CS & IT ENGINEERING

COMPUTER NETWORKS

TCP & UDP

Lecture No-14

By- Ankit Doyla Sir

TOPICS TO BE COVERED

- Congestion Control in TCP

Congestion control in TCP

TCP congestion control

1980's
Jacobson's

$W_c \rightarrow$ congestion window

$W_S = 1MB$

$MSS = 1KB$

$W_R = 1MB$

No. of segments $= \frac{1M\cancel{B}}{1K\cancel{B}}$

$= \frac{2^{20}}{2^{10}} = 2^{10}$

No. of segments = 1024

No. of segments = 1024

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

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

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)

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

$W_C = 1$

$W_S = \min\{W_C, W_R)$

$W_S = \min(1, 1024)$

$W_S = 1$

$W_C = 2$

$W_S = \min(W_C, W_R)$

$W_S = \min(2, 1024)$

$W_S = 2$

$W_C = 4$

$W_S = \min(W_C, W_R)$

$W_S = \min(4, 1024)$

$W_S = 4$ ......

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

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

256, 512 (TH), 513, 514, 515, 516, 517

518, ... 1024, 1024, 1024

$$W_C: \underbrace{1, 2, 4, 8, 16, 32, 64, 128, 256, 512}_{1023}, 513, 514, 515$$

$$W_S = \min\{W_C, W_R)$$

$$= \min\{513, 1)$$

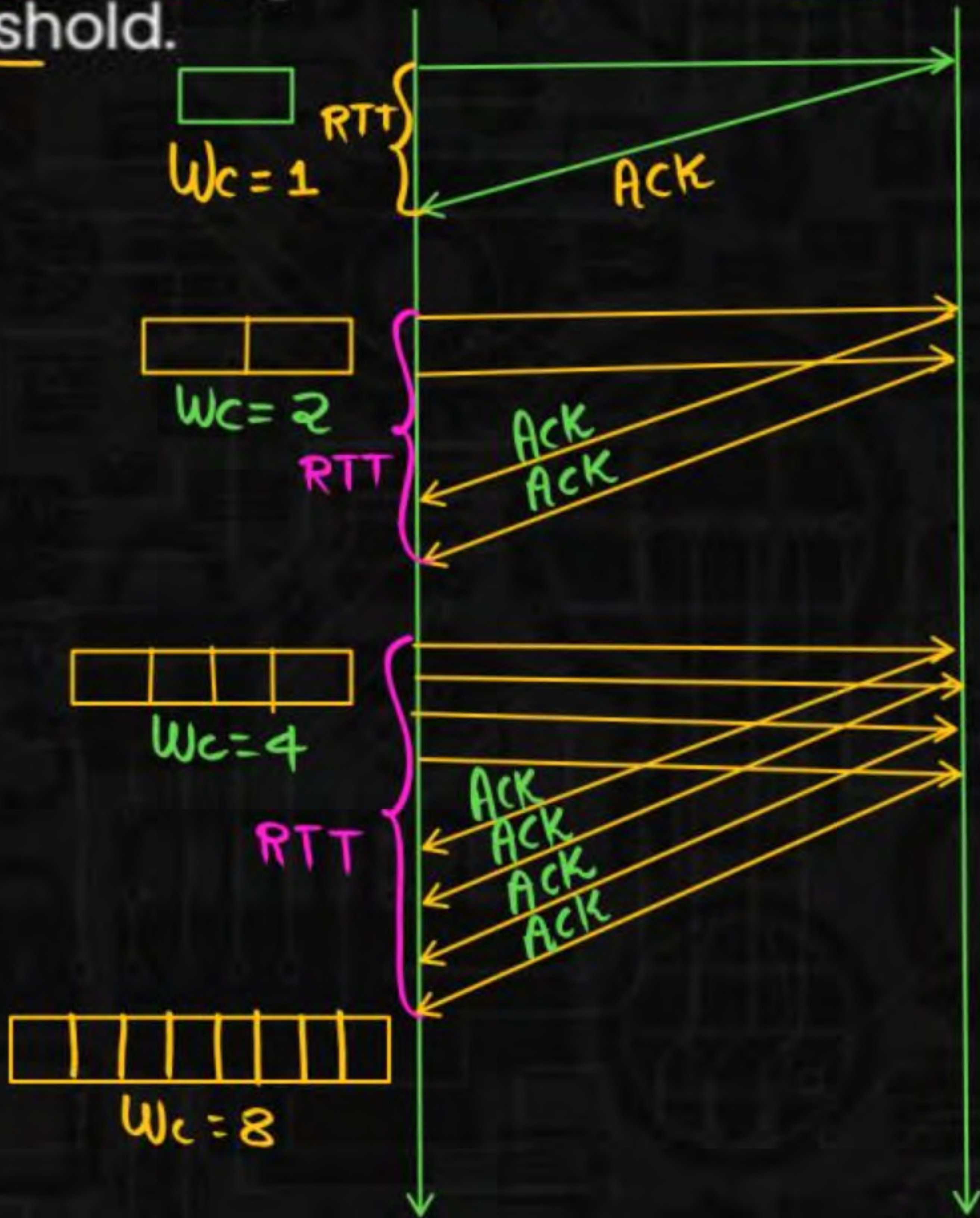
$$\boxed{W_S = 1}$$

Congestion Control Algorithm

It has 3 phases
i. Slow start (exponential Increase)
ii. Congestion Avoidance (Additive Increase)
iii. Congestion Detection (multiplicative decreases)

(i) Slow start Phase:

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



Slow start Phase

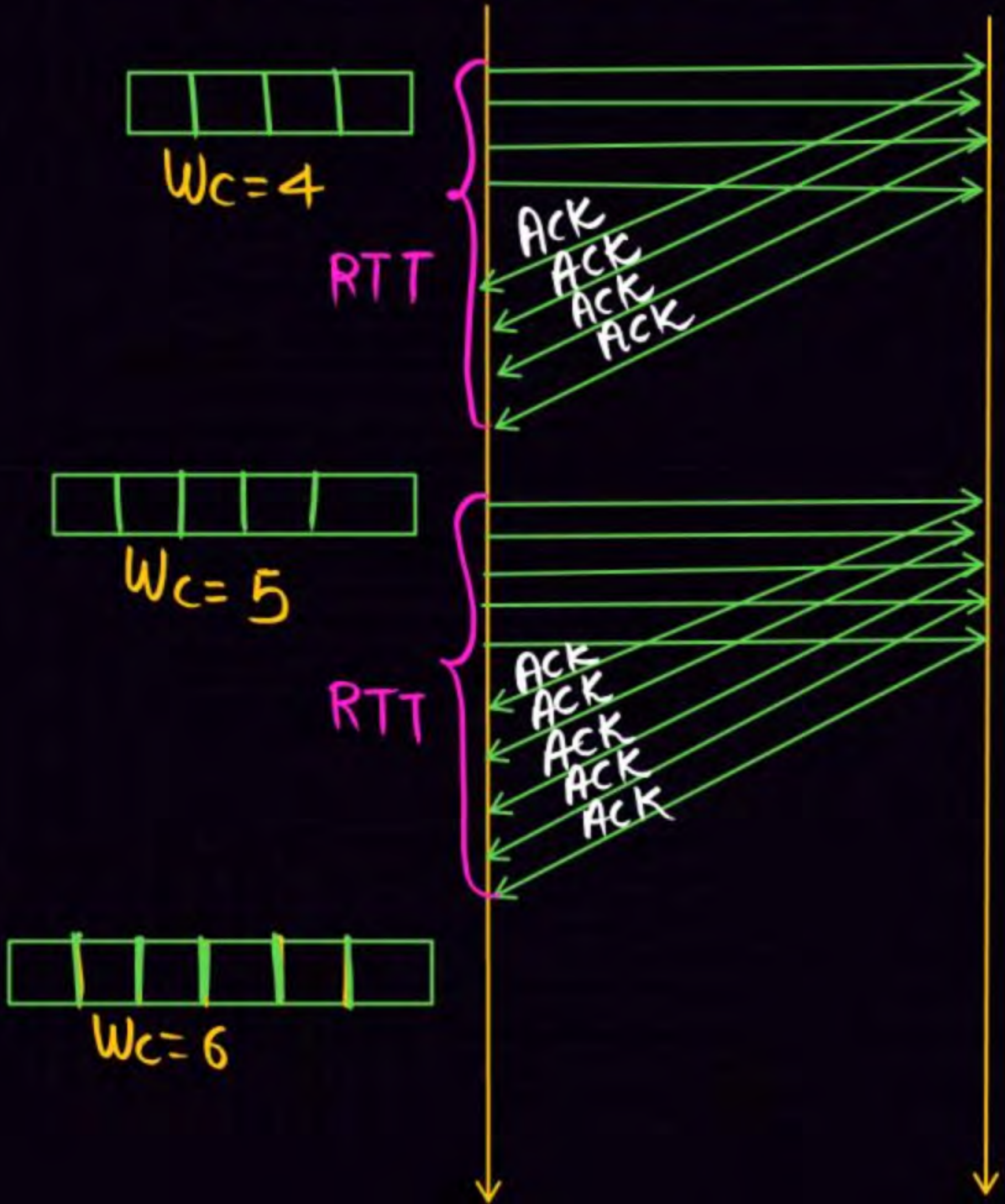
(i) After one RTT congestion window will be double in slow start phase

(ii) 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 Phase



$$4+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}+\frac{1}{4}=4+1=5$$

$$5+\frac{1}{5}+\frac{1}{5}+\frac{1}{5}+\frac{1}{5}+\frac{1}{5}=5+1=6$$

Congestion Avoidance

(i) After one RTT the congestion window will be increased by one only

(ii) If an Ack arrives $\boxed{W_c = W_c + \frac{1}{W_c}}$

Congestion Detection :

Congestion can be detected in two ways

i. Time out

ii. 3 duplicate ACK

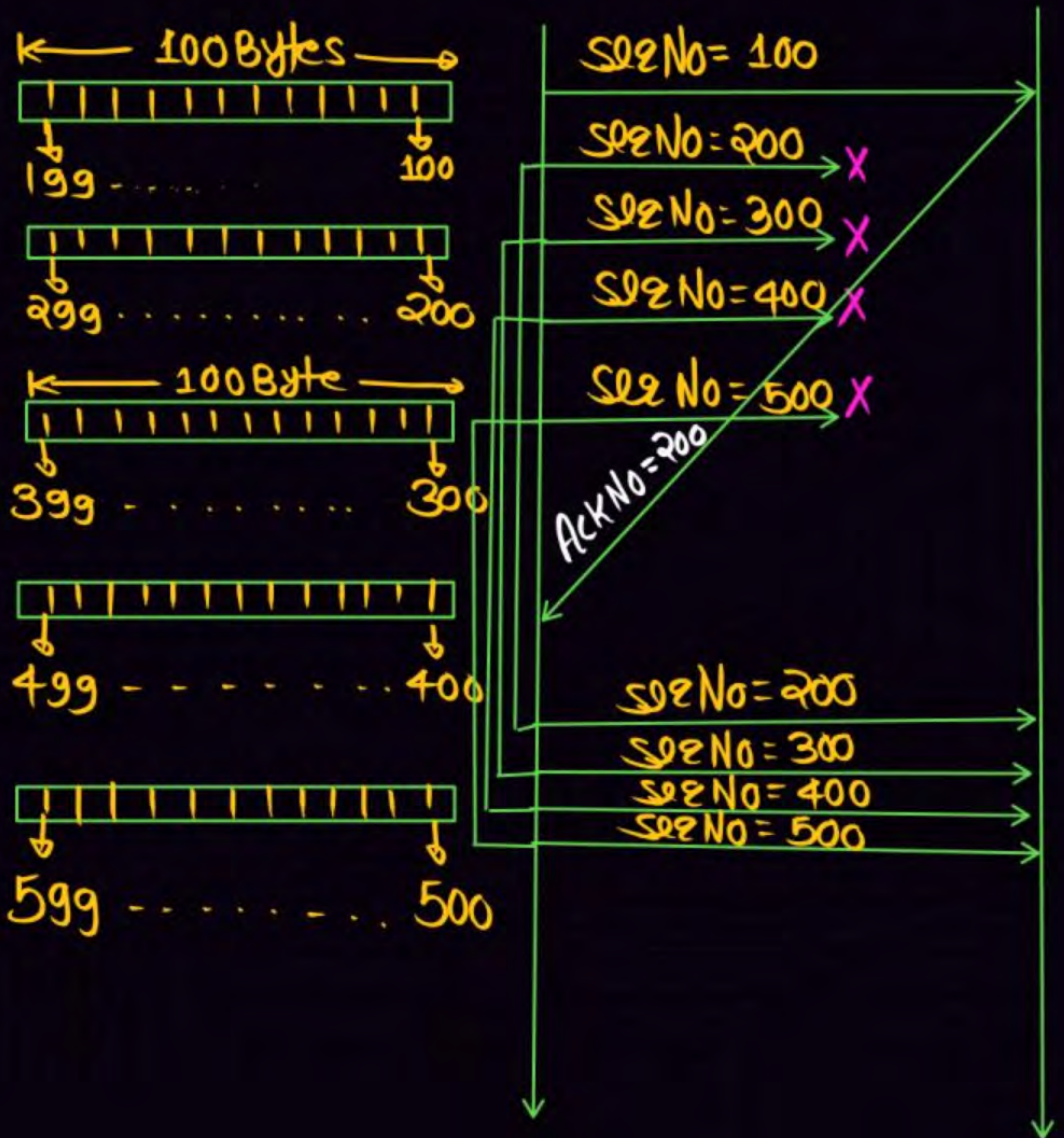
i. Time out Timer :

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

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

T.O 259, 1024, 1024



Time out timer indicate severe congestion condition

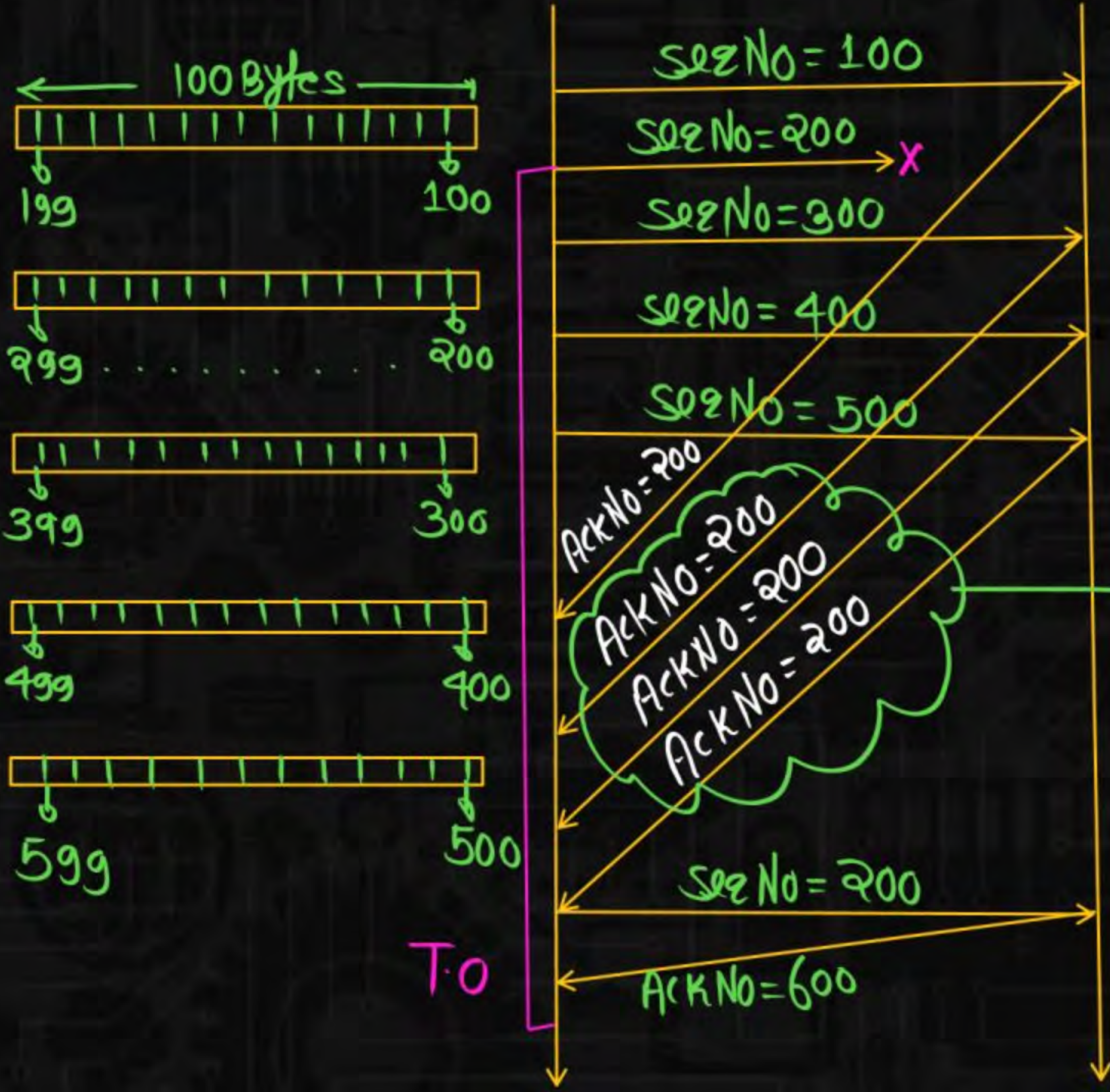
ii. 3 Duplicate ACK

3 Duplicate [redacted] 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

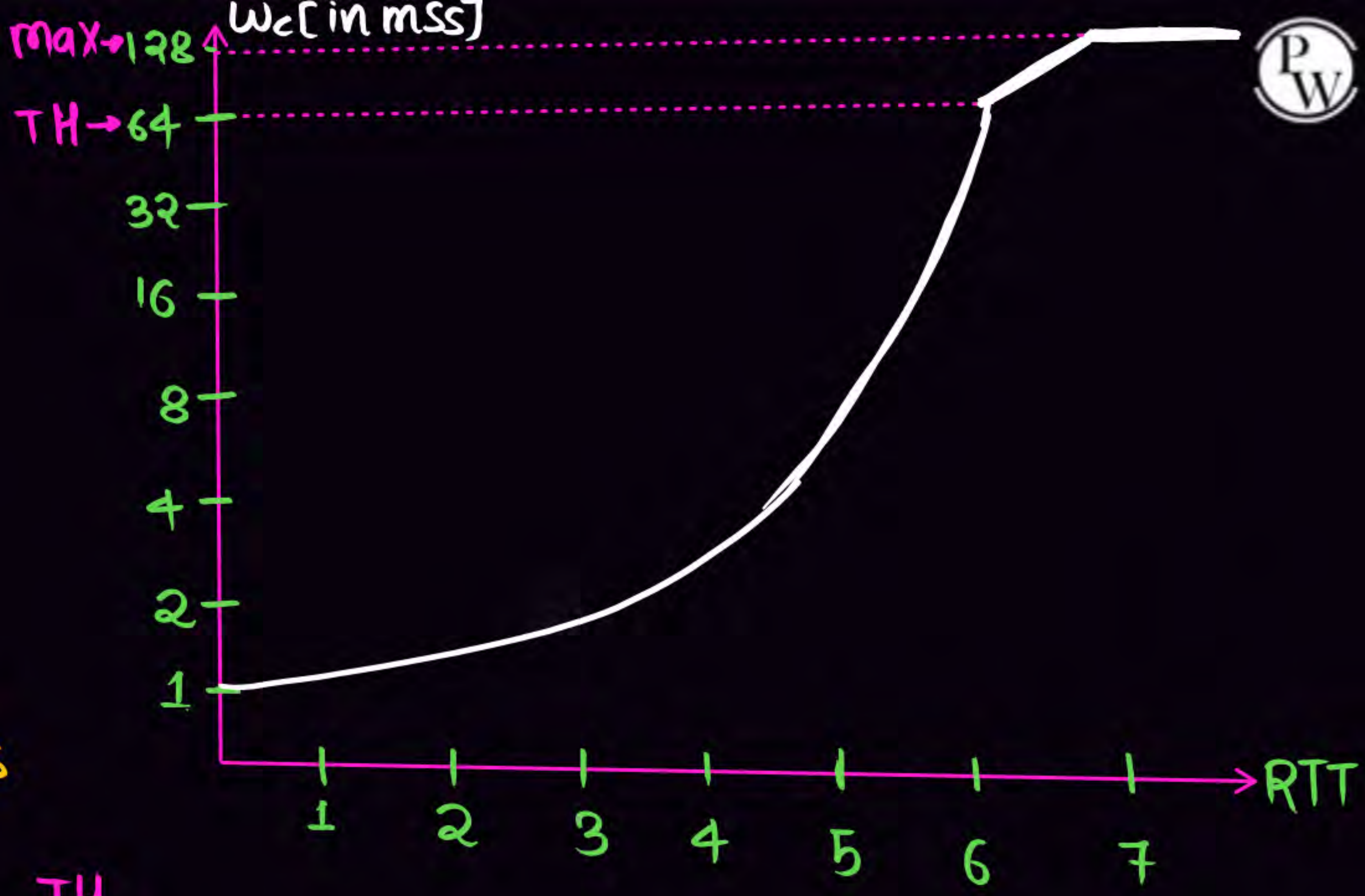
Q: $W_R = 128\,KB$

$MSS = 1\,KB$

$$\text{No. of segments} = \frac{128\,\cancel{KB}}{1\,\cancel{KB}}$$

No. of segments = 128

$$TH = \frac{1}{2} W_R = 64 \text{ segments}$$



TH

W_c: 1, 2, 4, 8, 16, 32, 64, 65

67, 68, . . . 128, 128, 128

Q: $W_R = 128KB$

$MSS = 1KB$

No. of segments $= \frac{128\cancel{KB}}{1\cancel{KB}}$

$= 128$

$TH = \frac{1}{2} W_R = 64$ segments

TH above 64; NTH = 34; NTH = 19

1, 2, 4, 8, 16, 32, 64, 65, 66, 67, 68 ↑ (3 duplicate ACK) 34, 35, 36, 37, 38 ↑ (T.O) 1, 2,

NTH = 10; NTH = 6

4, 8, 16, 19, 20 ↑ (3 duplicate ACK) 10, 11, 12 ↑ (T.O) 1, 2, 4, 6, 7, 8, 9, ... 128, 128 ....

NTH = 9.5

19 ↑ 9, 10, 11, 12

3 duplicate
ACK

Problem Solving
on
Congestion control

Q1. Consider the following statements regarding the slow start phase of the TCP congestion control algorithm. Note that cwnd stand for the TCP congestion window and MSS denotes the Maximum Segment Size.

X i. The cwnd increases by 2 MSS on every successful acknowledgment.

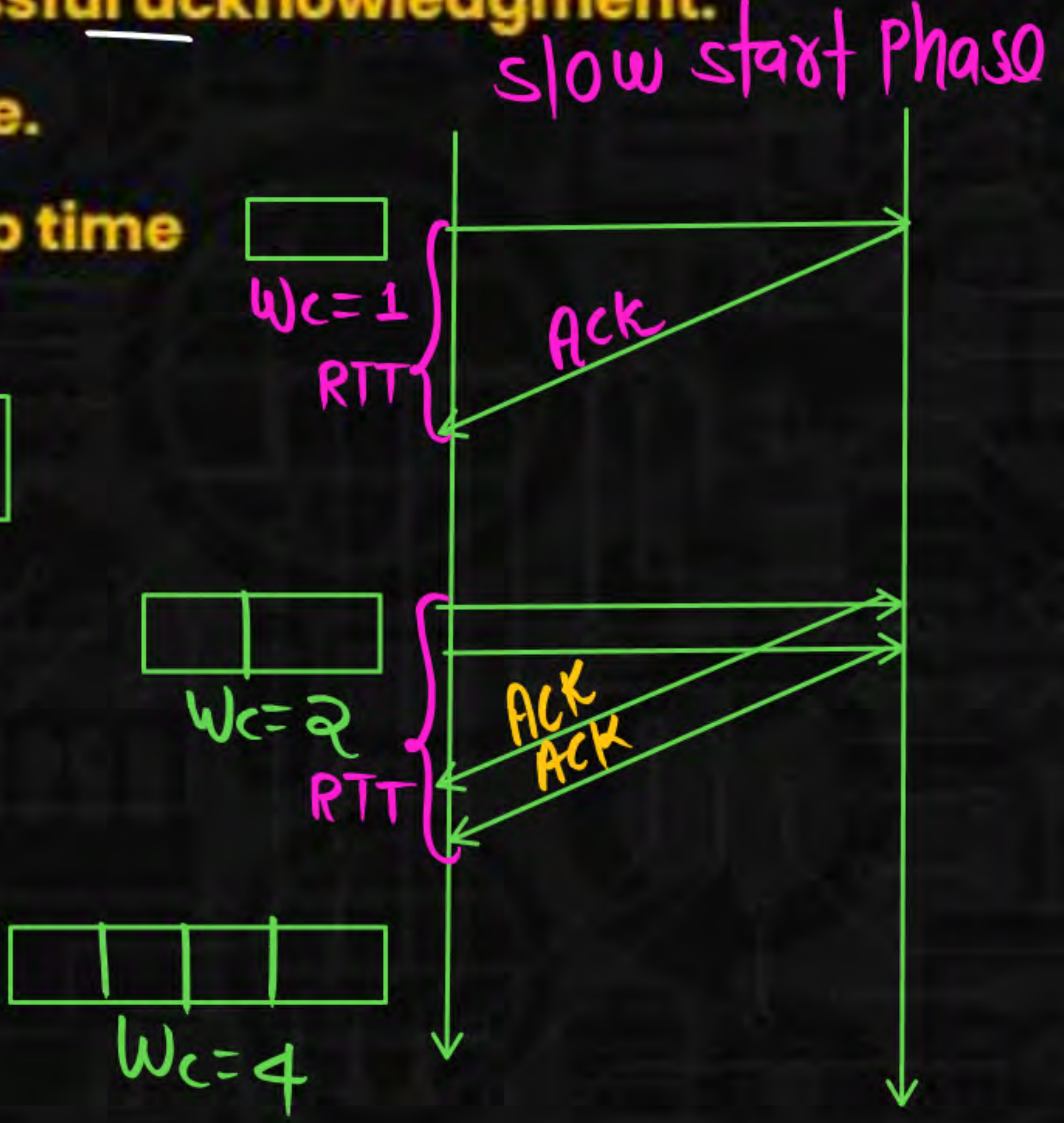
X ii. The cwnd approximately doubles on every successful acknowledgment.

X iii. The cwnd increases by 1 MSS every round trip time.

✓ iv. The cwnd approximately doubles every round trip time

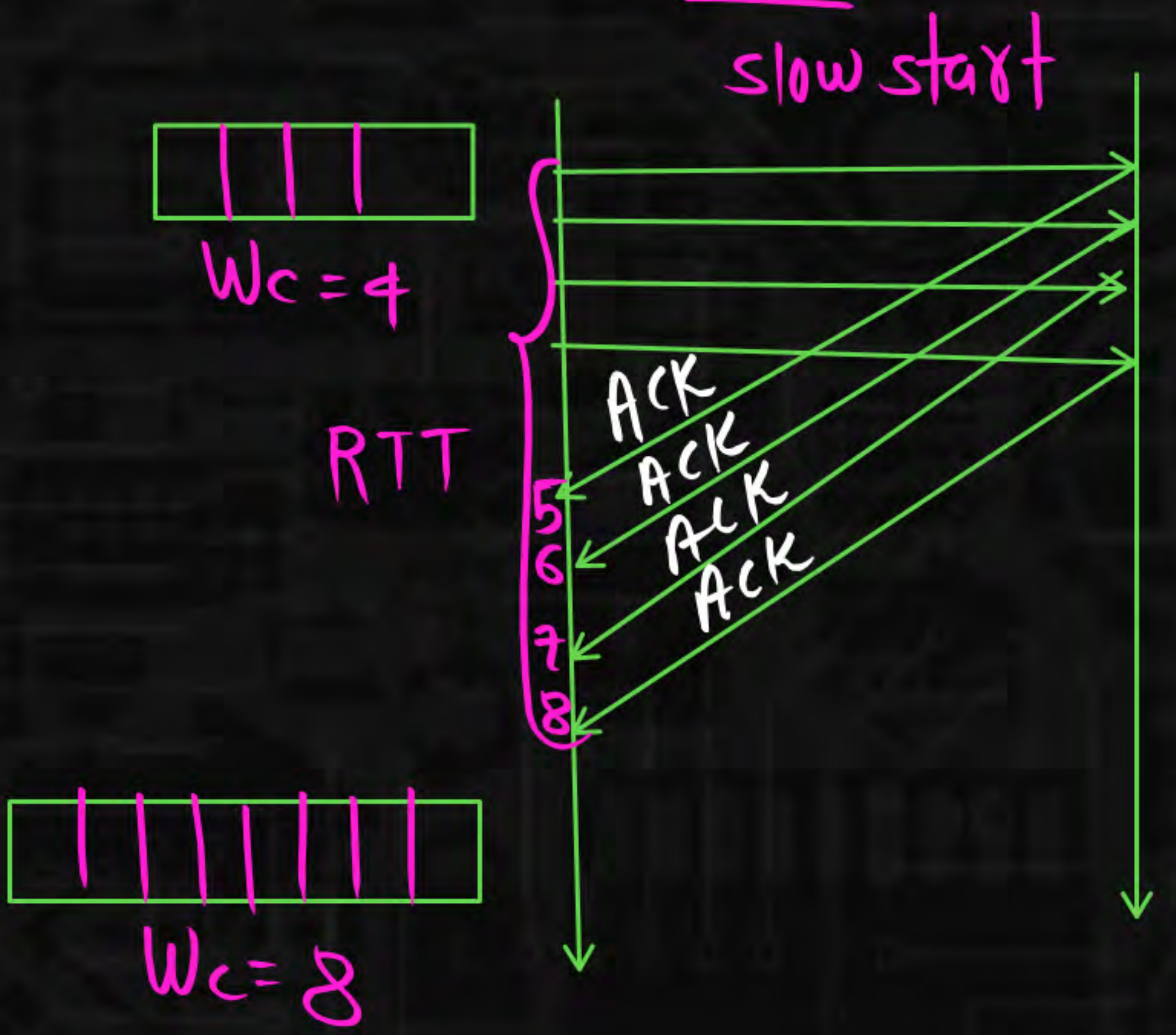
Which one of the following is correct? (GATE – 2018)

A. Only (ii) and (iii) are true
B. Only (i) and (iii) are true
✓ C. Only (iv) is true
D. Only (i) and (iv) are true



Q 2. In slow start phase of TCP congestion control, current congestion window size (CWND) is 4 MSS and sender gets 4 successful ACK of segments (no any outstanding ACK) then what should be the value of CWND? (NIELIT) 2017

A. 5 MSS
B. 8 MSS
C. 16 MSS
D. 64 MSS



Q 3. If receiver window size is 16000 Byte and maximum segment size is 1000 Byte then after how many RTT sender will send full window----------? 11

Sol^n:

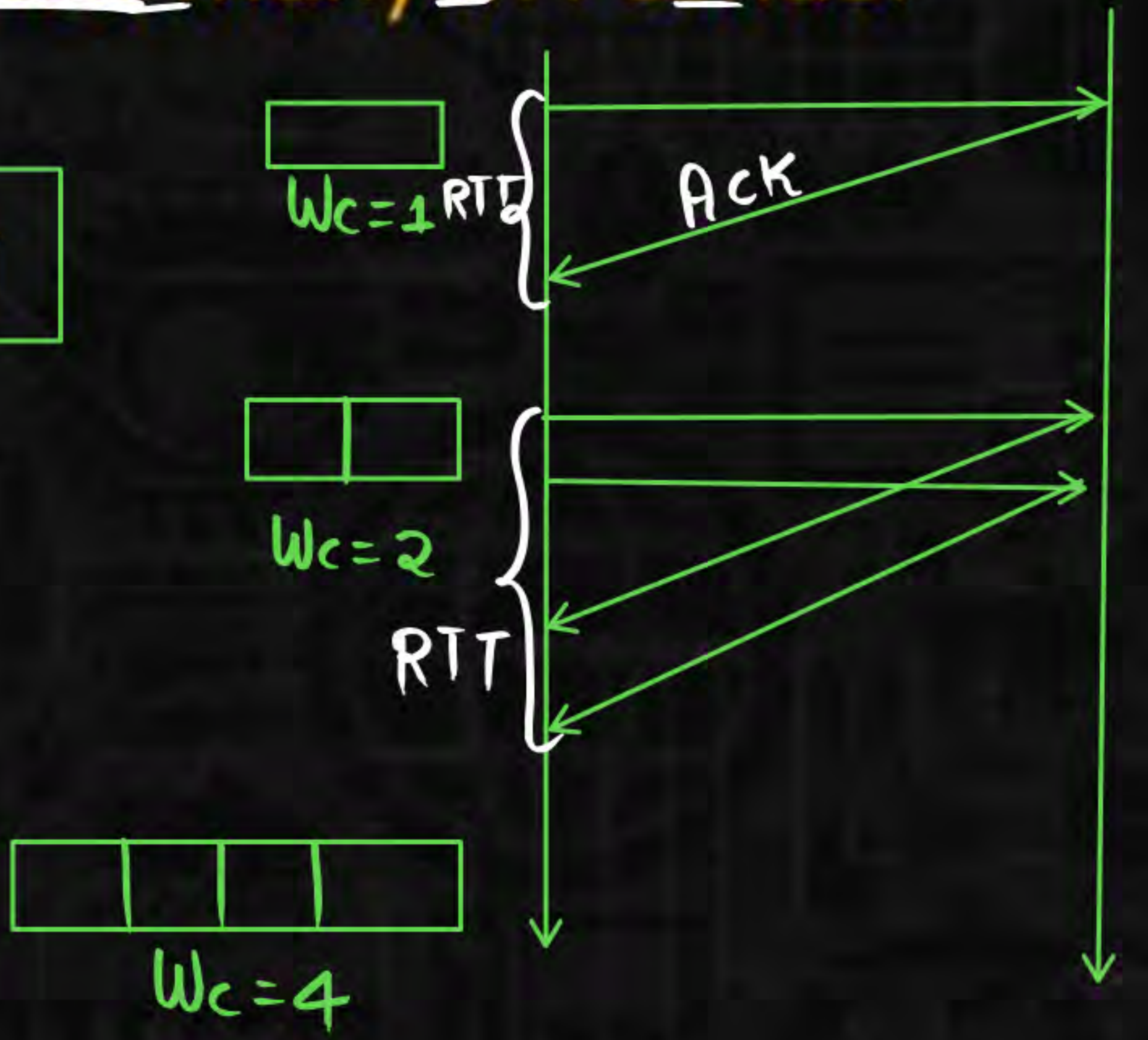
$\boxed{1 \to RTT}$

$W_R = 1600$ Byte

$MSS = 1000$ Byte

No. of segments $= \dfrac{16000\cancel{B}}{1000\cancel{B}} = 16$

$TH = \frac{1}{2} W_R = 8$ segment



RTT

TH

1 | 2 | 4 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16

1 2 3 4 5 6 7 8 9 10 11

Q 4. If receiver window size is 8000 Byte and maximum segment size is 1000 Byte then after how many RTT sender will send full window----6---?

$W_R = 8000$ Byte

$MSS = 1000$ Byte

No. of segments $= \frac{8000\cancel{B}}{1000\cancel{B}} = 8$

$TH = \frac{1}{2} W_R = 4$ segment

TH

1 | 2 | 4 | 5 | 6 | 7 | 8

1 2 3 4 5 6


THANK YOU
GW
SOLDIERS !