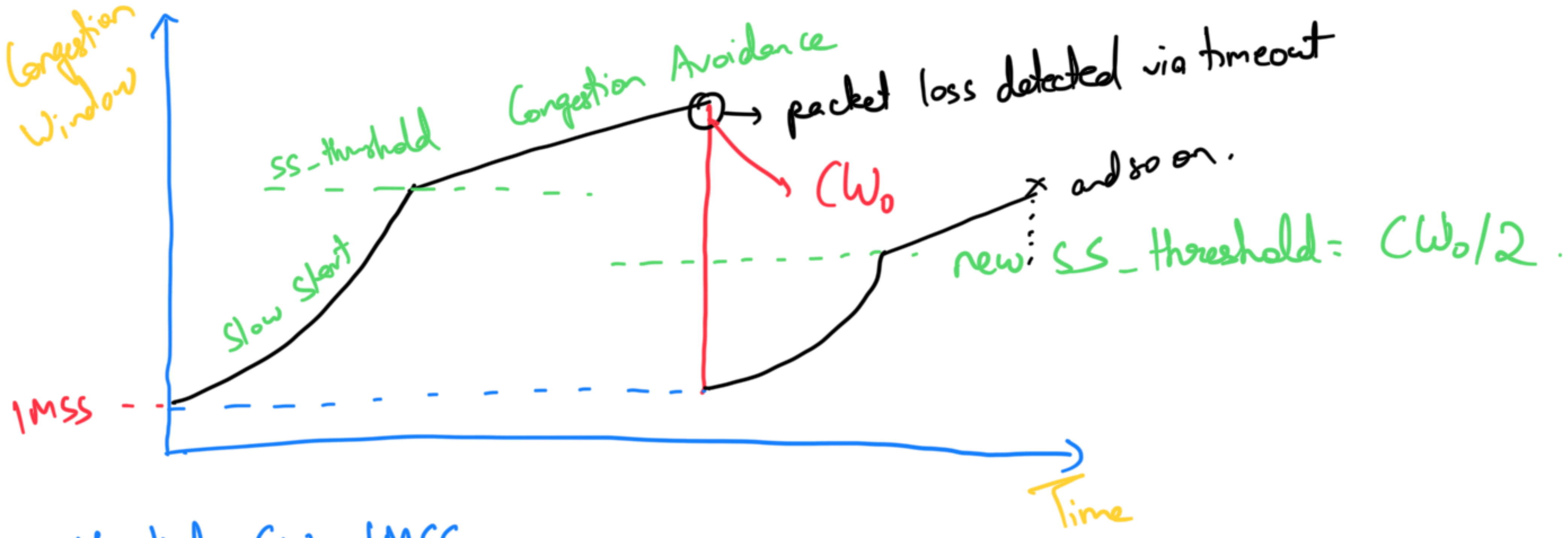
CS348 Notes
TCP Ta(hoe) Reno
Video Numbers: 26

OjMaha

I have prepared these notes by watching the videos from Networks Playlist. The following notes may be asynchronous and irrelevant to what Prof. Vinay teaches in class (cuz I do not pay attention during lectures lol). Further, these notes might not cover *everything* as explained in the video lectures. Consider these to be a supplemental read :). If you find any errors, do notify me so they can be edited.

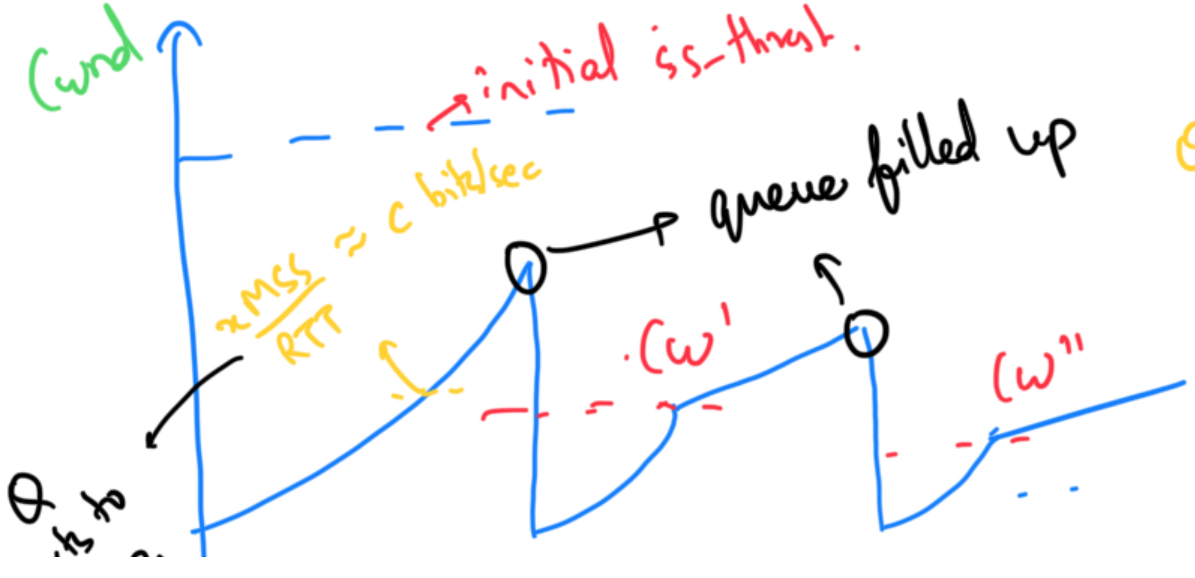
TCP TAHOE → decrease to 1MSS.

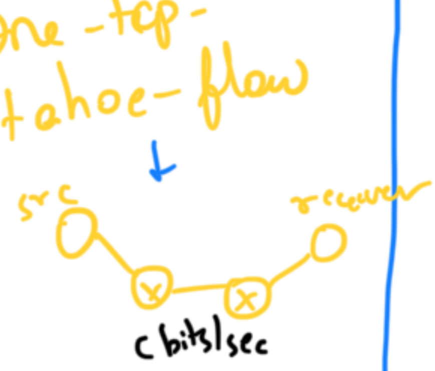


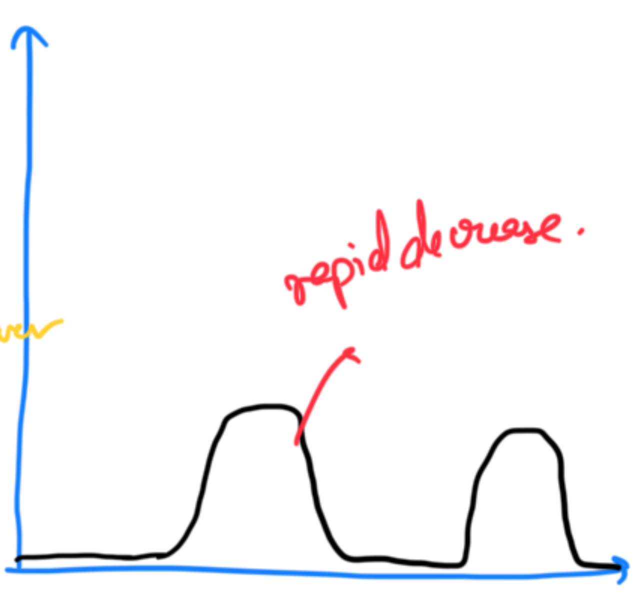
Initial CW = 1MSS

CW += 1 MSS for each recd ACK (slow start)

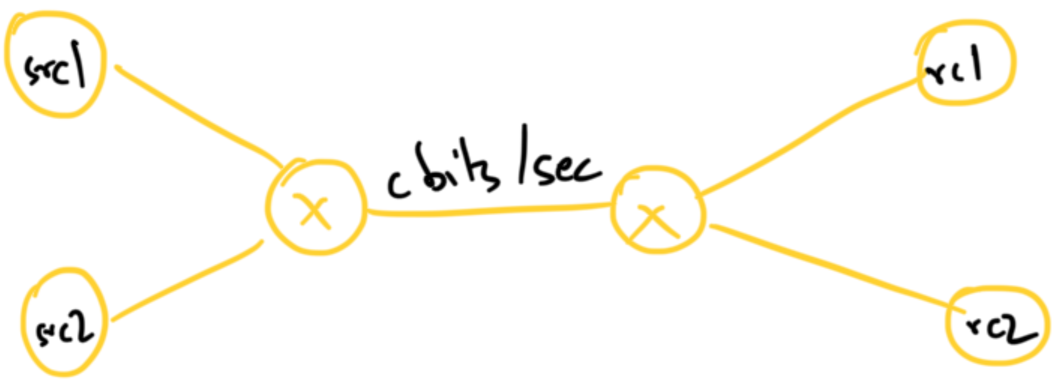
CW += $(MSS)^2/CW$ if $CW \geq ss\_thresh$ for each recd ACK. (Congestion Avoidance)







Two-TCP-tahoe flow:



The TCP connections need not start together. Say flow 2 starts later. Now, packet loss is detected when $c_1+c_2>c$. Then both reduce their cwnd and ss_threshold. But since the connections started in diff points of time; there could be a huge diff. b/w c_1 & c_2. One may be in slow start and other maybe in Congestion Avoidance. Thus, one of the connections may suffer great decrease in cwnd upon (say flow 2) "convergence". This means that it is possible flow 2 never suffers packet losses eventually as its rate is low so whenever it is being routed, the queue has space for it.

If we have multiple flows through the same bottleneck, it is tricky to predict what may happen.

STOP / RELOAD:

STOP & RELOAD:

Sometimes, it might occur that a few initial packets are inadvertently lost. This leads to a sharp decrease in the cwnd and ss_threshold. Thus, it might occur that "slow start" phase is skipped (or run very less) and avoidance begins which is very slow despite there being a lot of bandwidth. Thus, upon reloading the webpage, the data packets are resent, protocol is reset, now the first few packets reach and it quickly downloads.

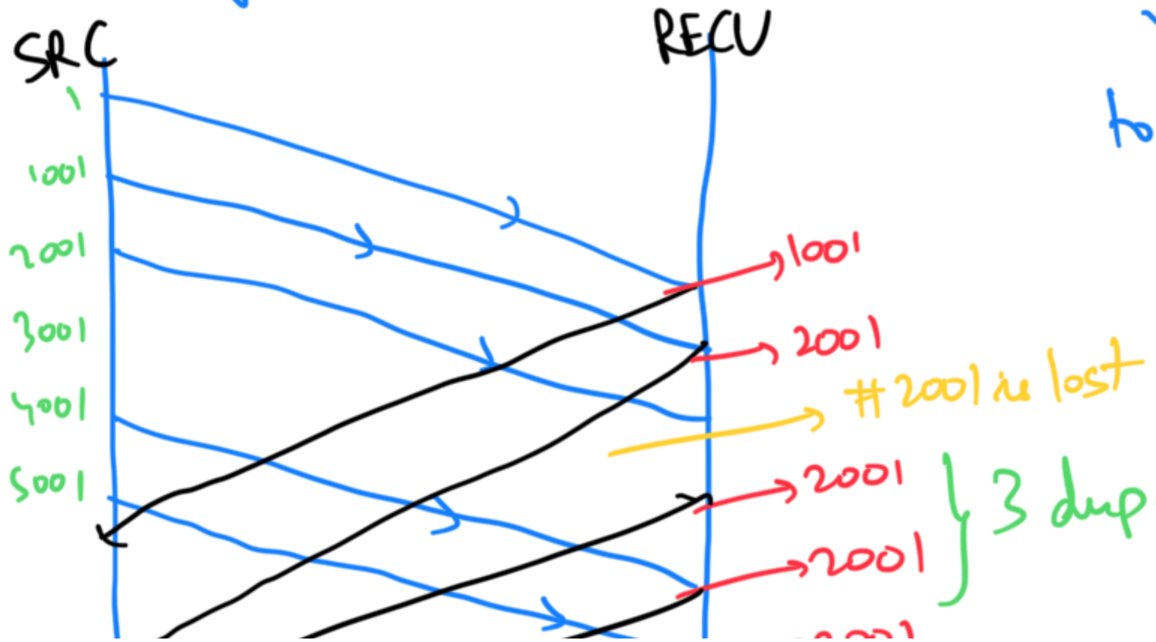
TCP RENO

Fast Retransmit: (triple duplicate acks)

→ disable timer. reset it while retransmitting pkt.

Retransmit segment if 3 DUP ACKs received. (alternate way to detect pkt loss)

→ to accomodate for packet re-ordering due to delays.



3 dupacks so retransmit #2001.

Fast Recovery:

* Note that for timeout loss, CW = 1MSS still like in Tahoe.

But when triple duplicate ack loss is detected, $ss\_threshold = CW_0/2$.
$CW = CW_0/2$. (instead of 1MSS)

↓
intuition: some packets are still getting through, so no need to decrease CW that aggressively. Timeout loss means you aren't even receiving dup acks. So the situation is really bad. Moye Moye.

RFC 5681: Congestion Control.

In slow start, if we receive an ACK which acknowledges 'N' bytes of new data; cwnd += min(N, MSS).

In congestion avoidance, for every ACK that acknowledges new data;
$cwnd += (mss)^2 / cwnd$

cwnd += (MSS) / cwnd

retransmission

RFC 6298: Timeout.

Initial RTO = 1 sec or higher

min (cwnd, rwnd)

Loss by Timeout ⇒ ss_threshold = max(window/2, 2 * MSS)

⇒ RTO = min(2 * RTO, max_RTO) (exponential backoff)