01fe18bcs211
H19

8c) In process to process communication the processes are identified by port numbers.

Ports numbers range from 0-65535. They have 16-bit address as identifier. Port addressing in unique within the host.

To extend host-to-host delivery service provides by network layer port numbers is very much necessary.

They are analogous to door. the data moves from these ports to network from processes.

Well-known ports range from 0-1023 addresses. Usually assigned to servers.

Ex: HTTP: 80
FTP: 21
SMTP: 25

Ports that are for user purpose and are not well-known is ephemeral ports ranging from 1024-65535.

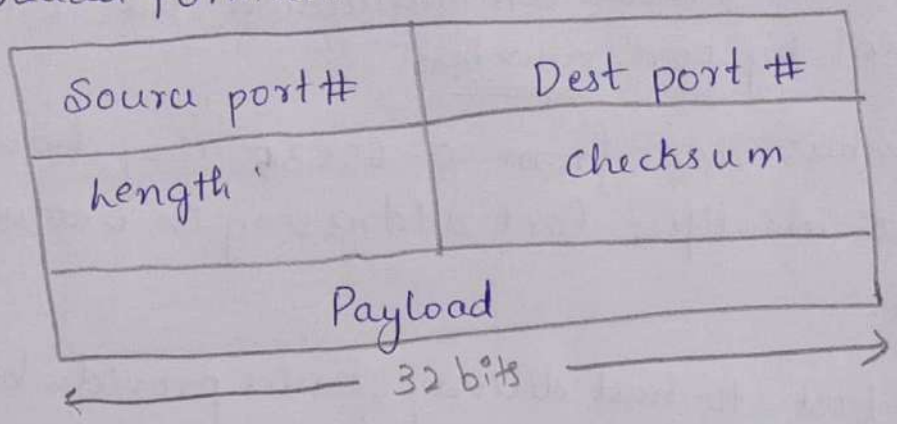
Well-Known TCP ports:

1) HTTP 80
2) SMTP 25
3) FTP 21
4) RPC 111
5) Telnet 23.

Well-known UDP ports

1) DNS 53
2) Daytime 13
3) SNMP 161
4) NTP (network time protocol) 123.
5) Chargen 19

UDP header format:



i) Source port number = 0X CA 45 = 51781

ii) Destination port number = 0X 000D = 13 (Daytime)

iii) Total length (header + data) = 0X001C = 28 bytes

iv) length of data = Total length - header length.

= 28 - 8

= 20 bytes.

8b) Datagram = 2800 bytes. Header = 20 bytes

MTU = 780 bytes.

Identification number = 312.

How many fragments ?

$$\text{No. of fragments} = \text{ceil}\left[\frac{\text{Datagram size} - \text{header size}}{\text{MTU size} - \text{header size}}\right]$$

$$= \text{ceil}\left(\frac{2800-20}{780-20}\right)$$

$$= \text{ceil}(3.6578)$$

= 4 fragments are required.

fragment no.	No. of bytes	Id	Offset	flag
1	760	312	0	1
2	760	312	95	1
3	760	312	190	1
4	500	312	285	0

Offset = $\frac{\text{first byte}}{8}$

These are the valuese Id, Offset and flags in IP datagram.

8a) Ipv4 datagram:

(i) TTL value = 0x25 = 37 more routers the packet can travel to

(ii) Source address = 0x8C6E0302

Destination address = 0x D41E 0F02

(iii) Size of header = 0x05 * 4d
= 5 * 4
= 20 bytes.

Total length = 0x0074 = 116 bytes

Size of data = Total length - header = 116 bytes - 20 = 96 bytes.

7a) Three-way-handshaking

Step1: The application layer tells its transport layer that it wants to connect to another host or server. Then the transport layers makes a TCP segment in which SYN bit is set. Therefore it is called SYN segment. The sequence number is set as client_isn. This sequence number is randomly choses to avoid that more than one packet in network carry the same client_isn. The transport layer passes the SYN TCP segment to the IP layer which makes the datagram and puts it on the Internet. It requests for connection.

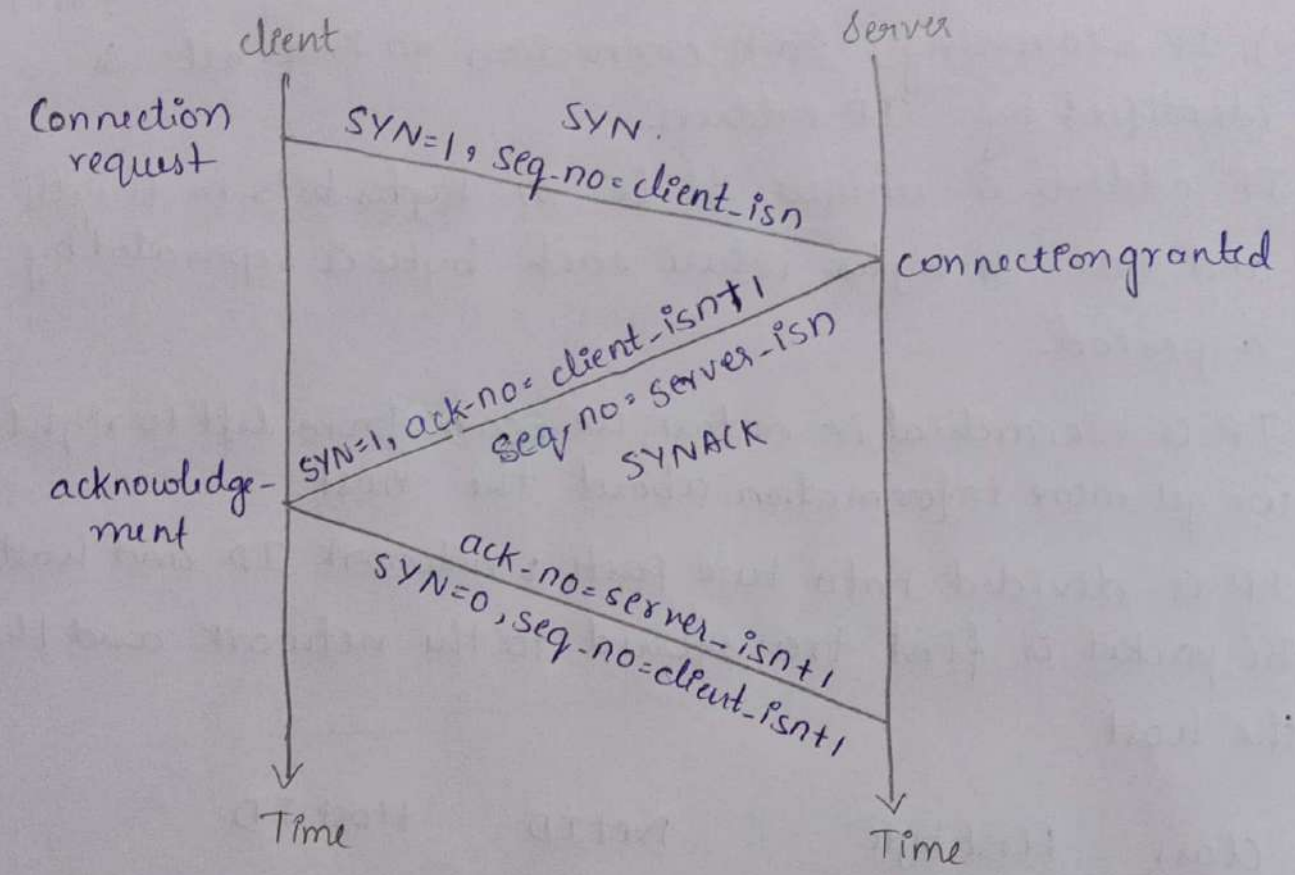
Step 2: The server recieves the SYN segment and grants the connection using SYNACK segment where SYN is set to 1 and ACK is set as client_isn + 1. The sequence number now is server_isn.

Step 3: The client recieves the segment SYNACK and sees that the connection is granted and sends back an acknowledgement to server where SYN=0, sequence number = client_isn+1 ACK = server_isn + 1

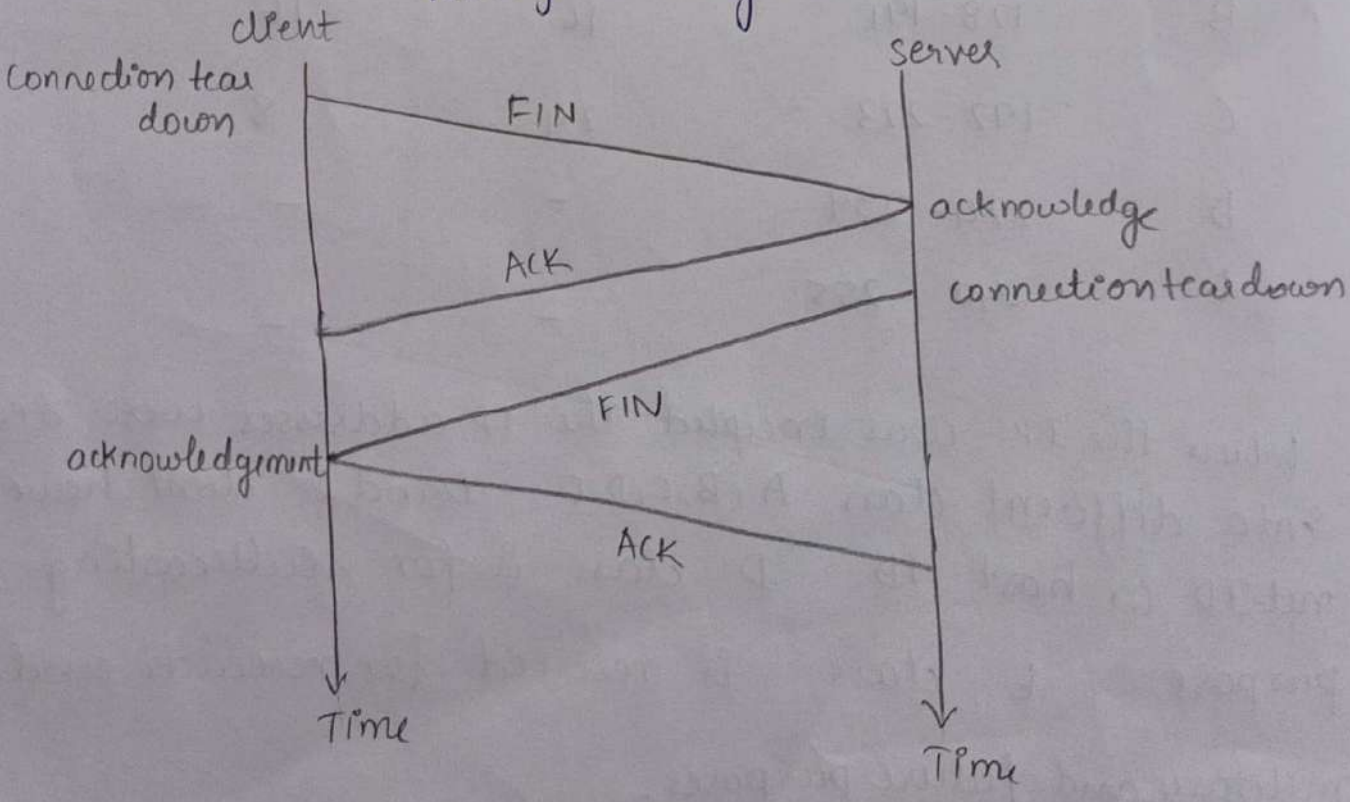
Connection - tear down

* The client sends the FIN segment to request for connection teardown and then recieves an ACK from server. The server also sends FIN segment to client and gets back an acknowledgement. Thus connection is teared down.

01fe18bcs211
H19



TCP connection establishment through 3-way handshake.



7c) IP addressing: Each connection to internet is identified by IP address.

IP address is unique. It is 32-bits, bits in length and has 4 bytes. where each bytes is separated by a period.

IP is hierarchial in nature we scan from left to right we get more information about the host.

IP is divided into two parts: network ID and host ID. The packet is first transferred to the network and than the host

Class	First byte	NetID	Host ID
A	0 - 127	8	24
B	128 - 191	16	16
C	192 - 223	24	8
D	224 - 239	-	-
E	240 - 255	-	-

When the IP was incepted the IP addresses were divided into different class A, B, C, D, E. D and E dont have net ID or host ID. D class is for multicasting purpose. E class is reserved for research and military and future purposes.

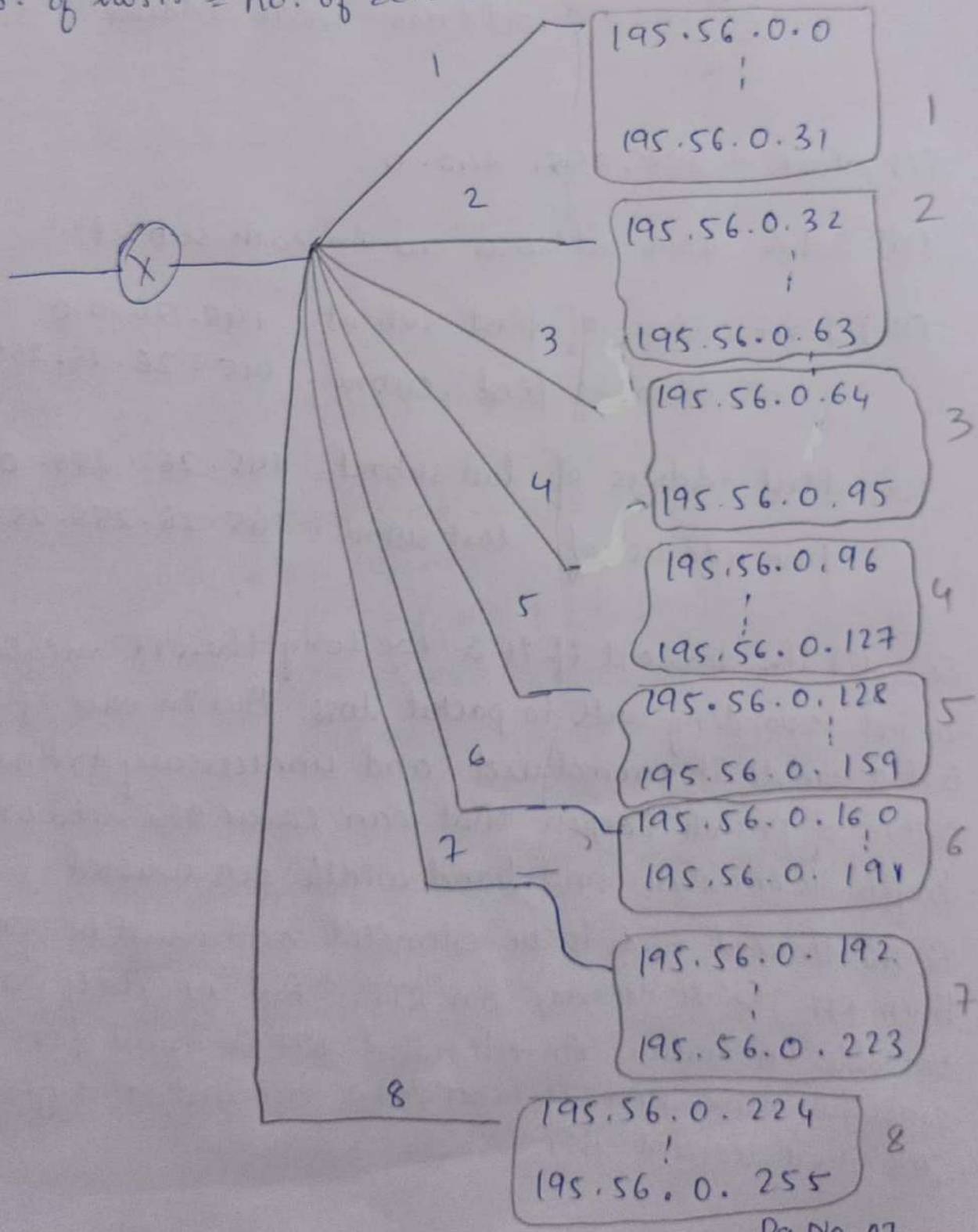
Address space is 2^{32}.

Granted address 195.56.0.0. which belongs to class C.

No. of subnets is 7. nearest $2^3 = 8$. (3 bits more)
(last byte = 111 00000

Mask = 255.255.255.224

No. of hosts = no. of zeroes in mask = 5 = 32 (hosts)



6b) Address block available with organization.

$2^{32-mask} = 2^{32-16} = 2^{16}$ address.

The administration wants 16 subnets.

$$\frac{2^{16}}{24} = 2^{12} \text{ address} = 4096 \text{ address}.$$

(i) Mask = 255.255.240.0.

(ii) Total 4096 addresses are in each subnet.

(iii) First addres of first subnet = 145.26.0.0
Last address of first subnet = 145.26.15.255

(iv) First address of last subnet = 145.26.240.0
Last address of last subnet = 145.26.255.255

6c) (i) The timeout if it is too long then it might be not responding well to packet loss. But in case if it is too small it's premature and unnecessary transmissions of packets happen that can cause the network traffic to increase and band width gets wasted So the timeout needs to be estimated and must be greater than RTT. So it depends on RTT. But RTT itself is variable in nature. An estimated RTT is must for smoother curve and it takes into account and gives priority to recent RTT.

01fe18bcs211

H19

(ii) Estimated RTT = $(1-\alpha)$ * estimated RTT + α * sample RTT

DevRTT = $(1-\beta)$ * DevRTT + β | estimated RTT - sample RTT |

Timeout = Estimated RTT + 4 * DevRTT

(iii) Sample RTT = 320 msec $\alpha = 0.125$

Estimated RTT = 300 msec. $\beta = 0.25$

Dev RTT = 51 msec.

Estimated RTT = $(1 - 0.125) * 300 + 0.125 * 320$.

$= 262.5 + 40$

$= 302.5$ ms

Dev RTT = $(1 - 0.25) * \text{DevRTT} + \beta$ | estimated RTT - sample RTT |

$= 38.25 + 0.25(14.25)$

$= 38.625 \text{ms} + 4.375$

$= 42.625$ ms

Timeout Interval = Estimated RTT + 4 * Dev RTT

$= 302.5 \text{ms} + 4 * 42.625$

$= 473$ ms.

6a) RDT 2.0 protocol:

FSM sender: The sender waits for call from above application layer when data is recieved sender makes packets. adds checksum value and sends it. Then it waits for ACK or NAK. Checksum is used to detect bit errors which might happen in Rdt 2.0. so if every thing is OK ACK is sent and sender goes back to the state

where it waits for application layer call.
If NAK is recieved then there is some error so NAK is recieved by sender. It remains in the waiting state for NAK or ACK as long as it recieves atleast one of them.

RDT 2.0 Reciever side:

The reciever has only one state. It is always waiting for the call from below when it recieves the packets it checks the error using checksum. If packet is found to be corrupt it makes NAK packet and sends to the reciever.

If there is no error or corruption in the packet then. ACK is sent from reciever to sender. Reciever extracts data after removing headers. delivers data to the application.

ACK and NAK's are used for feedback signals. ARQ or Automatic Repeat Request is used which increases the Efficeency:

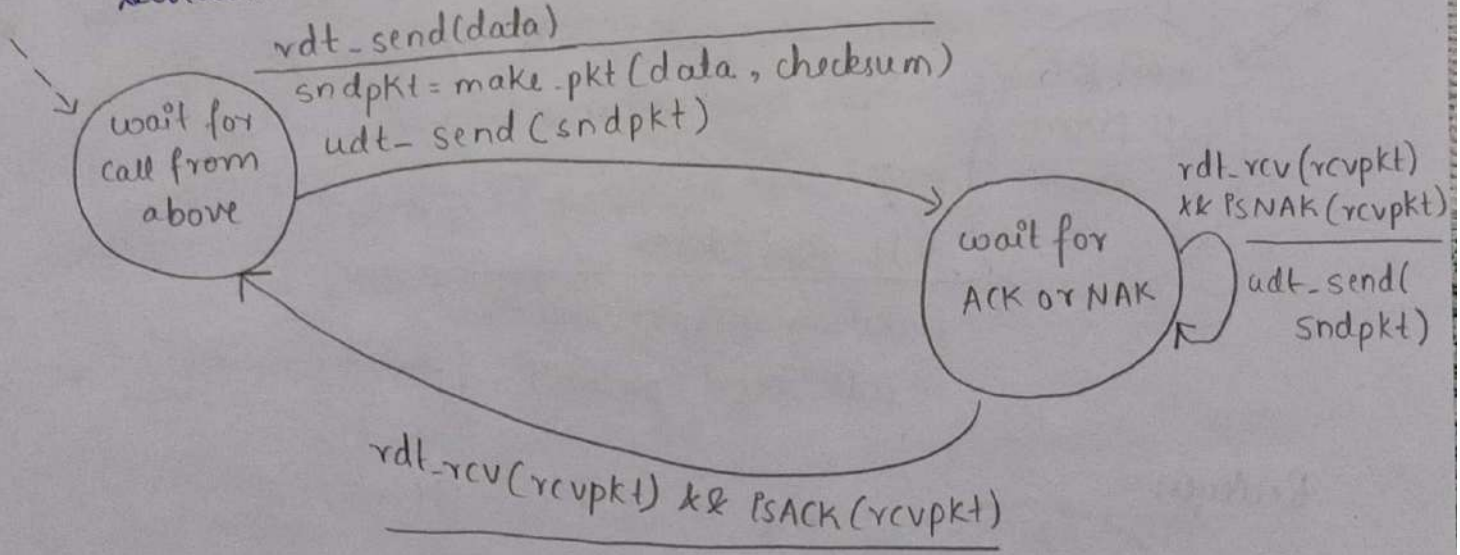
01fe18bcs211
Sneha K. Majjigudda

Rdt 2.0

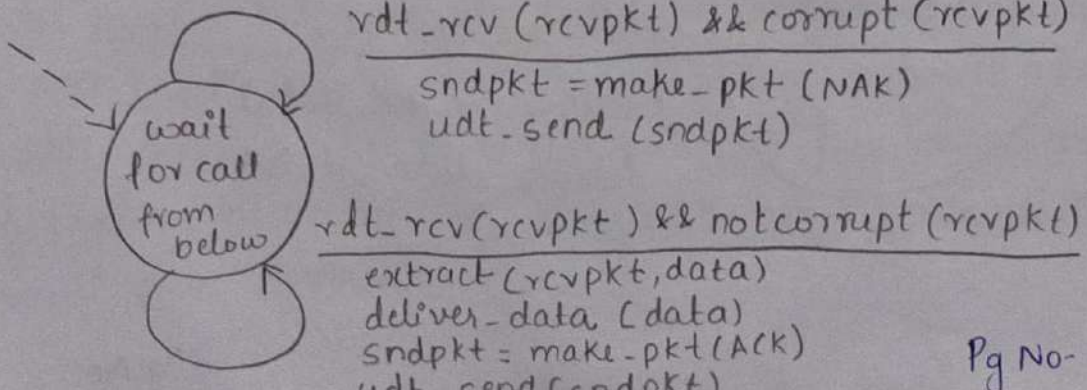
* There is channel with bit errors
* Use checksum to detect bit errors
* Error recovery: By giving feedback to sender (Control msgs)
  - acknowledgements (ACK): receiver explicitly tells sender that packet recieved is OK
  - negative acknowledgements (NAK): reciever explicity tells sender that pkt had errors
  - Sender retransmits packet on recipt of NAK.

FSM: Finite state machine is separate for receiver and sender
Reciever has 1 state while sender has two states

Sender:



Reciever.



9a) * NAT or network address translation is designed so that it allows a router to modify packets to allow for multiple devices to share a single public IP address

* Purpose:
  * The avaiable no. of global IP's are very less and customers are very more.

Merits:
* It is used address conservation as it allows a organization/ISP/ user to have a large set of addresses internally and one or a small set of addresses globally
* Security: NAT router hides details of home/private network from Internet. The private network is transpaint to the rest of Internet but the rest of Internet only sees the NAT router with global address.

The private networks have special IP's

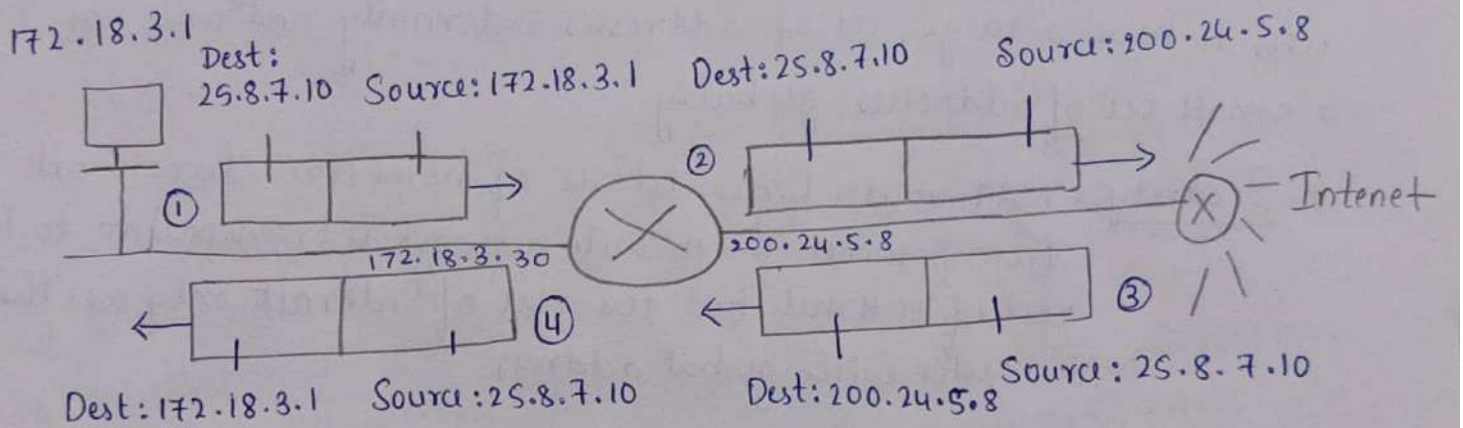
Range			Total
10.0.0.0	to	10.255.255.255	2^{24}
172.16.0.0	to	172.31.255.255	2^{20}
192.168.0.0	to	192.168.255.255	2^{16}.

* Private addresses are unique inside the organization and the organization can use this without the permission of Internet authorities
* Routers dont forward the packet with special ip addresses as its distination addrss

Working:

* The NAT enabled router has two sides LAN side and WAN side.

	NAT Translation table.	
	LAN SIDE	WAN SIDE
	172.18.3.1, 3412	200.24.5.8, 5007



* NAT router behaves to the outside world as single device with IP address.

Working:

Step 1: Consider one host with IP address 172.18.3.1 in the local area network wants to a message to a host 25.8.7.10. So the packet is created. Source address is 172.18.3.1, Soura port = 3412, Destination address is 25.8.7.10 and destination port is 80.

Step 2: Packet leaving the host 172.18.3.1 reaches the NAT router LAN side. The router recieves this packet, generates a new source port number for the datagram suppose 5007 and replaces the soura IP address with its WAN side address 200.24.5.8 and replace original source port number to 5007. NAT chooses a a port number that is not in the translation table. Adds entry to translation table.

Step3: The datagram then reaches the destination and the web server responds back to destination address which WAN side address of NAT router.

So the source address = 25.8.7.10

soure port = 80

Destination address = 200.24.5.8

Destination port = 5007.

And sends this packet to network layer which puts datagram on network

Step4:

When the datagram reaches the NAT router, the router indexes the NAT translation table using destination port number and destination IP address to obtain appropriate IP address and port number. in LAN.

Then it rewrites the datagrams destination address and destination port number and forwards it in LAN.

Destination address = 172.18.3.1

Destination port = 3412.

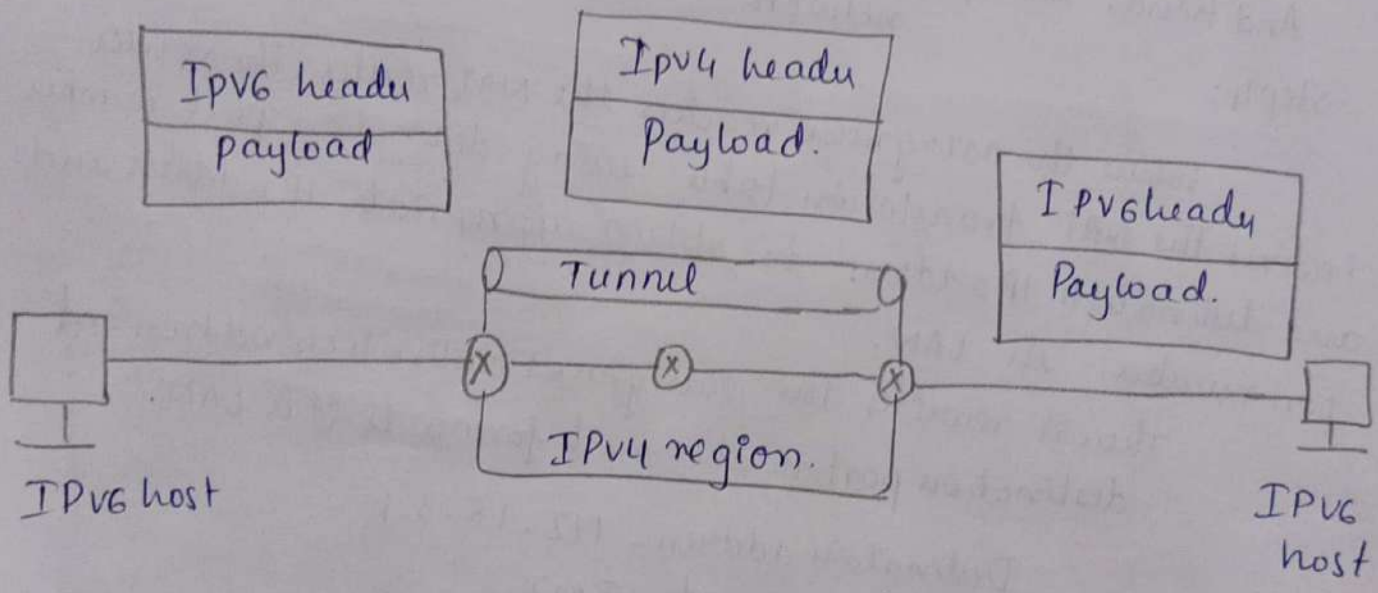
Source IP = 25.8.7.10

Soure port = 80

* The NAT router and local hosts gets IP through DHCP
* The NAT router gets its IP address from ISP's - DHCP server
* the router runs a DHCP server to provide addresses to computers within NAT-DHCP-router-controlled private network address space.
* All packets leaving LAN to larger Internet has source IP of WAN side of router. All packets entering LAN must have destination IP as WAN side of router

01fe18bcs211
4119

9b) When two hosts using IPV6 wants to communicate with each other and the packet must pass through the region of IPV4. To pass through this region, the packet must have IPV4 address. So the IPV6 packet is encapsulated in an IPV4 packet when it enters the region, and it leaves its capsule when it exists the region. It seems as if IPV6 packet goes through a tunnel at one end and emerge out at other end



The same happens when two IPV4 hosts wants to communicate over IPV6 region. IPV4 packet is encapsulated into IPV6 and passes the region to reach the IPV4 host.

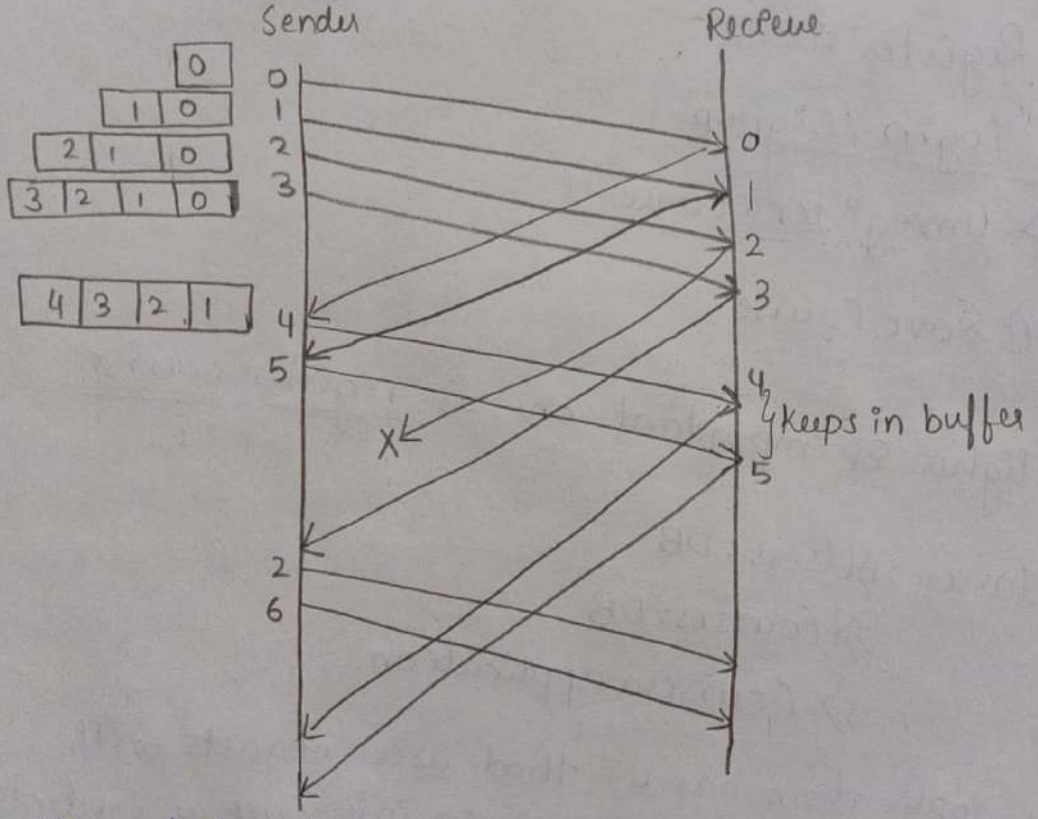
7b)

Go Back - N	Selective Repeat
* Retransmission is more	* Retransmission is less.
* Cummulative ACK's are taken into account	* Cummulative and indepen ACK's are used
* Packets are accepted in inorder	* Packets are accepted out-of-order.
* Bandwidth utilization is low	* Bandwidth utilization is high.
* Receiver window size = 1	* Receiver window size = N-1 N = window size of sender.
* Less efficient	* More efficient.
* Simple	* More Complex.
* Multiple frames are sent when errors or losses occur	* It retransmits only individual frame containing error.
* Cummulative based.	* NAK based
* Diagrams and example are shown next	* Diagrams and example shown next

01fe18bcs211
H19

Selective-Repeat ARQ

* It is sliding window protocol
* Here only erroneous or lost frames are retransmitted, while correct frames are recieved and buffered
* the reciever while keeping track of a sequence numbers, buffers the frames in memory and sends NACK for only frame which is missing or damaged.
* the sender will send / retransmit packet for which NACK is recieved



* Consider window size = 4.
* Consider 11 frames

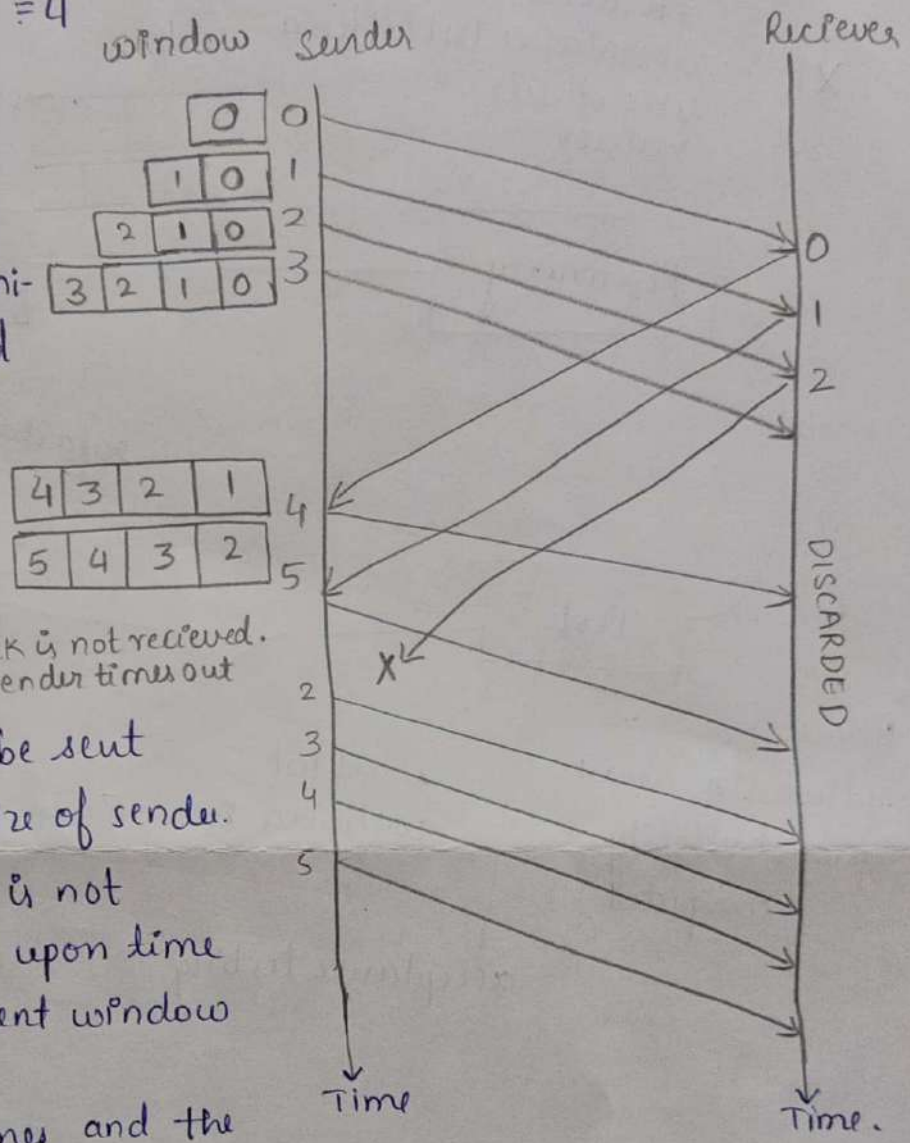
10	9	8	7	6	5	4	3	2	1

* Foramus and sliding window in figure

Go-Back-N ARQ

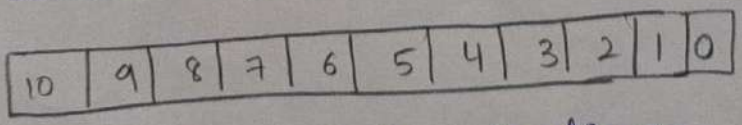
N=4 Window size = 4

* It is used in noisy channels
* Go-Back-N ARQ uses the concept of protocol pipelining i.e the sender can send multiple frames before receiving the ACK for first frame.



It goes to start of current window ← ACK is not received. Sender times out

* No. of frames that can be sent depends on the window size of sender.
* If the acknowledgment is not received within an agreed upon time period, all frames in current window are transmitted.
* There are finite no. of frames and the frames are numbered in sequential manner.
* The size of the sending window determines the sequence no. of the outbound frames. (No. of bits used for window size).
* Example: Consider 11 frames



Frames and sliding window in figure.