Congestion Control

Context

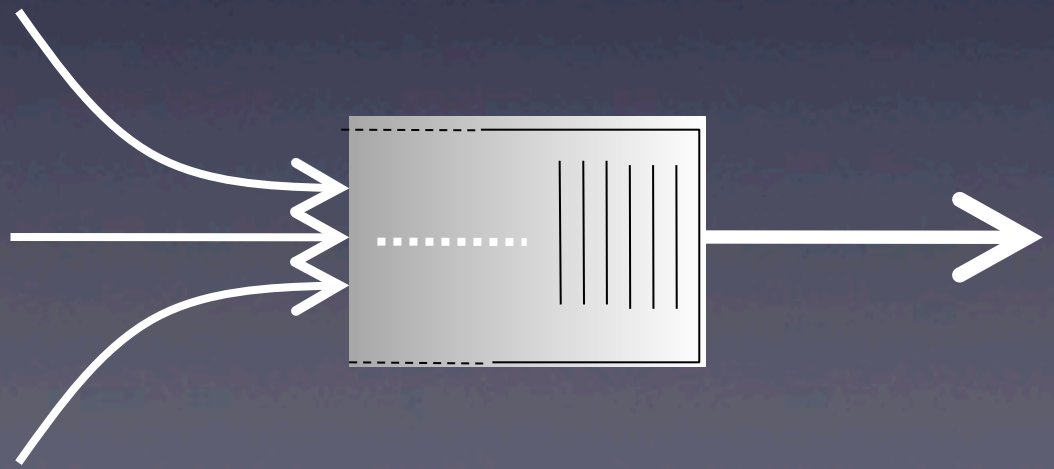
- TCP is the dominant transport protocol in today's Internet
  - Web page loads, BitTorrent transfers, some video streaming, some of Skype
- Embodies some of Internet design principles
  - packet switching (i.e., no state on switches)
  - smart host, dumb network

TCP Mechanisms

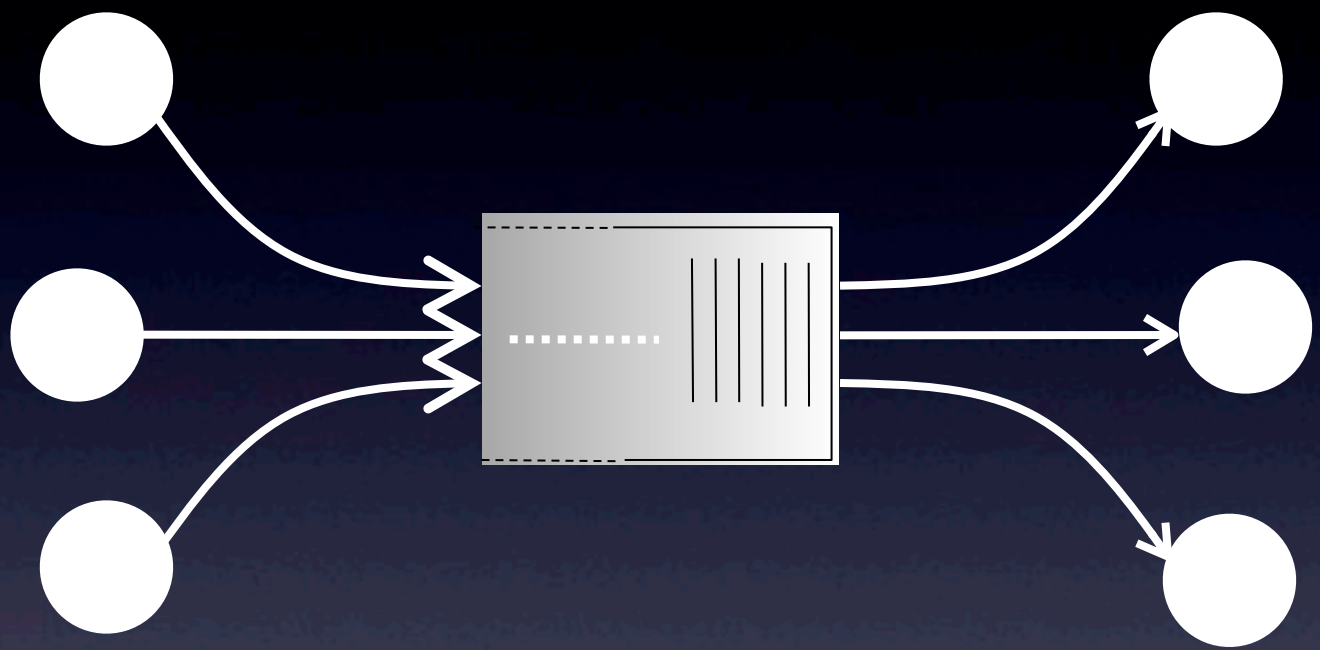
- Flow control: prevent sender from overwhelming the receiver
- Reliable delivery: lost packets are retransmitted
- Congestion control: react to network congestion

Congestion Control

- Best effort delivery on the Internet
  - Let everybody send, try to deliver what you can, and drop the rest
  - If many packets arrive in a short period of time, router buffers fill up and packets are lost
  - Loss indicates congestion; so does increased delay



Congestion Control



- What should be the goals of congestion control? (Or what is an ideal congestion control protocol?)

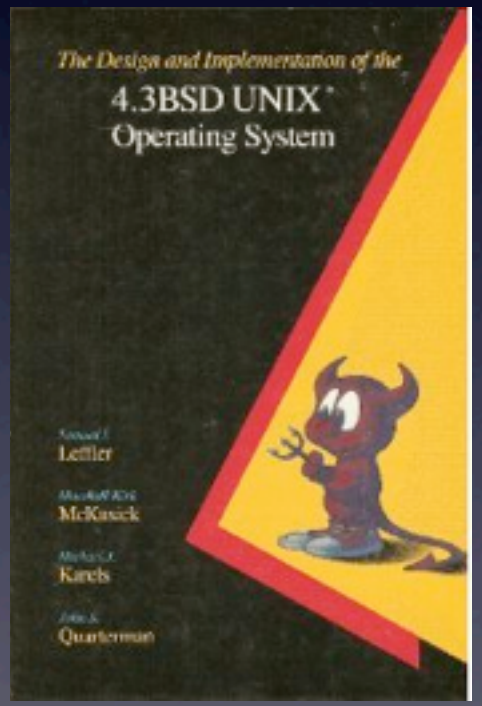
Observations

- Congestion is inevitable, and arguably desirable
- If packets are dropped, then retransmissions can make congestion even worse
- When packets are dropped, they waste resources

Context

Van Jacobson

- Formerly at LBL
- Internet pioneer
- Now at PARC
- Inventor tcpdump, traceroute



Michael J. Karels

- Very involved in BSD development
- Replaced Bill Joy as developer

Cited more than 6,000 times.

Context

TCP didn't work well under congestion (circa 1988)

Congestion collapse:

- Breakdowns in performance noted in 1986 on NSFNet.
- 40Kb/s links operating as slow as 32b/s.
- NSFNet was a forerunner of today's Internet backbone (from 1986 to 1995).

Main contributions

Seven new algorithms:

1. RTT Variance estimation
2. Exponential retransmit timer backoff
3. Slow-start
4. More aggressive receiver ack policy (delayed ACK or not)
5. Dynamic window sizing on congestion
6. Karn's algorithm (accurate RTT)
7. Fast retransmit

Packet Conservation

'Conservation of packets' principle:

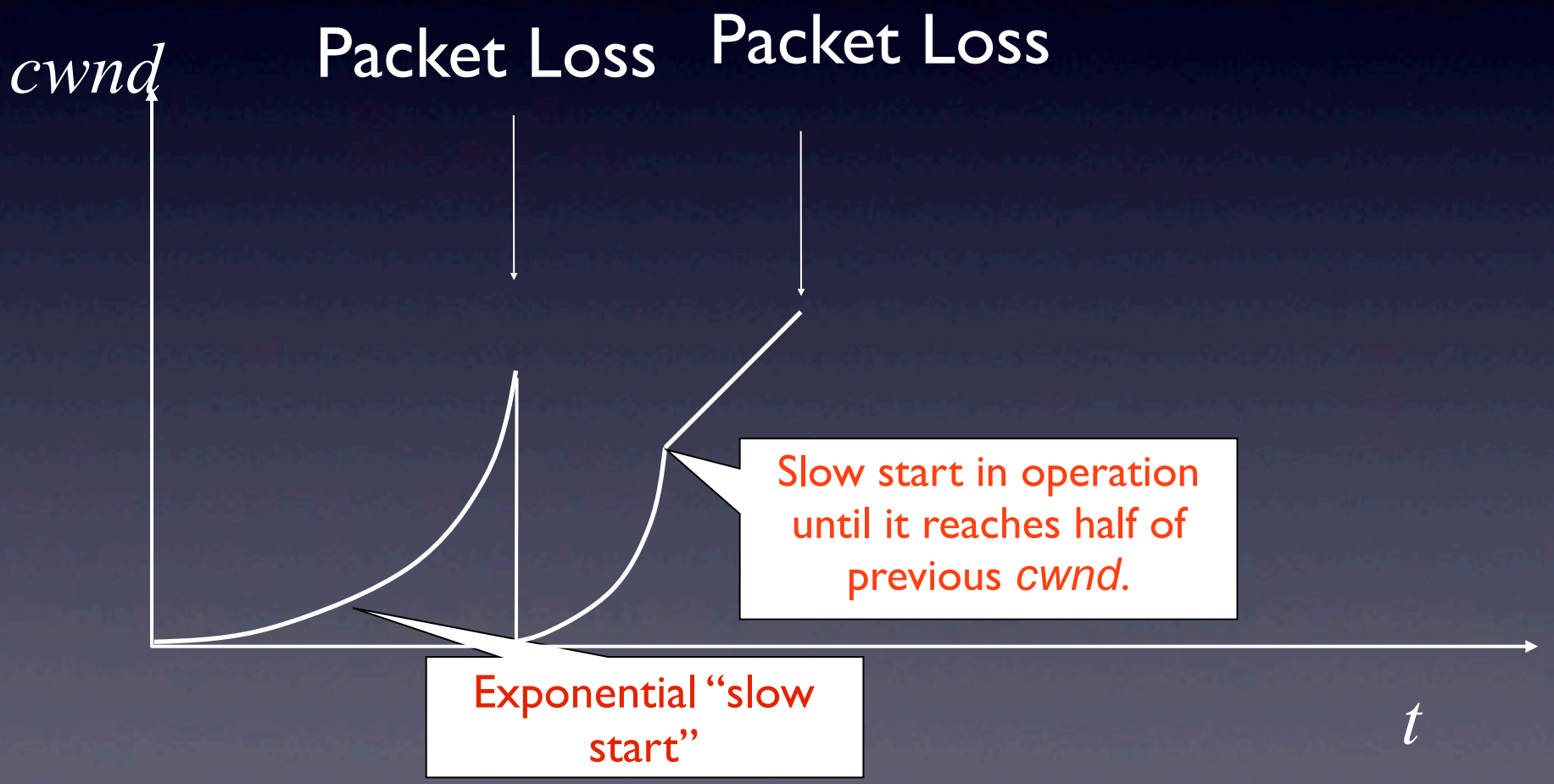
- For a connection 'in equilibrium', i.e., running stably with a full window of data in transit…
- A new packet shouldn't be put into the network until an old packet leaves.

TCP Animation

- http://guido.appenzeller.net/anims/

Slow-start + AIMD (Tahoe)

Window size = min(advertised window, *cwnd*)

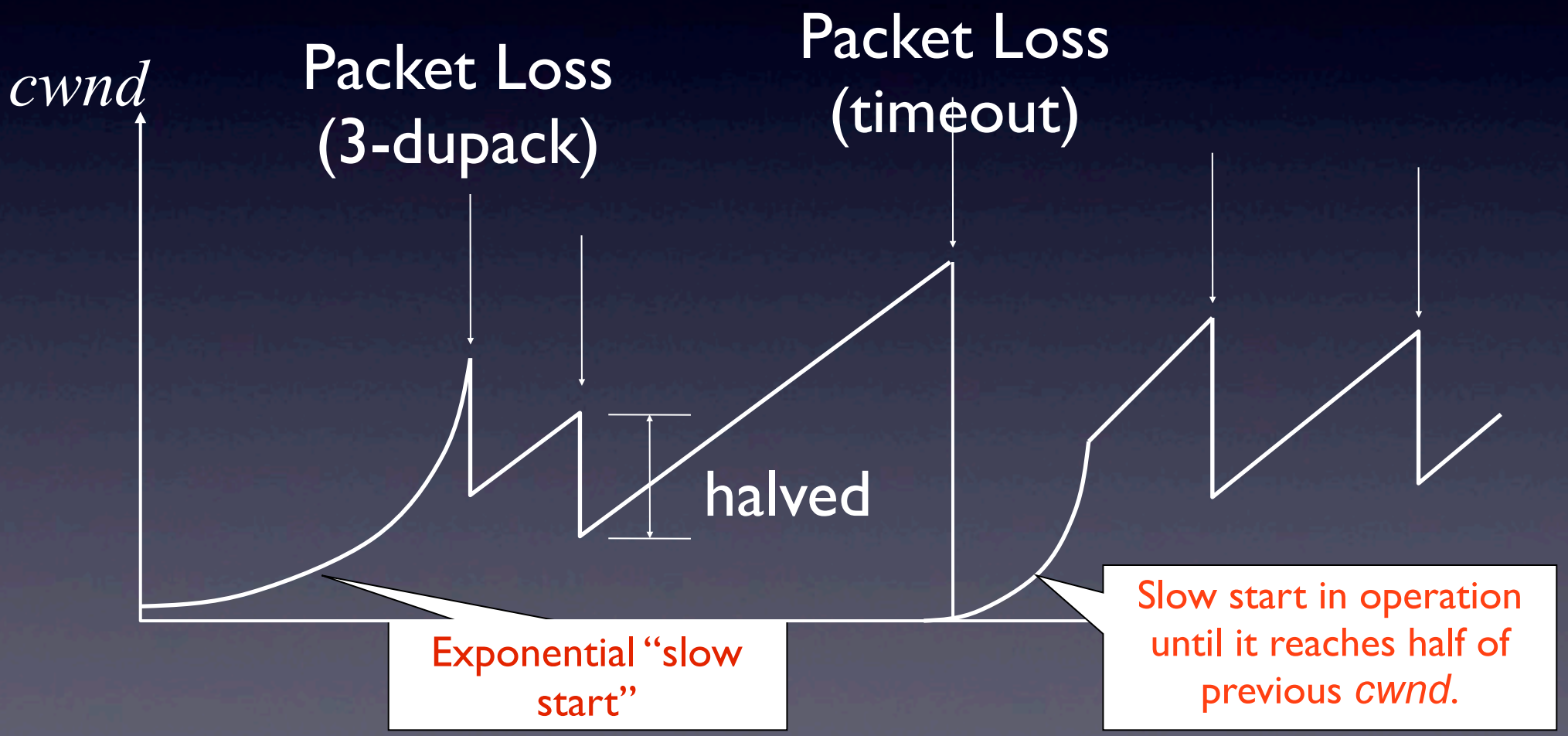


TCP Reno

- Enhancements include:
  - early detection of packet loss using 3-dupack (where acks are cumulative ACKs)
  - fast retransmit of such packets
  - fast recovery of congestion window

Slow-start + AIMD (Reno)

Window size = min(advertized window, *cwnd*)



TCP NewReno

- During fast recovery:
  - keep track of last unacked pkt when entering fast recovery
  - on every dupack, inflate congestion window by 1 pkt
    - start sending out new packets while fast retransmit is in flight
  - when last packet is acked, return to congestion avoidance -- set cwnd back to value set when starting fast recovery

Getting to equilibrium
Slow-start

Q: What is slow-start trying to accomplish?

Q: How long does it take slow-start to reach equilibrium?

Q: How does AIMD compare to AIAD, MIMD, and MIAD?

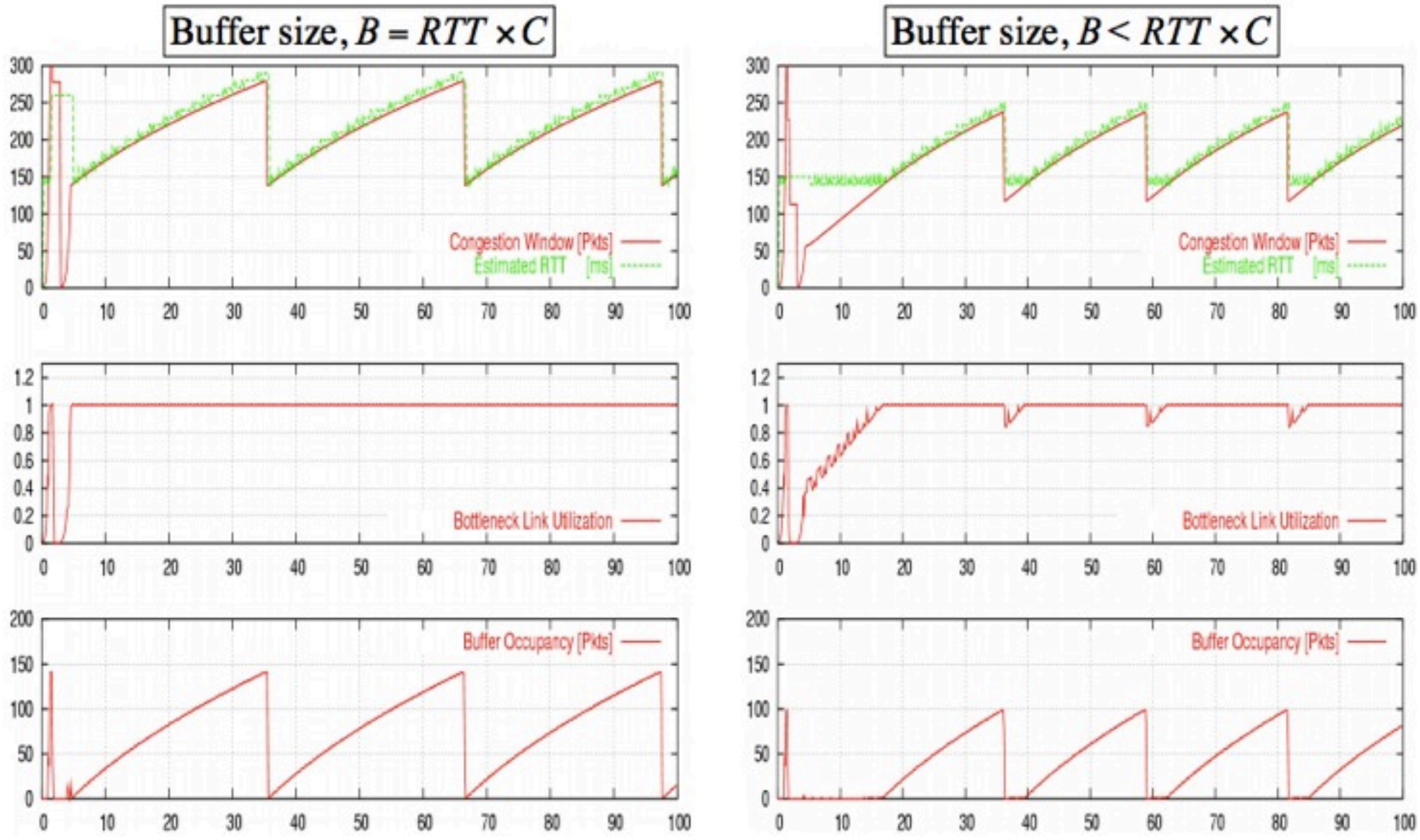
RTT Variation Estimate

Q: Why is it important to estimate RTT well?

Q: How did they improve RTT estimation?

Q: Can we do better than cumulative ACKs?

Sizing Router Buffers



TCP properties

What are TCP goals for:

- Long-lived flows
- Short-lived flows
- The network operator

Q: How well does it accomplish these goals?

Meta Comments

Style

Q: What do you think of the style of the paper

TCP Challenges

- TCP early designs were in 80s and 90s
- What are the new challenges for TCP in today's world?

Course Wrapup

- Covered key systems topics:
  - concurrency, transactions, distributed systems, file systems, large-scale storage, DHTs, MapReduce, TCP
- Combination of both classical and recent papers
- Insights into what makes building computer systems hard
  - Performance issues, availability, consistency, fault-tolerance, etc.
- More interest? Take 551/552/561