

CS & IT ENGINEERING

COMPUTER NETWORKS

TCP & UDP

Lecture No-10



By- Ankit Doyla Sir



TOPICS TO
BE
COVERED



Silly window syndrome

Silly window Syndrome

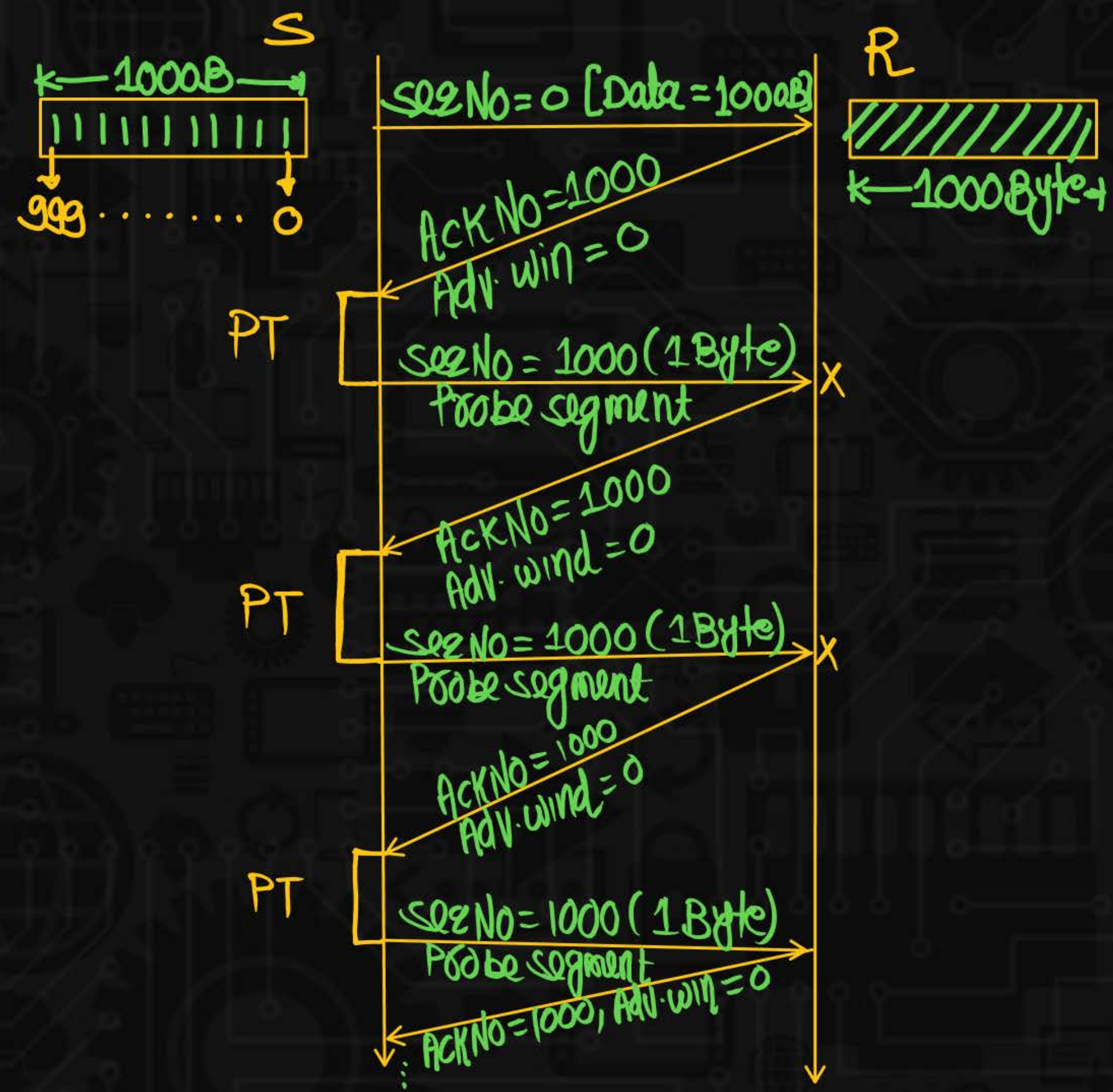
Silly window syndrome:

- > A serious problem can arise in the sliding window operation when either the sending application program creates data slowly or the receiving application program consume data slowly, or both.
- > Any of these situations results in sending of data in very small segments, which reduce the efficiency of the operation.
- > For example if TCP send segments containing only one byte of data, it means a 41 byte datagram (20 byte of TCP header and 20 byte of IP header) transfer only one byte of user data.
- > Here the overhead is $41/1$, which indicates that we are using the capacity of network very inefficiently.
- > The efficiency is even worse after accounting for the datalink layer and physical layer overhead. (Header)
- > This problem is called silly window syndrome

Reasons for silly window syndrome:

1. Whenever receiver announces that its receiving capacity is zero it leads to silly window syndrome.
2. Whenever sender produce only one byte at a time it leads to silly window syndrome.
3. Whenever receiver consume only one byte at a time it leads to silly window syndrome.

1.



2

S

D = 1 Byte

Adv. wind = 999 B

Data = 1 Byte

Ack

Data = 1 Byte

Ack

Data = 1 Byte

Ack

Data = 1 Byte

Ack

Silly window syndrome

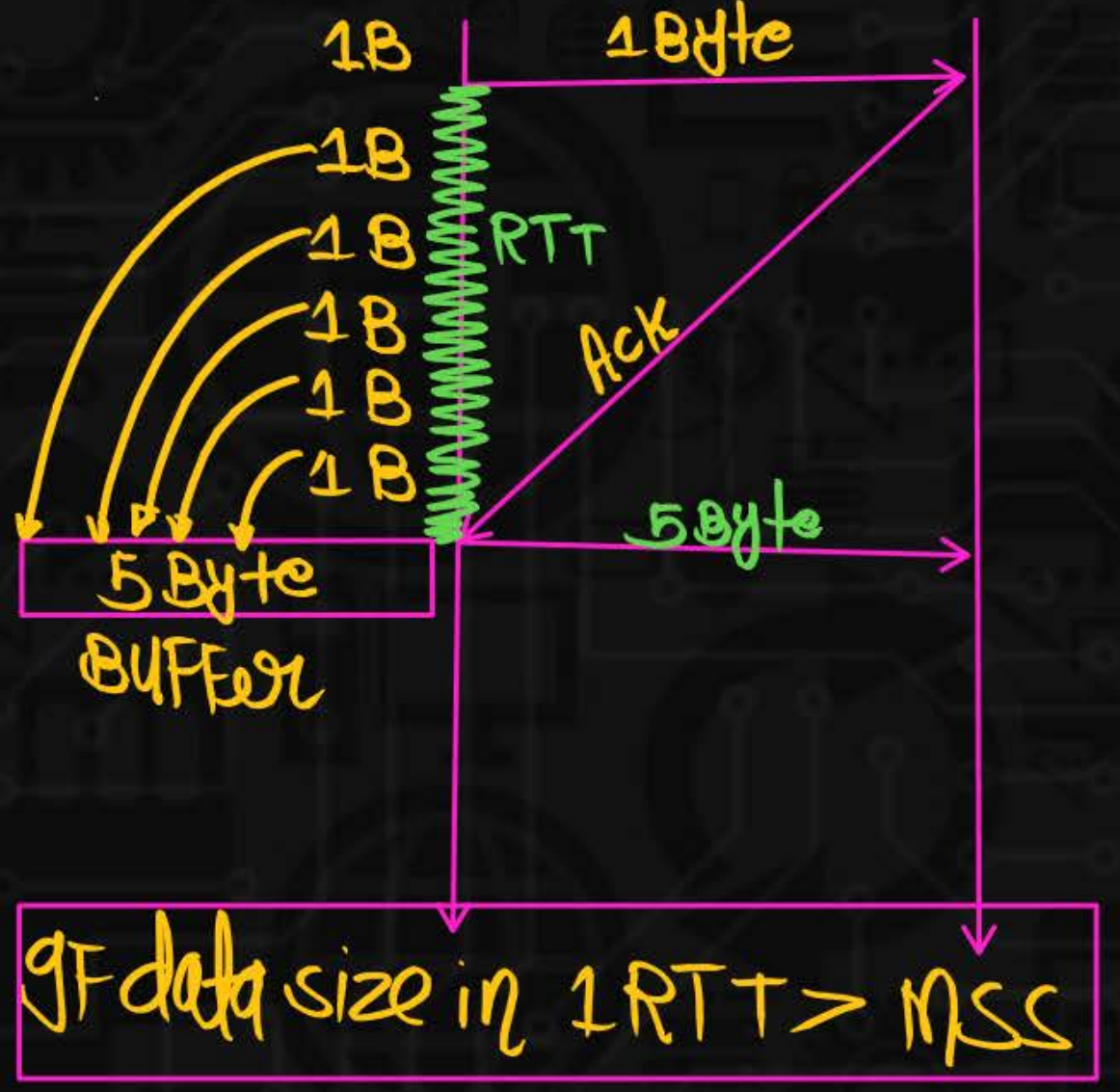
R

999



← 1000 B →

Nagle's Algorithm



1 Byte/sec

RTT = 100 sec

MSS = 200 Byte

In one RTT = 100 Byte

So we transfer 100 Byte

If data size in 1 RTT < MSS

1 Byte/sec

RTT = 100 sec

MSS = 50 Byte

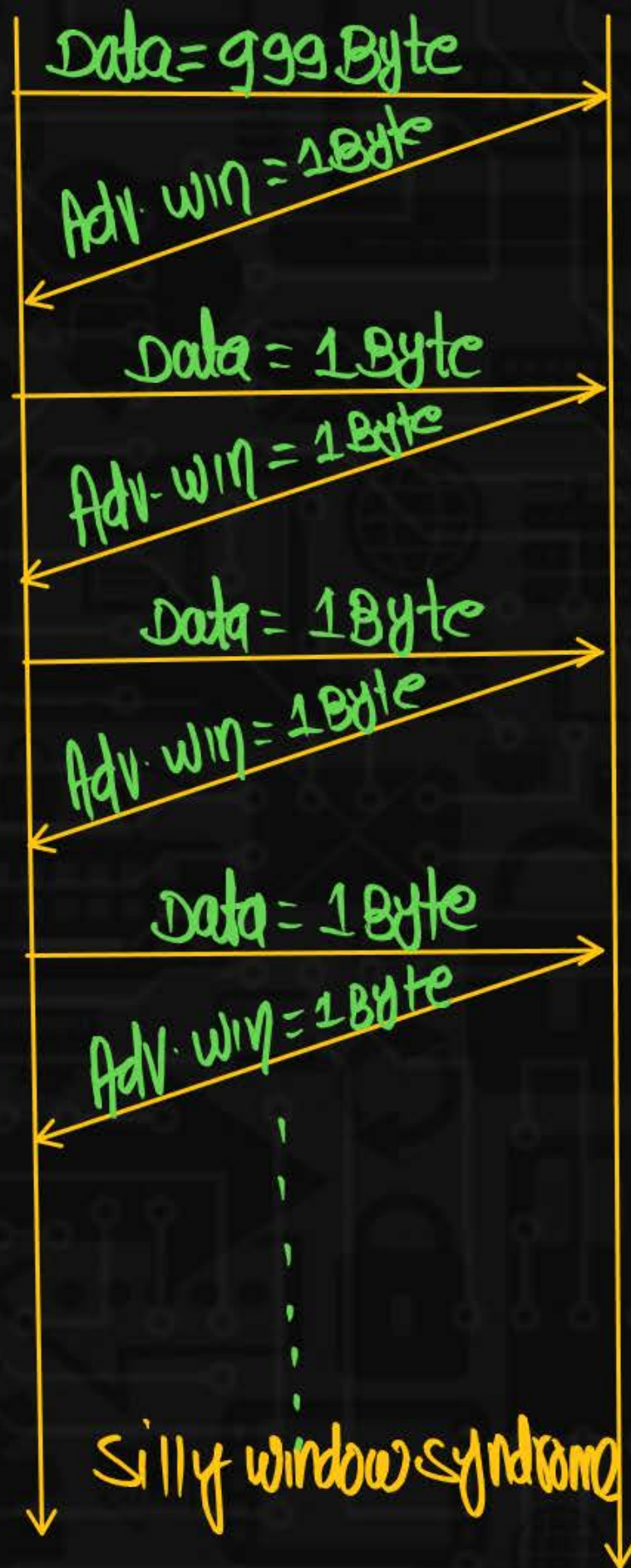
In one RTT = 100 Byte

So we transfer 50 Byte

data size in one RTT > MSS



3



Clark's solution

$\frac{1}{2}$ Buffer empty
or
1 MSS

Nagle's algorithm:

1. The sending TCP send first piece of data it receive from application program even if it is only one byte.
2. After sending the first segment, the sending TCP accumulates data in the output buffer and waits until either the receiving TCP sends an acknowledgement or until enough data have accumulated to fill a maximum size segments. At this time the sending TCP can send the segment

Clark's Solution:

Two solutions has been proposed by clark's

First solution: Receiver send the acknowledgement as soon as the data arrive , but to announce a window size of zero until either there is a enough space to accommodate a segment of maximum size or until atleast half of the receiver buffer is empty.

Second solution: (Delay acknowledgement)

When a segment arrives, it is not acknowledged immediately. Receiver wait until there is a decent amount of space in its buffer before acknowledging the arrived segment.

