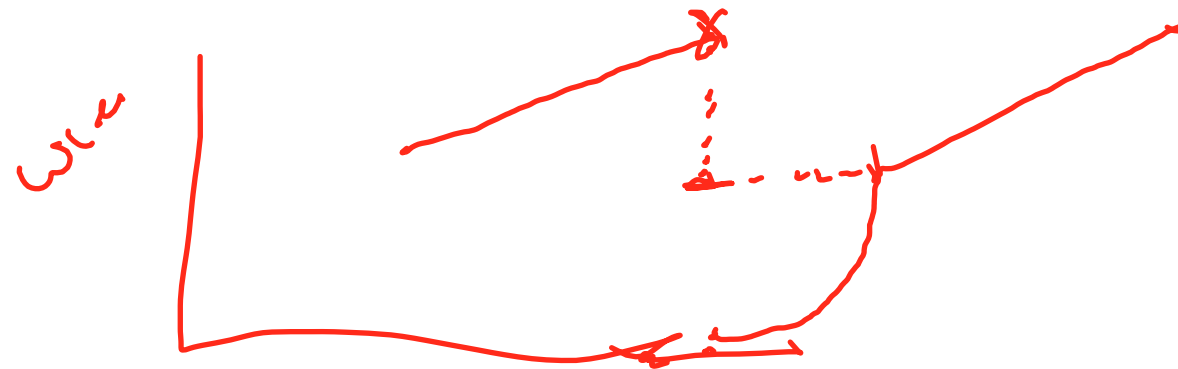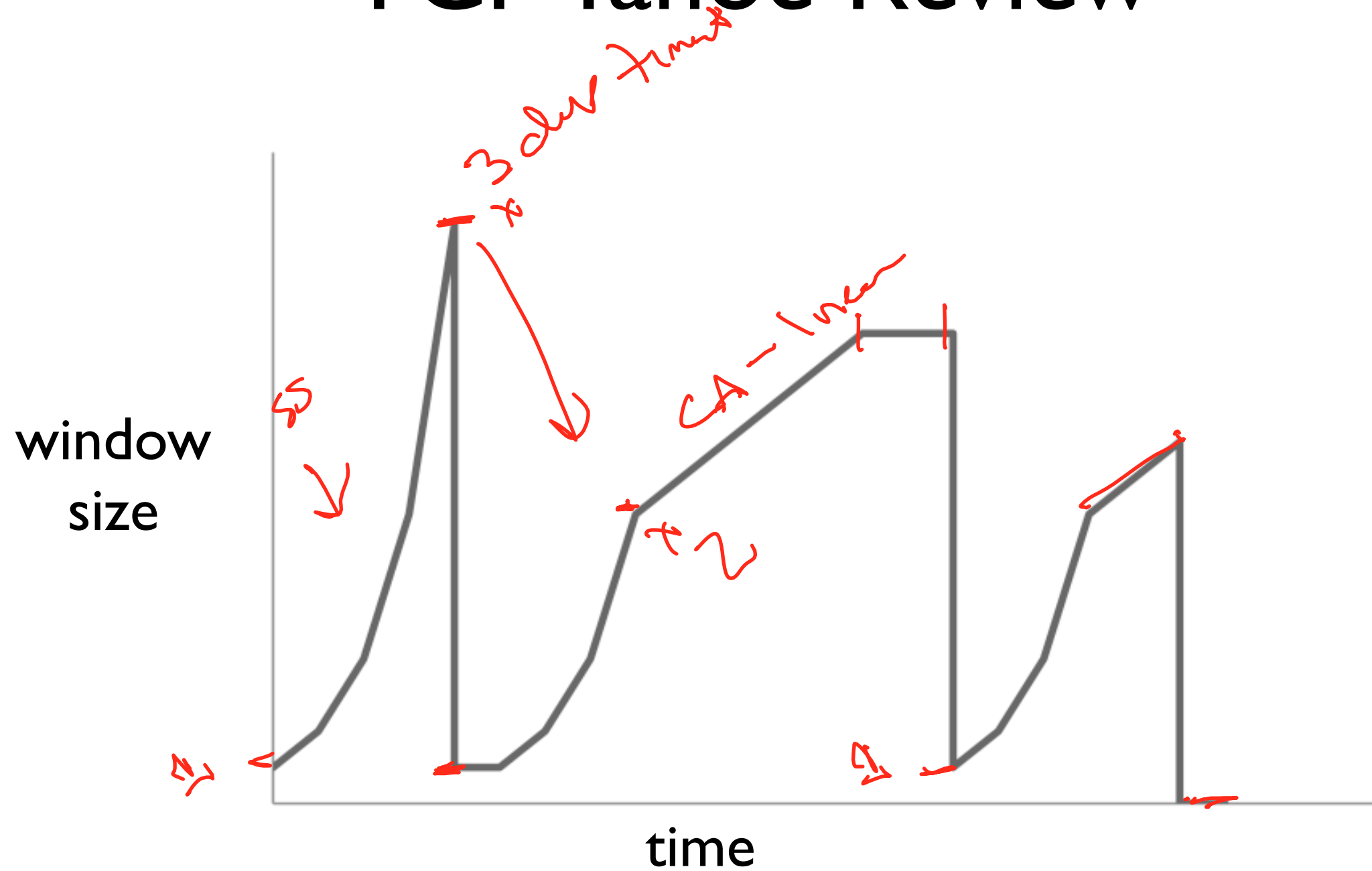# Congestion Control III

Performance improvements: TCP Reno, TCP NewReno

TCP Tahoe

- On timeout or triple duplicate ack (implies lost packet)
  - ▸ Set threshold to congestion window/2
  - ▸ Set congestion window to 1
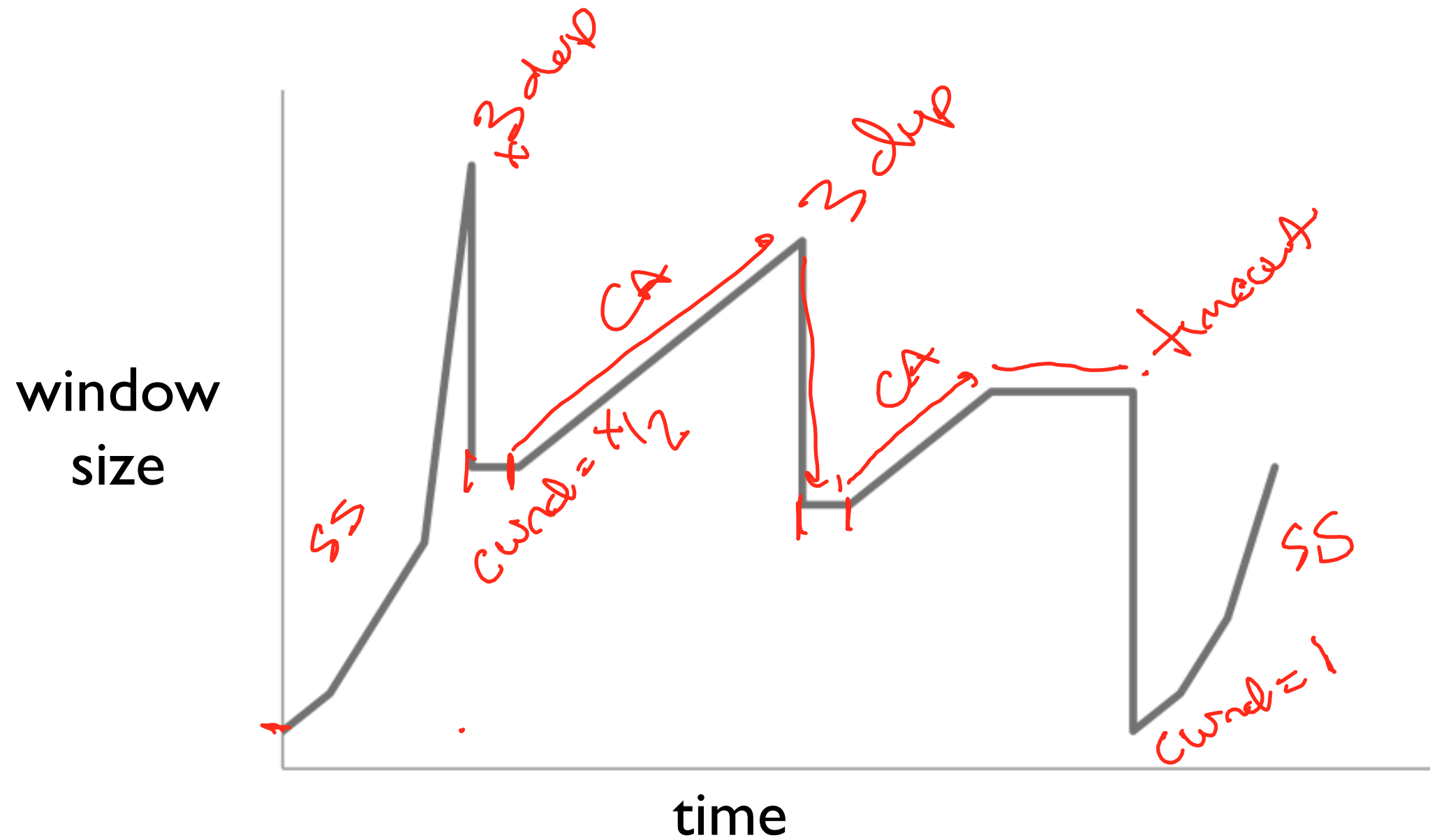 - ▸ Enter slow start state

TCP Tahoe Review
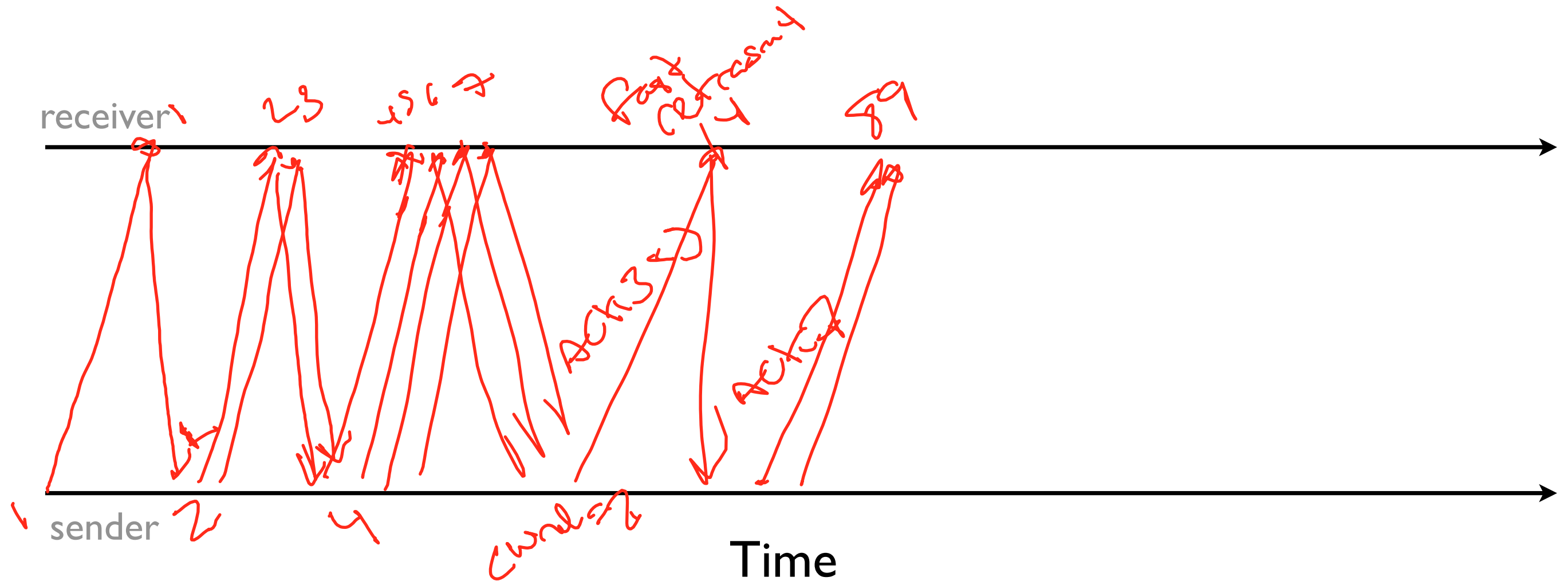

window size

time

TCP Reno

- Same as Tahoe on timeout
- On triple duplicate ack
  - ▸ Set threshold to congestion window/2
  - ▸ Set congestion window to congestion window/2 (fast recovery)
  - ▸ Retransmit missing segment (fast retransmit)
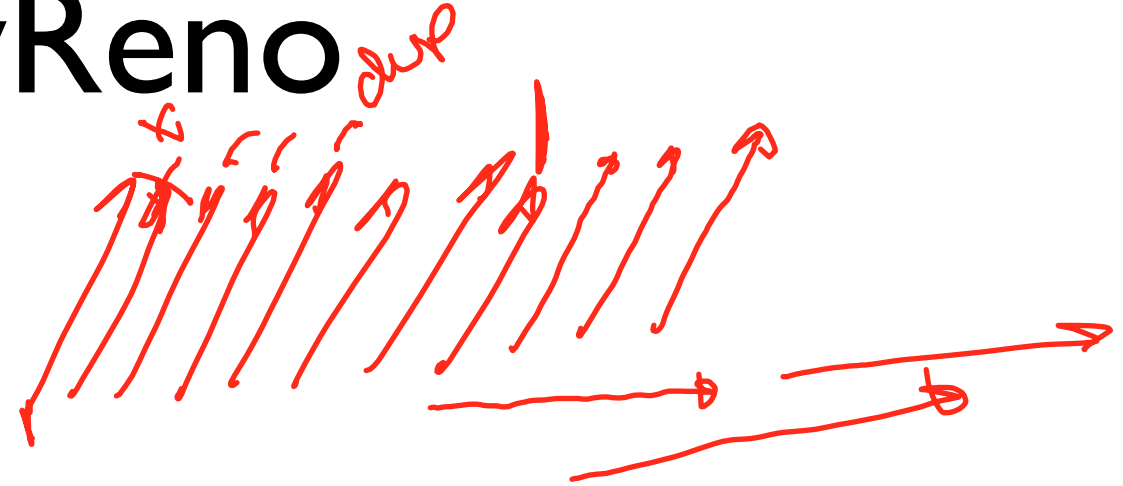 - ▸ Stay in congestion avoidance state

TCP Reno

window
size

time

TCP Reno Example

6

TCP NewReno

- Same as Tahoe/Reno on timeout
- During fast recovery
  - ▸ Keep track of last unacknowledged packet when entering fast recovery
  - ▸ On every duplicate ack, inflate congestion window by maximum segment size
  - ▸ When last packet acknowledged, return to congestion avoidance state, set cwnd back to value set when entering fast recovery
  - ▸ Start sending out new packets while fast retransmit is in flight

TCP NewReno Behavior
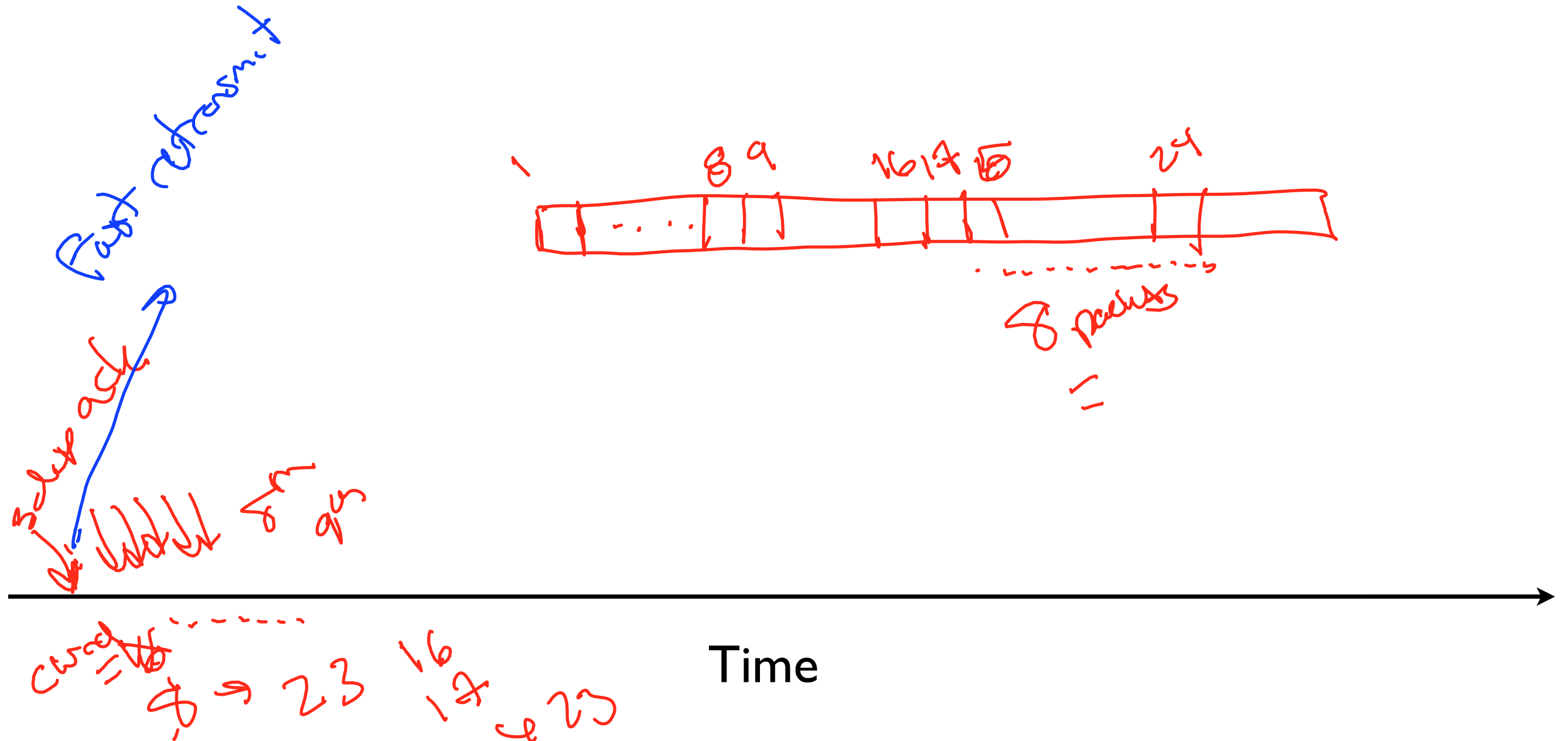
Congestion Control

- One of the hardest problems in robust networked systems
- Basic approach: additive increase, multiplicative decrease
- Tricks to keep pipe full, improve throughput
  - Fast retransmit (don't wait for timeout to send lost data)
  - Congestion window inflation (don't wait an RTT before sending more data)