
Congestion Avoidance.

Slow Start

SST = 8

queueing delay
Packet Drop.

\[ CW = \frac{SST}{2} \]

A1MD: Additive increase Multiplicative decrease of the CW.

TCP optimization: Transmit every alternate ACK

A_A_A_A...

Alternate ACK
When ACK $A_{20}$ comes back:

Pretend ACKs $A_{11}$ to $A_{20}$ are coming one by one.

$CW = 2$ $\leftarrow A_{11}$

$CW = 3$ $\leftarrow A_{12}$

$CW = 5$ $\leftarrow A_{14}$

Enter cong. avoidance.

$CW = 5 \frac{1}{5}$ $\leftarrow A_{15}$

$CW = 5 \frac{2}{5}$ $\leftarrow A_{16}$

$CW = 6$ $\leftarrow A_{19}$

$CW = 6 \frac{1}{6}$ $\leftarrow A_{20}$
Duplicate ACKs (DupACKs).

Upon receiving DupACK, Tx thinks: Some kind of reordering has happened. Let's wait & watch. Increase DupACK count ++.

If DupACK count = 3, TCP enters "Fast Recovery Phase".
Fast Recovery Phase.

Let's immediately get to Cong. Avoidance.

\[ \text{SST} = \frac{\text{CW}}{2} \]
\[ \text{CW} = \text{SST} \]
\[ \text{CW} = \text{CW} + 3 \]

Send pkt Rx is expecting

When more DwPacks arrive:

\[ \text{CW} = \text{CW} + 1 \]

End of "Fast Recovery":

when new (non-DupAck) arrives,
let's clean slate and start from Cong. avoidance. So

\[ \text{CW} = \text{SST} \text{ and DwPacks} = 0 \]
Note: SST can increase with time depending on where TCP encounters drop or 3 DupsACKs.