Congestion Avoidance and Control

Van Jacobson

Lawrence Berkeley Laboratory

Michael J. Karels

University of California at Berkeley

November, 1988

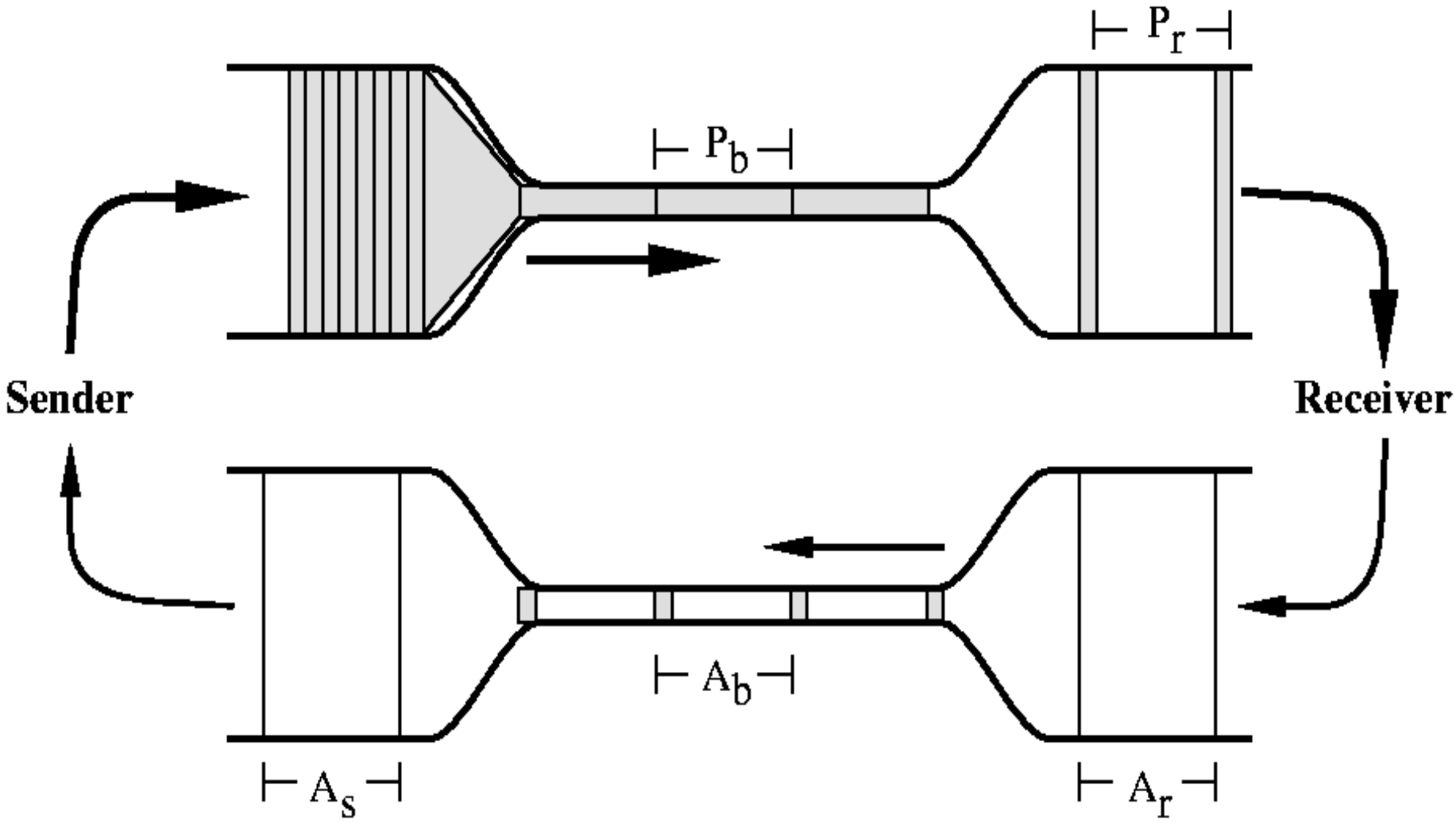
Outline

- Slow-start
- Round-trip time timing
- Congestion avoidance

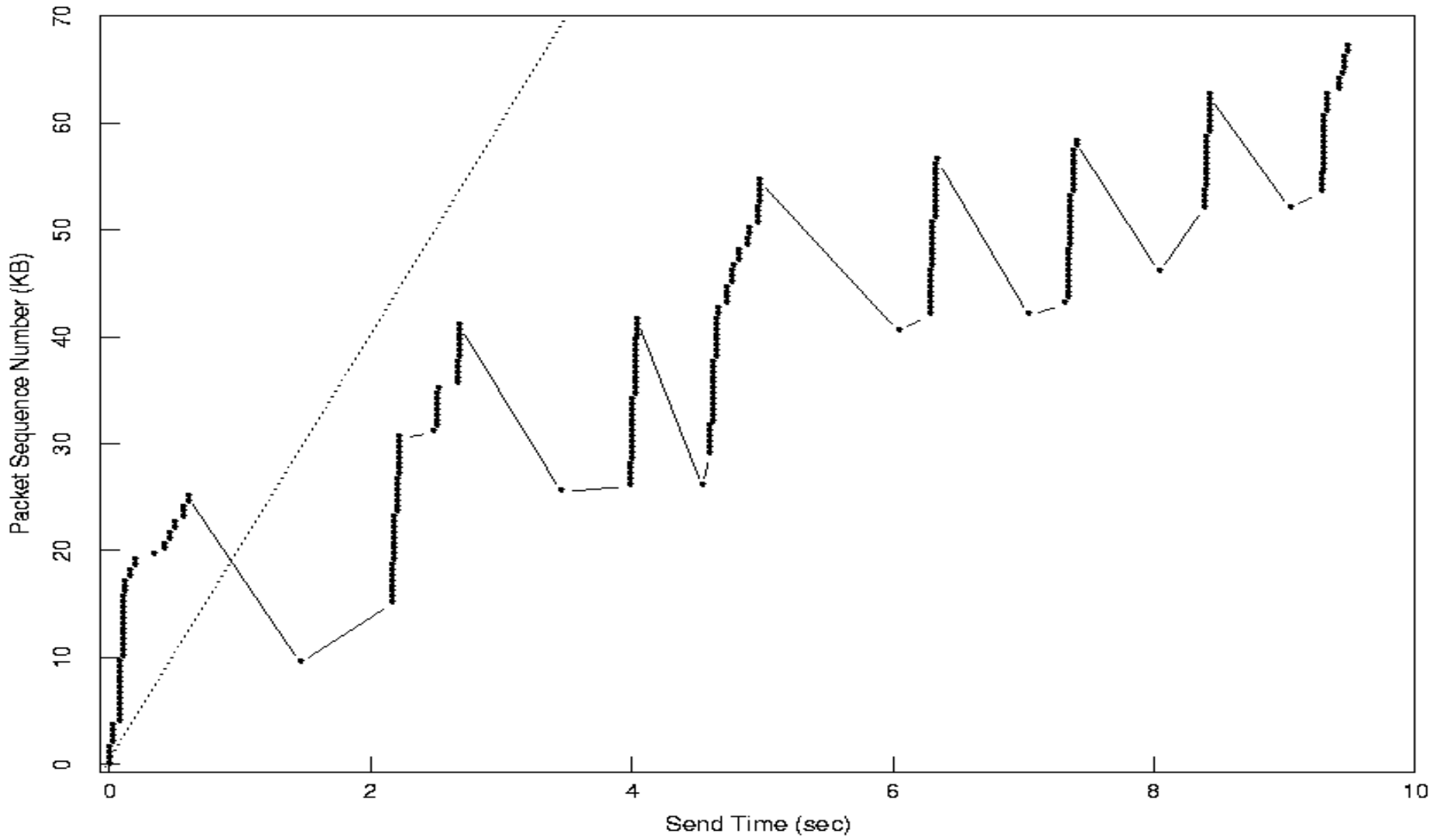
Packet Conservation Fails

- The connection doesn’t get into equilibrium
- A sender injects a new packet before an old packet has exited
- The equilibrium can’t be reached because of resource limits along the path

Window Flow Control
‘Self-clocking’



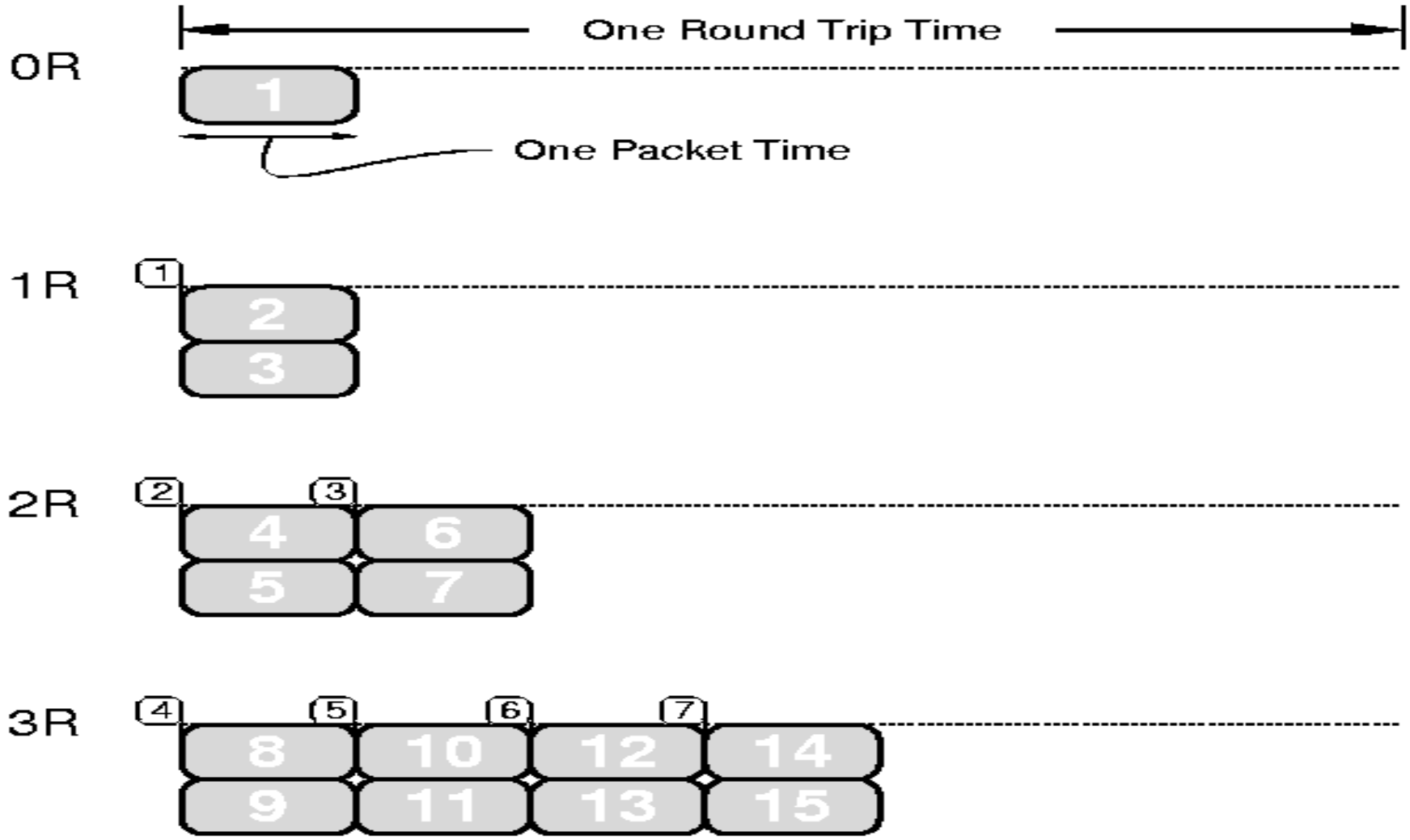
Startup behavior of TCP without Slow-start



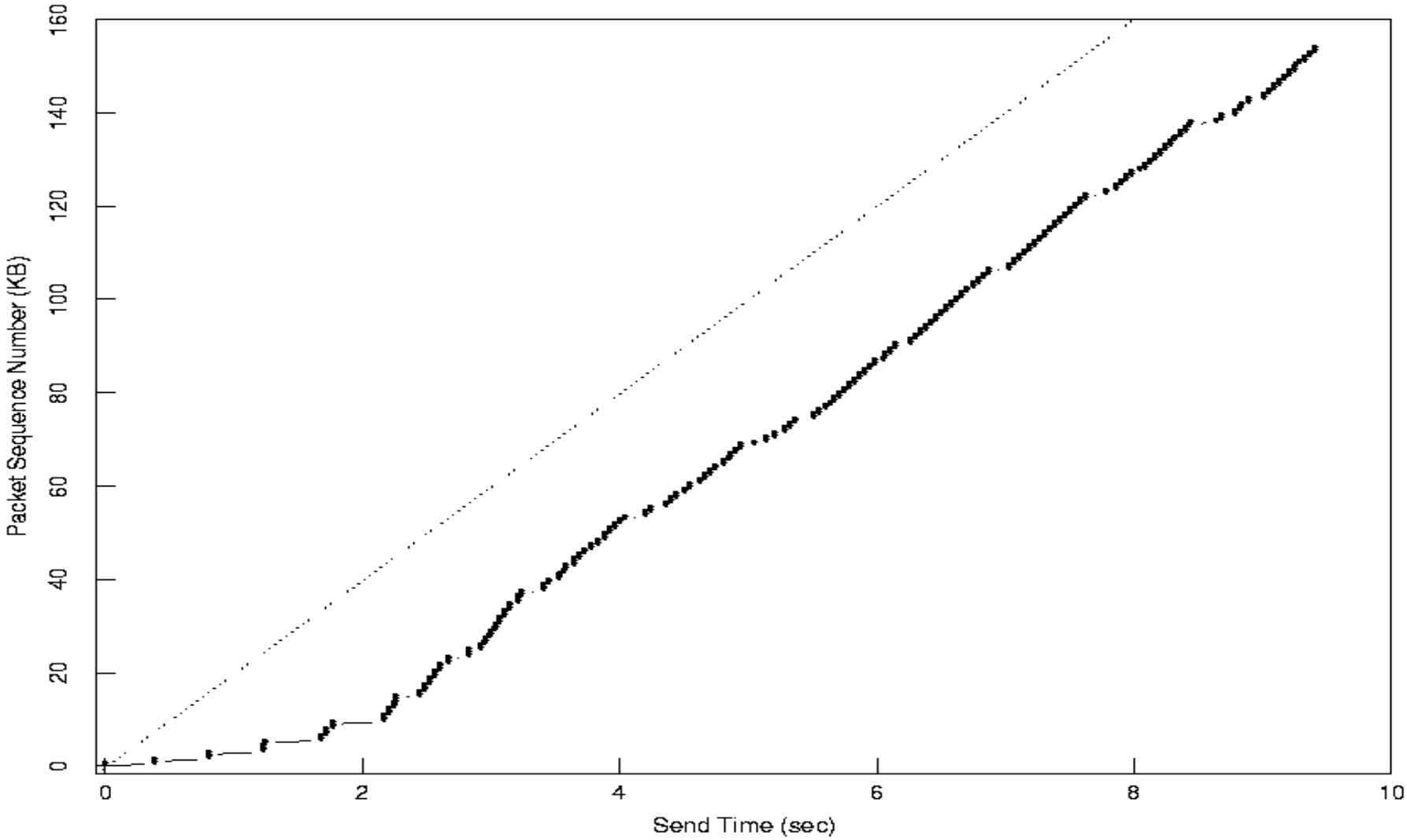
Getting to Equilibrium: Slow-start

- Add a *congestion window*, cwnd, to the per-connection state
- When starting or restarting after a loss, set cwnd to one packet
- On each ack for new data, increase cwnd by one packet
- Sending the minimum of the receiver's advertised window and cwnd

The Chronology of a Slow-start



Startup behavior of TCP with Slow-start



Conservation at equilibrium: round-trip timing

- Good round trip time estimator
  - TCP protocol specification suggests

$$R \leftarrow \alpha R + (1 - \alpha)M, \quad \alpha = 0.9$$

$$\text{rto} \leftarrow \beta R, \quad \beta = 2$$

- Backoff after a retransmit

A fast algorithm for RTT mean and variation

- Err $\equiv$ m – a
- a $\leftarrow$ a + gErr
- v $\leftarrow$ v + g(|Err| - v)
  - m: new measurement of RTT
  - a: average RTT
  - g = 1/8

Window adjustment interaction with round-trip timing

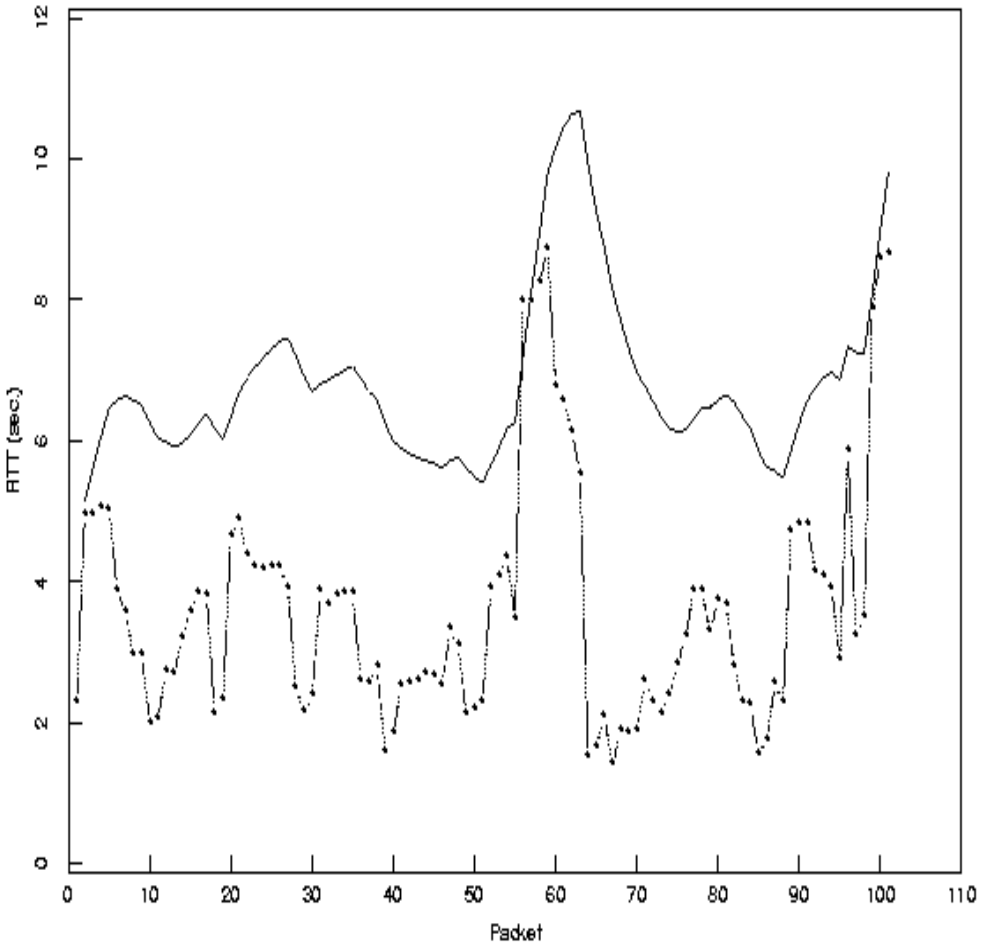
- Bandwidth-dominated
- In worst case, $R_{i+1} = 2R_i$
  - $V_i = R_i - R_{i-1} = R_i / 2$
  - $RTO_i = R_i + 4 V_i$

    $= 3 R_i$

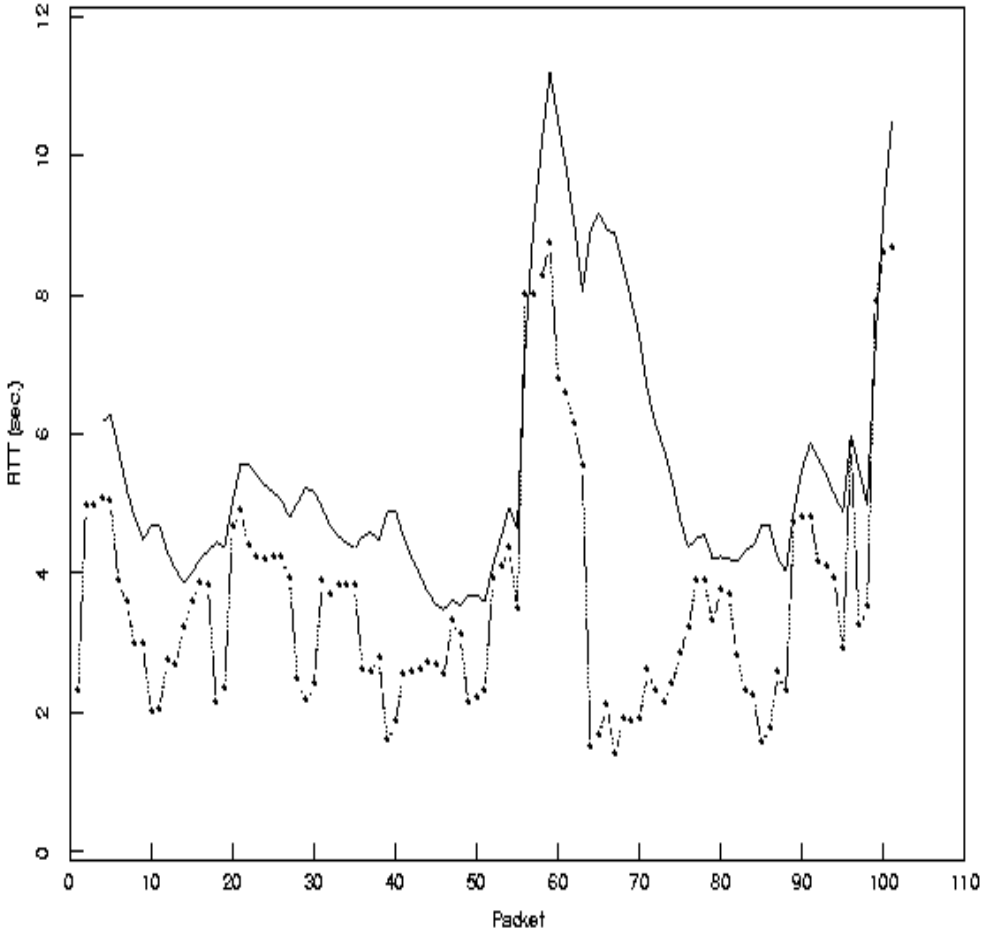
    $> 2 R_i$

    $> R_{i+1}$

Performance of retransmit timer



RFC 793

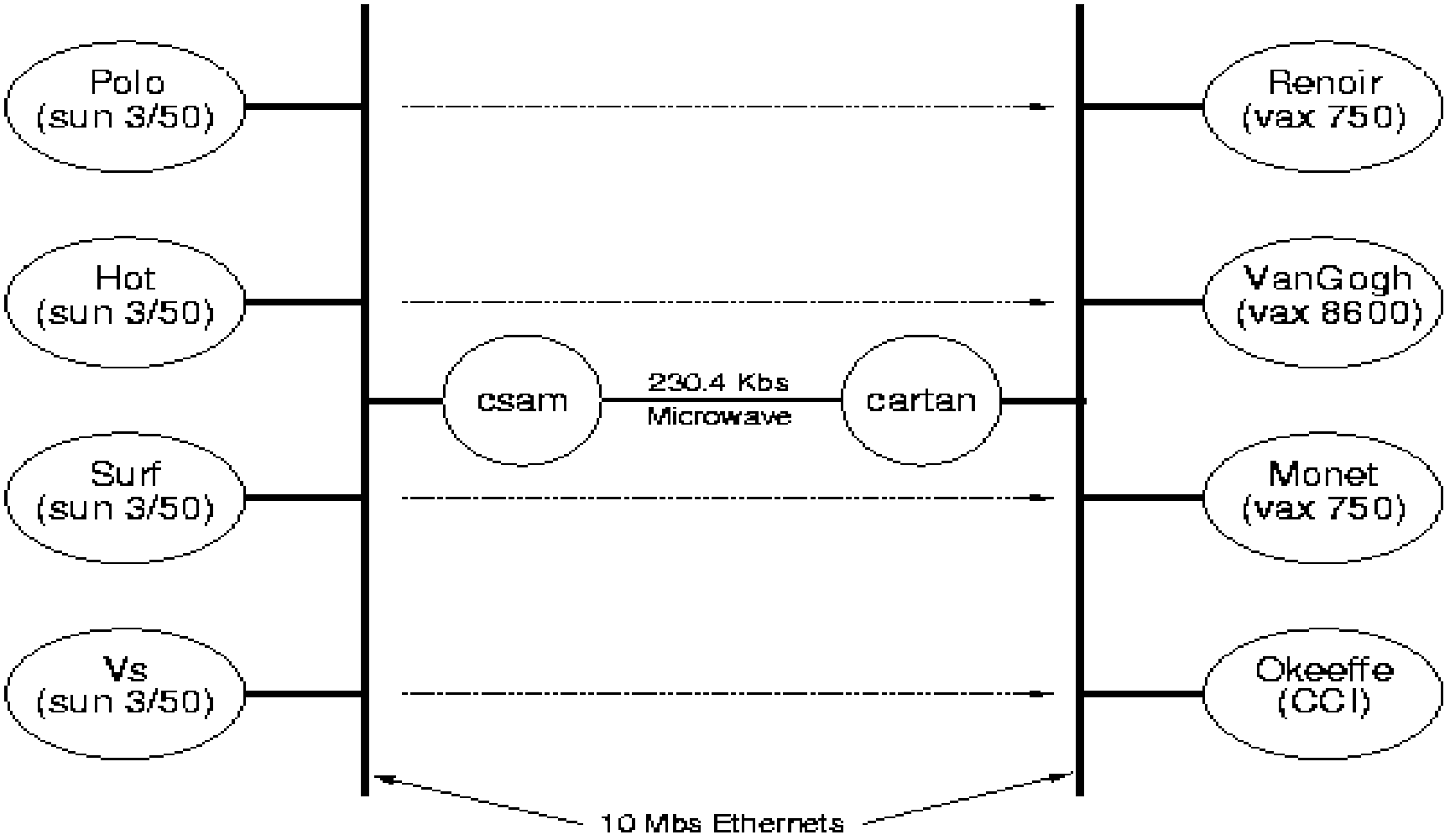


Mean+Variance

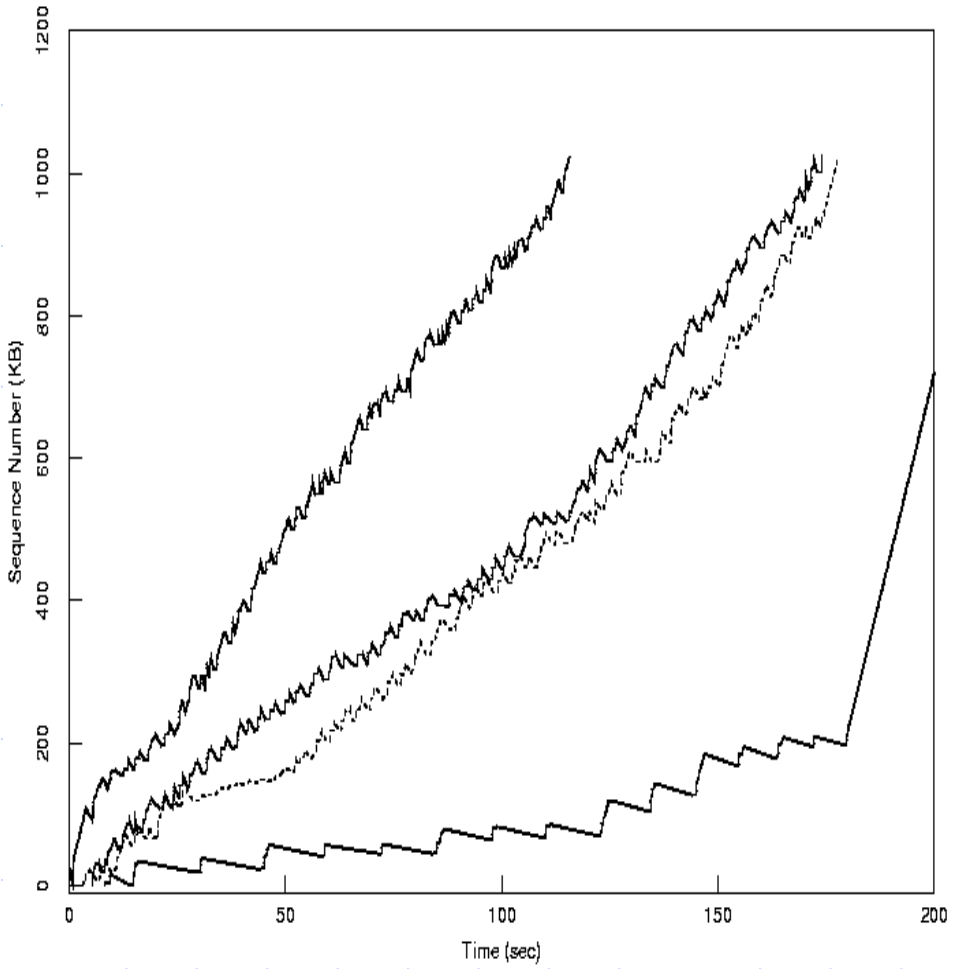
Adapting to the path: congestion avoidance

- Signal the congestion
- Sender's policy
- Congestion avoidance
  - On any timeout, $cwnd = cwnd/2$
  - On each ack for new data, $cwnd += 1/cwnd$
  - Sending the minimum of the receiver's advertised window and *cwnd*

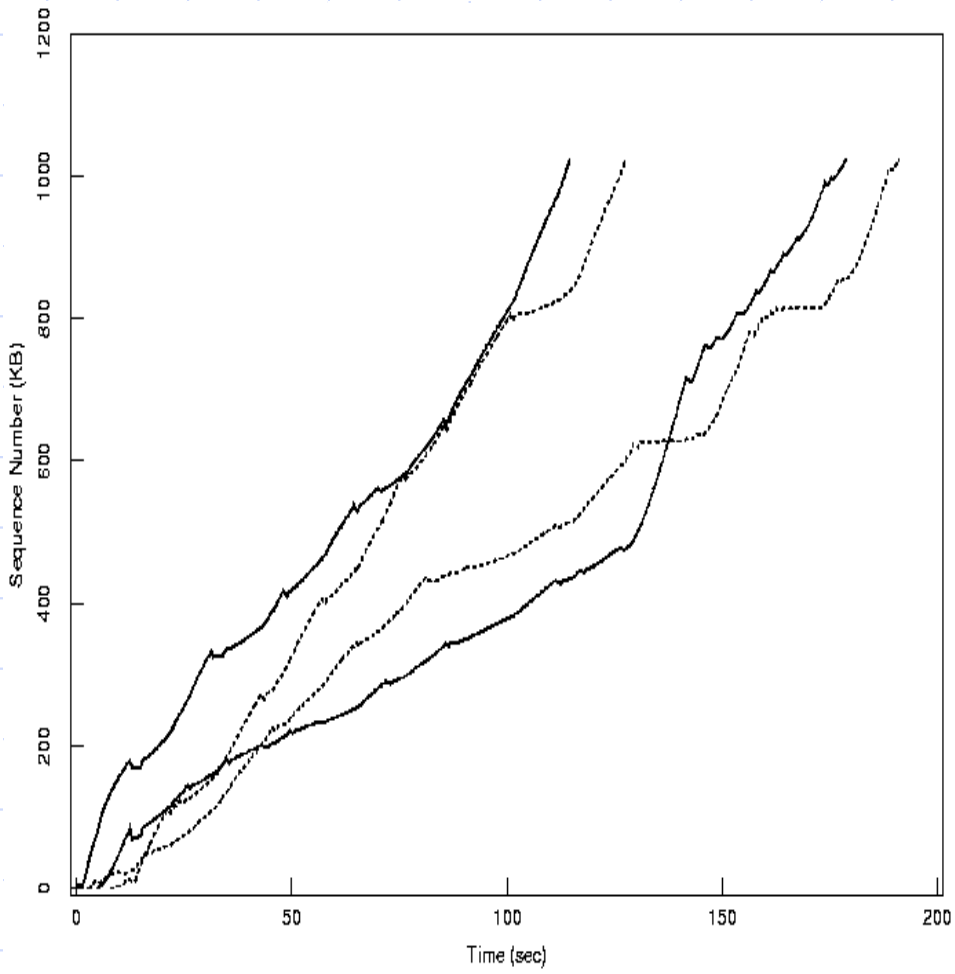
Multiple conversation test setup



Multiple, simultaneous TCP

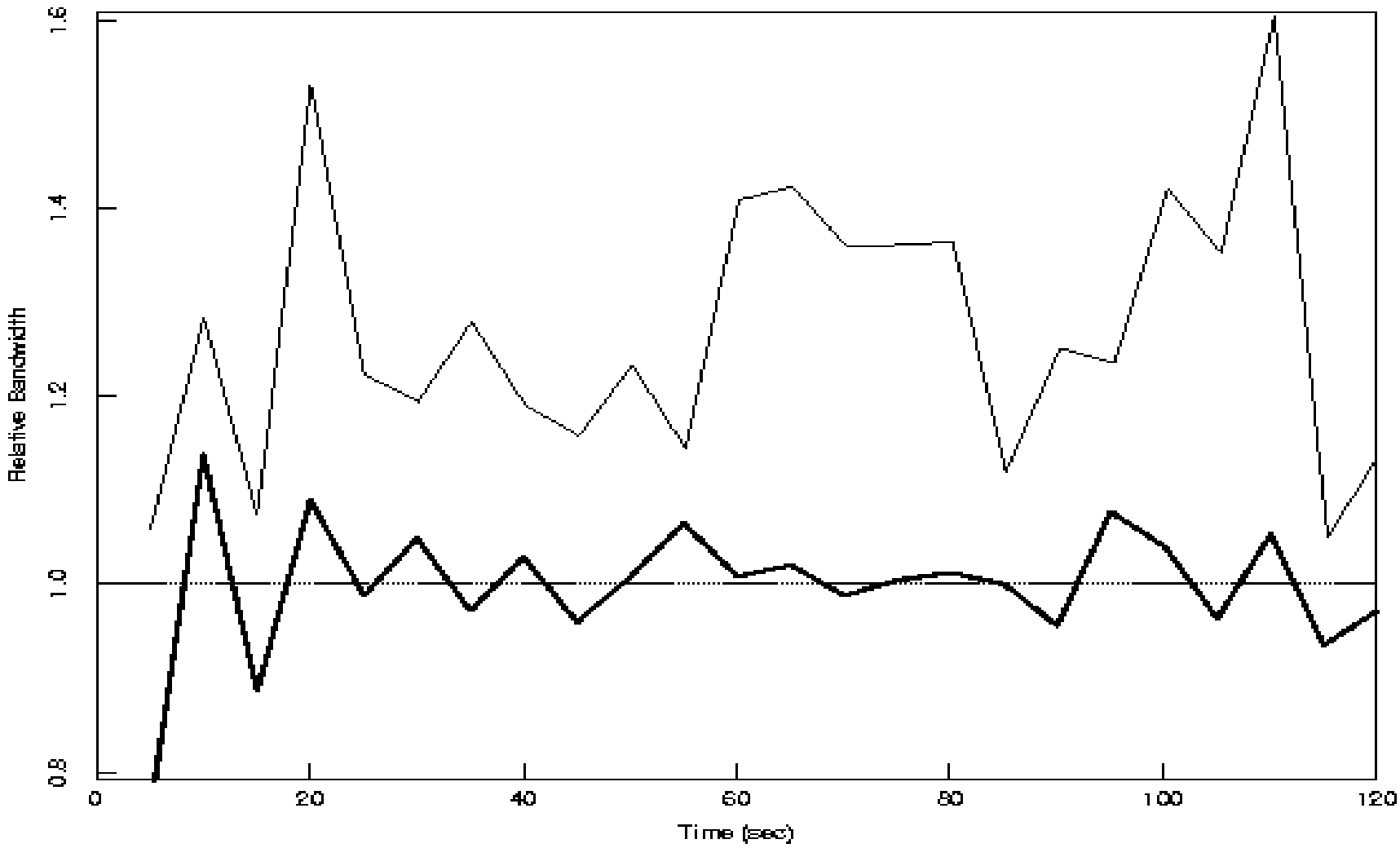


w/o congestion avoidance

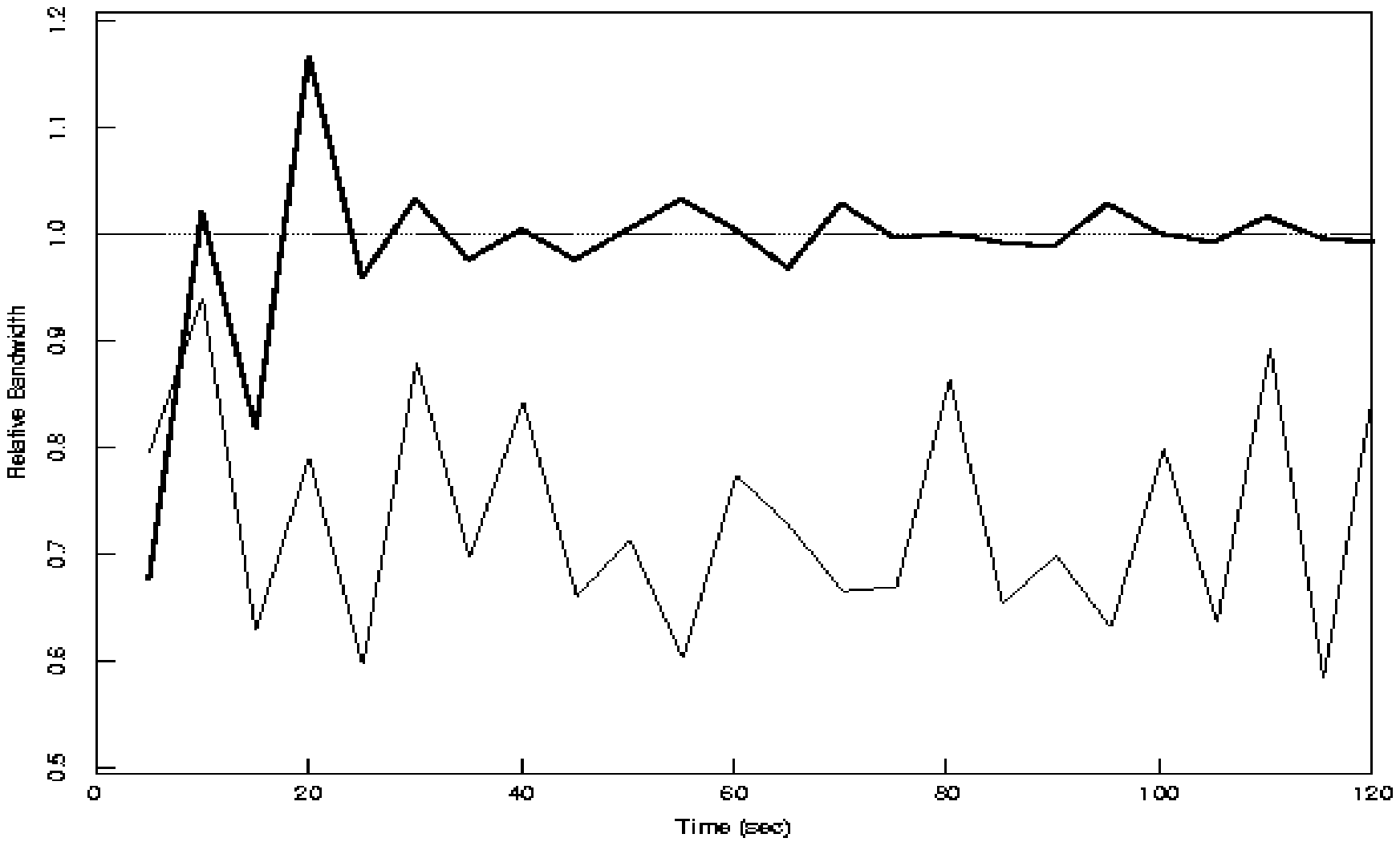


with congestion avoidance

Total bandwidth used by old and new TCPs



Effective bandwidth of old and new TCPs



Slow-start with Congestion avoidance

```
if (cwnd < ssthresh)
        cwnd += 1;
else
        cwnd += 1/cwnd;
```

Future work:
gateway side of congestion control

- Ensure fair sharing of the capacity
- Using packet drops as a congestion signal