TCP performance for short-lived sources

Matthew Roughan
AT&T Labs - Research
roughan@research.att.com

Ashok Erramilli
Qnetworx
Erramilli@qnetworx.com

Darryl Veitch
EMUlab – Ericsson and the University of Melbourne
d.veitch@ee.mu.oz.au

TCP and Congestion Control

- October 1986, Internet had its first congestion collapse
- Link LBL to UC Berkeley
  - 400 yards, 3 hops, 32 Kbps
  - throughput dropped to 40 bps
  - factor of ~1000 drop!
- 1988, Van Jacobson proposed TCP flow control

Window Flow Control

- TCP seeks to
  - Achieve high utilization
  - Avoid congestion
  - Share bandwidth
- Window flow control
  - Source rate = $\frac{W}{RTT}$ packets/sec
  - Adapt W to network (and conditions)

    W = BW x RTT

TCP Congestion Control

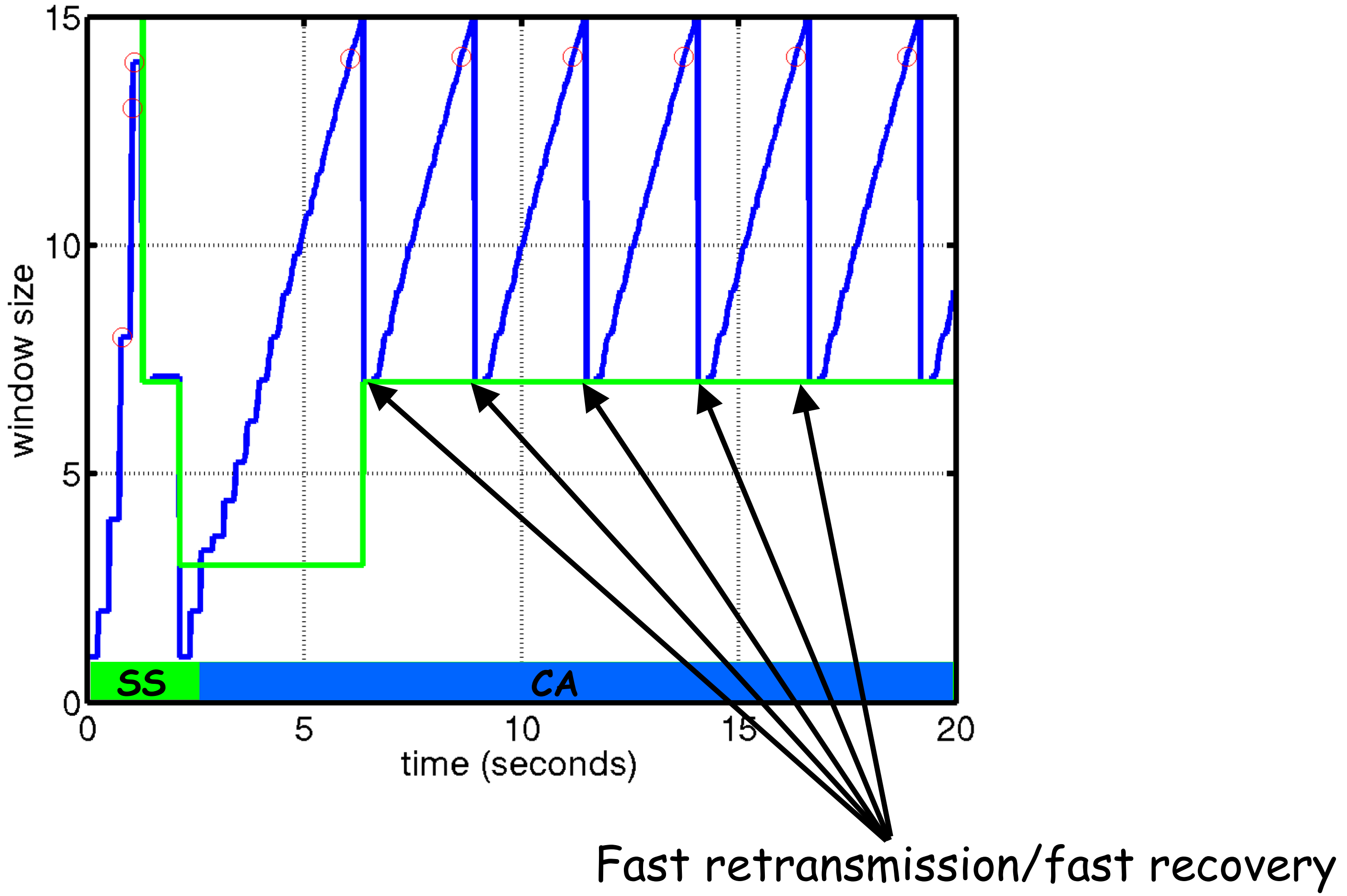
- Has four main parts
  - Slow Start (SS)
  - Congestion Avoidance (CA)
  - Fast Retransmit
  - Fast Recovery

Tahoe

Reno

- ssthresh: slow start threshold determines whether to use SS or CA
- Assume packet losses are caused by congestion
  - detected by timeouts and dup. ACK

TCP Reno



15
10
5
0
window size
SS
CA
0
5
10
15
20
time (seconds)
Fast retransmission/fast recovery

$1/\sqrt{p}$ Law – for CA

- Equilibrium window size $w_s = \frac{a}{\sqrt{p}}$
- Equilibrium rate $x_s = \frac{a}{D_s\sqrt{p}}$
- Empirically constant $a \sim 1$
- Verified extensively through simulations and on Internet
- References
    - T.J.Ott, J.H.B. Kemperman and M.Mathis (1996)
    - M.Mathis, J.Semke, J.Mahdavi, T.Ott (1997)
    - T.V.Lakshman and U.Mahdow (1997)
    - J.Padhye, V.Firoin, D.Towsley, J.Kurose (1998)
    - J.Padhye, V.Firoin, D.Towsley (1999)

Calculating Performance

- Single link, capacity C, buffer B
  - Window size: w = f(p)
  - Loss rate: p = g(w; C,B)
  - Find w*: w* = f(g(w*; C,B))
- Example:
  - Window size: $w = 1/\sqrt{p}$
  - Loss rate approx. $p = \frac{[w-C]^+}{w}$

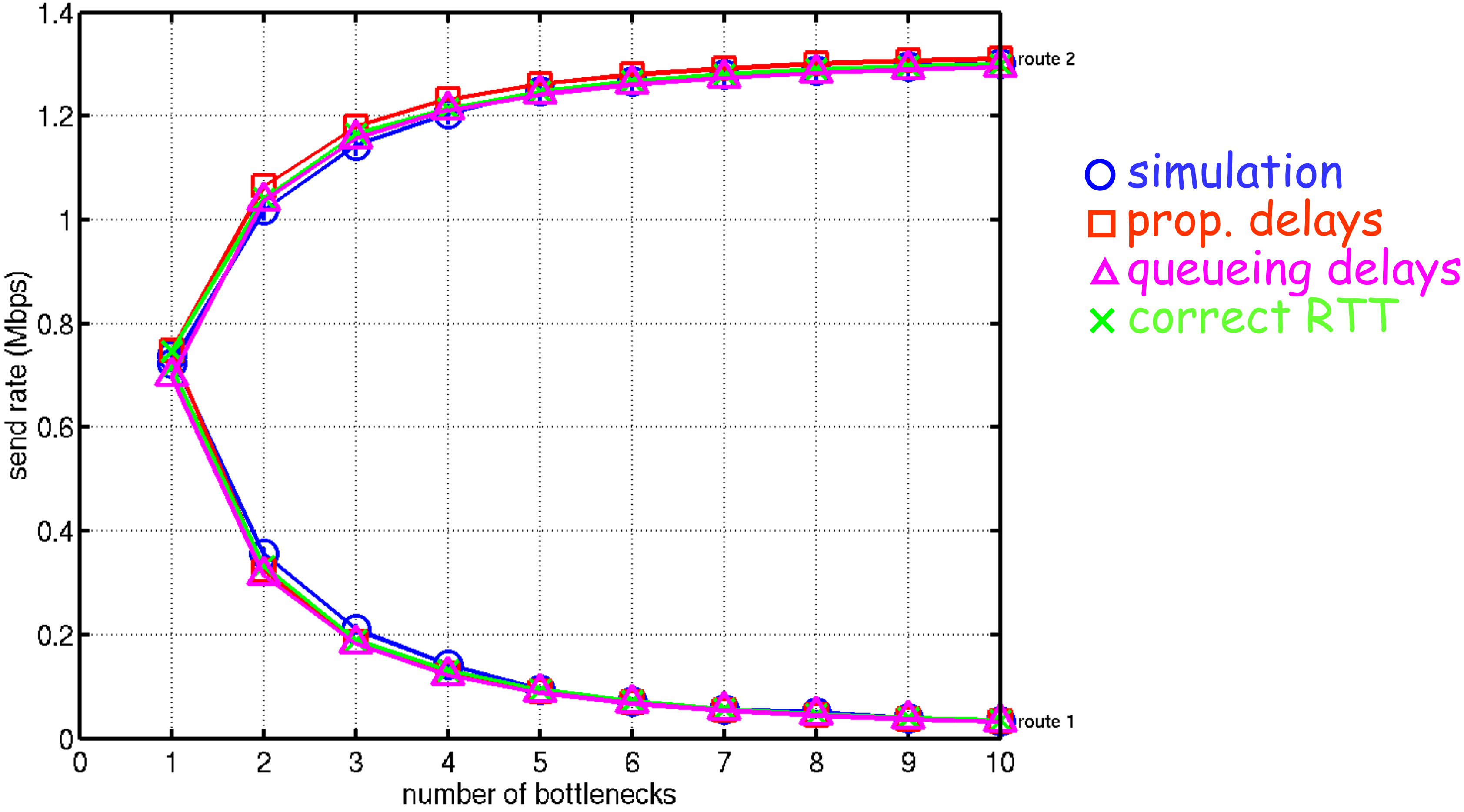
$$w^* = \frac{C + \sqrt{C^2 + 4}}{2}$$

Fixed Point Models

- Mean field theory
  - Solve for a particular source given the mean field
  - Use single source to approximate the mean field
- Generalize previous example
  - Multiple sources
  - Network
    - various routes, RTTs, capacities, …
  - Arbitrary functions f, and g
- Solve using
  - Repeated substitution
  - Newton-Raphson

Numerical Example

- Send rates



Short-lived sources

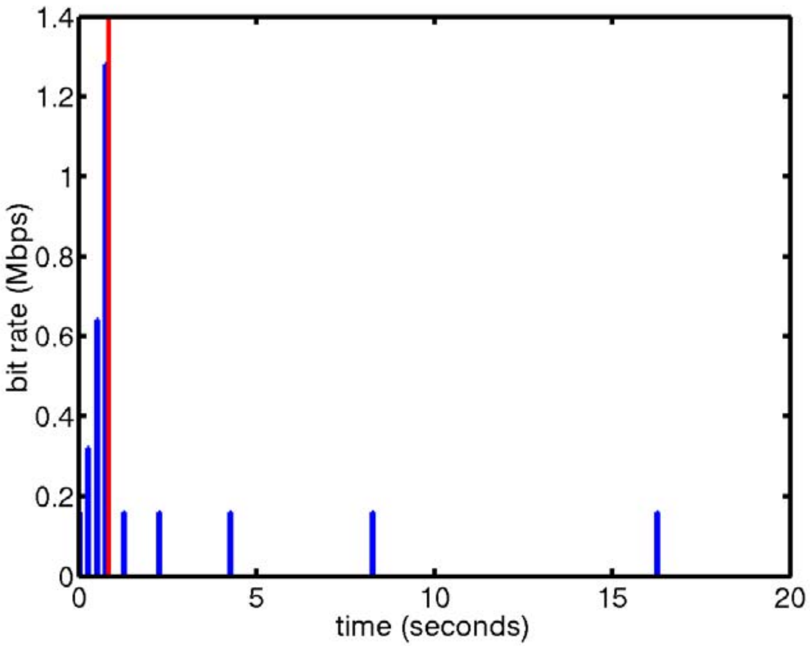
- Heavy-tailed distribution of flow sizes
  - Some really big files elephants
  - Many small files mice
- CA model only good for elephants
  - Short lived sources always in Slow Start
  - M/G/1 processor sharing suggested
  - Really we need a new model, e.g.
    - Cardwell, Savage and Anderson, Infocom 2000
    - Sikdar, Kalyanaraman and Vastola, IPCCC 2001
    - Mellia, Stoica and Zhang, IEEE Communications Let. 2002

New approach

- Use the loss rate to estimate transfer latency (e.g. from Cardwell *et al*)
- Use transfer latency to compute the number of sessions in progress
  - M/G/1 processor sharing queue (for number of sources)
- Use the number of sessions in progress (and their duration) to estimate the load and thence the loss rate
  - M/G/1/K FIFO model (for packets in each buffer)

Successive Timeouts

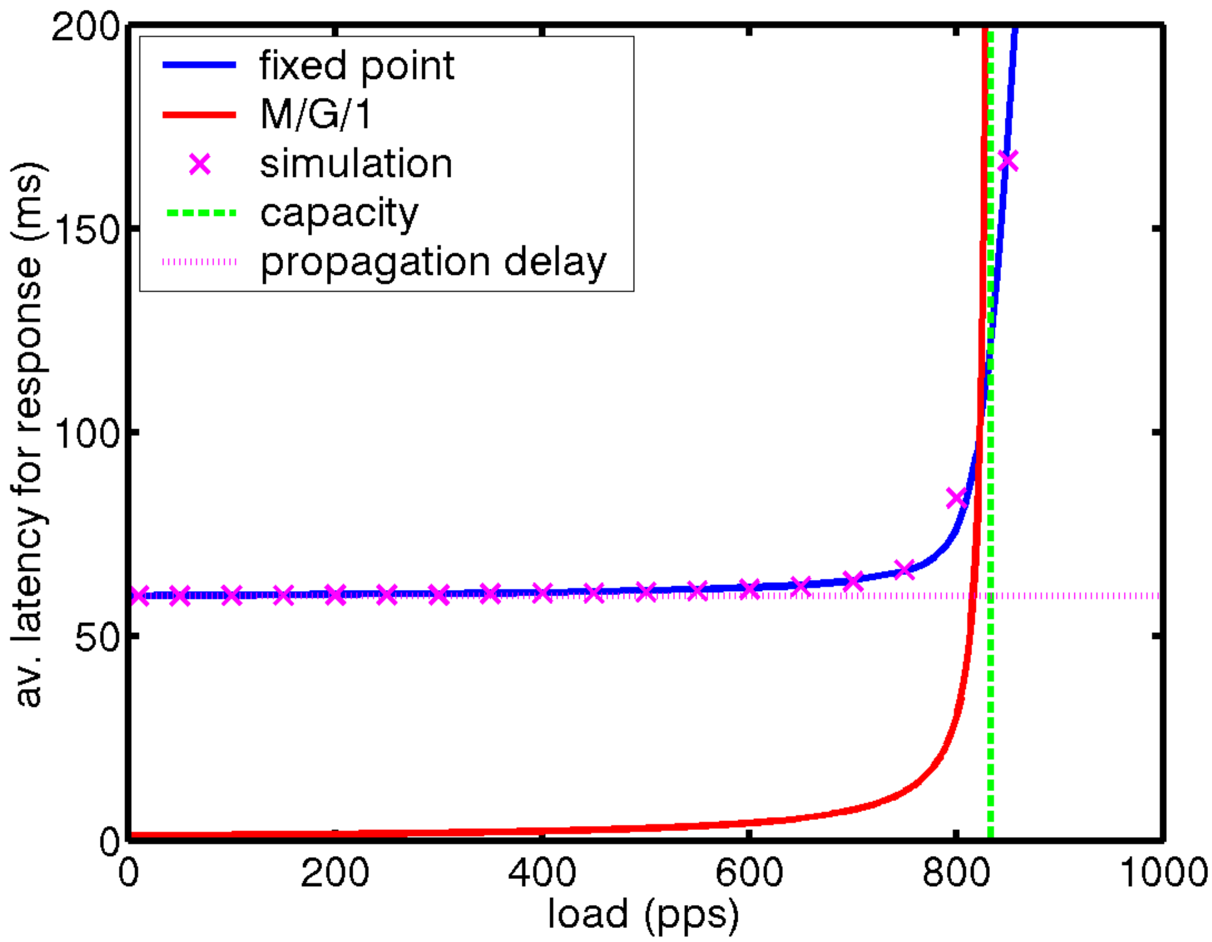
- When there is a timeout, double the RTO
- Keep doing so for each lost retransmission
  - Exponential back-off
  - Max 64 seconds[1]
  - Max 12 restransmits[1]



1 - Net/3 BSD

Simple example

- Poisson arrivals of single packet transfers



Results

- Processor sharing
  - Doesn't get latency right for low load
    - (can't get RTT)
  - Asymptote at capacity
- Even so result is not responsive to congestion!
- Can get a good measure from fixed point approach

Conclusion

- Can use fixed point methods to estimate performance for TCP flow controls
  - Persistant case (based on CA)
  - Short-lived case (based on SS)
- Nice because they generalize to networks
- Need to understand limitations of SS models for TCP window flow controls
  - RTT estimation used in RTO computation
  - In BSD simple because of 500ms timer