TCP Congestion Control

Christian Grothoff

Berner Fachhochschule

December 9, 2017

TCP Congestion Control

- Flow control forces the sender to stop if the receiver is too slow
- Congestion control forces the sender to slow down if the network is too slow

Key issue:

How fast can we (re-)transmit frames?

The congestion window

- Limits the total number of *unacknowledged* packets that may be in transit
- Initial window is (traditionally) two maximum segment sizes (MSS)
- Modern initial congestion window is (up to) ten MSS
- Window size changes over time based on network conditions
- Window size also limited by receiver window (flow control)

Slow start

- Initial congestion control mode after setup or timeout
- Increases congestion window by 1 MSS for every packet acknowledged
- Mode ends if congestion window exceeds *ssthresh* limit

Slow start *doubles* the window size every RTT.

Slow start is *very aggressive* $\Rightarrow$ Worst name ever!

Congestion avoidance

- Congestion control mode after *ssthresh* limit
- Increases congestion window by 1 MSS every RTT *unless* duplicate ACKs are received

Congestion avoidance grows much slower than “slow” start.

Dealing with trouble: TCP Tahoe

Upon three duplicate ACKs or timeout:

- Retransmit packet indicated by ACK ("fast retransmit")
- *ssthresh* set to half of current congestion window
- Congestion window reset to 1 MSS
- Resume with slow start

Dealing with trouble: TCP Reno

Upon three duplicate ACKs:

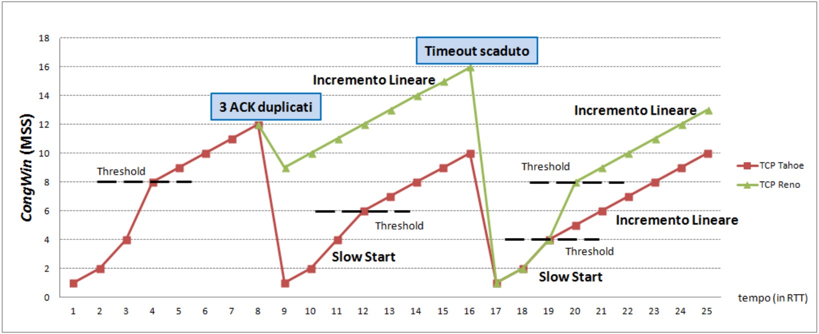
- Retransmit packet indicated by ACK ("fast retransmit")
- *ssthresh* set to half of current congestion window
- Congestion window reset to (new) *ssthresh*
- Continue with "fast recovery"

Timeout behavior same as TCP Tahoe.

Fast recovery

- Complete fast retransmit
- Wait for next acknowledgement
- Continue with congestion avoidance

Window size in pictures[1]



[1] CC-BY-SA: `https://commons.wikimedia.org/wiki/File:CongWin_in_TCP_Tahoe_e_Reno.png`

Diversity

Every other network research group has published a congestion control algorithm:

- TCP New Reno
- TCP Hybla
- TCP BIC
- TCP CUBIC
- TCP Westwood
- ...

Fairness

- Basis is additive increase, multiplicative decrease (AIMD) feedback control algorithm as seen in TCP Tahoe
- Multiple flows using AIMD will converge to use *equal amounts* of contended link
- Algorithms that compete fairly are called **TCP friendly**