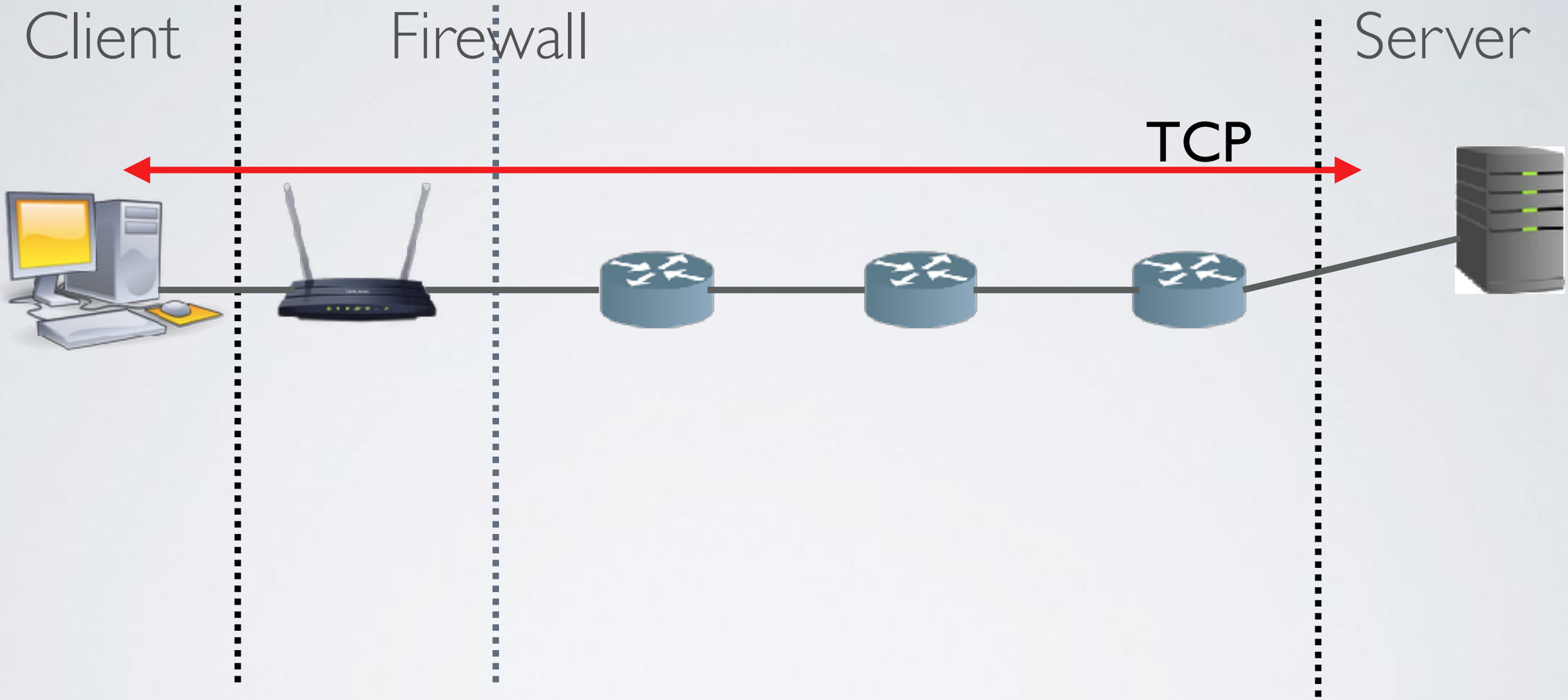


Cox does not show significant differences, was not affected by interconnection disputes

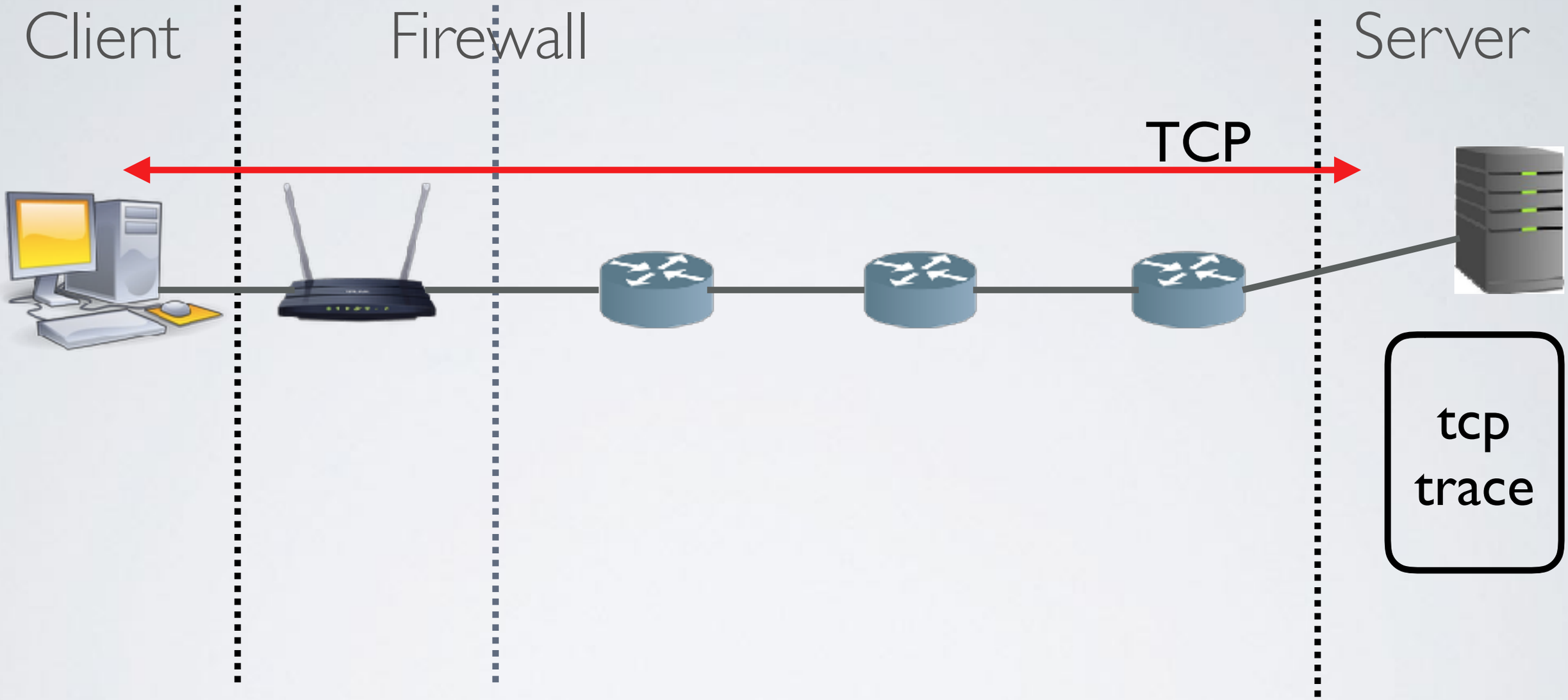
In-band Measurements

- Can we leverage an ongoing TCP connection for path measurements?
- e.g., where is the TCP flow bottlenecked? is the client's wireless access network the bottleneck? What is the capacity/available bandwidth of the path?
- Why in-band? No need to send external flows (which may be treated differently than the application)
- TCP flow has already punched a hole in the NAT at the client side

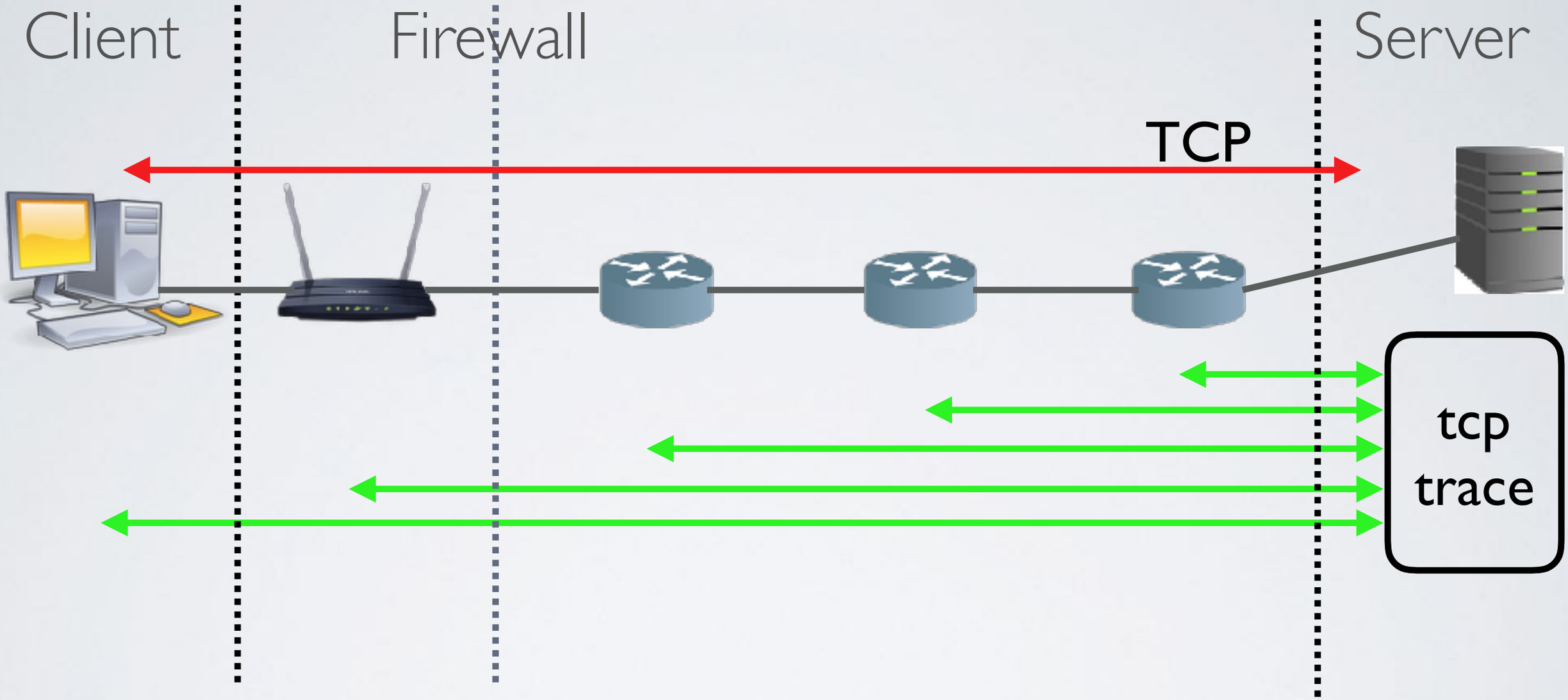
tcptrace



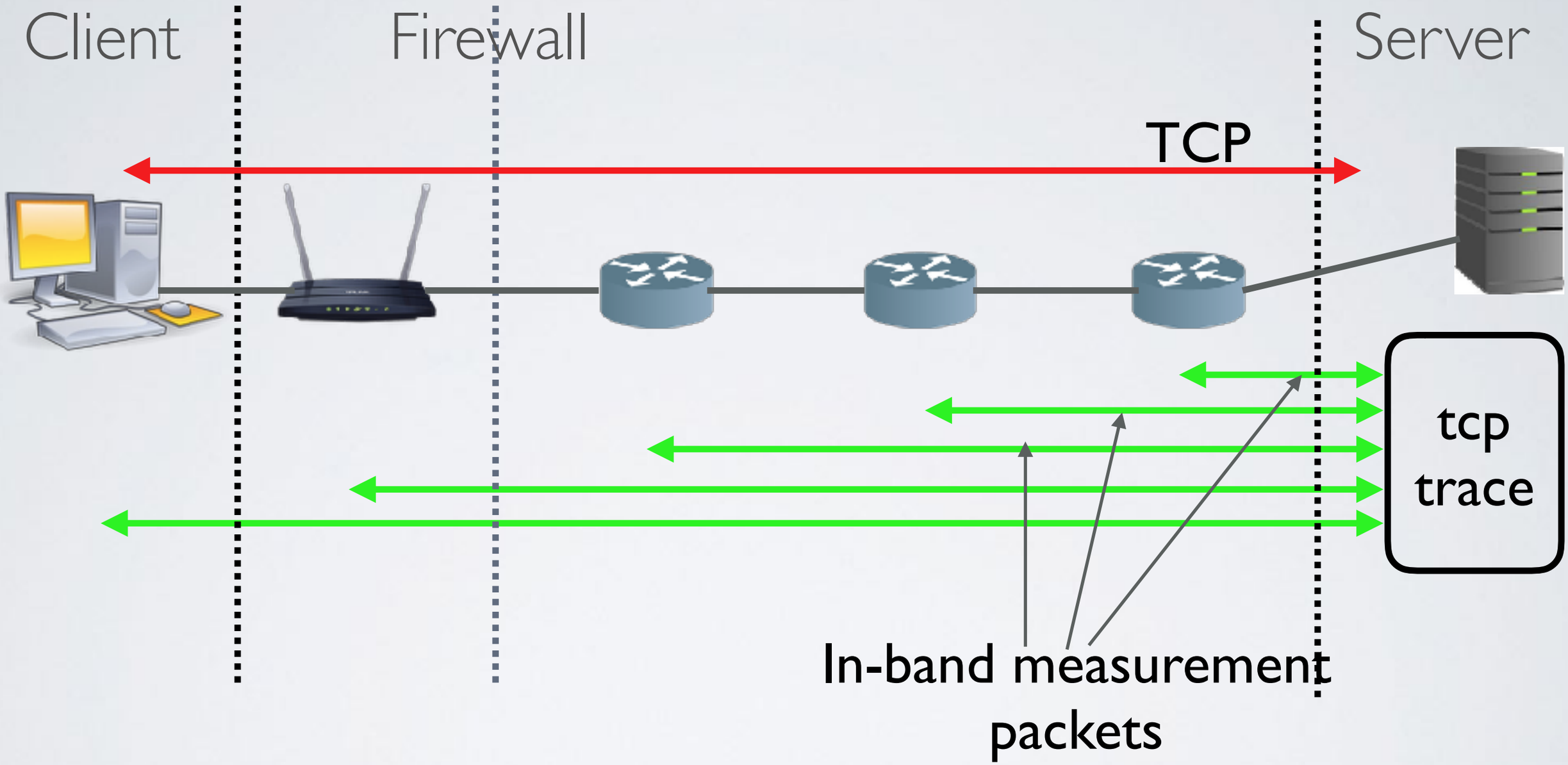
tcptrace



tcptrace



tcptrace


Client
Firewall
Server
TCP
tcp
trace
In-band measurement
packets

Tracetcp

- **In-band high-frequency traceroute:** injecting empty TTL-limited packets in the ongoing TCP flow
- Ability to observe the buffer building up at bottleneck
- Can measure to the client past the NAT, and observe wireless delays
- prototype at: https://github.com/ssundaresan/tracetcp (ask me for access)

Tracetcp

(more measurements in the works)

- Packet-pair and packet-train techniques to measure per-hop capacity and available bandwidth (in-band pathneck)
- Per-hop loss rate
- Main challenge: how to utilize packets from the TCP stream, and smartly insert measurement packets without affecting the ongoing flow

Thanks!
Questions?

amogh@caida.org