CS & IT ENGINEERING

COMPUTER NETWORKS

TCP & UDP

Lecture No-11

By- Ankit Doyla Sir

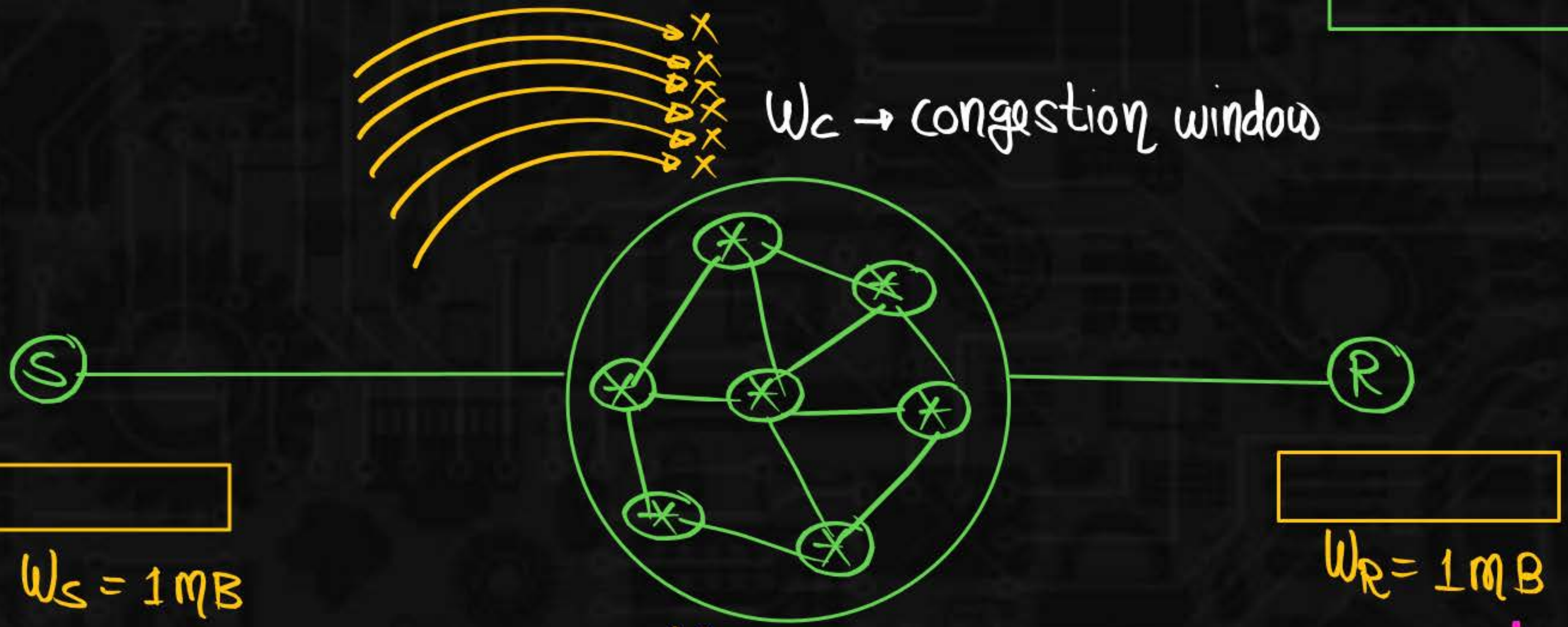
TOPICS TO BE COVERED

Congestion control in TCP

Congestion control
in
TCP

TCP congestion control

1980
Jacobson's



$W_S = 1\,MB$

$MSS = 1\,KB$

$W_R = 1\,MB$

No. of segments $= 1024$

No. of segments $= \dfrac{1\,MB}{1\,KB} = \dfrac{2^{20}\,\cancel{B}}{2^{10}\,\cancel{B}} = 2^{10}$

No. of segments $= 1024$

$$\boxed{W_S = \min\{\text{Receiver capacity}, \text{Network capacity}\}}$$

$$\boxed{W_S = \min\{W_R, W_C\}}$$

TCP congestion control

An Internet is a combination of networks and connecting devices (e.g., routers). A packet from a sender may pass through several routers before reaching its final destination. A router has a buffer that stores the incoming packets, processes them, and forwards them. If a router receives packets faster than it can process, congestion might occur and some packets could be dropped. When a packet does not reach the destination, no acknowledgement is sent for it. The sender has no choice but to retransmit the lost packet. This may create more congestion and more dropping of packets, which means more retransmission and more congestion. A point may be reached in which the whole system collapses and no more data can be sent. TCP therefore needs to find some way to avoid this situation.

Congestion Window

In TCP, the sender's window size is determined not only by the receiver but also by congestion in the network.
The sender has two pieces of information: the receiver-advertised window size and the congestion window size. The actual size of the window is the minimum of these two.
Actual window size = minimum (receiver window size, congestion window size)

$$\boxed{W_S = \min(W_R, W_C)}$$

$W_C = 1$

$W_S = \min(W_C, W_R)$

$W_S = \min(1, 1024)$

$W_S = 1$

$W_C = 2$

$W_S = \min(W_C, W_R)$

$W_S = \min(2, 1024)$

$W_S = 2$

$W_C = 4$

$W_S = \min(W_C, W_R)$

$W_S = \min(4, 1024)$

$W_S = 4$

$\vdots$

$W_R = 1024$ segment, $TH = \frac{1}{2} W_R = 512$ segment

W_C : 1, 2, 4, 8, 16, 32, 64, 128, 256

512, 513, 514, 515, 516, ... 1024, 1024, 1024

$$W_c: \underbrace{1, 2, 4, 8, 16, 32, 64, 128, 256, 512,}_{1023 \text{ segment}} 513, 514, \ldots$$

$$W_s = \min(W_c, W_R)$$

$$W_s = \min(513, 1)$$

$$\boxed{W_s = 1}$$

Congestion Control Algorithm

It has 3 phases

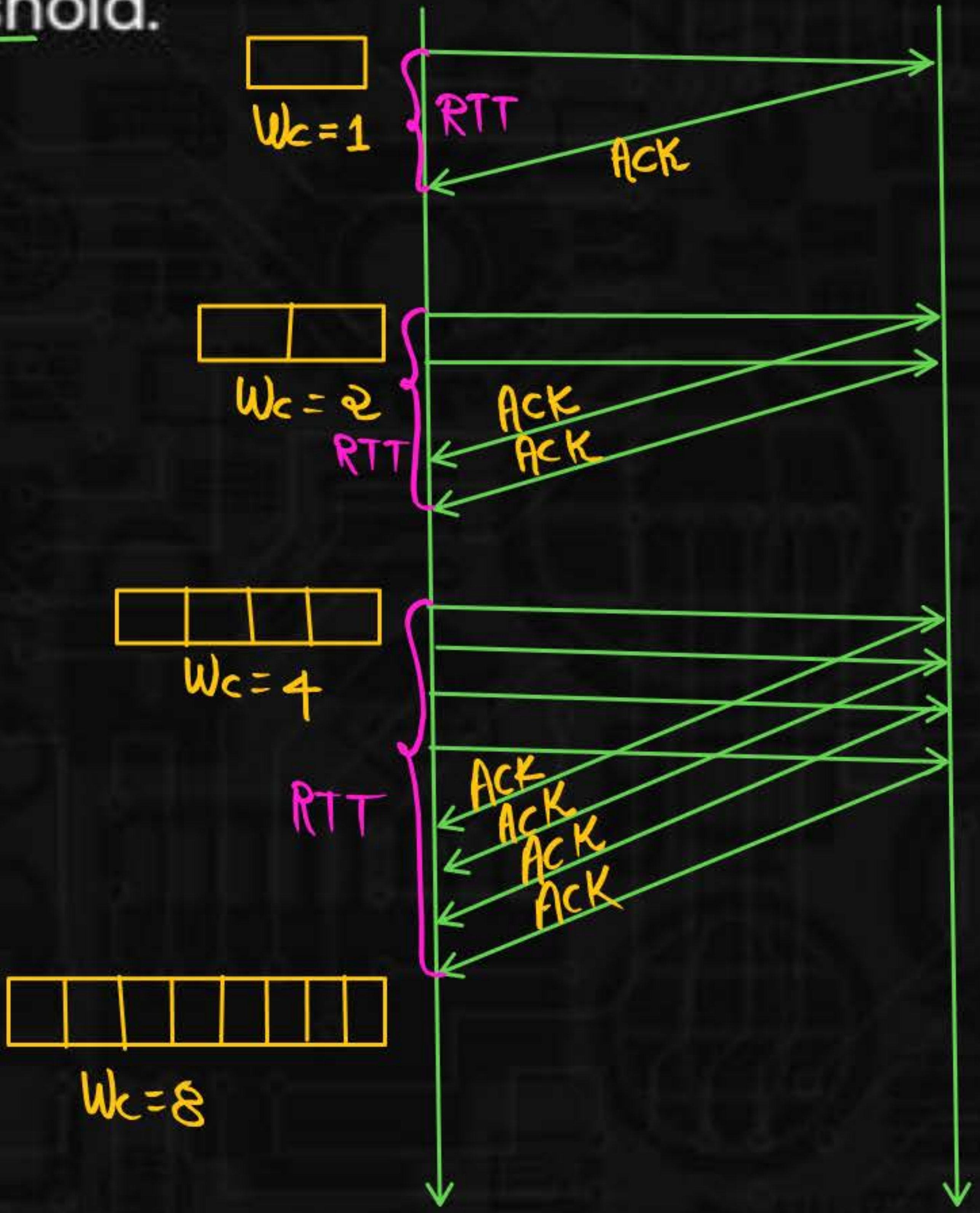
i. Slow start (exponential Increase)

ii. Congestion Avoidance (Additive Increase)

iii. Congestion Detection (multiplicative decreases)

Slow start Phase

In the slow start phase the size of the congestion window increases exponentially until it reach a threshold.



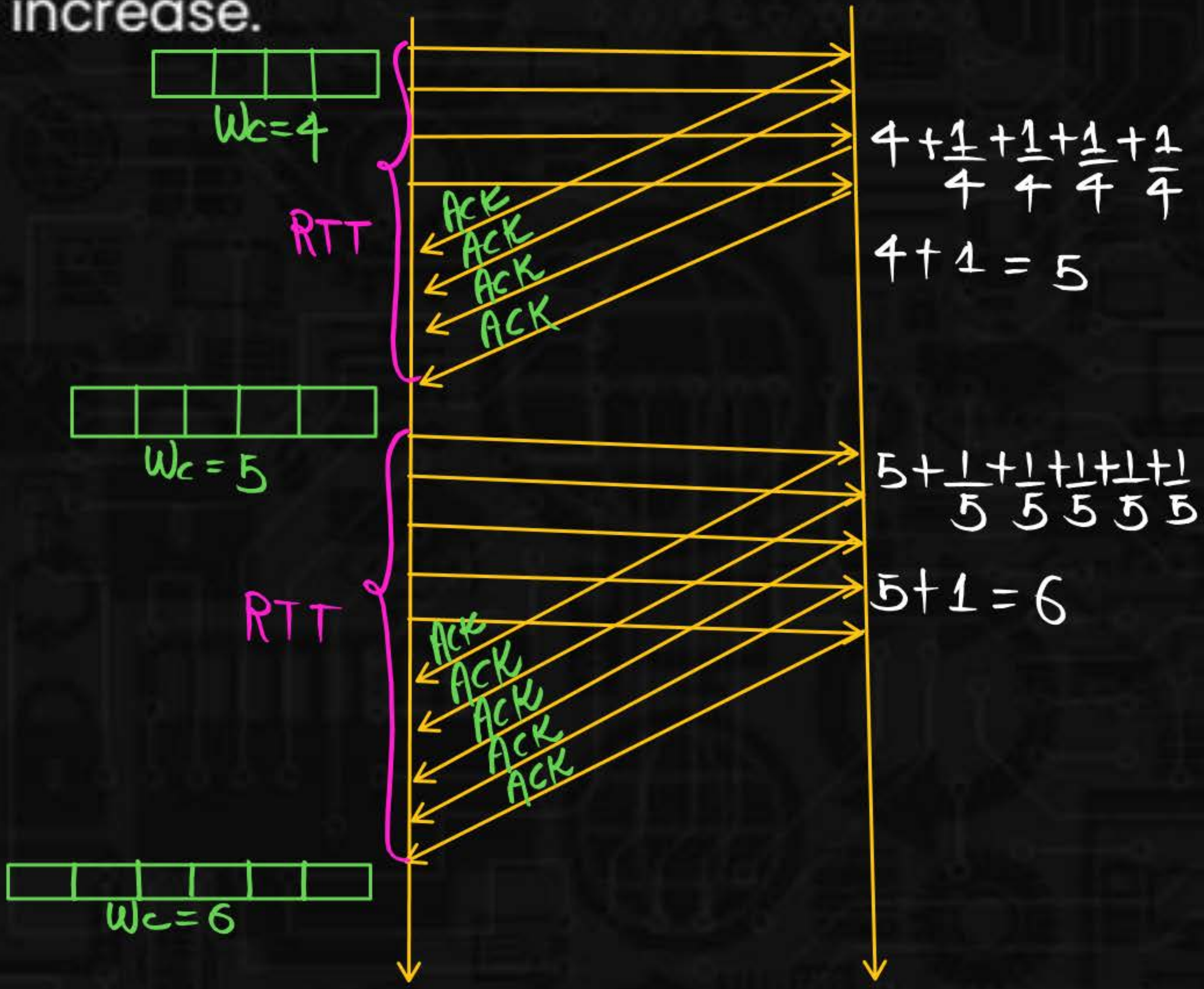
Slow start Phase

→ After one RTT congestion window will be double in slow start phase

→ If an Ack arrives then $\boxed{W_c = W_c + 1}$

Congestion Avoidance

To Avoid congestion before it happens we must slow down its exponential growth. In congestion Avoidance we use additive increase instead of exponential increase.



Congestion Avoidance

- After one RTT the congestion window will be increased by one only
- If an Ack arrives $\boxed{Wc = Wc + \frac{1}{Wc}}$

Congestion Detection

Congestion can be detected in two ways

i. Time out Timer

ii. 3 duplicate ACK

i. Time out Timer

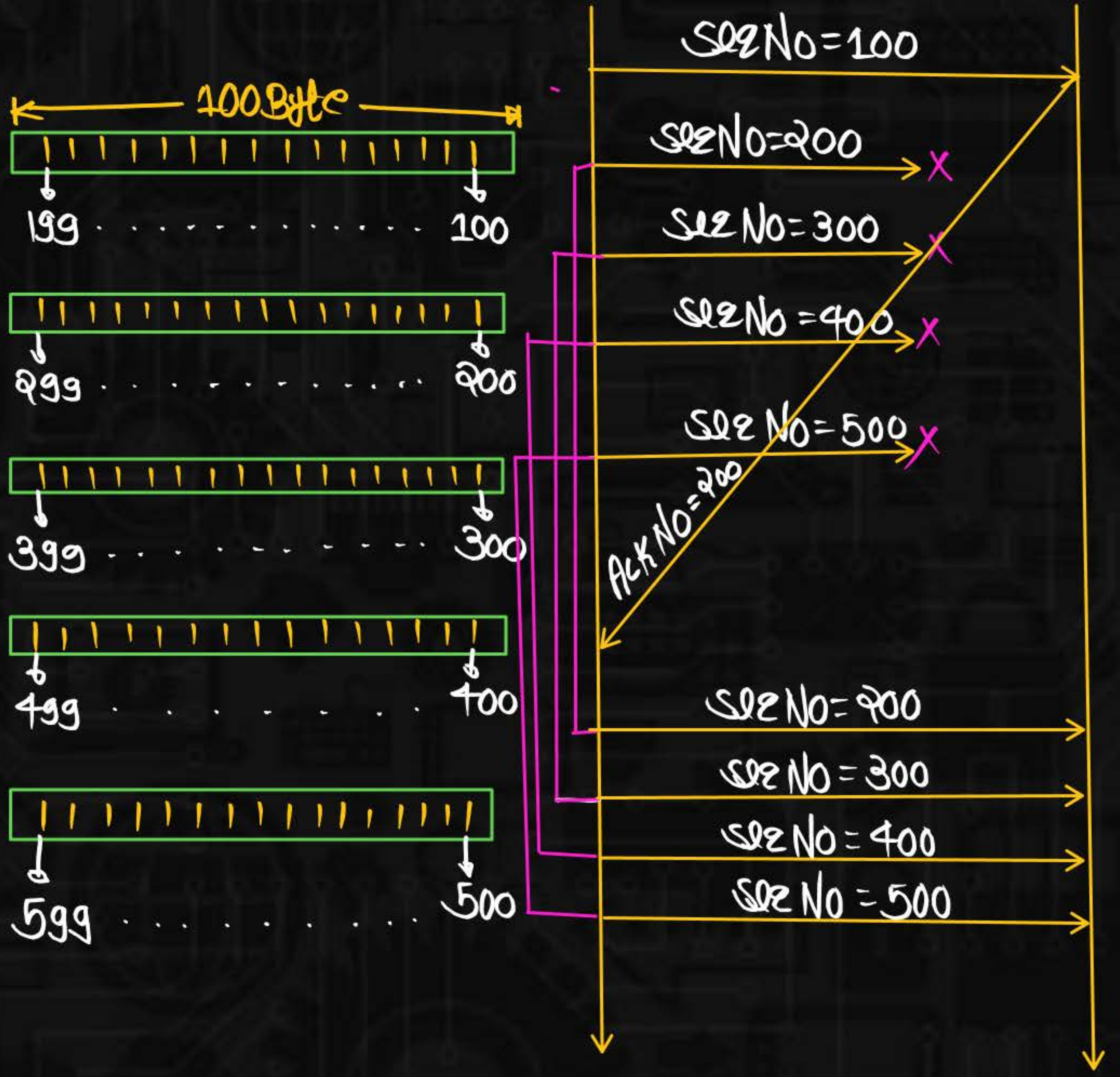
Time out timer indicate severe congestion condition. In this case the new threshold value is set to half of the current window size and next transmission starts from one segment and Algorithm enters in a slow start phase.

NTH = 256

TH

512 ↑ 1, 2, 4, 8, 16, 32, 64, 128, 256, 257, 258, 259, ... 1024, 1024

T.O



Time out timer indicate severe congestion condition

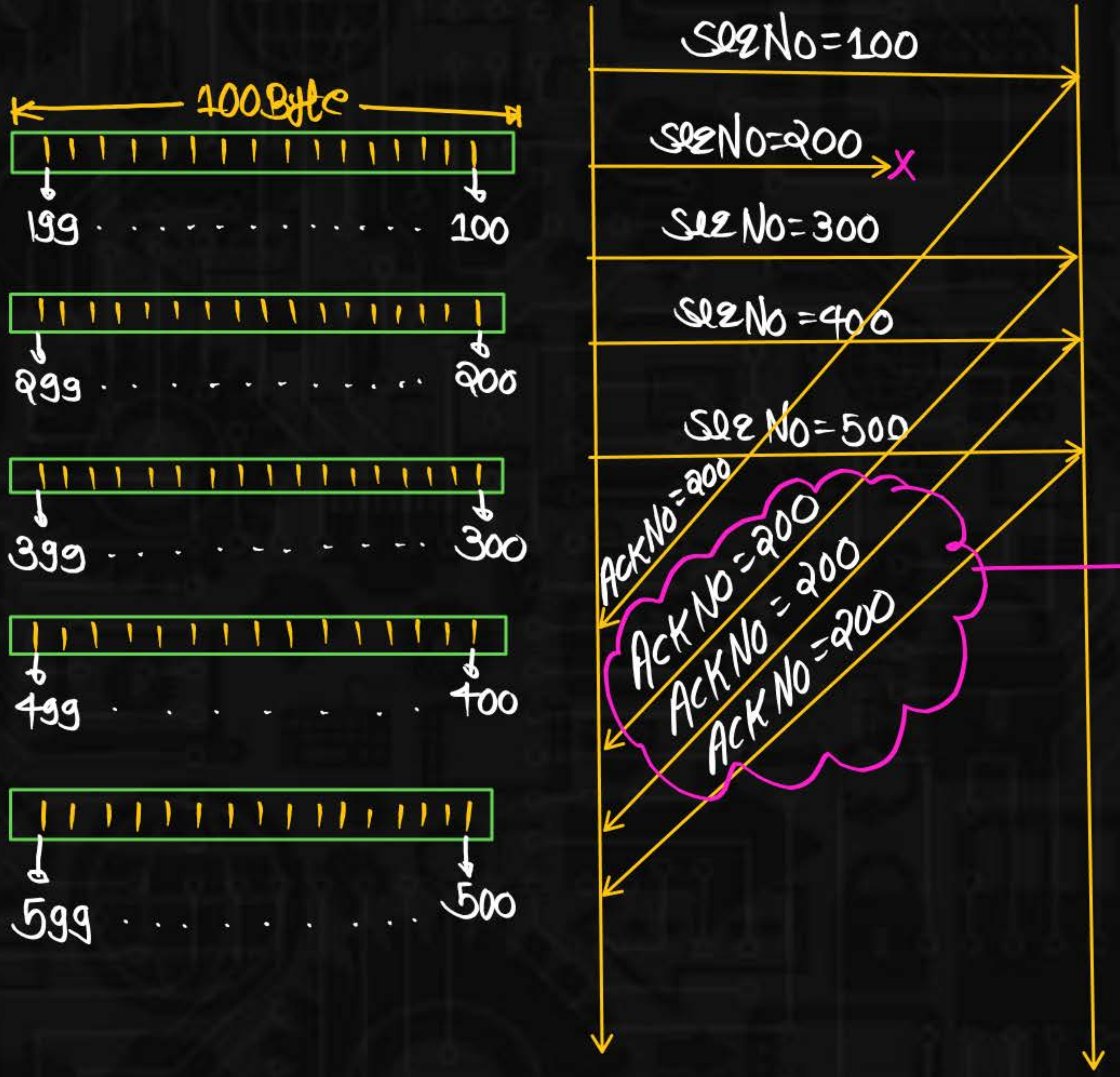
ii. 3 Duplicate ACK

3 Duplication Ack indicate mild congestion condition. In this case the new threshold value is set to half of the current window size and next transmission start from new threshold value and algorithms enters in a congestion avoidance phase.

NTH = 256

512 ↑ 256, 257, 258, 259, ... 1024, 1024 ....

3 duplicate Ack



3 duplicate Ack Indicate Mild congestion condition


THANK YOU
GW
SOLDIERS !