Modeling and Emulation of Internet Paths

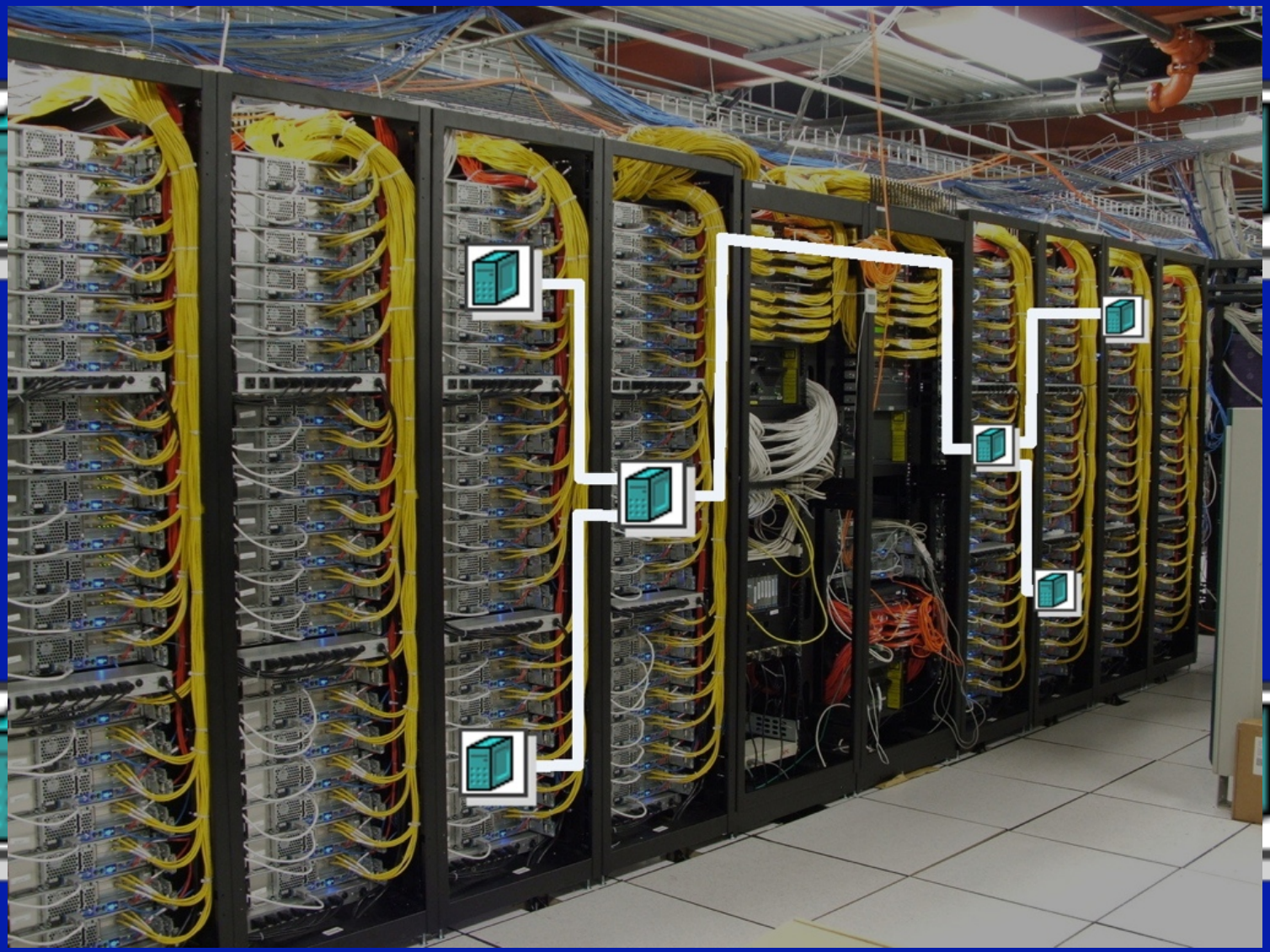
Pramod Sanaga, Jonathon Duerig,
Robert Ricci, Jay Lepreau

University of Utah

Distributed Systems

- How will you evaluate your distributed system?
  - DHT
  - P2P
  - Content Distribution

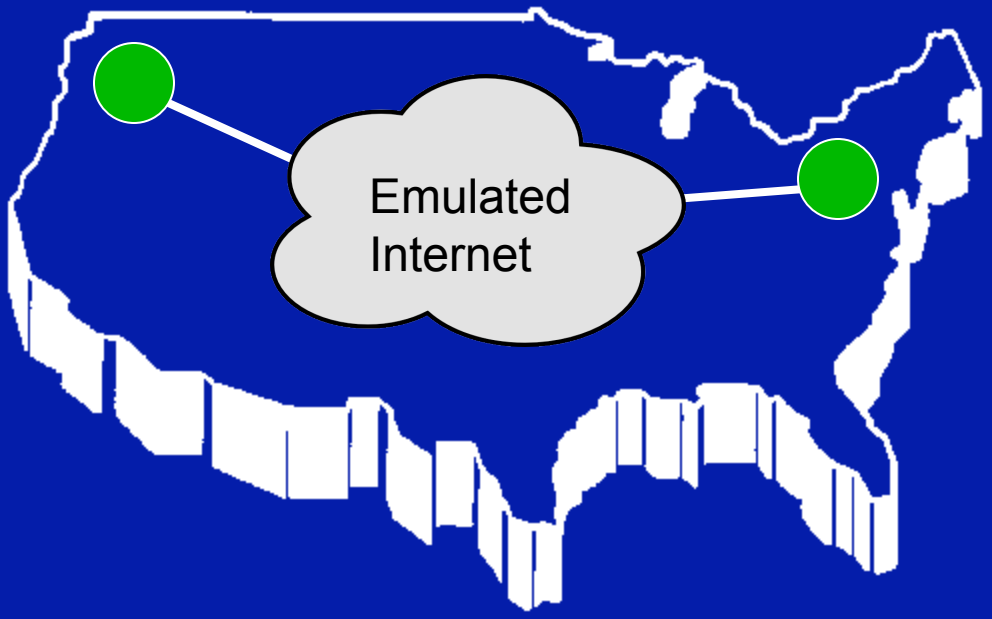
Network Emulation



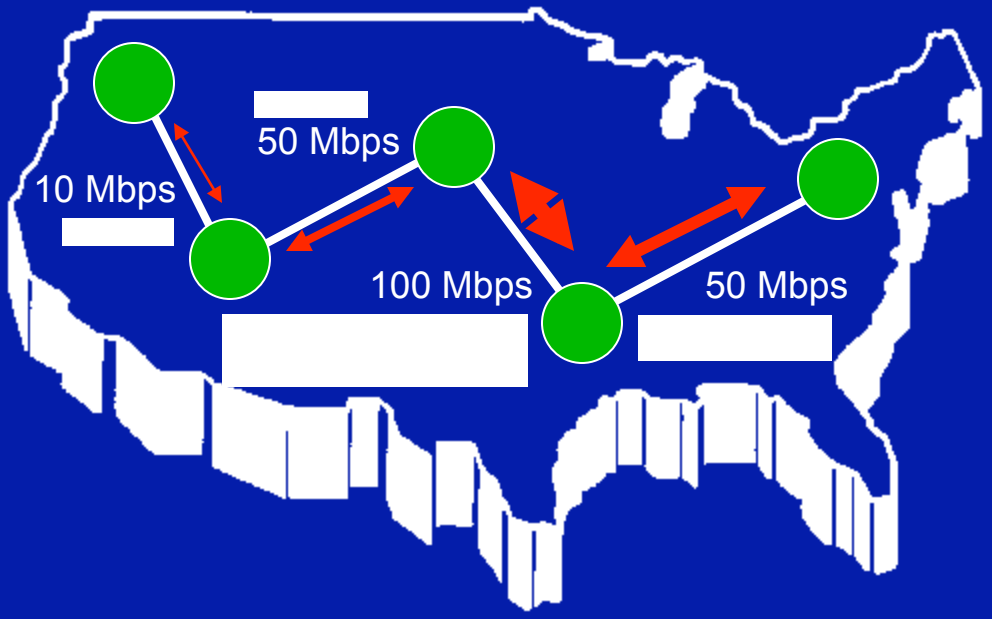
Why Emulation?

- Test distributed systems
  - + Repeatable
  - + Real
    - PCs, Applications, Protocols
  - + Controlled
    - Dedicated Nodes, Network Parameters
- Link Emulation

Goal: Path Emulation



Link by Link Emulation

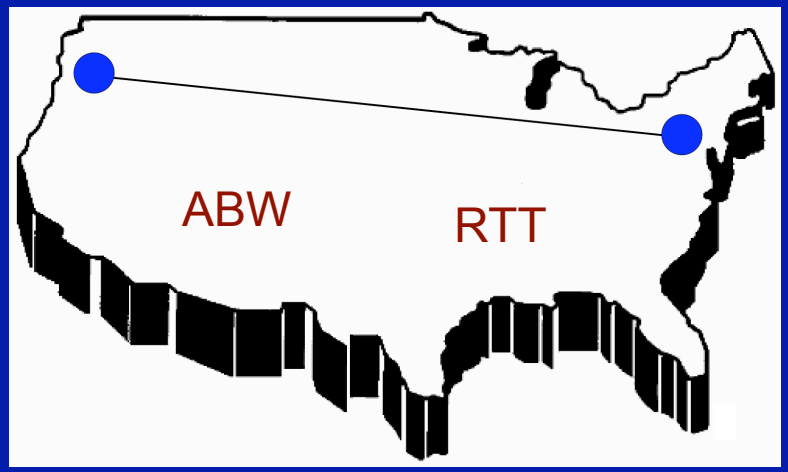


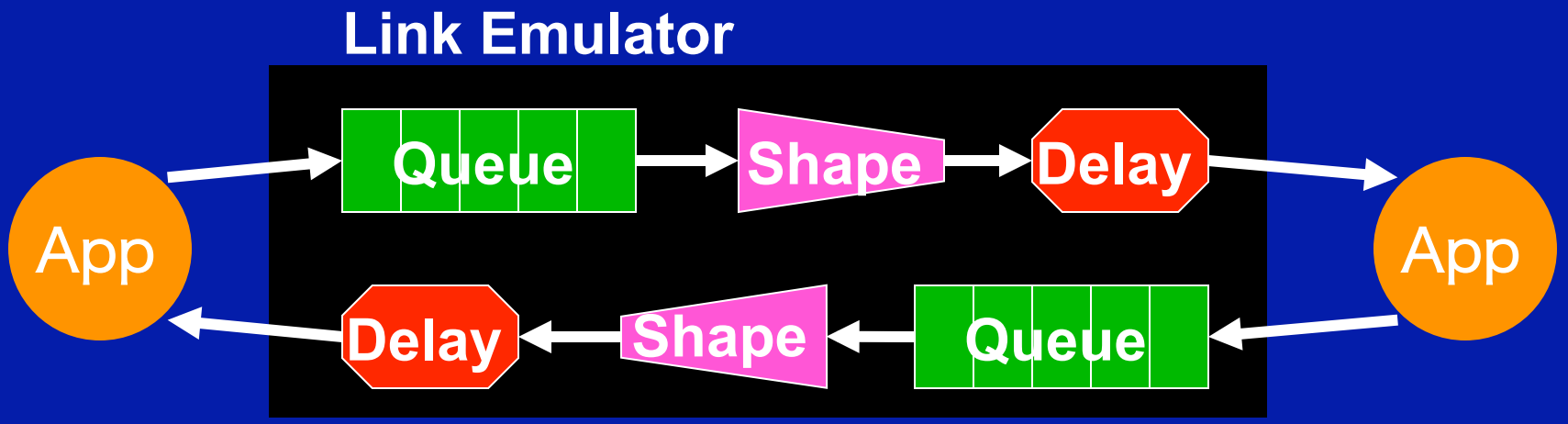
End to End Emulation

Contributions

- Principles for path emulation
  - Pick appropriate queue sizes
  - Separate capacity from ABW
  - Model reactivity as a function of flows
  - Model shared bottlenecks

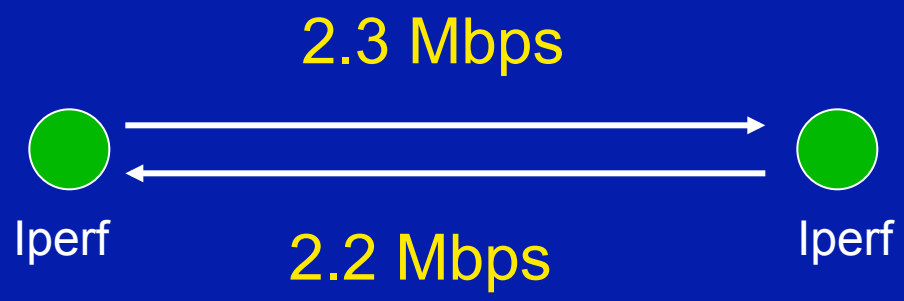
Obvious Solution





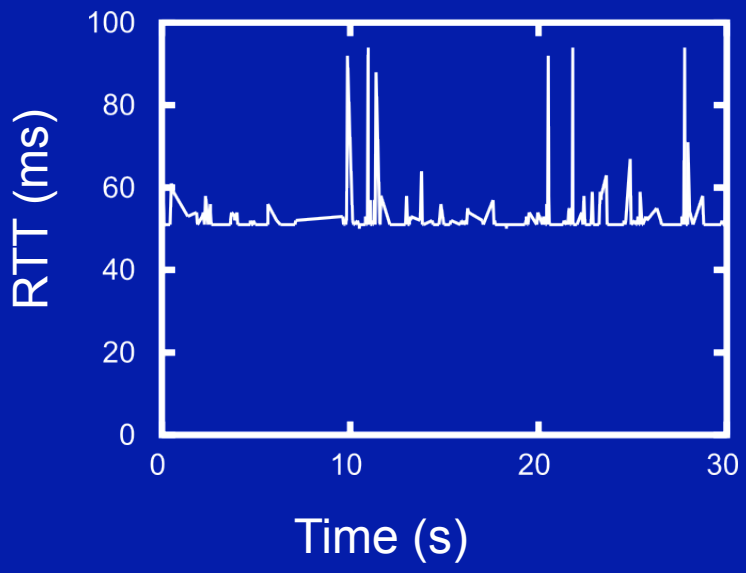
Obvious Solution (Good News)

- Actual Path
- Bandwidth Accuracy
  - 8.0% Error on forward path
  - 7.2% Error on reverse path

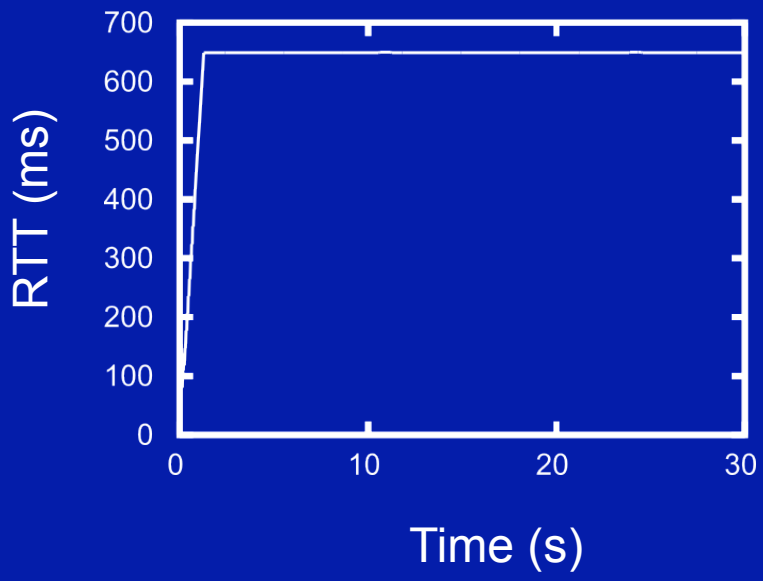


Obvious Solution (Bad News)

Real Path



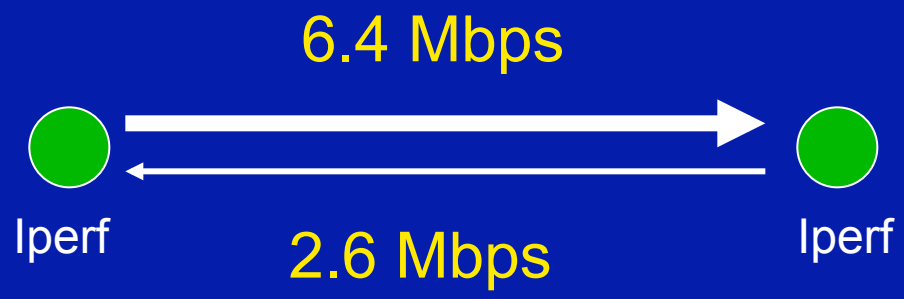
Obvious Emulation



Latency is an order of magnitude higher

Obvious Solution (More Problems)

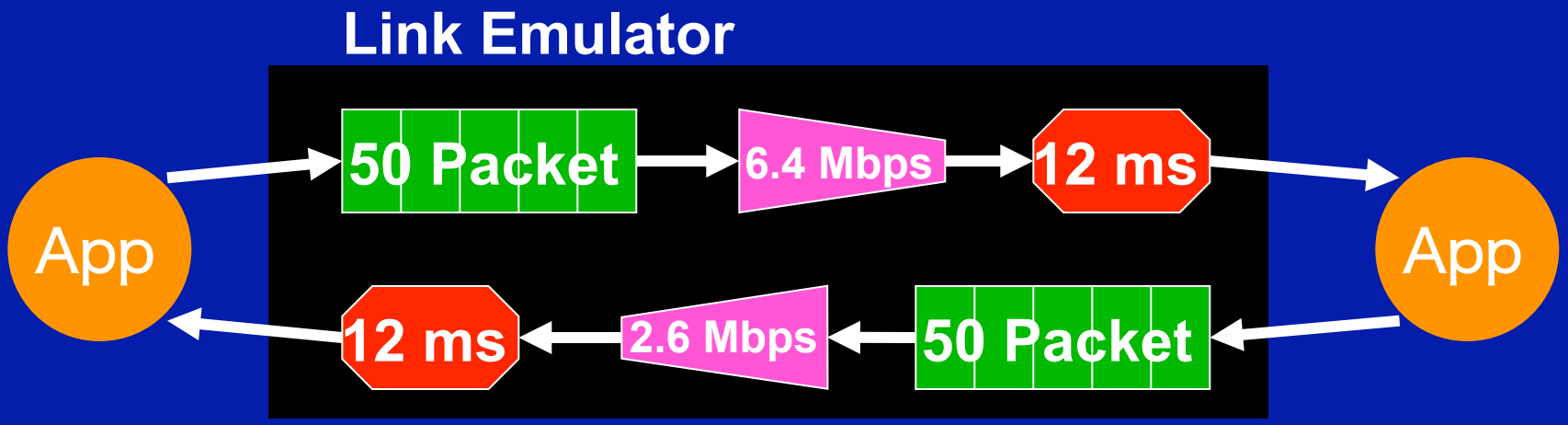
- Measure asymmetric path
- Bandwidth Accuracy
  - 50.6% error on forward path!
    - Much smaller than on real path
  - 8.5% error on reverse path



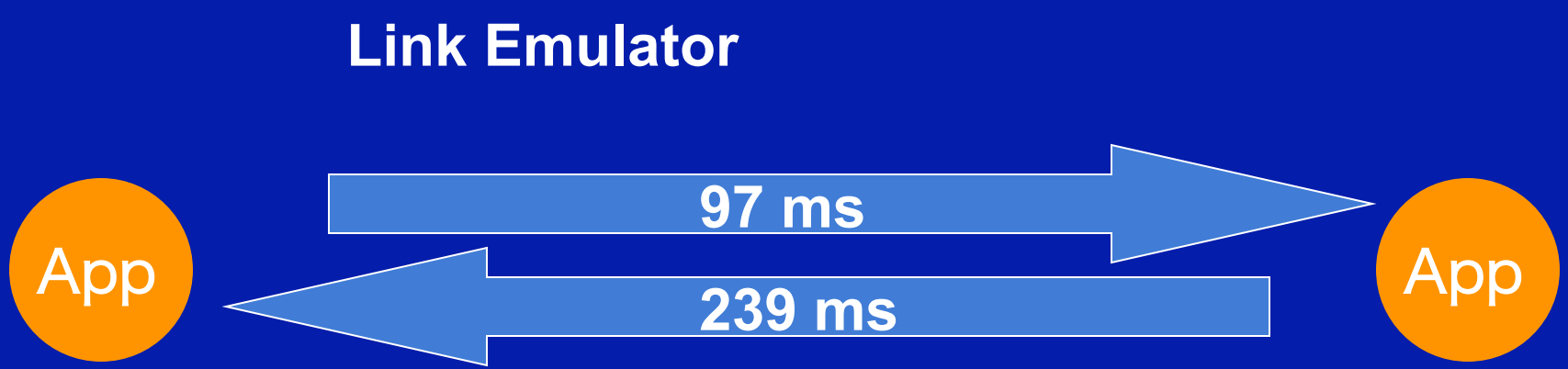
What's Happening?

- TCP increases congestion window until it sees a loss
- There are no losses until the queue fills up
- Queue fills up
- Delays grows until queue is full
- Large delays were queuing delays

What's Happening



Delays



How do we make this more rigorous?

Maximum Tolerable Queueing Delay

$$RTT_{\max} = \frac{w_{\max}}{ABW} - RTT_{base}$$

Max Tolerable Queueing Delay = Max Window Size / Target Bandwidth − Other RTT

Queue size cannot be limited above by delay

Upper Bound (Queue Size)

$$q_f + q_r \leq C \bullet RTT_{\max}$$

Total Queue Size

Capacity

Max Tolerable Delay

Upper limit is proportional to capacity

Upper bound

- Forward direction 13k
  - 9 packets
- Reverse direction 5k
  - 3 packets
- Queue size is too small
  - Drops even small bursts!

Lower Bound (Queue Size)

$$q > \sum_{f \in F} w_f$$

Total Window Size

Small Queues Drop Packets On Bursts

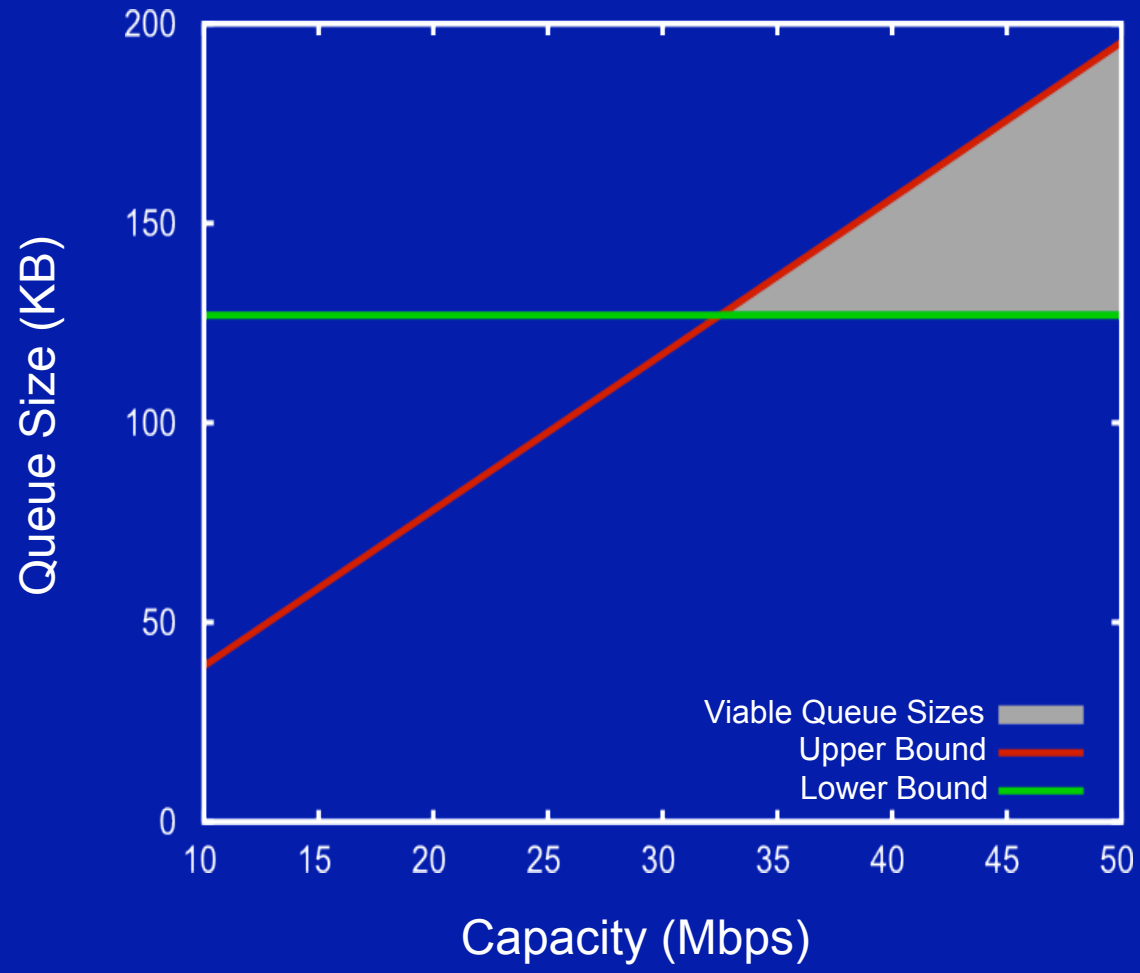
Can’t Fulfill Both Bounds

- Upper limit is 13k
- Lower limit is 65k
  - 1 TCP connection, no window scaling
- No viable queue size

Capacity Vs. ABW

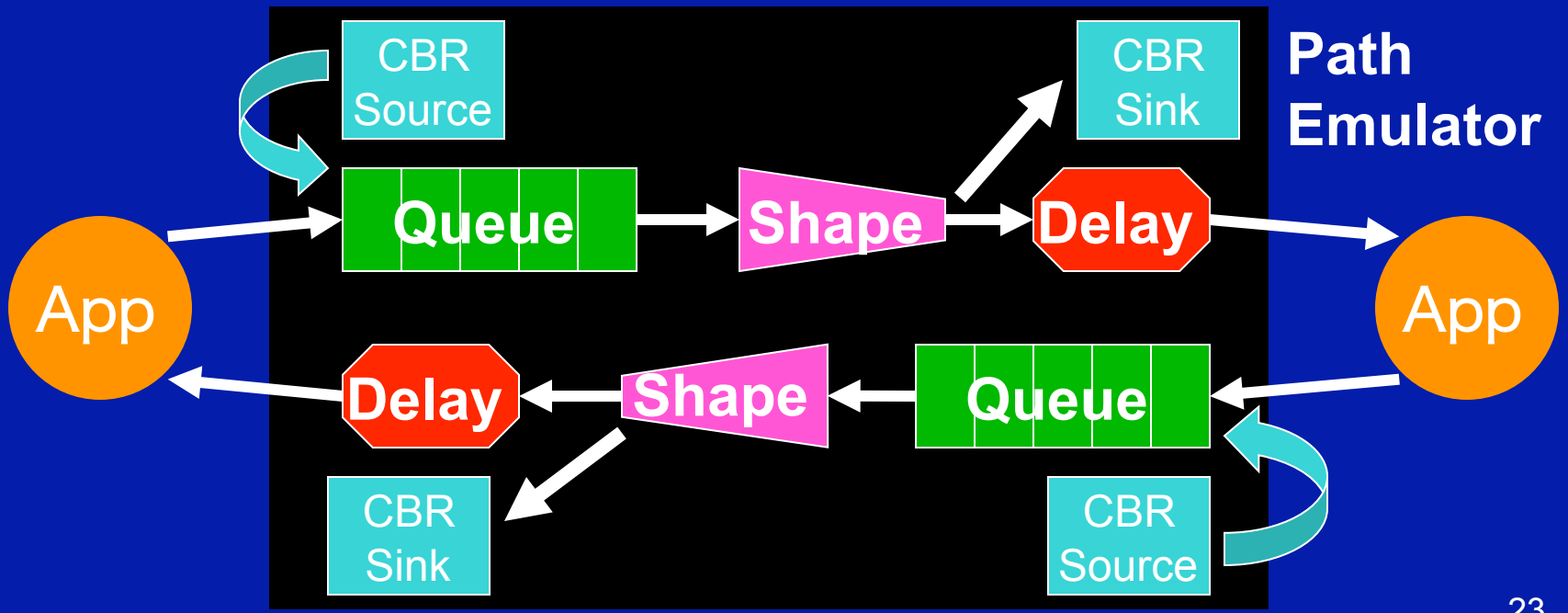
- Capacity is the rate at which everyone's packets drain from the queue
- ABW is the rate at which MY packets drain from the queue
- Link emulator replicates capacity
- Setting capacity == ABW interacts with queue size

Capacity and Queue Size



Our Solution

- Set queue based on constraints
- Set shaper to high bandwidth
- Introduce CBR cross-traffic



CBR Traffic

- TCP cross-traffic backs off
- CBR does not
- Background traffic cannot back off
  - If it does, the user will see larger ABW than they set

Reactivity

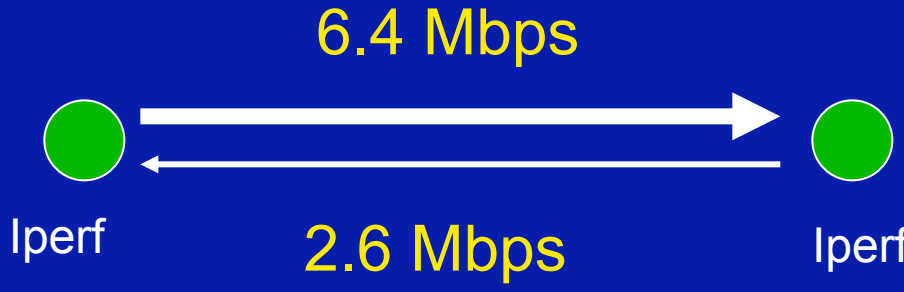
- Reactive CBR traffic?!?!?!
- Approximate aggregate ABW as function of number of foreground flows
- Change CBR traffic based on flow count

$$ABW = f(|F|)$$

Does it work?

Testing Bandwidth

	Obvious Error	Our Error
Forward	50.6 %	4.1 %
Reverse	8.5 %	5.0 %

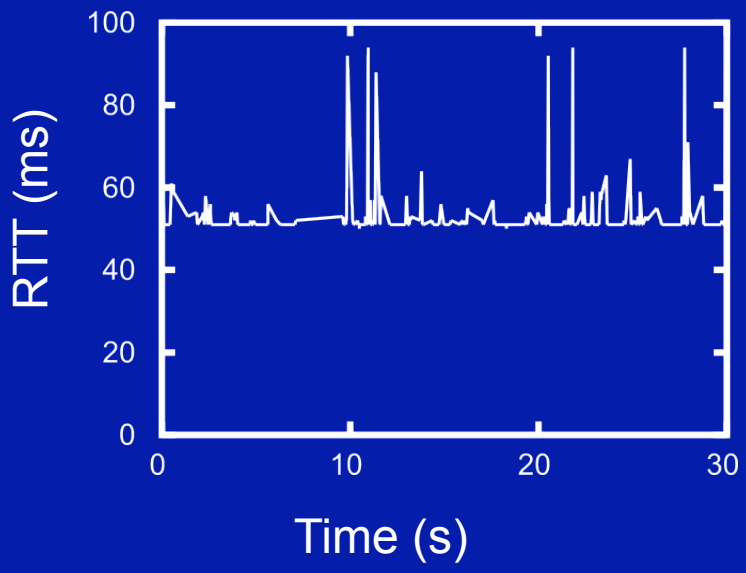


More Bandwidth Tests

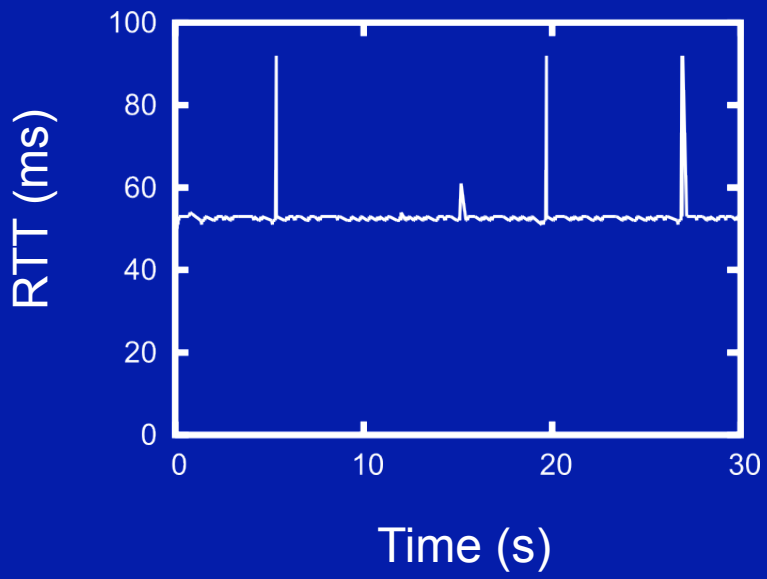
Forward	Reverse	Link Error	Path Error
2.3 Mbps	2.2 Mbps	8.0 %	2.1 %
4.1 Mbps	2.8 Mbps	31.7 %	5.8 %
6.4 Mbps	2.6 Mbps	50.6 %	4.1 %
25.9 Mbps	17.2 Mbps	20.4 %	10.2 %
8.0 Mbps	8.0 Mbps	22.0 %	6.3 %
12.0 Mbps	12.0 Mbps	21.5 %	6.5 %
10.0 Mbps	3.0 Mbps	66.5 %	8.5 %

Testing Delay

Real Path



Path Emulator

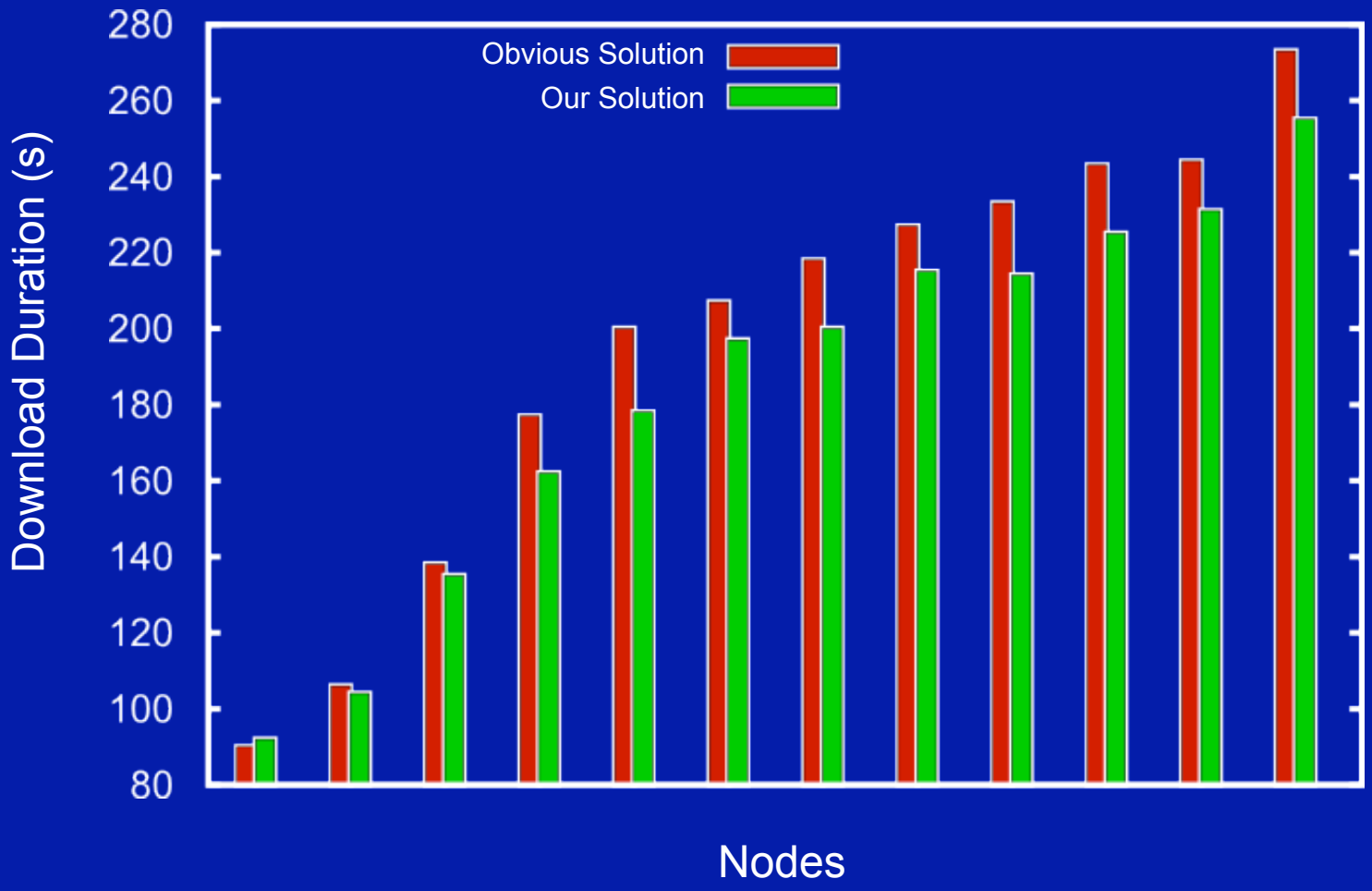


Obvious solution was an order of magnitude higher

BitTorrent Setup

- Measured conditions among 13 PlanetLab hosts
- 12 BitTorrent Clients, 1 Seed
- Isolate capacity and queue size changes

BitTorrent


Obvious Solution
Our Solution
Download Duration (s)
280
260
240
220
200
180
160
140
120
100
80
Nodes

Related Work

- Link Emulation
  - Emulab
  - Dummynet
  - ModelNet
  - NIST Net

Related Work

- Queue Sizes
  - Apenzeller et al (Sigcomm 2004)
    - Large number of flows
      - Buffer requirements are small
    - Small number of flows
      - Queue size should be bandwidth-delay product
      - We build on this work to determine our lower bound
  - We focus on emulating a given bandwidth rather than maximizing performance

Related Work

- Characterize traffic through a particular link
  - Harpoon (IMC 2004)
  - Swing (Sigcomm 2006)
  - Tmix (CCR 2006)
- We use only end to end measurements and characterize reactivity as a function of flows

Conclusion

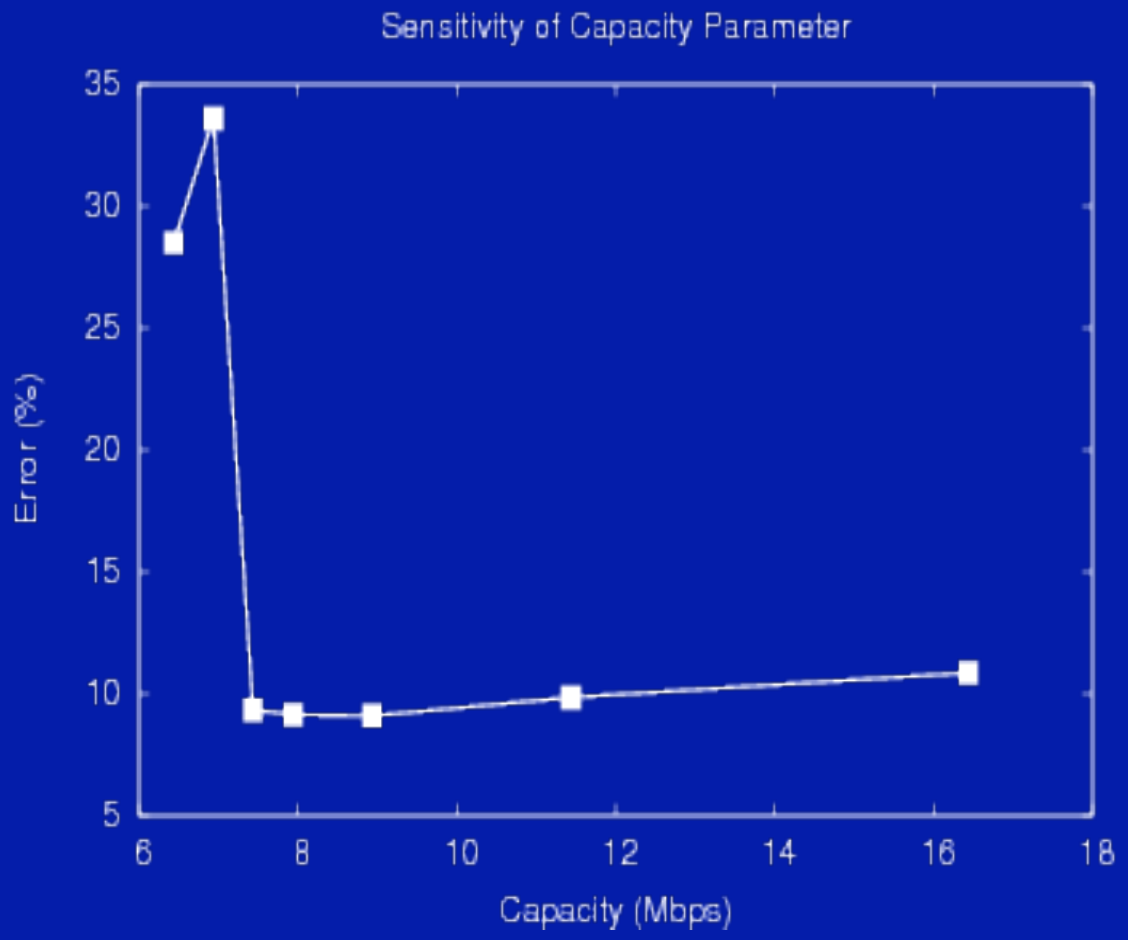
- New path emulator
  - End to end conditions
- Four principles combine for accuracy
  - Pick appropriate queue sizes
  - Separate capacity from ABW
  - Model reactivity as a function of flows
  - Model shared bottlenecks

Questions?

- Available now at www.emulab.net
- Email: duerig@cs.utah.edu

Backup Slides

Does capacity matter?



Scale

Shared Bottlenecks are Hard

Stationarity