

TCP Versions: Reno

Kameswari Chebrolu

Break

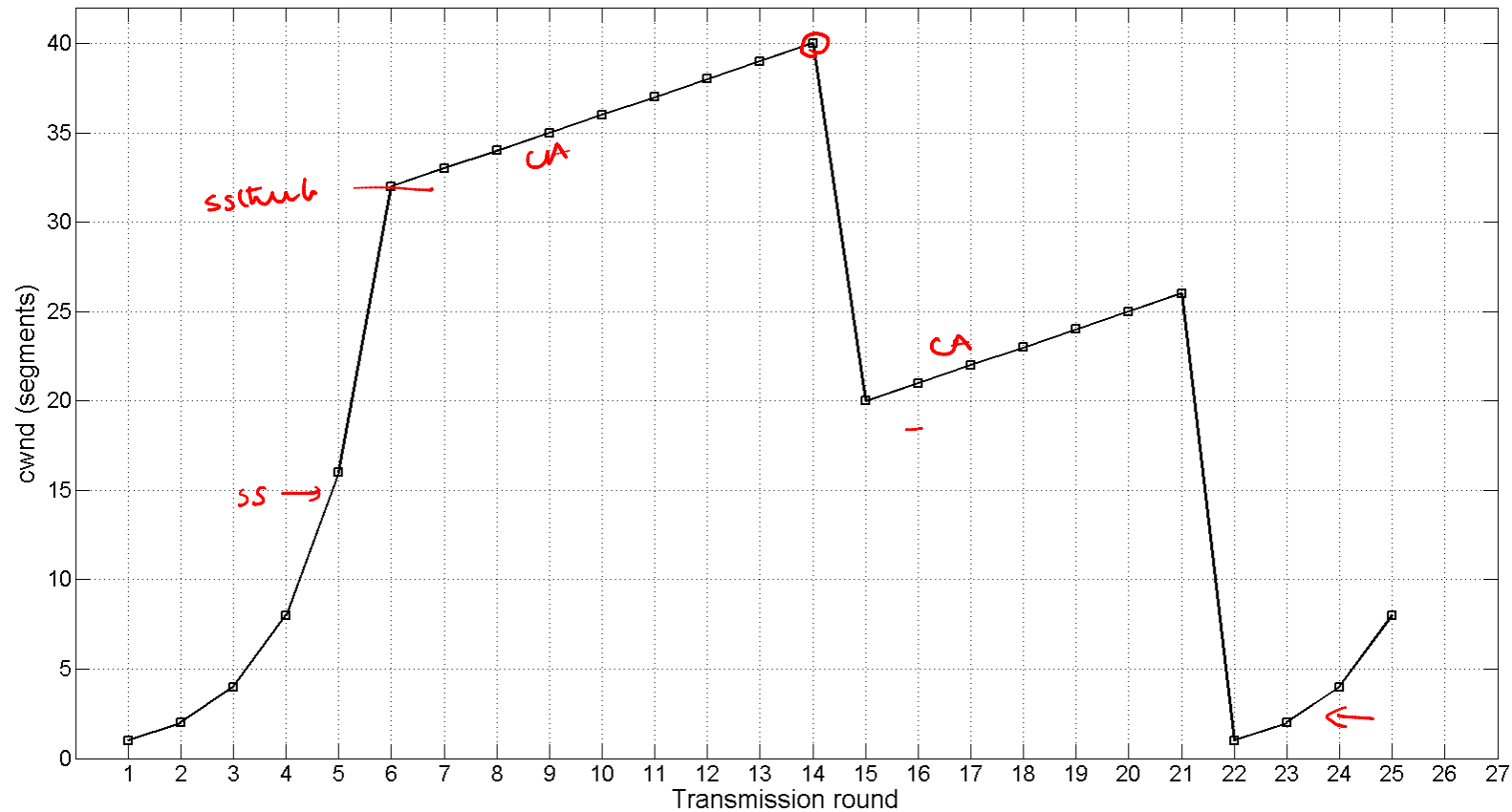


TCP Reno

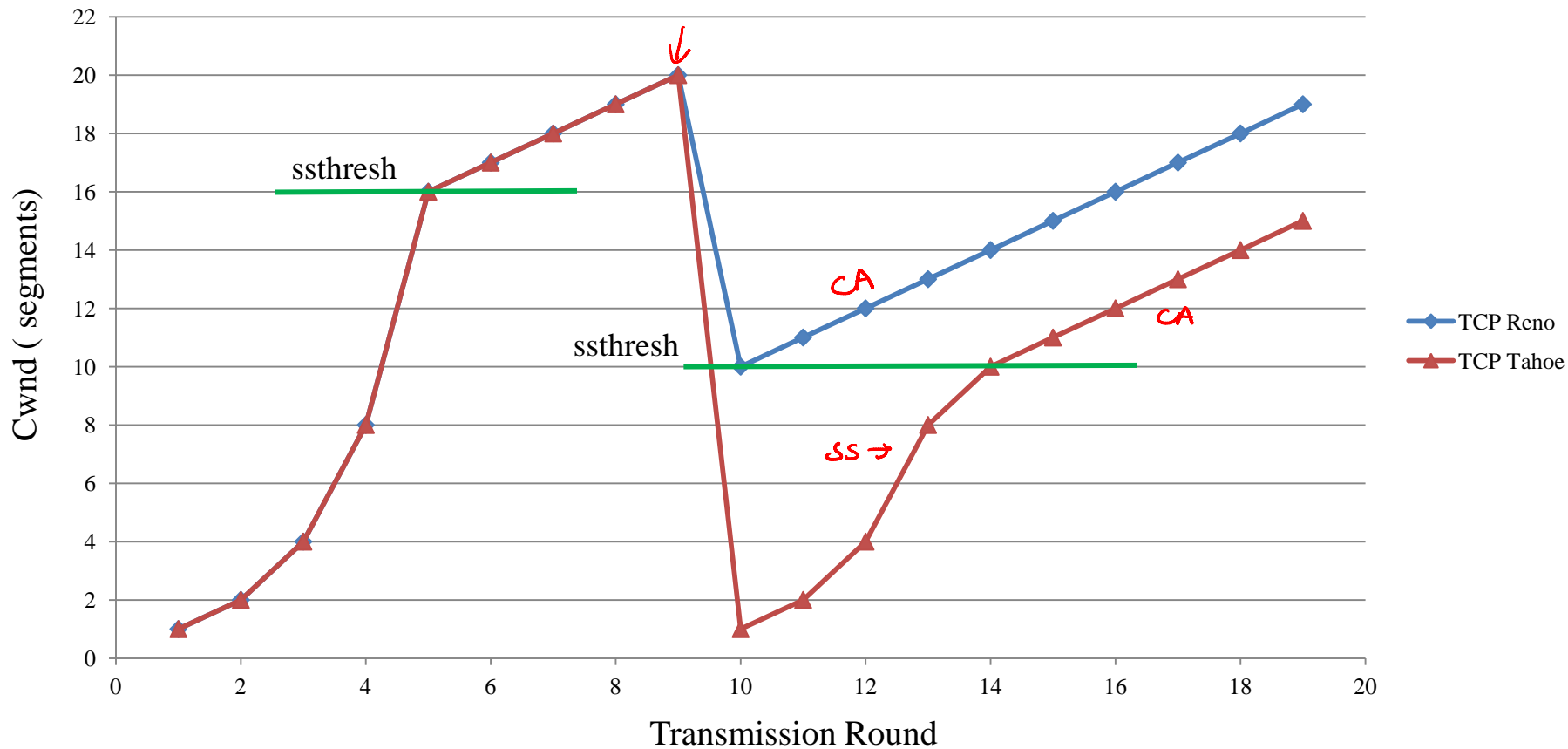
- Incorporates two new mechanisms: Fast Retransmit and Fast Recovery
- Fast Retransmit: Retransmit packet at sender after 3 duplicate acks → Packet is lost
reorder ← 1 or 10?
– Cut the window by half (loss event)
– Avoids having to time-out which keep the link idle for longer duration

Fast Recovery

- On 3rd dupack, retransmit packet, $ssthresh = \max(\underline{cwnd/2}, 2)$; $cwnd = ssthresh + \underline{3}$
- Another dupack, $cwnd = \underline{cwnd} + 1$; transmit packet if allowed by $cwnd$
- On ack acknowledging new data, $cwnd = \underline{ssthresh}$, invoke congestion avoidance (linear increase in $cwnd$ now on)

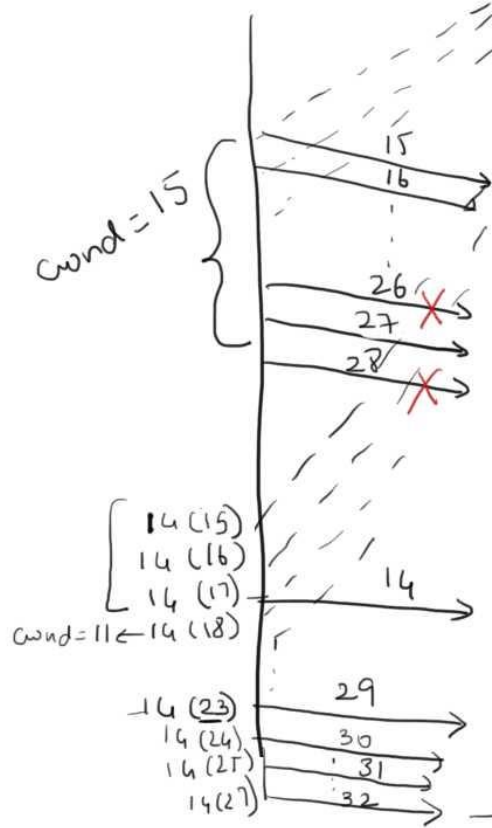
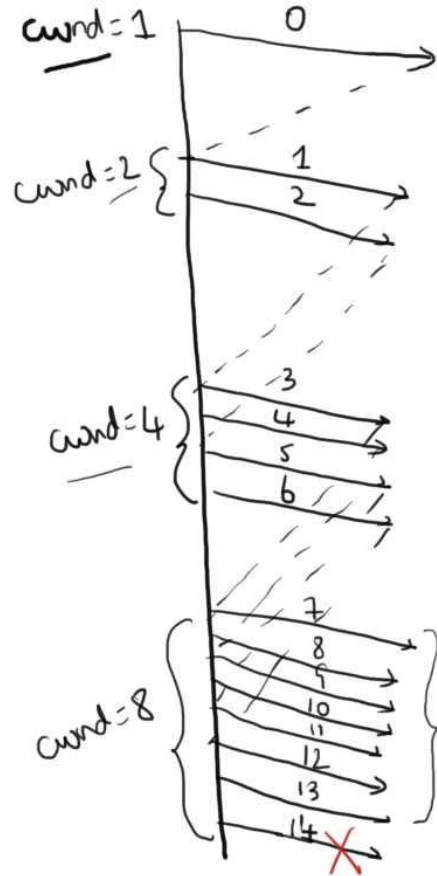


Comparison

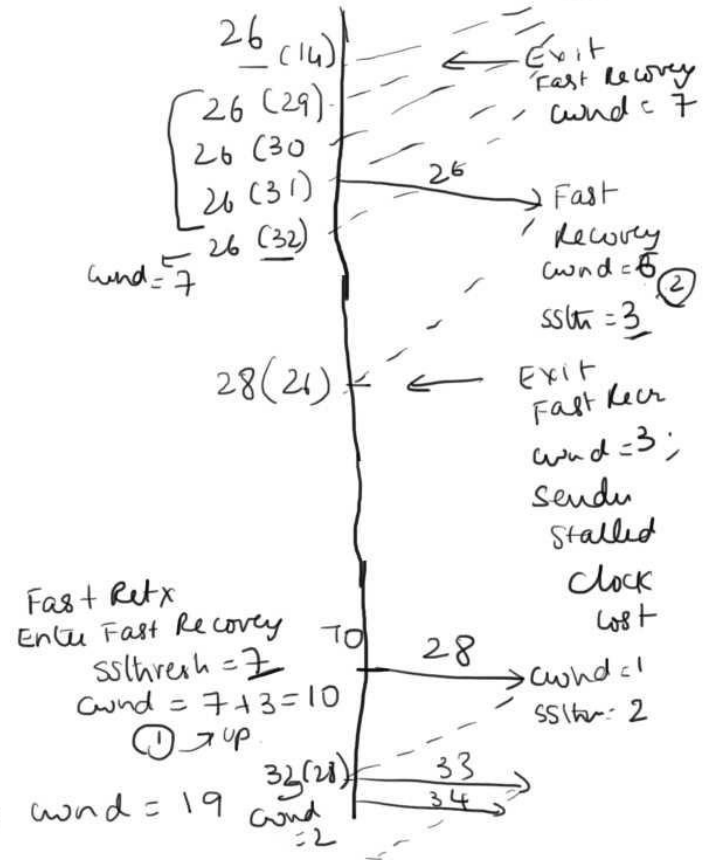


x (4) → who generated
L) asking for

TCP Reno



(Not drawn to scale) ① 14, 15-28
② 26 to 32
Pkt in Pipe 15



Other Versions

- TCP NewReno: Handles multiple losses per congestion window better (high loss rate scenarios)
- TCP Vegas: Uses packet delay to signal congestion than loss event
- TCP SACK: Employs selective acknowledgments

Summary

- TCP Tahoe: Go-back-N with slow start and congestion avoidance *Fast Retransmit*
 - Loss recovery slow; timeouts drain pipe
- TCP Reno: improves upon Tahoe
 - Better loss recover via duplicate acks (fast retransmit)
 - Prevents draining of pipe after fast retransmit (avoids slow-start)
- Ahead: Sliding window, flow control, and other miscellaneous things