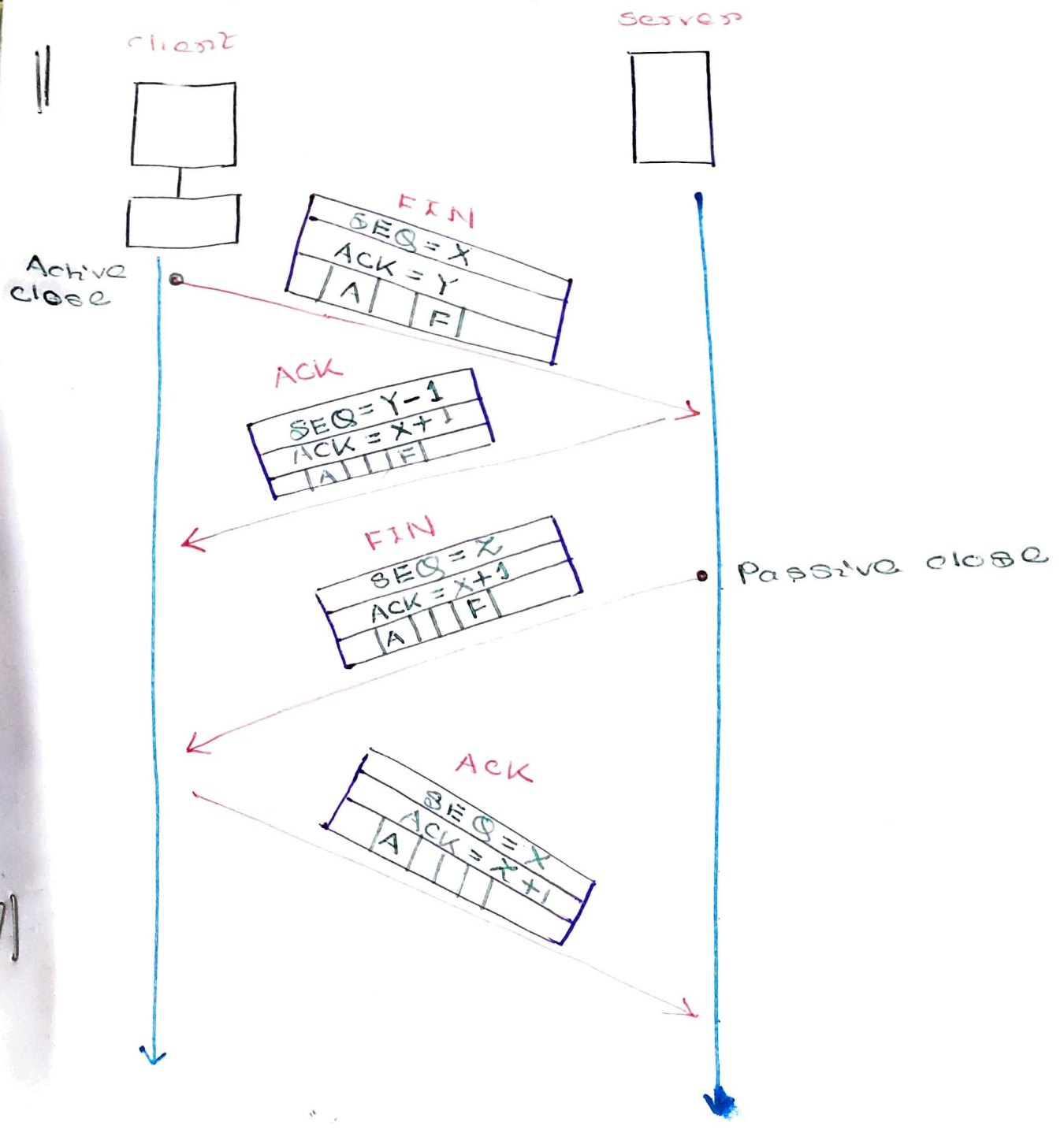
Four way handshaking :-



Sliding window:- window is used for flow control at the receiver.

Flow control is usually done in two layers:-

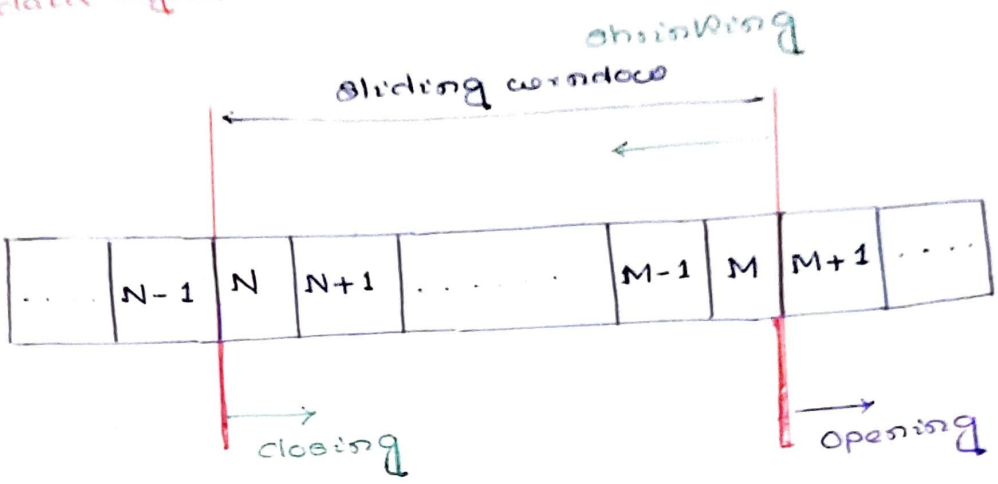
1. Transport layer.
2. Datalink layer.

Now we are in Transport layer:-

- TCP uses a sliding window to handle flow control.
- From Go-back-N Selective Repeat
  - does not use NAK.
  - The receiver holds out-of-order segments
- TCP sliding window is of variable size. where as datalink layer sliding window is of fixed size.
- TCP Sliding window is byte-oriented. whereas datalink layer sliding window is of frame oriented.

The window takes care about

① the data bytes which are sent
② the data bytes which are sent and acknowledged
③ the data bytes which are yet to sent



window size = minimum (rwnd, cwnd)

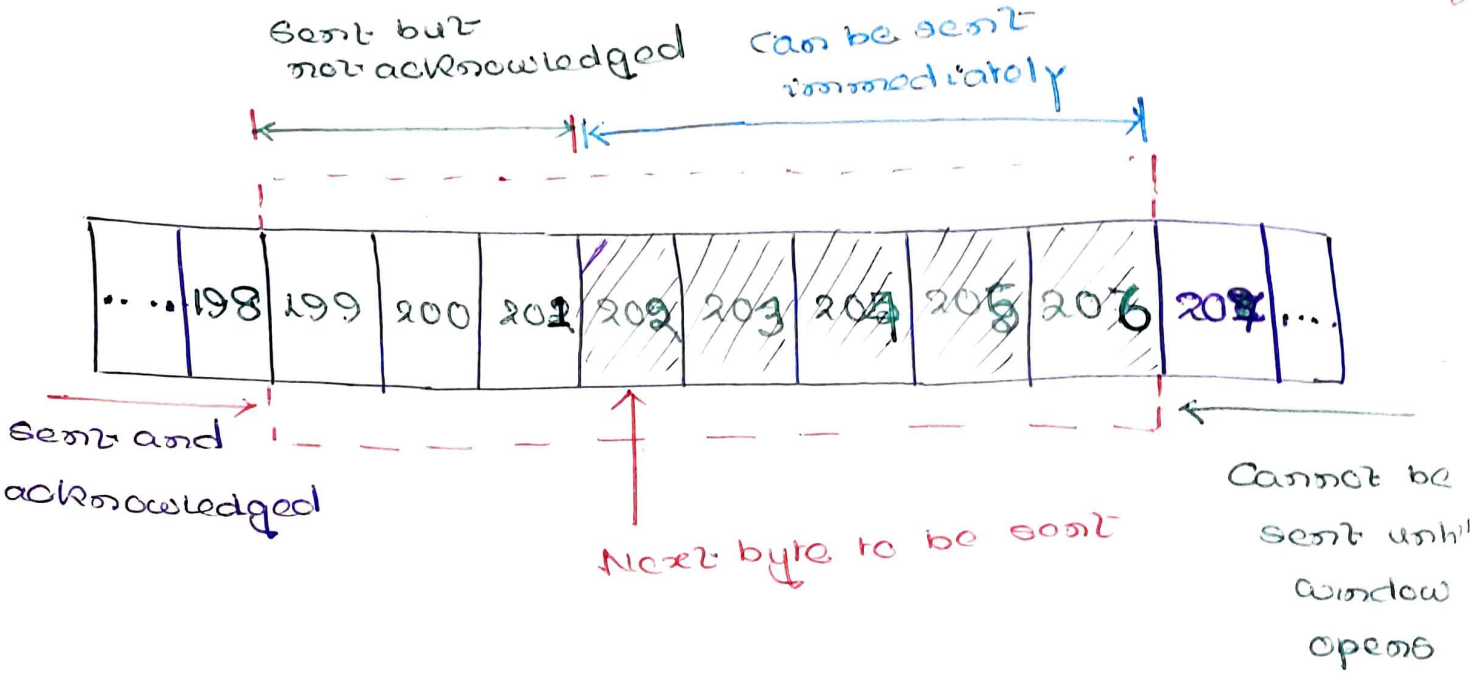
rwnd → receiver window size
cwnd → congestion window size

The window size is of variable length.
The data which are sent but not have been acknowledge and the data which are to be sent immediately as will be in the window.
Suppose the N number segment is sent and get acknowledged now the window will move towards (N+1) and now the window can allow the (M+1) number segment to be sent in the queue.

- Congestion window is the ~~pre~~ feature of intermediatary devices such as router or switches. Suppose the receiver has a window size of 50. But the router through which many segments of other ongoing communications are processing, so there is a congestion.

Window size = minimum (20, 8) = 8

- 20 → receiver window size
- 8 → congestion window size (size of network window)



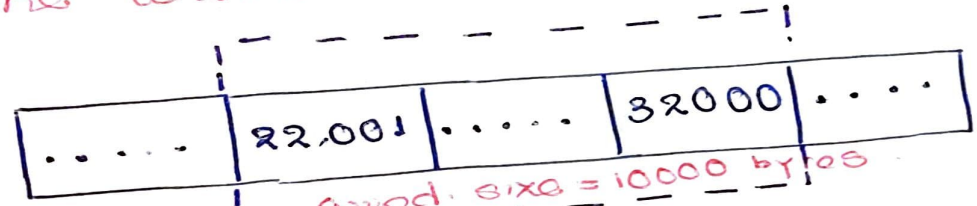
- window size = min (rwnd, cwnd).
- The source does not have to send a full window's worth of data.
- The window can be opened or closed by the receiver, but should not be shrunk.
- The destination/receiver can send an ACK at any time.
- The sender can send 1-byte segment even after the window is shut down by the receiver side.

Q. What is the size of the window for host A if the value of rcvwnd (receiver window) is 3000 bytes and the value of cwnd (congestion window) is 3500 bytes?

⇒ We know, window size = min (rcvwnd, cwnd)
= min (3000, 3500)
= 3000 bytes

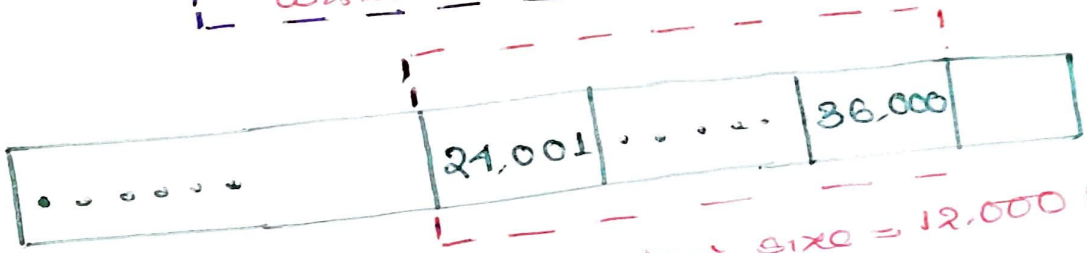
Q. A TCP connection is using a window size of 10,000 bytes and the previous acknowledgement number was 22,001. It receives a segment with acknowledgement number 24,001 and window size advertisement of 12,000. Draw a diagram to show the situation of the windows before and after.

22,001
+ 10,000
32,001
32,000



before receiving the segment

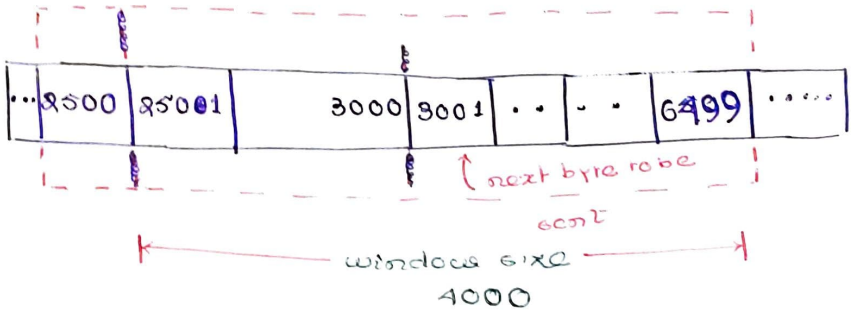
24,001
+ 12,000
36,001
36,000



Q. A window holds bytes 2001 to 5000.
The next byte to be sent is 3001.
Draw a figure to show the situation of the window after the following two events:-

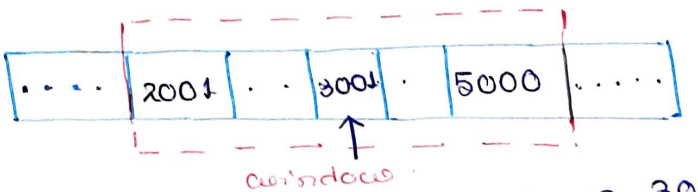
(a) An ACK segment with the acknowledgment number 2500 and window size advt. 1000 is received.

(b) A segment carrying 1000 byte is sent.

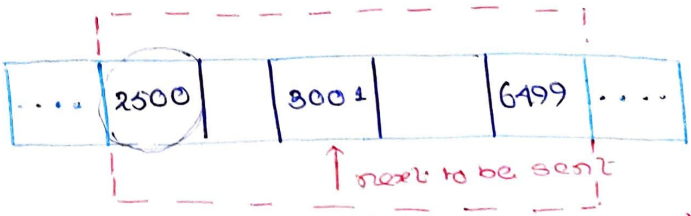


2501
+ 1000
6501
− 1
6500

Initial Figure:- The window holds byte 2001 to 5000.

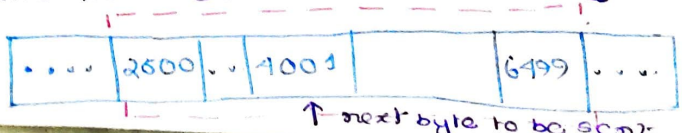


The next data byte to be sent is 3001
An acknowledgement for 2500 is received and the new window size is 4000.



2500✓
+4000✓
6500
− 1
6499

* A segment carrying 1000 byte is sent.
means from 3001 next 1000 bytes is sent

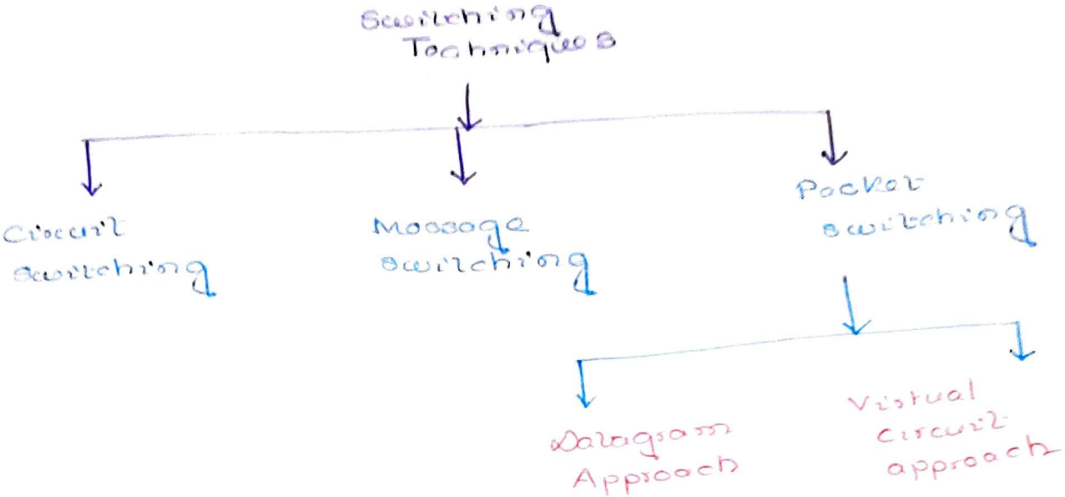


3001
+1000
4001

From 3001 to 4000 total 1000 byte sent.

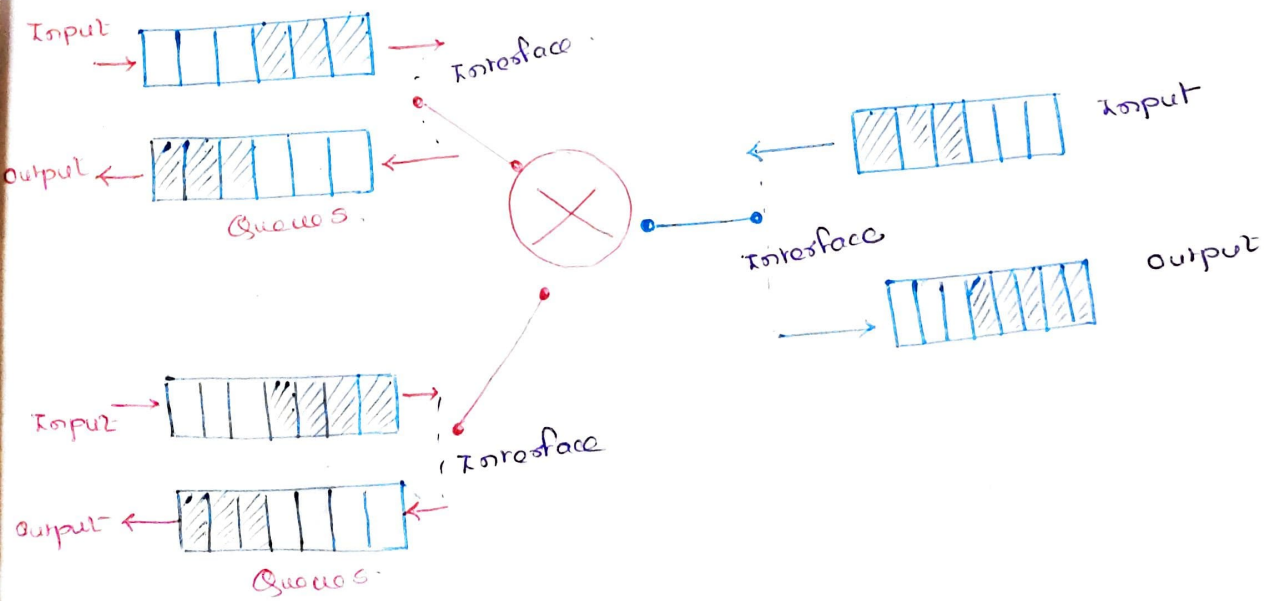
Congestion:-

We know there are three types of switching



Internetwork communication is not possible without routers. But routers has limited memory capacity. Routers has memories in form of queues.

Suppose a movie hall has limited amount of seats. As there are no prior seat booking is allowed. If the number of people arrived at the hall is large than the no. of available seats then the congestion occurs. Same way congestion in computer networks occurs at the routers/switches as routers maintains buffering - queues before and after processing the data.



(1) The packets are put at the end of the output queue.

(2) Processing the packets have the queue. The packet should be send in the right/ correct destination. For this the routing has to follows the routing rules.

(3) The router will put the packets in the output queue.

* When all the memory of the input queue is filled then congestion occurs. If the arrival rate is very high then the processing rate must be congestion occurs departure rate is slow in comparison to the processing rate then more will be congestion at the output queue.

Congestion control :-

(1)

Open loop (Before happening) (prevention method)	closed loop (After happening) (Removing method)
→ Retransmission policy	→ Back pressure
→ Windows policy	→ choke packet
→ Acknowledgement policy	→ Implicit signalling
→ Discarding policy	→ Explicit signalling
→ Admission policy	

Open loop congestion control mechanism :-

- Retransmission policy :-
- As reliability is ~~g~~ guaranted in TCP so till the time acknowledgement is received by the receiver the data packets will be retransmitted.
- Retransmission will also be done incase of loss or corrupted or damaged packets.
- But this will increase congestion.

- A good retransmission policy can help
- The TCP's retransmission policy is designed to prevent or alleviate congestion.

Window Policy:-

(1) window at sender may also affect congestion
Suppose, it is busy or congested network
if the window size at the sender is
larger than sender will be sending
more no. of data packets which can
again make the network more congested.

(2) SR (Selective Repeat) windowing is more
better than Go-back-N window
In Go-back-N if one segment is corrupted
or missing or damage then the entire
window is retransmitted, whereas in
Selective Repeat window only the missing/
corrupted segment is retransmitted again.

(3) This duplication in Go-back-N may worse
congestion.

(4) Selective Repeat retransmits only lost or
corrupted packets.

Acknowledgement Policy:-

(1) Ack policy imposed by the receiver may also
affect congestion.
Suppose, there are 10 (N=10) packet if the
network is already congested and acknowledg
ment has to be sent for each of the 10 packets
then the network will be loaded by both
data and also with acknowledgment packets.

(2) So either then sending Ack for each individual
packet if acknowledgement can be sent
for N=10 packets together, than
fewer Acks impose less load.

Discarding policy:-

- A good discarding policy
  if loss sensitive data packets can be discarded then by discarding flow loss important on sensitive data packets where the integrity of the data sent is still acceptable, then this type of discarding policy can make the network less congested.

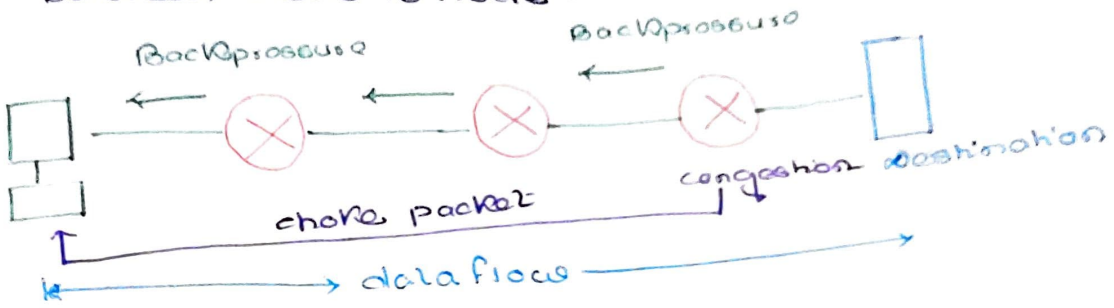
- Ex:- audio transmission

Admission policy:-

- QoS (Quality of Service) mechanism prevents congestion in virtual circuit network. → (virtual ckt network means no physical connection is established before transmission of data packets).
- Switches in a flow first check the resource requirement of a flow before admitting it to the network.
- Routers can also deny if there is a possibility of future congestion.

closed-loop congestion control:-

→ Backpressure mechanism is a virtual connection control mechanism ~~in node~~ between node to node.



Cannot be applied to datagram.
Opposite direction to the dataflow.
It will inform the ~~up~~ previous upstream node about the congestion in one node.

choke packet:-

choke packet is not a node-to-node mechanism.
choke packet is directly sent between the congested node and the source. It is a special type of packet where it will not pass through the intermediatary node and will be sent directly to the source for flow control. choke packet ~~ICMP message~~ will take help of ICMP messages.

Implicit signalling:-

There will be not implicit communication between the congested node and the source to between the congested node and intermediate nodes. The source will get ~~no~~ to know about the congestion by No ACK (acknowledgment) packet received or delayed ACK.

Explicit signalling :-

There will be explicit between congested node and source. Here signals will be included in the data packets itself.

- Routers can discard less sensitive packets than carry data.
- There are two types of Explicit signalling

(1) Backward signalling.

(2) Forward signalling.

Congestion control in TCP –

Input

congestion occurs to the input queue when the packet arrival rate is larger than the processing rate of the router.

Congestion occurs at the output queue when the packet despatch rate is slower than the processing rate of the routers.

In both the cases there will be congestion at the routers.

Who decided the window size?

window size = min (rwnd, cwnd)

rwnd → receiver window size

cwnd → routers window size

which the set by the sender after consulting to the intermediatary device (e.g- routers) and the receiver.

Congestion control in TCP :—

Three congestion polic

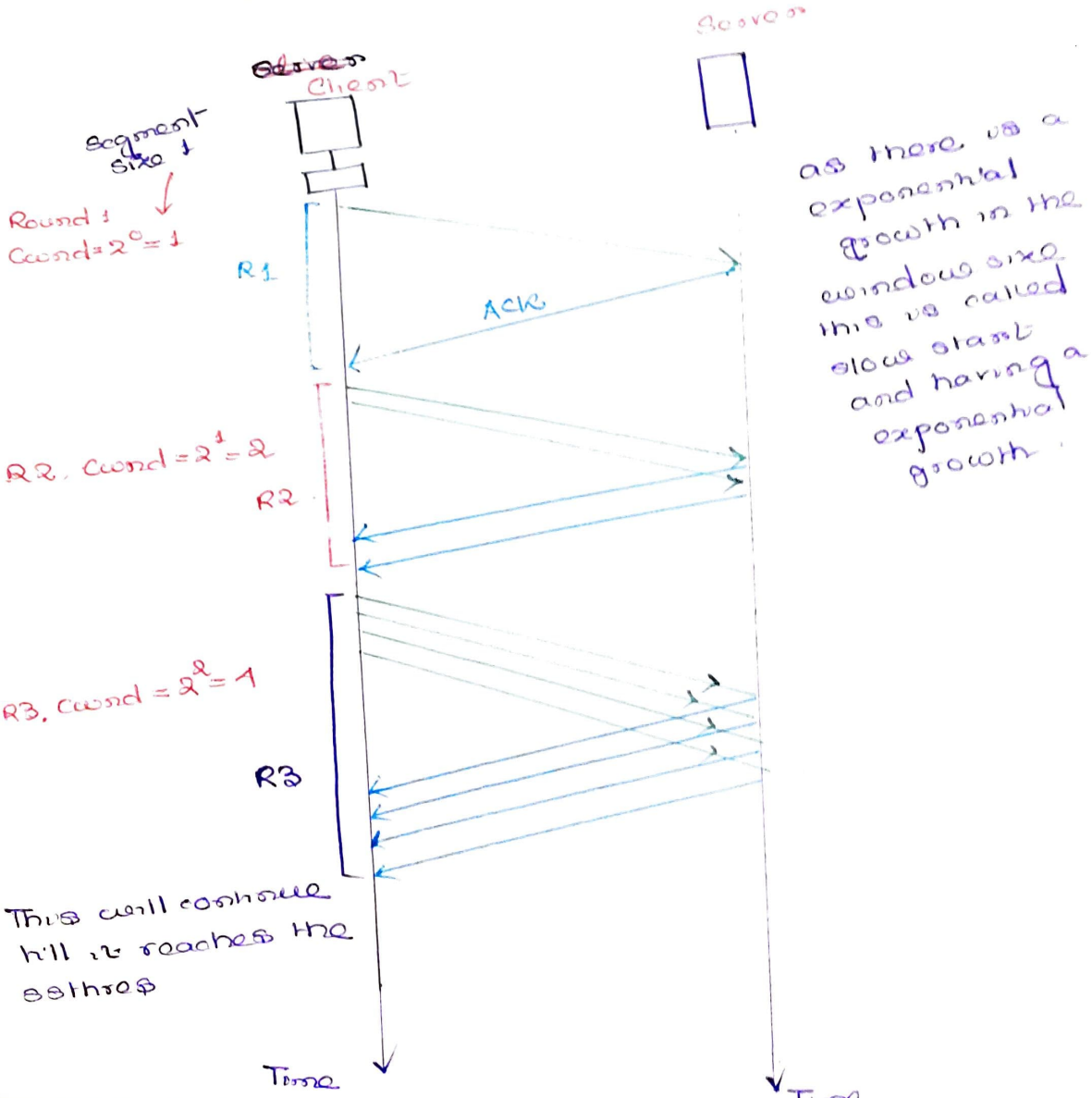
1. Slow start
2. Additive increase
3. Multiplicative decrease.

Slow start Algorithm :-

1. The size of the congestion window (cwnd) starts with one maximum segment size (mss).
2. The MSS is determined during TCP connection establishment.
3. The window starts slowly but grows exponentially.
4. In the slow start algorithm, the size of the congestion window increases exponentially until it reaches a threshold.

Assumptions:-

1. The value of rwnd > cwnd and hence the window size = cwnd
2. Each segment is acknowledged individually



Some important points about slow start:

1. Slow start cannot continue exponentially. There must be a threshold to stop this exponential growth.
2. The sender keeps track of a variable named ssthresh (slow-start threshold) which is the size of the window in bytes reached at the beginning of slow start steps and the slow start stops and the congestion avoidance phase starts. (i.e. the Additive increase algorithm starts)
3. In most implementations the value of ssthresh is $(2^{16}-1)$ i.e. 65,535 bytes.

Additive Increase Algorithm :—

Congestion avoidance — Additive Increase

There are three phases in congestion control in TCP

1. Slow start
2. Additive Increase (Congestion Avoidance)
3. Multiplicative Decrease (Congestion Detection)

In slow-start process cwnd increases exponentially by a power of 2. This exponential growth must be slowed down, to avoid congestion before it happens.

Now for congestion avoidance additive increase will be implemented. When cwnd reaches the threshold the slow start steps and the additive phase begins.

In this algorithm, each time the whole window of segments is acknowledged (one round) the size of the congestion window is increased by 1.

Suppose in the earlier diagram (new diagram). suppose the ssthresh is 4, i.e. after receiving the acknowledgement of round 3 i.e. R3 now the next window size will not be 2^3 (i.e. 8) it will be $(2^2=4+1)$ i.e. 5.

In the congestion avoidance algorithm, the size of the congestion window increases additively until congestion is detected.

Congestion detection - Multiplicative Decrease

Slow start - exponential increase in cwnd

additive increase - linear increase " "

The cwnd must be decreased, if there is a congestion

Now sender can guess if there is a congestion if the ack message is timed out or delayed

Retransmission of the segments have to be done

Retransmission can occur in two cases :-

(1) when a timer times out or

(2) when three ACKs are received.

- Threshold is dropped to one-half, a multiplicative decrease. when the cwnd reached the ssthresh (suppose the value is 20) and congestion is sensed by the sender and retransmission has to be done, then this time the ~~ssth~~ cwnd size will be decrease to $\left(\frac{1}{2} \times 20 = 10\right)$

Now after congestion occurs Tcp implementations have two reactions :-

Reaction 1 :-

1. It sets the value of the threshold i.e. ssthresh to one-half of the current window size
2. It sets cwnd to the size of one segment i.e $2^0 = 1$
3. It starts the slow-start phase again.

Reaction 2 :-

1. It sets the value of the threshold i.e. ssthresh to one half ~~of~~ of the current window size.
2. It sets the cwnd to the value of threshold
3. It starts the congestion avoidance phase i.e. the additive increase, that is the cwnd will be increase by 1 after each round.