New TCPs

염익준

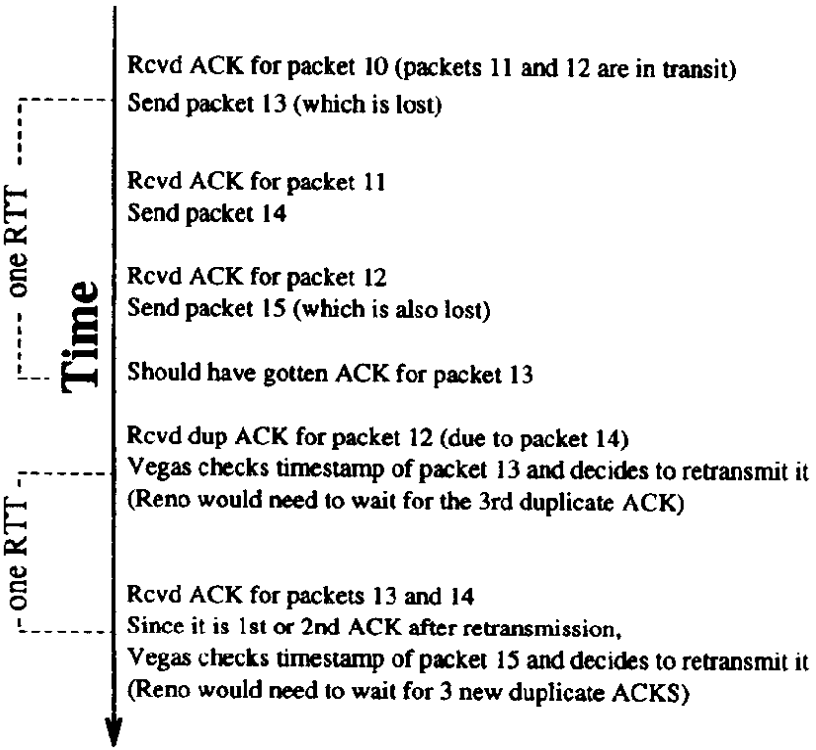
TCP Vegas

Problems with TCP Reno

- TCP Reno uses two mechanisms to detect packet losses:
  - Triple duplicated ACKs
  - Timeout
- Triple duplicated ACKs often fails to be triggered due to either
  - Losses in burst
  - Small window
- Timeout needs unnecessarily long delay.
- Congestion control in Reno
  - Need to create packet losses to find the available bandwidth of the connection
  - Continually congesting the network
  - Creating losses for other connections sharing the link.
  - Oscillations

New Retransmission Mechanism

- Upon receiving a duplicated ACK or an ACK for a retransmitted packet, Vegas checks the time interval after the previous packet of the just ACKed packet was sent.
- If the time interval is greater than the timeout value, then the packet is retransmitted without waiting triple duplicated ACKs.
- Only decreasing CWND if the retransmitted packet was sent after the last decrease.

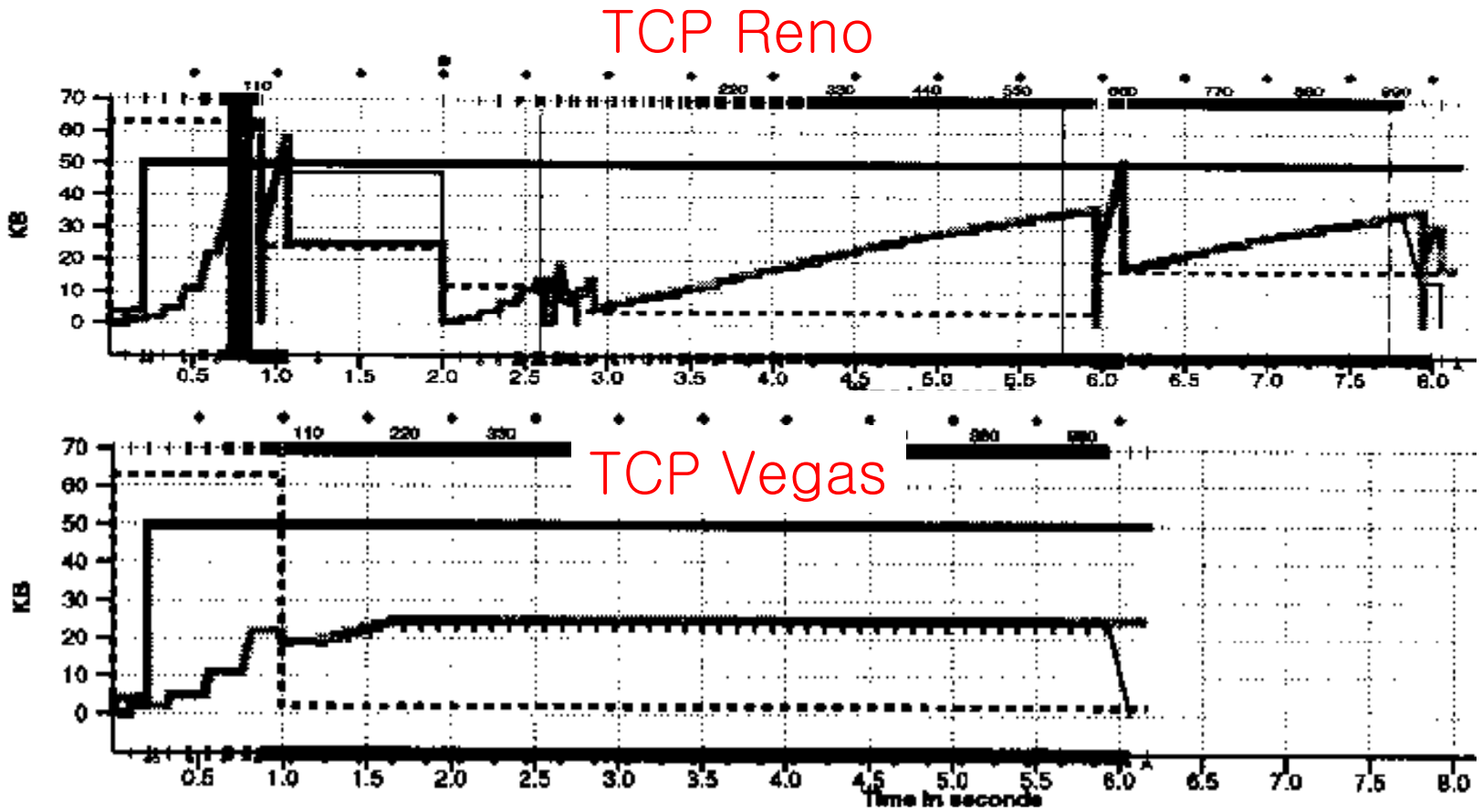


Congestion Avoidance in Vegas

- Trying to alleviate TCP's oscillation.
- Estimating available bandwidth based on the difference between expected and actual flows rates.

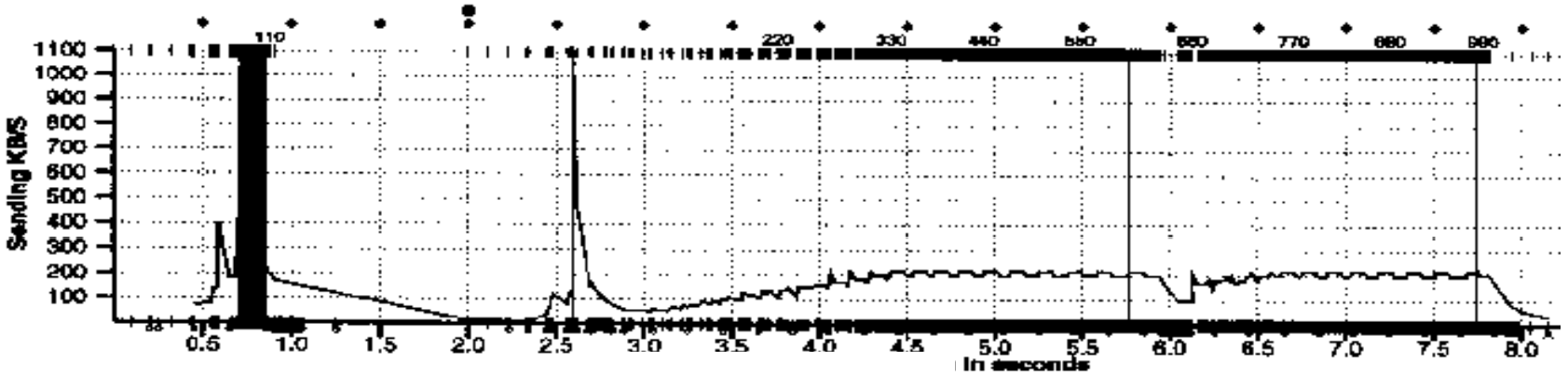
```
Upon an ack arrivals:
  Diff = (Expected - Actual) BaseRTT
    where Expected = CWND/BaseRTT and
          Actual   = CWND/ActualRTT.
              CWND + 1 if Diff < α
    CWND =    CWND - 1 if Diff > β
              CWND     otherwise
```

Congestion Window

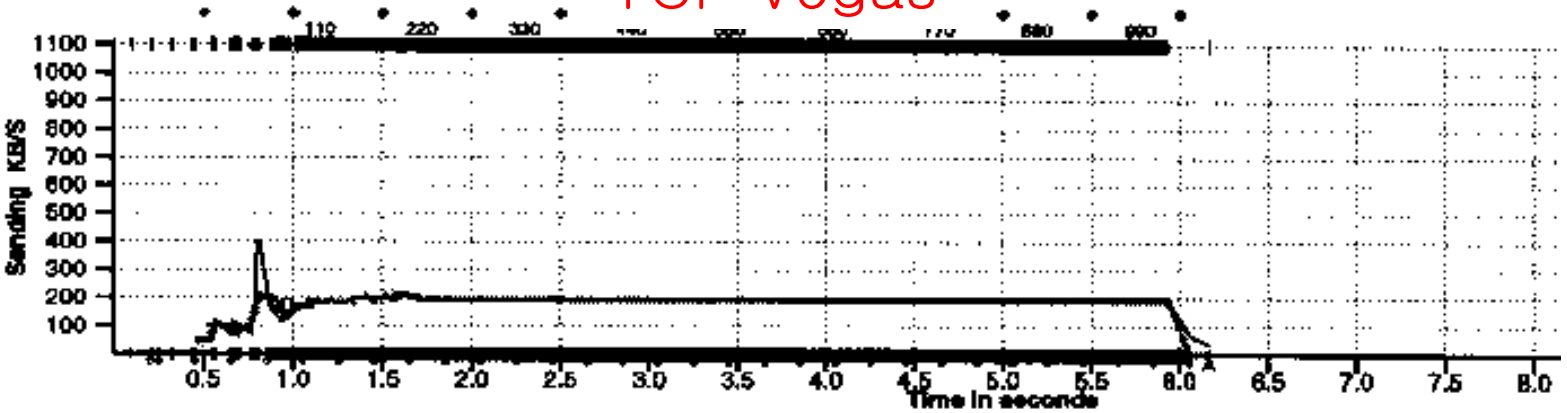


Sending Rate

TCP Reno

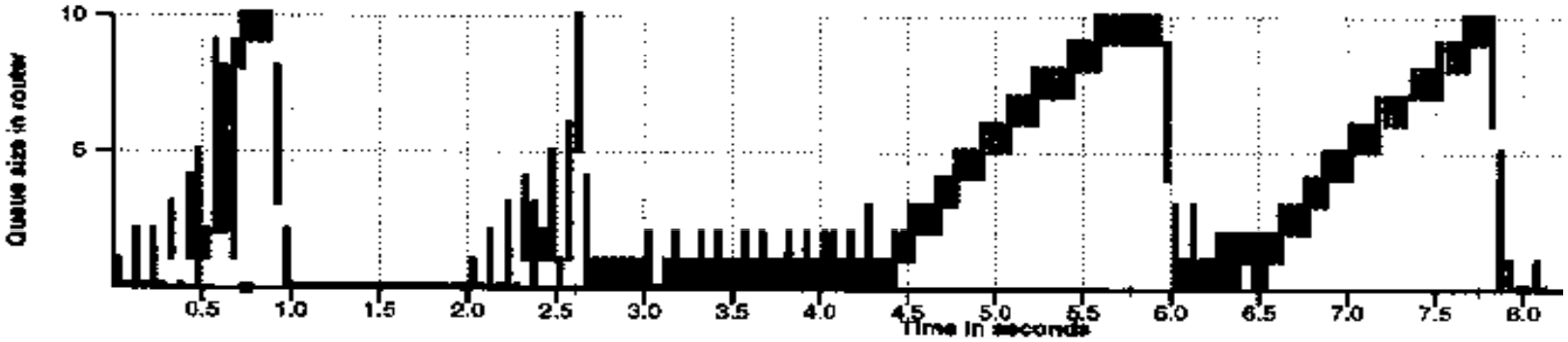


TCP Vegas

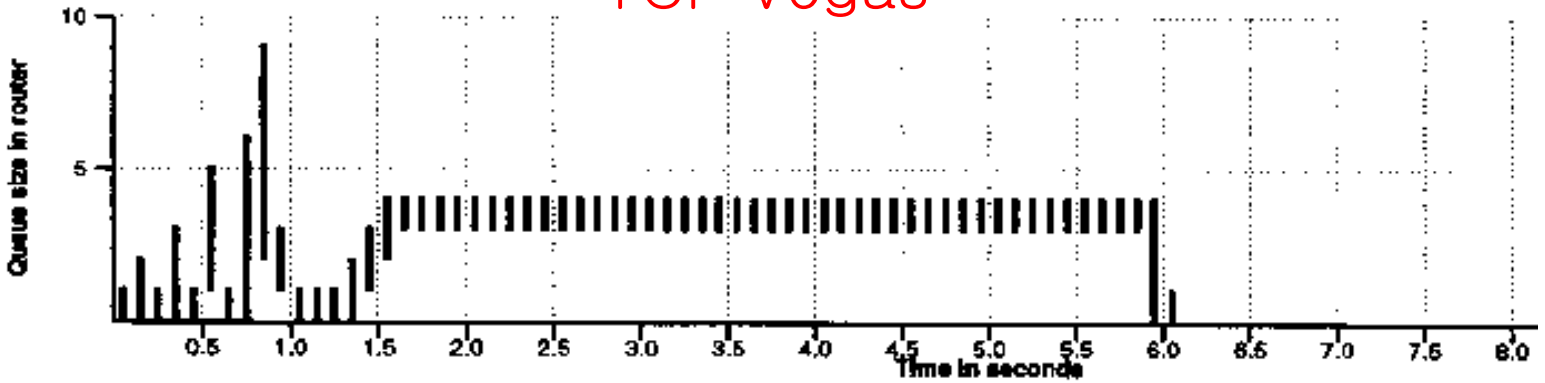


Queue Length

TCP Reno



TCP Vegas



Slow-Start in Vegas

- Slow start in Reno pumps packets in burst and eventually creates packet losses.
- Vegas proposes two alternatives:
  - Maintain a window for two RTTs and compare them to decide if increasing the window.
  - Send probe packets to estimate the available bandwidth similarly in packet-pair technique.

TCP Westwood

Problem of TCP in Wireless Links

- A packet loss is considered as an indication of congestion.
- Reacting to the loss by reducing the window.
- In wireless links, not all the losses mean the congestion.
- Some of them may be due to channel errors.
- may suffer from under-utilization.
- Ideal solution
  - keep the window upon losses due to channel errors.
  - reduce the window only upon congestion losses.
- how to distinguish them??

TCP Westwood

- do not attempt to distinguish the reasons of packet losses.
- instead, measures the available bandwidth
- and uses it for resizing the window upon packet losses.
- with a hope to keep the window upon channel errors since the available bandwidth does not be changed.

Bandwidth Estimation

- b = the amount of data sent in an RTT / RTT
- uses a low pass filter to smooth out.
- no additional traffic.

Congestion Control

- upon fast retransmission
  - ssthreshold = (BWE*RTTmin)/segment size
  - cwin = min(cwin/2, ssthreshold)
- upon timeout
  - ssthreshold = (BWE*RTTmin)/segment size
  - cwin = 1

TCPW Fairness

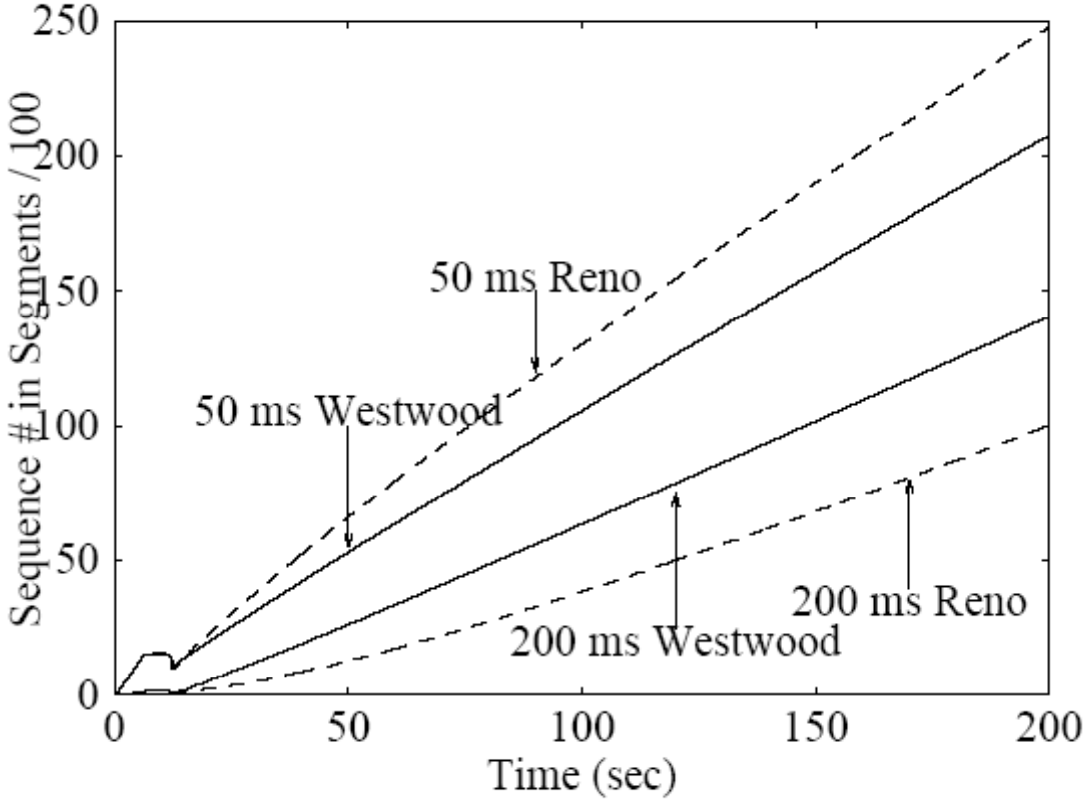


Figure 4. Sequence numbers vs. time for long and short RTT connections without RED

Over Lossy Link

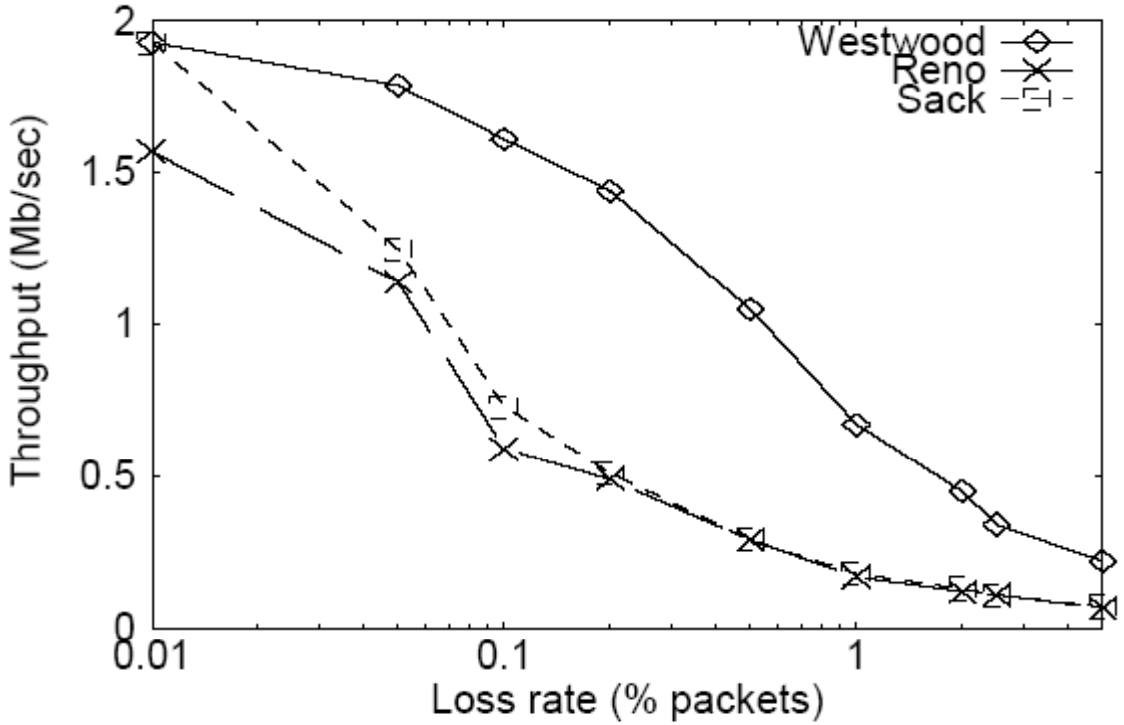


Figure 9. Throughput vs. error rate of the wireless link

CWIN and SSTHRESHOLD

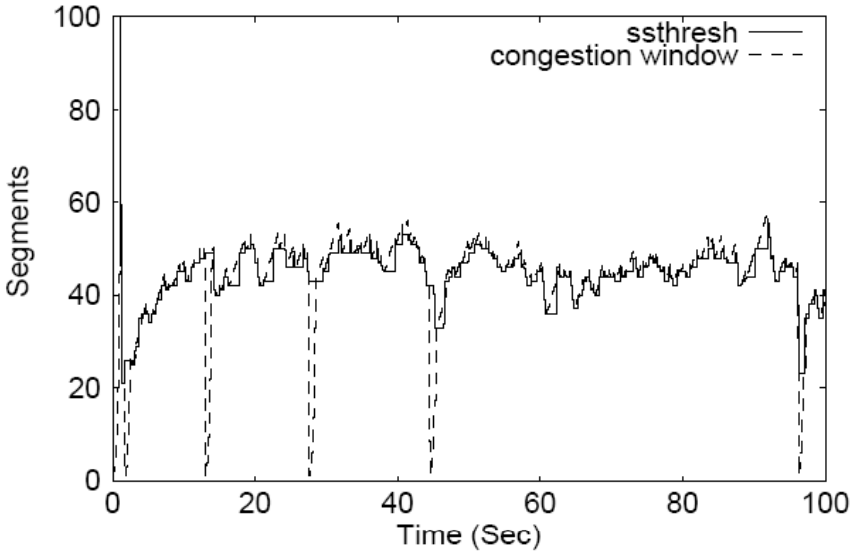


Figure12. TCP Westwood over lossy link—cwin and ssthresh

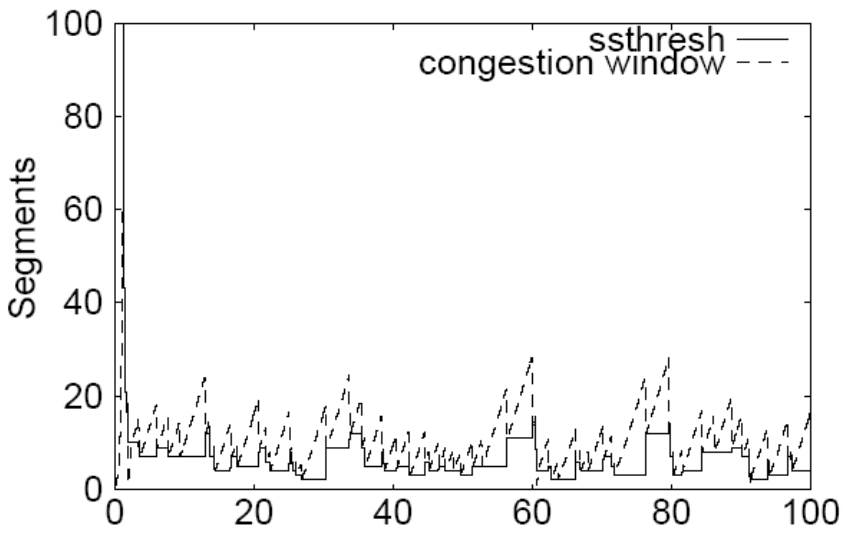


Figure 13. TCP Reno over lossy link—cwin and ssthresh

Discussion