Introduction to Computer Networks

COSC 4377

Lecture 10

Spring 2012

February 20, 2012

Announcements

- HW5 due this week
- HW deadlines
- Exam1 practice problems later today

Today's Topics

- HW5 discussions
- Transport Protocol
  - TCP Friendliness
  - Getting help from the network

Slow Start

Figure 3: Startup behavior of TCP without Slow-start

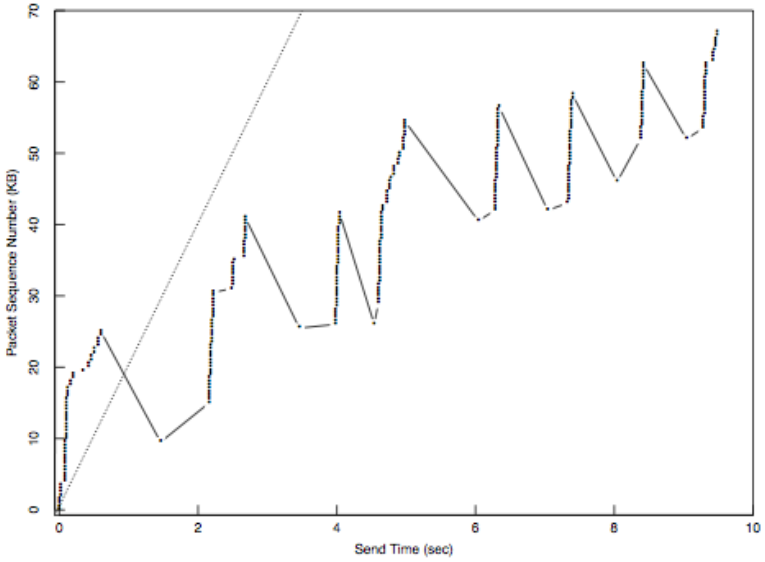
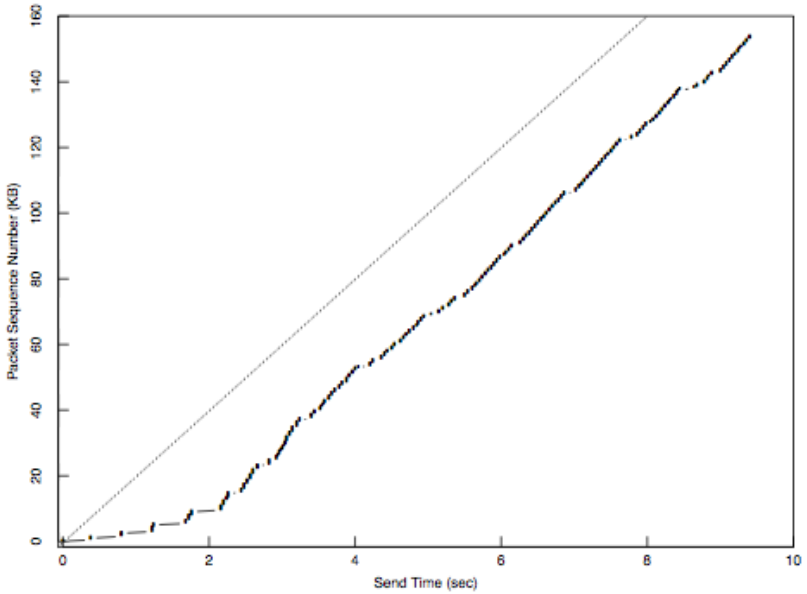
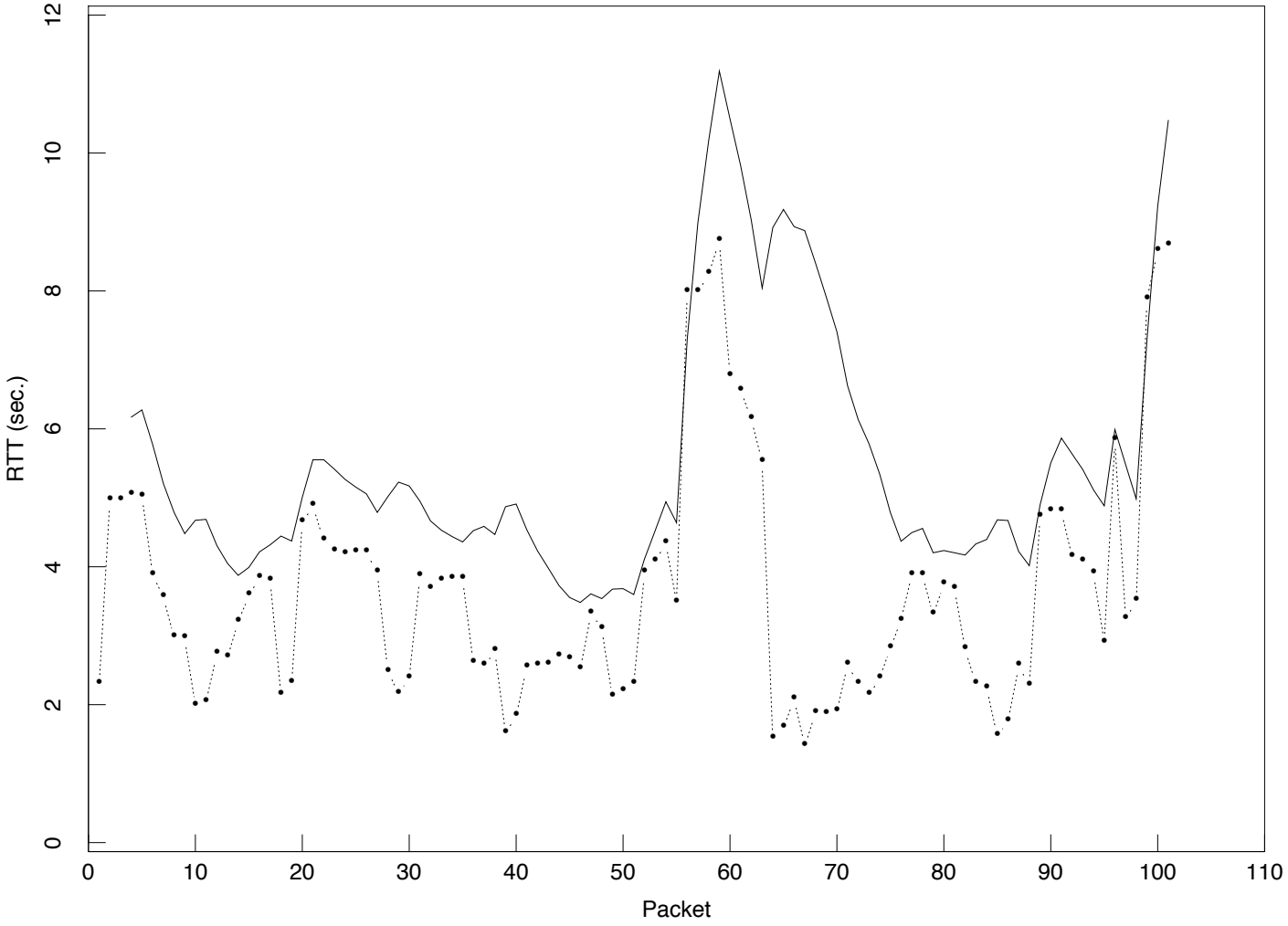


Figure 4: Startup behavior of TCP with Slow-start

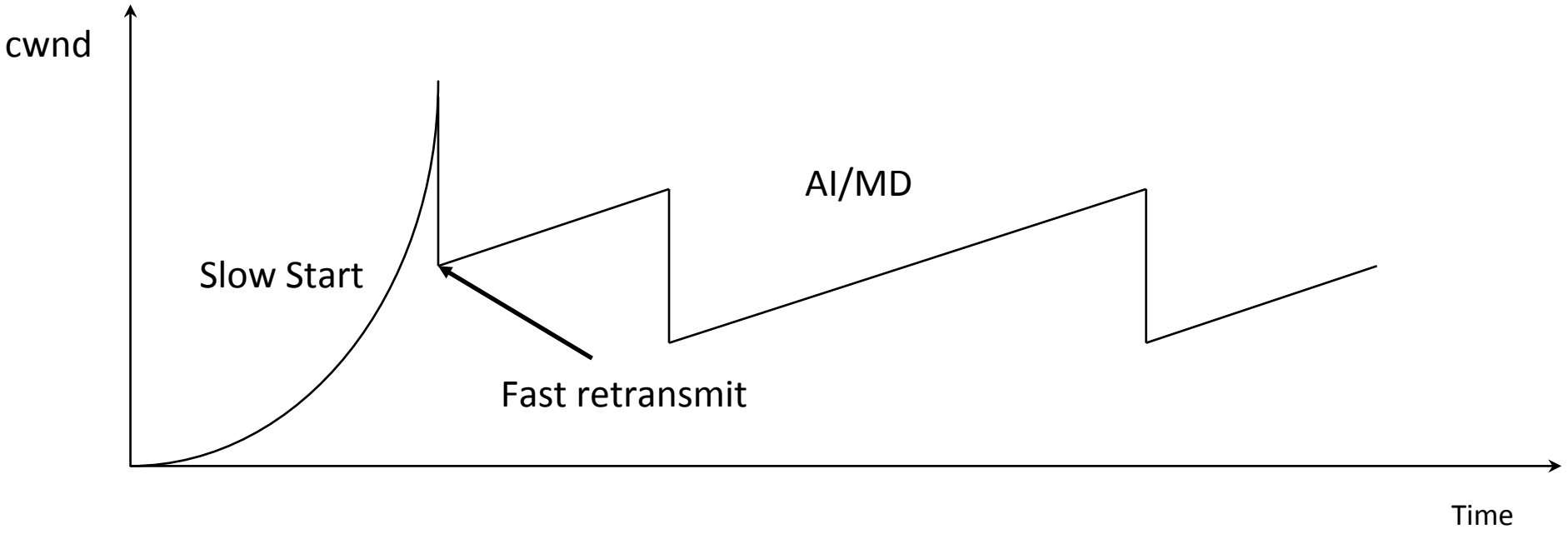


From [Jacobson88]

Tahoe RTT Estimation

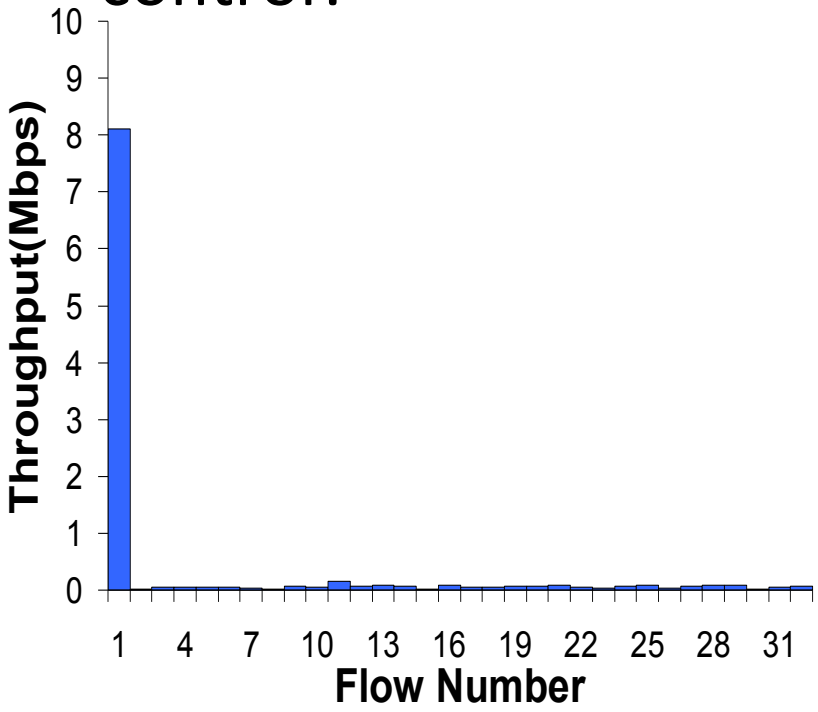


Fast Recovery and Fast Retransmit



TCP Friendliness

- Can other protocols co-exist with TCP?
  - E.g., if you want to write a video streaming app using UDP, how to do congestion control?



1 UDP Flow at 10MBps
31 TCP Flows
Sharing a 10MBps link

TCP Friendliness

- Can other protocols co-exist with TCP?
  - E.g., if you want to write a video streaming app using UDP, how to do congestion control?
- Equation-based Congestion Control
  - Instead of implementing TCP's CC, estimate the rate at which TCP would send. Function of what?
  - RTT, MSS, Loss
- Measure RTT, Loss, send at that rate!

TCP Throughput

- Assume a TCP connection of window W, round-trip time of RTT, segment size MSS
  - Sending Rate $S = W \times MSS / RTT$ (1)
- Drop: W = W/2
  - grows by $MSS\ W/2\ RTTs$, until another drop at $W \approx W$
- Average window then 0.75x*S*
  - From (1), $S = 0.75\ W\ MSS / RTT$ (2)
- Loss rate is 1 in number of packets between losses:
  - Loss = $1 / (1 + (W/2 + W/2+1 + W/2 + 2 + \ldots + W)$
    $= 1 / (3/8\ W^2)$ (3)

TCP Throughput (cont)

– Loss = $8/(3W^2)$

$$\Rightarrow W = \sqrt{\frac{8}{3 \cdot Loss}} \qquad (4)$$

– Substituting (4) in (2), $S = 0.75\ W\ MSS\ /\ RTT$,

$$\text{Throughput} \approx 1.22 \times \frac{MSS}{RTT \cdot \sqrt{Loss}}$$

- Equation-based rate control can be TCP friendly and have better properties, e.g., small jitter, fast ramp-up…

$$W = \sqrt{\frac{8}{3p}} \tag{1}$$

Substitute W into the bandwidth equation below:

$$BW = \frac{data\ per\ cycle}{time\ per\ cycle} = \frac{MSS * \frac{3}{8}W^2}{RTT * \frac{W}{2}} = \frac{MSS/p}{RTT\sqrt{\frac{2}{3p}}} \tag{2}$$

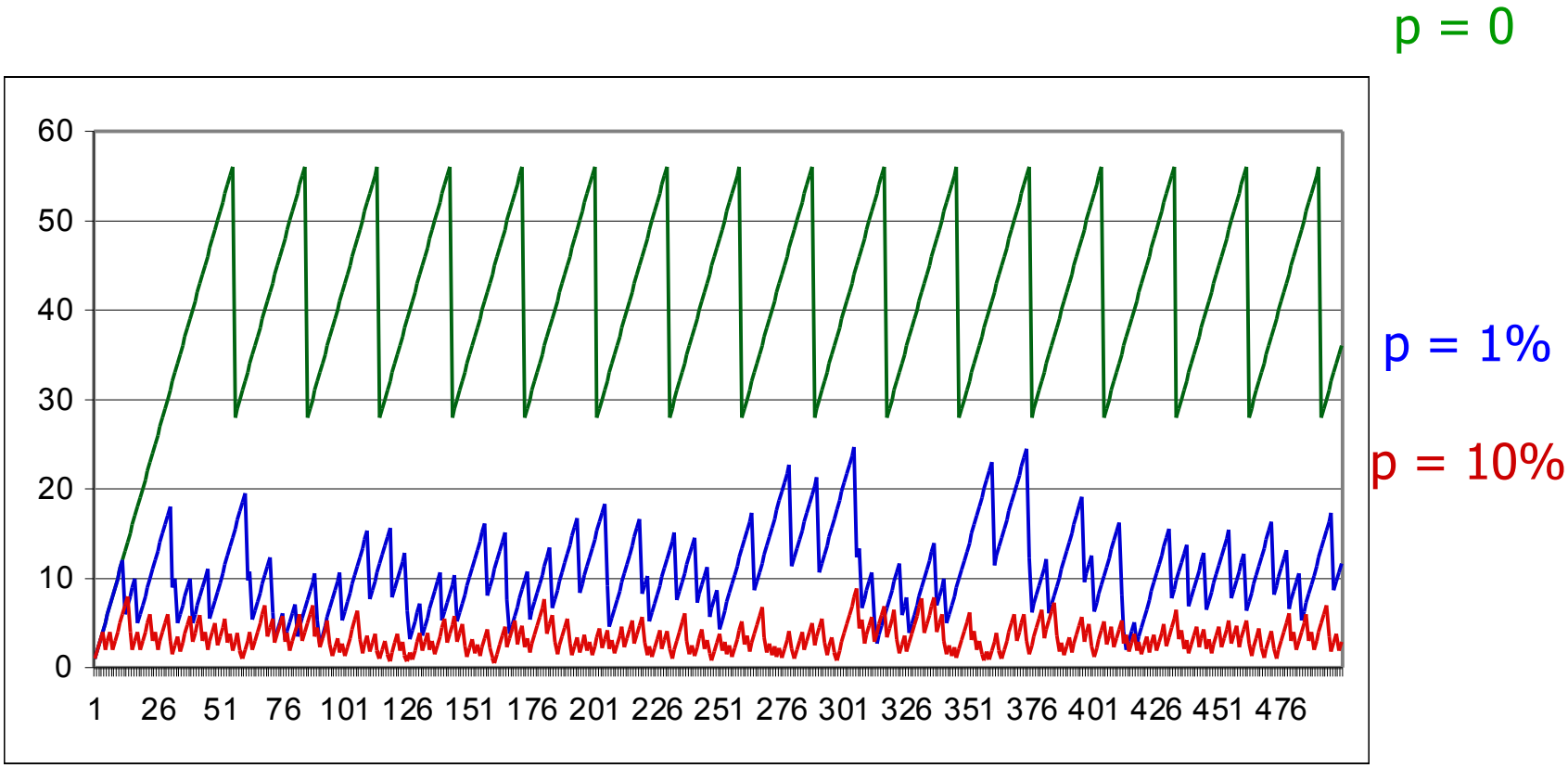
Collect the constants in one term, $C = \sqrt{3/2}$, then we arrive at:

$$BW = \frac{MSS}{RTT}\frac{C}{\sqrt{p}} \tag{3}$$

[Mathis et al. 97]

What Happens When Link is Lossy?

- Throughput ≈ 1 / sqrt(Loss)



What can we do about it?

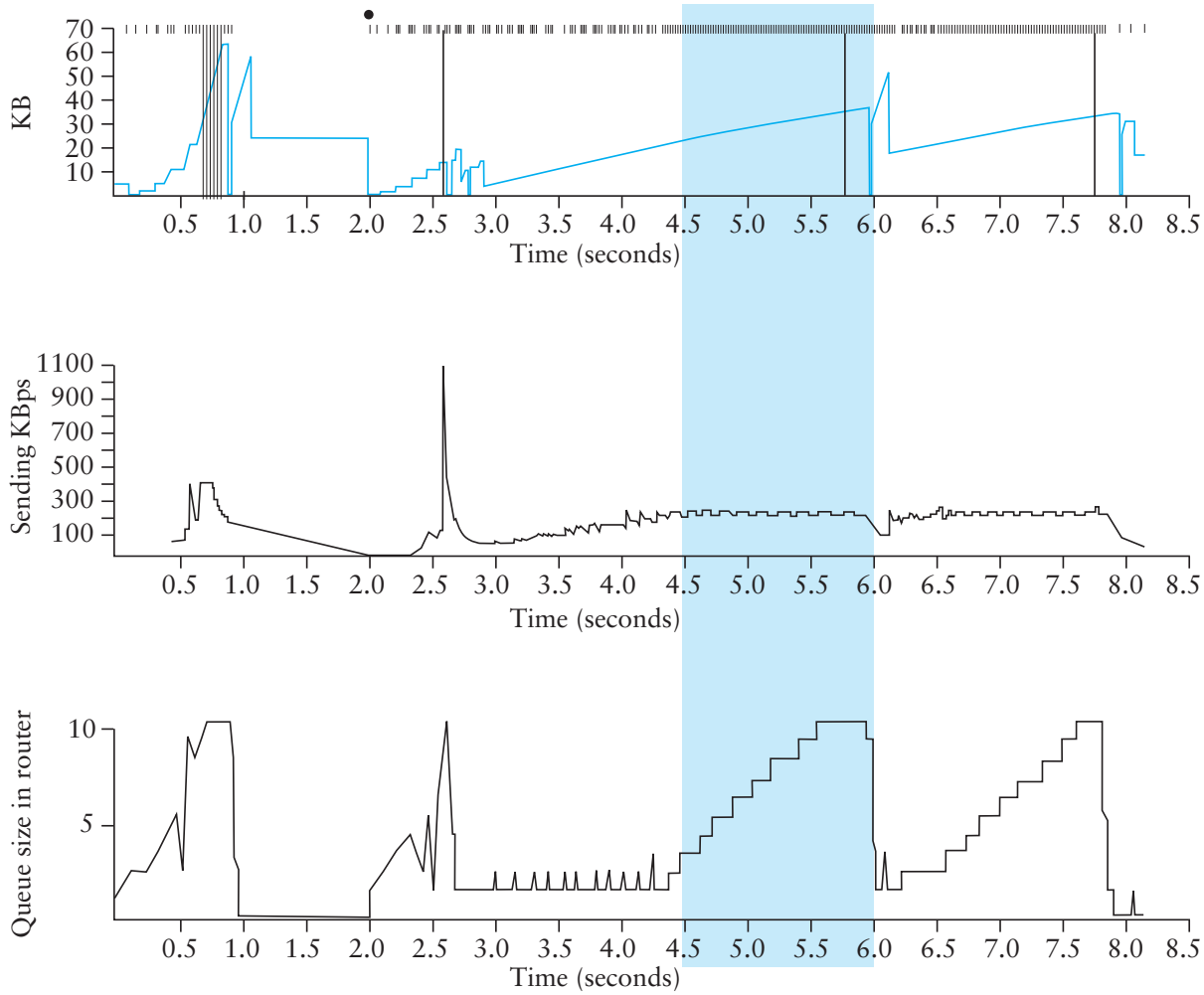
- Two types of losses: congestion and corruption
- One option: mask corruption losses from TCP
  - Retransmissions at the link layer
  - E.g. Snoop TCP: intercept duplicate acknowledgments, retransmit locally, filter them from the sender
- Another option:
  - Tell the sender about the cause for the drop
  - Requires modification to the TCP endpoints

Congestion Avoidance

- TCP creates congestion to then back off
  - Queues at bottleneck link are often full: increased delay
  - Sawtooth pattern: jitter
- Alternative strategy
  - Predict when congestion is about to happen
  - Reduce rate early
- Two approaches
  - Host centric: TCP Vegas
  - Router-centric: RED, DECBit

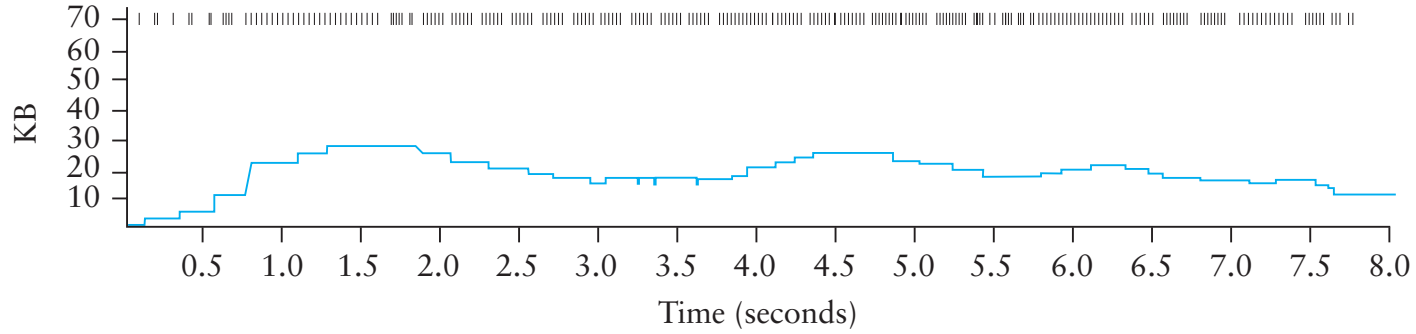
TCP Vegas

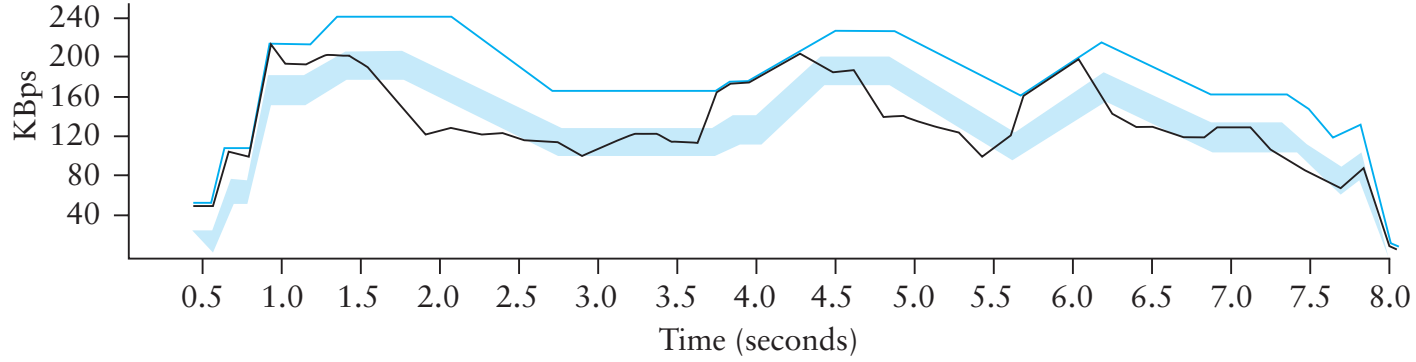
- Idea: source watches for sign that router's queue is building up (e.g., sending rate flattens)



TCP Vegas

- Compare Actual Rate (A) with Expected Rate (E)
  - If E-A > β, decrease cwnd linearly : A isn't responding
  - If E-A < α, increase cwnd linearly : Room for A to grow





Vegas

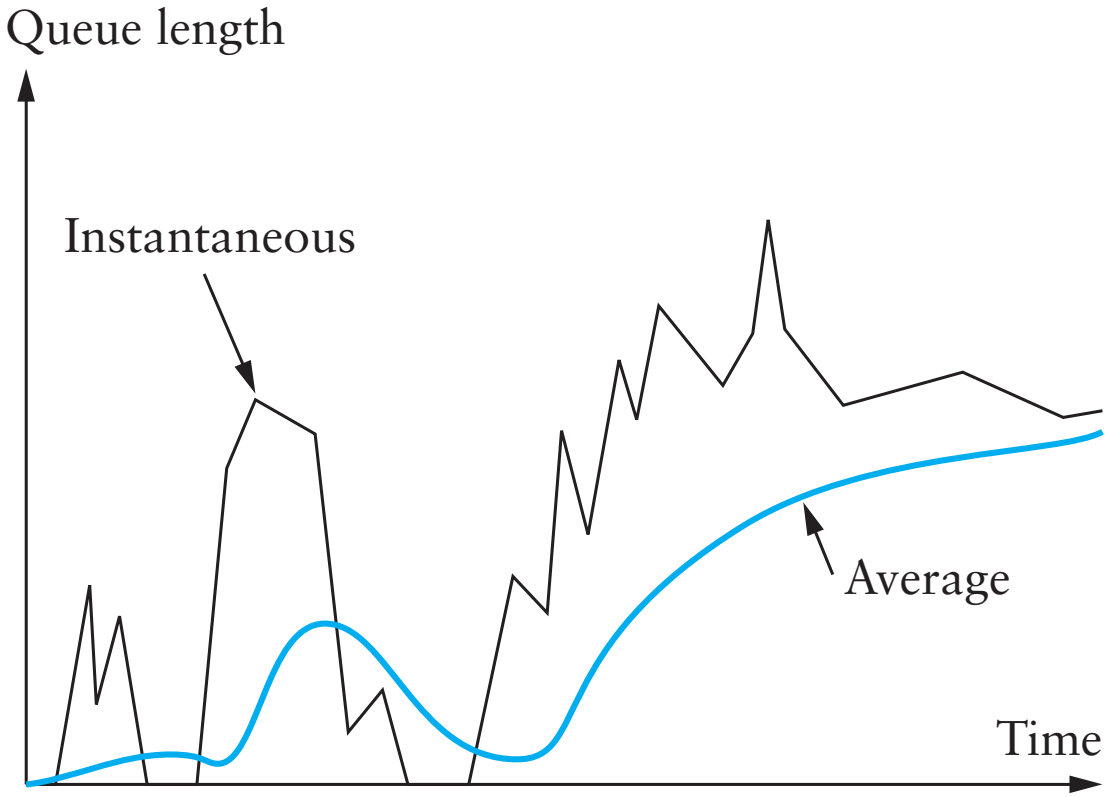
- Shorter router queues
- Lower jitter
- Problem:
  - Doesn't compete well with Reno. Why?
  - Reacts earlier, Reno is more aggressive, ends up with higher bandwidth…

Help from the network

- What if routers could *tell* TCP that congestion is happening?
  - Congestion causes queues to grow: rate mismatch
- TCP responds to drops
- Idea: Random Early Drop (RED)
  - Rather than wait for queue to become full, drop packet with some probability that increases with queue length
  - TCP will react by reducing cwnd
  - Could also mark instead of dropping: ECN

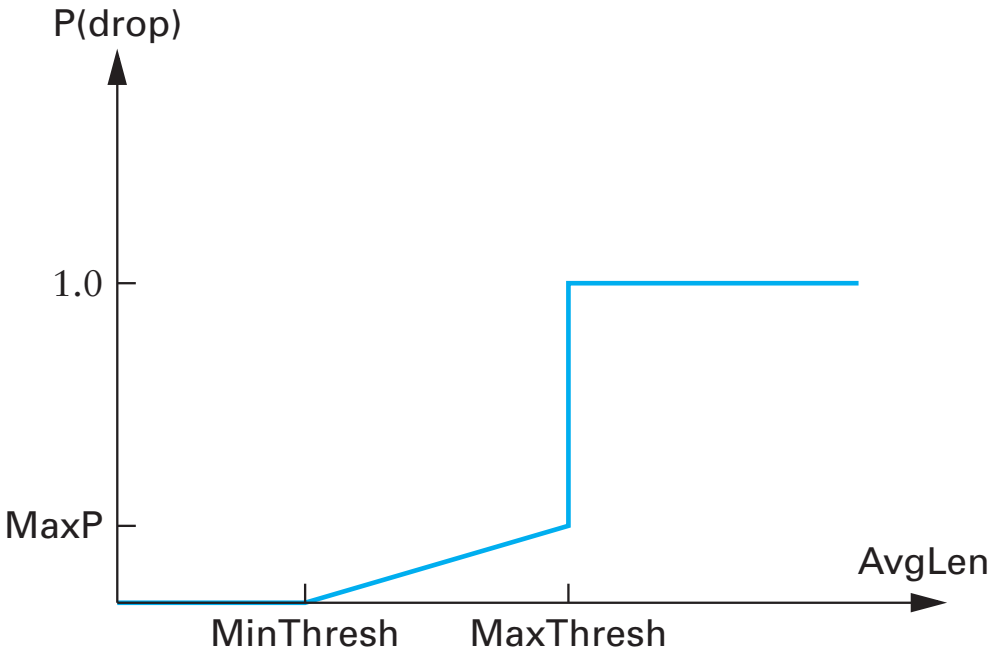
RED Details

- Compute average queue length (EWMA)
  - Don't want to react to very quick fluctuations



RED Drop Probability

- Define two thresholds: MinThresh, MaxThresh
- Drop probability:



- **Improvements to spread drops (see book)**

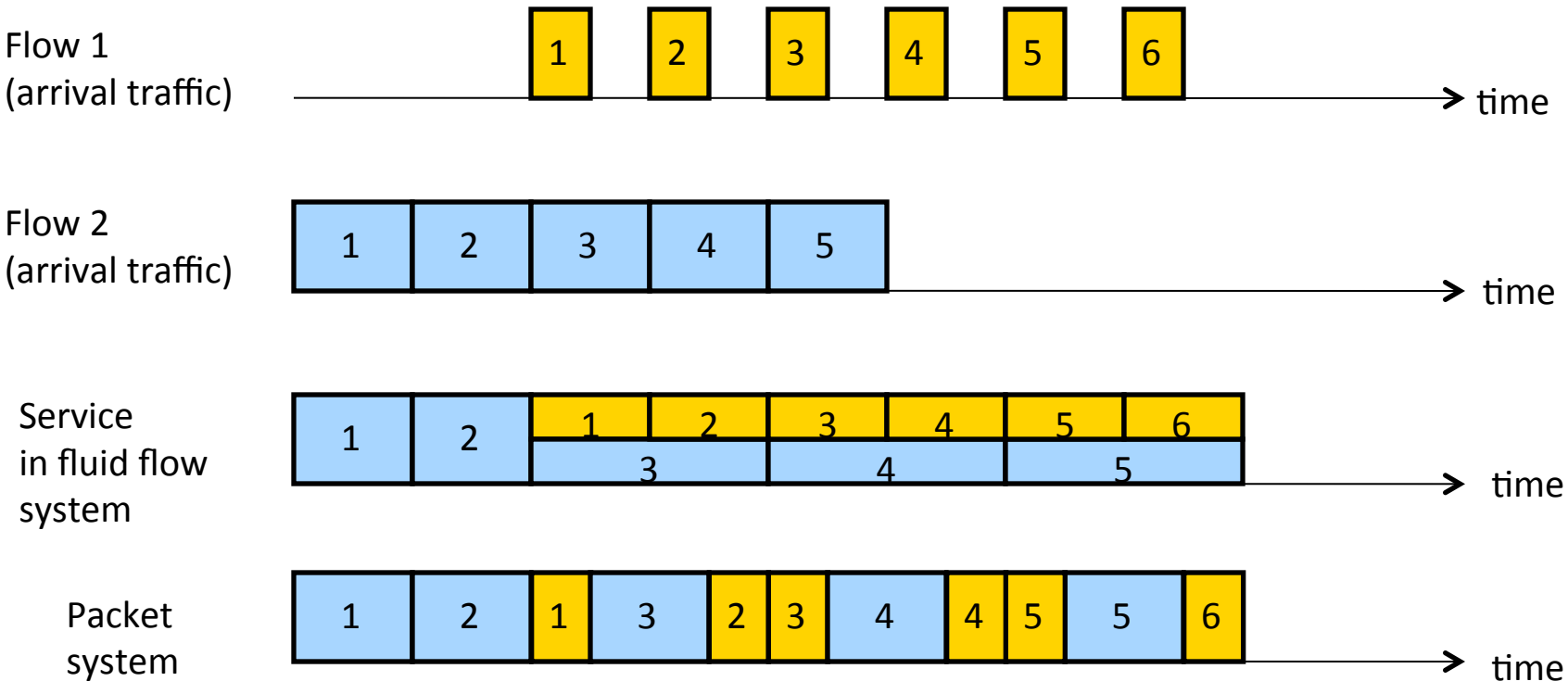
RED Advantages

- Probability of dropping a packet of a particular flow is roughly proportional to the share of the bandwidth that flow is currently getting
- Higher network utilization with low delays
- Average queue length small, but can absorb bursts
- ECN
  - Similar to RED, but router sets bit in the packet
  - Must be supported by both ends
  - Avoids retransmissions optionally dropped packets

More help from the network

- Problem: still vulnerable to malicious flows!
  - RED will drop packets from large flows preferentially, but they don't have to respond appropriately
- Idea: Multiple Queues (one per flow)
  - Serve queues in Round-Robin
  - Nagle (1987)
  - Good: protects against misbehaving flows
  - Disadvantage?
  - Flows with larger packets get higher bandwidth

Example


Flow 1
(arrival traffic)
1
2
3
4
5
6
time
Flow 2
(arrival traffic)
1
2
3
4
5
time
Service
in fluid flow
system
1
2
1
2
3
4
5
6
3
4
5
time
Packet
system
1
2
1
3
2
3
4
4
5
5
6
time