

TCP



Matthew Roughan
AT&T Labs - Research
roughan@research.att.com

Ashok Erramilli
Qnetworx
Erramilli@qnetworx.com

Darryl Veitch
EMUlab – Ericsson and the University of Melbourne
d.veitch@ee.mu.oz.au

TCP Congestion Control

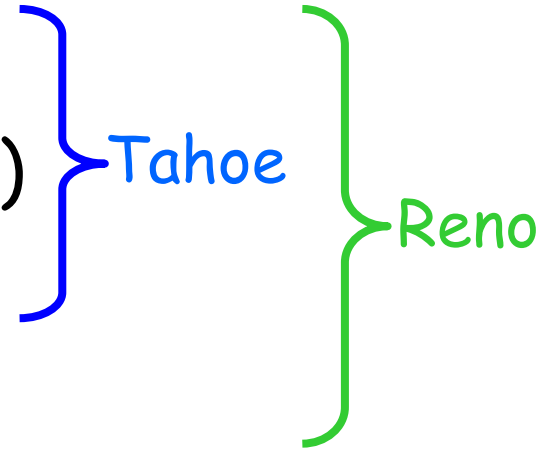
- Has four main parts

- Slow Start (SS)

- Congestion Avoidance (CA)

- Fast Retransmit

- Fast Recovery



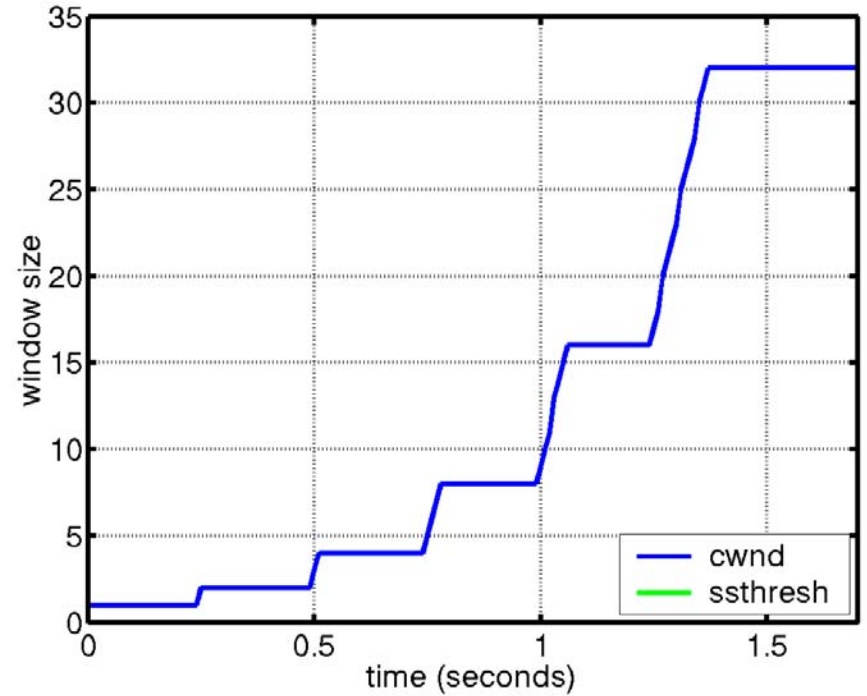
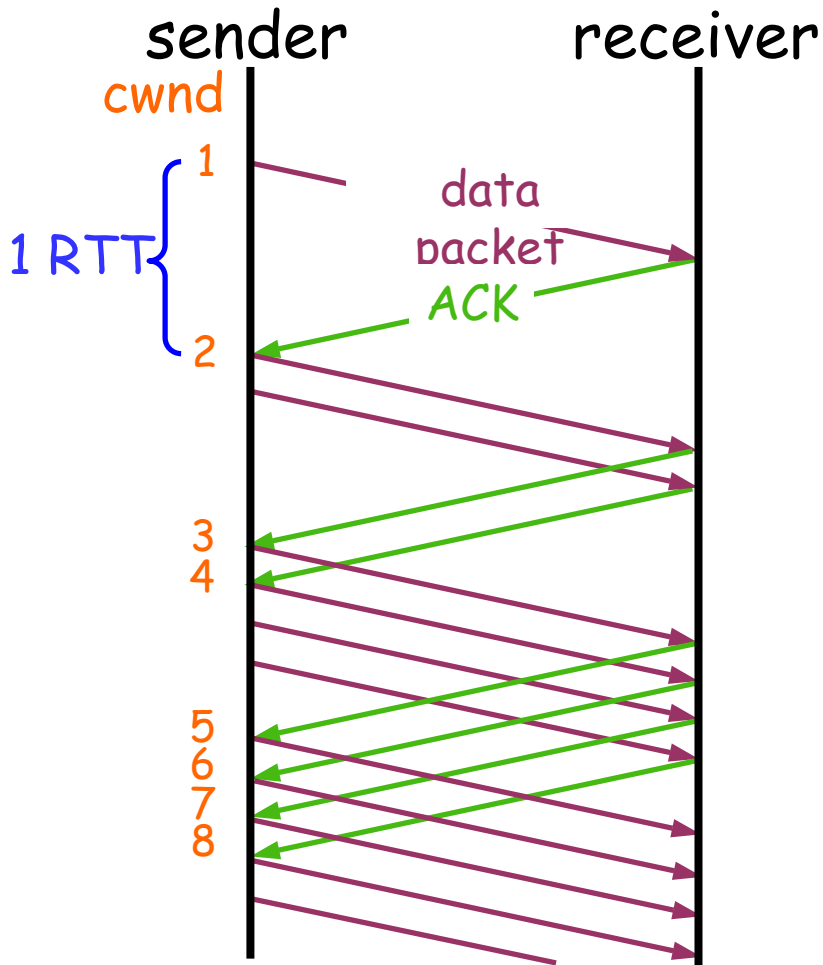
- **ssthresh**: slow start threshold
determines whether to use SS or CA

- Assume packet losses are caused by congestion

Slow Start

- Start with $cwnd = 1$ (slow start)
- On each successful ACK increment $cwnd$
 $cwnd \leftarrow cwnd + 1$
- Exponential growth of $cwnd$
each RTT: $cwnd \leftarrow 2 \times cwnd$
- Enter **CA** when $cwnd \geq ssthresh$

Slow Start

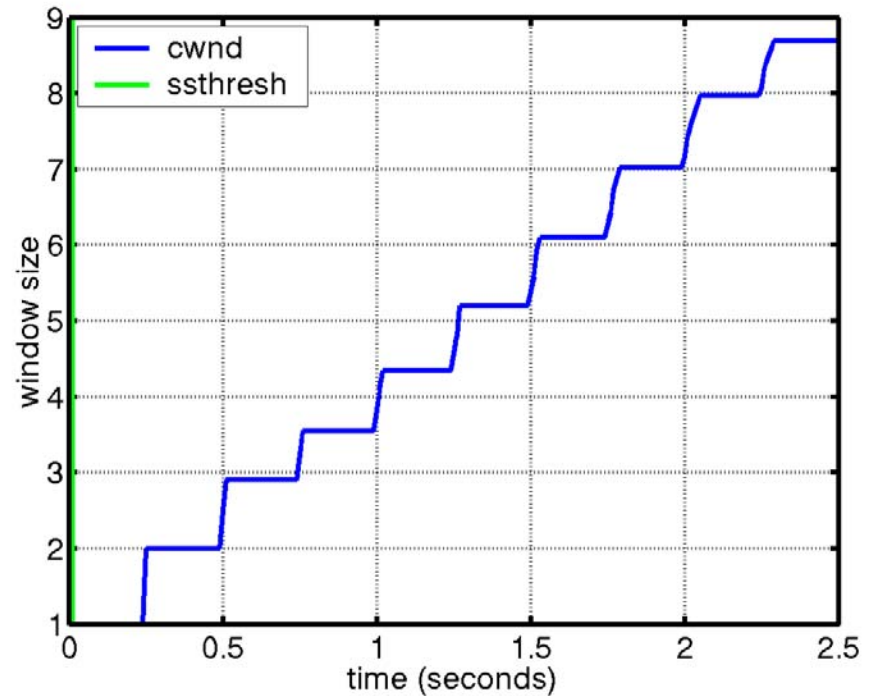
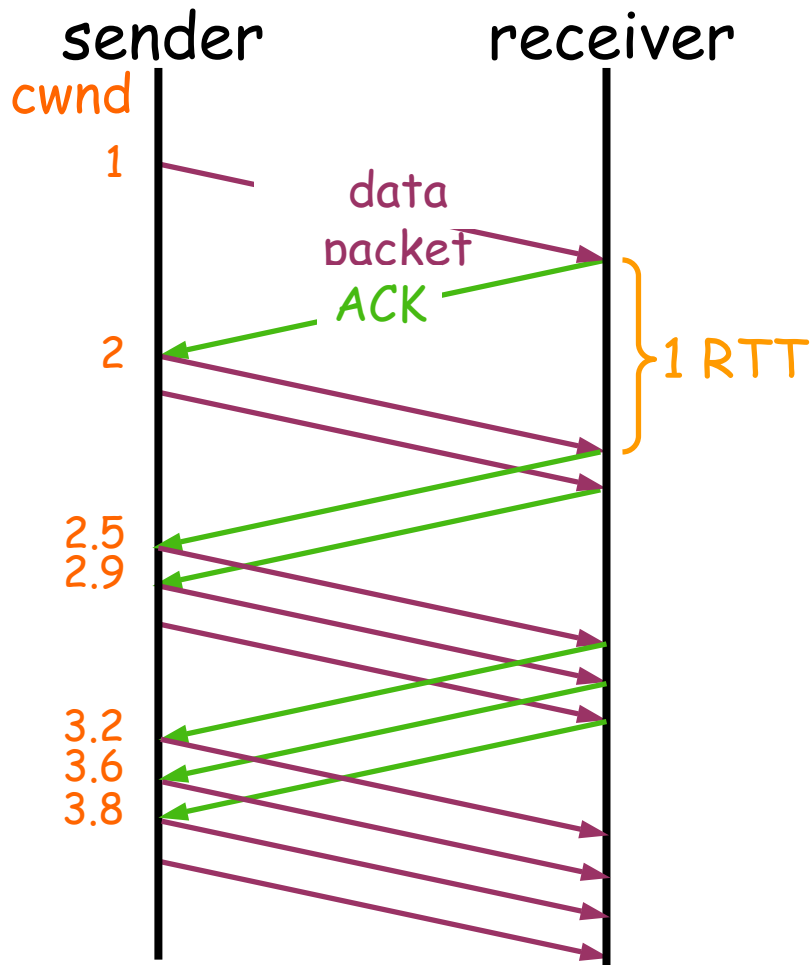


$cwnd \leftarrow cwnd + 1$ (for each ACK)

Congestion Avoidance

- Starts when $cwnd \geq ssthresh$
- On each successful ACK:
 $cwnd \leftarrow cwnd + 1/cwnd$
- Linear growth of $cwnd$
each RTT: $cwnd \leftarrow cwnd + 1$

Congestion Avoidance

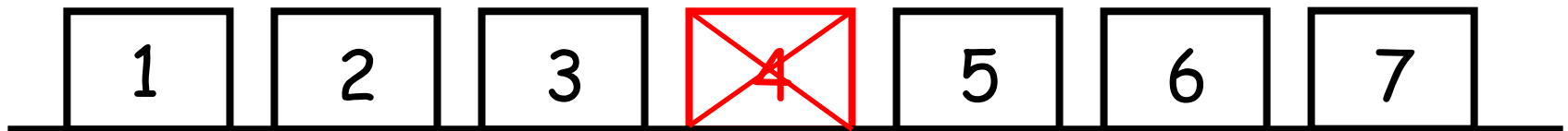


$cwnd \leftarrow cwnd + 1/cwnd$ (for each ACK)

Packet Loss

- **Assumption:** loss indicates congestion
- Packet loss detected by
 - Retransmission TimeOuts (RTO timer)
 - Duplicate ACKs (at least 3)

Packets



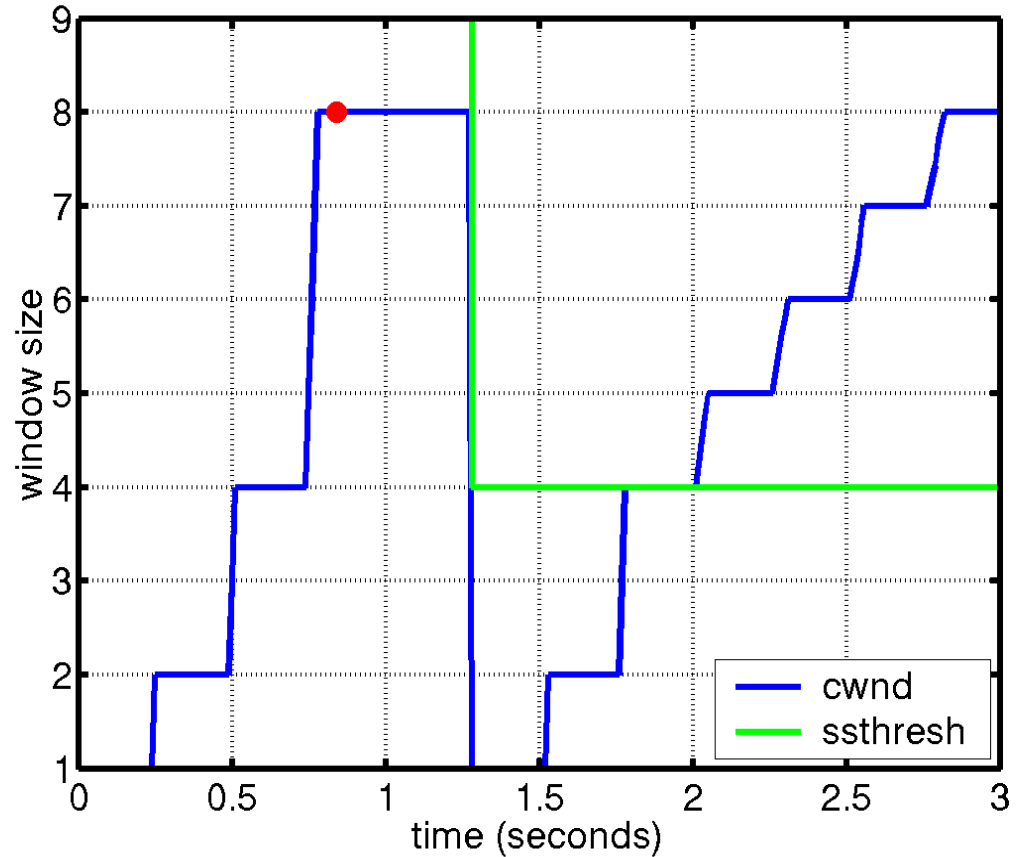
Acknowledgements



Timeout

$ssthresh \leftarrow cwnd/2$

$cwnd = 1$



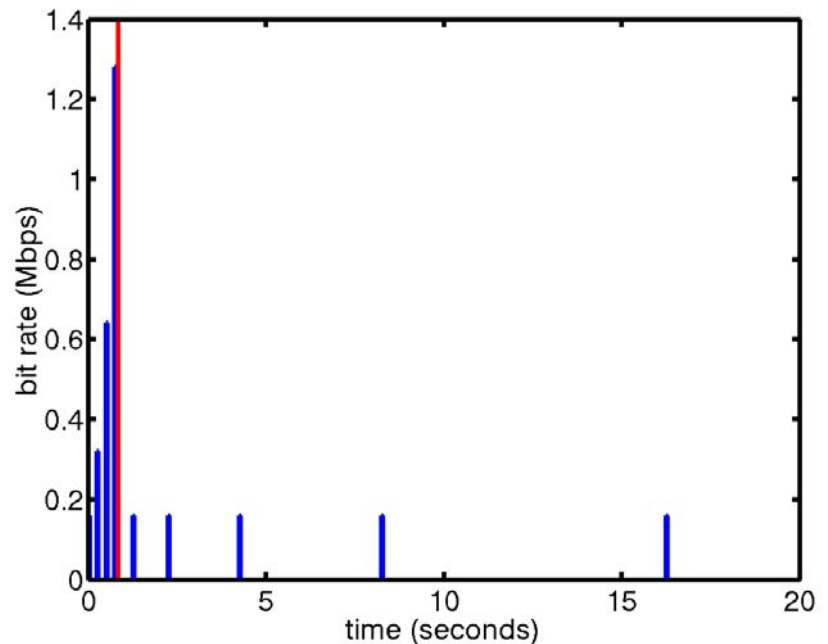
Fast Retransmit

- Wait for a timeout is quite long
- Immediately retransmits after 3 dupACKs without waiting for timeout
- Adjusts ssthresh
 - $flight_size = \min(awnd, cwnd)$
 - $ssthresh \leftarrow \max(flight_size/2, 2)$
- Enter Slow Start ($cwnd = 1$)

Successive Timeouts

- When there is a timeout, double the RTO
- Keep doing so for each lost retransmission
 - Exponential back-off
 - Max 64 seconds¹
 - Max 12 retransmits¹

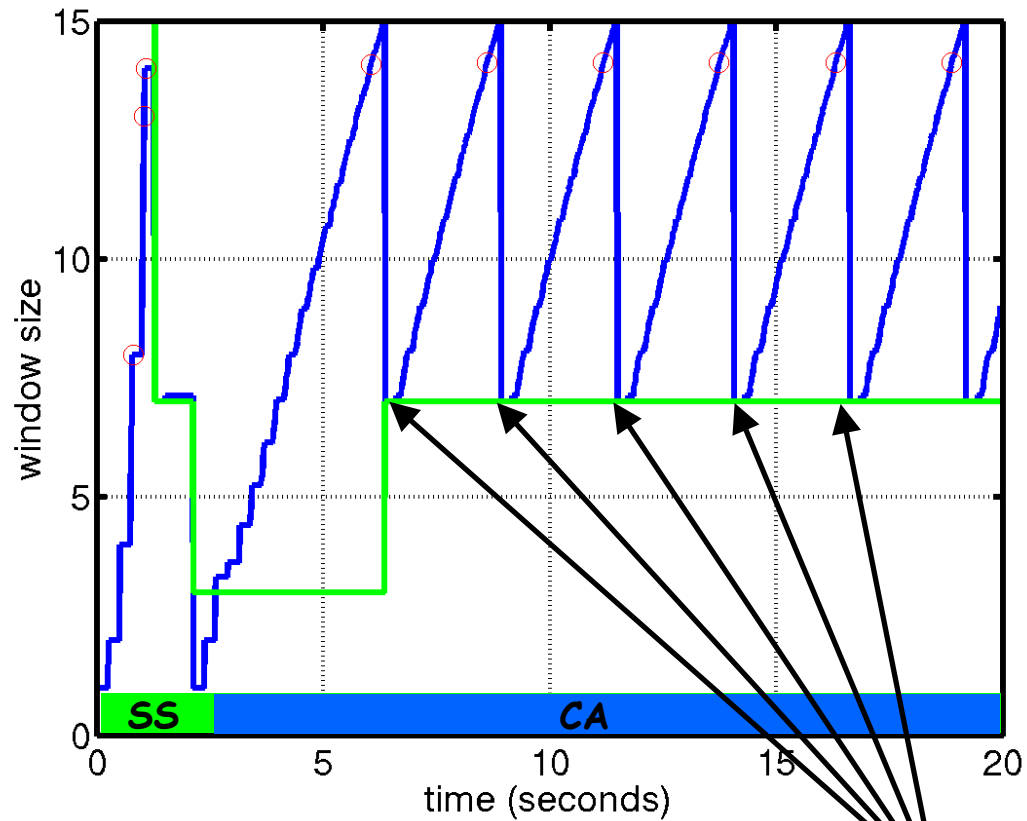
1 - Net/3 BSD



Fast recovery

- Motivation: prevent 'pipe' from emptying after fast retransmit
- Idea: each dupACK represents a packet having left the pipe (successfully received)
- Enter FR/FR after 3 dupACKs
 - Set $ssthresh \leftarrow \max(\text{flightsize}/2, 2)$
 - Retransmit lost packet
 - Set $cwnd \leftarrow ssthresh + ndup$ (window inflation)
 - Wait till $W = \min(\text{awnd}, cwnd)$ is large enough; transmit new packet(s)
 - On non-dup ACK (1 RTT later), set $cwnd \leftarrow ssthresh$ (window deflation)
- Enter CA

TCP Reno



Fast retransmission/fast recovery