Congestion Control

goal: provide an end-to-end feedback mechanism which causes senders to adapt to a fair share of the bottleneck link
what is the bottleneck link?

what is the maximum bandwidth acheived by a TCP connection: $BW = \frac{window}{RTT}$

- window constrains the sending rate
- corresponds to number of network buffers
- congestion control is the problem of adapting the window
- $win_{effective} = MIN(win_{congestion}, win_{advertised})$

references:
“Congestion Avoidance and Control” - Jacobson, Karels - Sigcomm 1988
RFC2581- M Allman et. al.

Congestion Control - some basic definitions

cwnd the effective congestion window

ssthresh the current minimum bound on a reasonable congestion window

congestion event a loss which occurs within the timeframe of the current congestion window

packets in flight the number of packets that have been sent but not yet acknowledged

convervation of packets dont inject new data until we are fairly certain that a packet has left

Slow Start

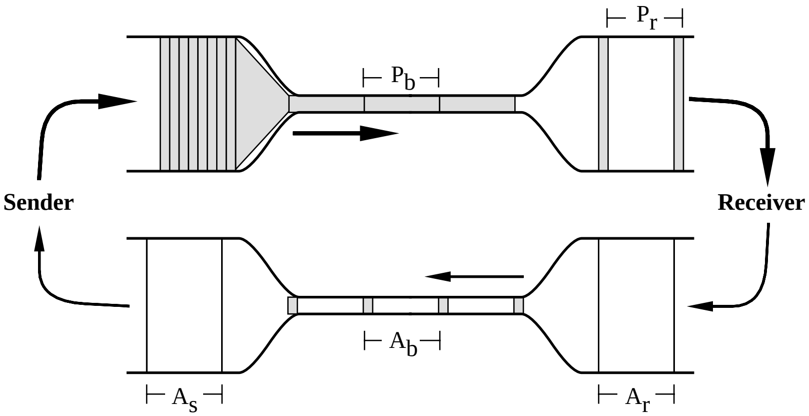
how should we start sending?

- initial window size may be greater than number of available buffers
- set `cwnd` to 1, and `ssthresh` to ∞
- increase by 1 for every ack
- “slow start” is actually exponential

Congestion Avoidance

- linear increase, multiplicative increase
- once $\texttt{cwnd} = \texttt{ssthresh}$ probe the network more slowly
- linear: for each ack increase $\texttt{cwnd}$ by $\frac{1}{\texttt{cwnd}}$
- on a timeout, set $\texttt{ssthresh} = \frac{pif}{2}$
- on a timeout, set $\texttt{cwnd} = 1$

Ack Pacing



- limiting the window isn't enough to stop bursts from occuring
- each (non-duplicate) ack advances the window by one segment
- this naturally smooths out transmissions to the bottleneck capacity
- slow start is necessary to start pacing
- so idle connections restart in slow start

Fast Retransmission

- single losses are catastrophic for performance
- duplicate acks indicate that a segment was missing
- we can retransmit that segment immediately after 3
- set `ssthresh` to $\frac{pif}{2}$
- set `cwnd` equal to `ssthresh` $+ 3$

RTT Estimation

what timeout interval should we use?

- if larger than the real RTT, performance suffers
- if smaller than the real RTT, we have excessive retransmission
- solution: adapt retransmit timer based on ACK measurements
- use a weighted moving average to smooth out sampling noise

$RTT_{new} = (1 - \alpha)RTT_{old} + \alpha Measurement$ where α is called the gain and determines how responsive the moving average is