Computer Networks

Network Performance Parameters :

1) Throughput / Efficiency / Bandwidth :

Bandwidth → Bitrate / Datarate / channel capacity / Tx rate / Tx speed

* Throughput = Total Data Size / Total Time
  - Throughput → max datarate / effective datarate
* efficiency = $\eta = \frac{T}{B}$ → max link utilization
* $T \le B$

2) RTT (Round Trip Time) → 2-way latency

$RTT \cong 2 \times T_p$

* Throughput = $\frac{L}{T_x + 2T_p}$
* Total time = $T_x + 2T_p$
* $\eta = \frac{T_x}{T_x + 2T_p} = \frac{1}{1+2a}$

3) Bandwidth - Delay Product : [The no of bits that can be placed into the channel]

- For FD link → $B \times R$
- For HD link → $B \times T_p$

4) Delay / Time →
- Transmission Time $(T_x) = \frac{L}{B}$ (Msg size / Bandwidth)
- Propagation Time (1-way latency) $(T_p) = \frac{d}{v}$ (distance b/w S & R / speed of propagation)
  - $v = 3 \times 10^8$ m/s
- $T_{ack}, T_{proc}, T_{que}, T_{qproc}$.. → ng

$T_p = \frac{d}{v}$ (wired) $T_p = \frac{2d}{v}$ (wireless)

* for fiber link, $v = \frac{2}{3} \times 3 \times 10^8$ m/s $= 2 \times 10^8$ m/s

5) Jitter ~~~~ If the link is fiber link or coaxial cable then the speed of propagation is 2/3 x speed of light in vacuum.

ISO/OSI reference Model (7 layers)

- N/w friendly layer
  - PL (Physical layer)
  - NL
  - DLL
- User friendly layer
  - AL
  - Pres - L
  - Sumion L
- Transfer layer
  - TL
  - NL
  - DLL

• Address
- Port Addr. - TL
- MAC Addr. - DLL
- IP Addr. - NL

* $RTT = T_{d(data)} + 2 \cdot P_d + T_{d(ack)} + P_{rd} + \delta_d$

original formula

1. PDU : Frames
2. Flow control } Logical link layer
3. Error control }
4. Access control - MAC sublayer
5. Physical Addressing.
6. Node-to-node delivery.

Data Link layer → Logical link layer (LLC)
→ Mac layer (Media access layer)

- **Flow control:** ← stop & wait SW protocol

→ The process of stopping the fast sender from the slower receiver is referred as "Flow control".

- **Stop & wait flow control:**

Idea: After the transmission of one data frame, the sender "stops" the transmission and "waits" for Ack by setting the timer.

→ If the sender gets ACK within the timer, then, the sender transmit next frame. otherwise retransmit the same frame.

- Sender can transfer only one packet at a time
- ACK seq. no. indicates that (i) total frames received.
  (ii) Next expected frame seq. no.
- Operates in HD mode.

Drawback: No proper channel utilization.

- channel utilization / Efficiency: $\eta = \frac{T_x}{T_x + 2T_p} = \frac{1}{1+2a}$ $\left(\because a = \frac{T_p}{T_x}\right)$

- Throughput = $\frac{TDS}{TT} = \frac{L}{T_x + 2T_p}$

- If $\eta = 50\%$ then $L = BR$

$\propto \left(\eta = \frac{L}{L + BR}\right)$ put $\eta = \frac{1}{2}$ (BR: round trip time)

* ACK → Independent Ack (For every transmission, there should be an ACK)
→ Cummulative Ack (can be take place at any time)

- **Sliding Window Protocol:**

* n bit frame seq no. is used then the frame range in the window from 0 to $2^n - 1$
* Sliding window uses wrap around seq. no.
* Sender can transfer multiple packets.
* The window size shrinked - when sender transmits or receiver receives.
* The window size expanded - when sender got the ack or receiver sends an Ack.
* Pkt loss possible, to receiver this we use ARQ (Automatic Repeat Request) → Selective Repeat, GO Back N ARQ
* Sliding window uses Cummulative ACK and piggybacking (Data + ACK)

* SW Throughput = $\frac{TDS}{TT} = \frac{wL}{T_x + 2T_p}$ (w = window size, L = packet size)

* SW Efficiency = $\eta = \frac{wT_x}{T_x + 2T_p} = \frac{w}{1+2a}$

* Efficiency of SW:

Case-1: if $w \geq (1+2a)$ then $\eta = 100\%$ $\left[\because a = \frac{T_p}{T_x}\right]$

Case-2: if $w < (1+2a)$ then $\eta = \frac{w}{1+2a}$

Calculation of no of bits in Frame Seq. No. field:-

Step 1: Calculate RTT (FD link) (By default)
Calculate T_p (HD link)

Step 2: Calculate B×Delay product of windows (W_{bits}) (no. of bits)

$W_{bits} = B \times R$ (FD link)

$W_{bits} = B \times T_p$ (HD link)

Step 3: Calculate $W_{packets}$ (OWS) optimal window size

$$W_{packets} = \frac{BR}{L} = \frac{W_{bits}}{L} \quad \text{(FD link)}$$

$$W_{packets} = \frac{B \times T_p}{L}$$

Step 4: $W_{packets}$ (or) $OWS = 2^n$

where, n = no of bits in frame seq no. field.

Optimal window size = $1 + 2a$

Minimum No. req. = $1 + 2a$

Min no of bits req. = $\lceil \log_2(1+2a) \rceil$ in the seq. no field.

ARQ technique (To overcome the drawback of SWP)

* Can be used in 3 cases: Pkt lost, Ack lost, Ack delay

ARQ
- SWP
- Sliding Window
  - GBN
  - SR

GBN:-

* Recv. can accept only linear flow of data (Never accept out of order data)
* Recv. size is set to 1 ($W_R = 1$)
* NAK frame seq no. indicates from where the sender has to perform retransmission
* Sender window size + Recv. window size = Available frame seq no.

$$W_S + W_R \leq (ASN)$$

Case 1: $N-1+1 = N+1$

Case 2: $N-1+1 = N$

Case 3: $2^n - 1 + 1 = 2^n$

* $$\text{Efficiency} = \eta = \frac{W}{1+2a} = \frac{2^n - 1}{1+2a}$$

Disadv: No proper channel utilization due to retransmission.

Selective Repeat ARQ

- can ~~receive~~ receive out of order data
- NAK frame seq indicates which frame the sender has to retransmit.
- Sender window size + Receive window size = Available frame seq no.

⇝ $N/2 + N/2 = N$ * N = 2, ASN = 0 1 2 3.

⇝ $2^{n-1} + 2^{n-1} = 2^n$

⇝ $N+1/2 + N+1/2 = N+1$ if N is Maximum seq. no.

* **Efficiency** $= \eta = \frac{\omega}{1+2a} = \frac{2^{n-1}}{1+2a}$

- **Sliding Window**

S

. . 0 1 2 3 [0 1 2 3] 0 1 2 3 . . .

R

. . . 0 1 2 3 [0 1 2 3] 0 1 2 3 . . .

- **GBN ARQ**

S

. . 0 1 2 3 [0 1 2] 3 0 1 2 3 . . .

R

. . . 0 1 2 3 [0] 1 2 3 0 1 2 3 . . . .

- **SR ARQ**

S

. . [0 1] 2 3 0 1 2 3 0 1 2 3 . . .

R

. . . [0 1] 2 3 0 1 2 3 0 1 2 3 . . .

Summary

	Stop & Wait	GBN	SR
Efficiency	$\eta = \frac{1}{1+2a}$ or $\eta = \frac{T_d}{RTT}$	$\eta = \frac{N}{1+2a}$ or $\eta = \frac{N * T_d}{RTT}$	$\eta = \frac{W_S}{1+2a}$ or $\eta = \frac{W_S * T_d}{RTT}$
Throughput	$\frac{\text{Length of data Pkt}}{RTT}$ or $\eta * B$	$\frac{N * \text{Length of data Pkt}}{RTT}$ or $\eta * B$	$\frac{W_S * \text{Length of data Pkt}}{RTT}$ or $\eta * B$
Buffer	$\frac{1+1}{2}$	$\frac{N+1}{N+1}$	$\frac{N+N}{2N}$
Seq. No.		W_S W_R	W_S W_R
Seq No. = K bit		$2^K - 1$ 1	2^{K-1} 2^{K-1}

Error Control — In DLL error checks in bit error only

Error
- Single bit error
- Burst error (Multi bits)
  * Burst length: defines no of bits that are involved in starting bit position of the error to ending bit position of the error.

* using redundancy in DLL we control the error.

Redundancy Tech

VRC (Vertical Redundancy Check)
- parity checker
- Single bit error detection
- Parity
  - even (default)
  - odd

eg. 1001000 (Data)
↓ even parity
10010000 (Appended)

LRC (Longitudinal Redundancy Check)
- 2-D Check
- Divide the data into segments (Equal size)
- can calculate the parity column wise or row wise
- After generating the parity bit at recev. if the result is 0 then accept if non zero then reject.

CRC (Cyclic Redundancy Check) → Remainder at S
- In CRC, S can have two messages.
  (a) Original msg - M(x) - Dividend
  (b) Generator msg - G(x) - Divisor
- Append M(x) with (n-1) number of 0's (M(x) + (n-1) 0's)
- perform Modulo-2 division at the sender, the remainder (at the sender) is called as "CRC" whose length must be (n-1) bits
- While sending the msg, S can replace (n-1) 0's with CRC, Thus, the msg seems to (m(x) + CRC) which should be transmitted to the receiver.
- At recev. divide m(x) + CRC by G(x)
- if remainder 0 Accept otherwise reject

CheckSum

* CheckSum:-

Step 1: Segment the whole data.
Step 2: Perform addition op. on all segments of data.
Step 3: Until the result is carry free, keep on add the carry bit (if any generated) to LSB of result.
Step 4: Once the result is carry free, need to perform 1's complement.
Step 5: After 1's complement, the find result is called as "checksum"

Checksum
1000 1110 1100 0100

Recev
1000
1110
1100
0100
0111
11

Sender
10
1000
1110
1100
0100
0110
+1
0111
+1
1000
1's complement
0111

Some important point of CRC :-

1) If the generator has more than one term and coefficient of x^0 is 1, all single bit error can be detected.
2) If the generator cannot divide $x^t + 2$ (t between 0 and n-1) then all isolated double error can be detected.
3) A generator that contains a factor of $(x+1)$ and detect all the odd numbered errors.

A good polynomial generator needs to have the following characteristics :-

1) It should have atleast two terms.
2) The coefficient of the term x^0 should be 1.
3) It should not divide $x^t + 1$ for t between 2 and $(n-1)$
4) It should have the factor $(x+1)$.

Hamming Code

- used for both Error detection & Correction.
- It can do single bit Error detection & correction both, but, Multibit Error detection Only.
- 2 phases – Hamming code generation - sender
  Hamming code verification - Receiver

* Hamming distance :-
defines no of bits that are differ in b/w any 2 code words

codeword { $CW_1 = 1101$, $CW_2 = 1000$ } HD = 2

- A code with minimum hamming distance 'd' between its codewords can DETECT atmost '(d-1)' errors and can CORRECT atmost $\lfloor (d-1)/2 \rfloor$ errors.

- Hamming Code generation :-

(i) generated by sender
(ii) HC is combination of parity and data bits
(iii) Parity bits can be placed only in powers of 2 positions
(iv) If 'm' is the no of msg/data bits, then to find out the no of parity bits to be added, can be calculated using the following condition :

$$2^p \geq m + p + 1$$

m → msg (or data bits)
p → parity (or redundant)

Ex : msg/data = "1101" m = 4

p = 1, $2^1 \geq 4 + 1 + 1$, False
p = 2, $2^2 \geq 4 + 2 + 1$, False
p = 3, $2^3 \geq 4 + 3 + 1$, True

Since m = 4 and p = 3 then HC in 7

Struct" of HC :-

			2^2	2^1	2^0	
7	6	5	4	3	2	1
D_7	D_6	D_5	P_4	D_3	P_2	P_1
1	1	0	P_4	1	P_2	P_1
1	1	0	0	1	1	0

$P_1 \rightarrow (1, 3, 5, 7) \rightarrow$ LSB = 1
$(P_1, 1, 0, 1) \rightarrow P_1 = 1$

$P_2 \rightarrow (2, 3, 6, 7) \rightarrow$ middle bits 1
$(P_2, 1, 1, 1)$ $P_2 = 1$

$P_4 \rightarrow (4, 5, 6, 7) \rightarrow$ MSB = 1
$(P_4, 0, 1, 1)$ $P_4 = 0$

∴ Hamming code = 1100110 msg + parity

000	0
001	1
010	2
011	3
100	4
101	5
110	6
111	7

- Hamming code verification :

Error detection → Error Correction
with modification of HC
w/o modification of HC

PDU: PL - bits
DLL - frames
* packet format [H|D]

Framing

- Process of converting numeric char into frame. [H | D | T]

Framing

Static framing

- No predefined boundary to a frame (No head, No tail)
- The frame boundary created during framing.

ex: Char count

→ 5 4934 3 88 2 4 3 1
header

Not having that much data so just discard the bits

- No tail info.

Dynamic framing

- frame has predefined boundaries
- Head + Data + Tail

Char stuffing

Datalink Escape

[DLE STX | Data | DLE ETX]

Start of text — End of text

- Process of stuffing a ASCII char seq into the data field.
- HI - HIDLE
  HDLEI
  DLEHI
  (Any extra one sender can insert)

Bit stuffing

flag seq | Data | flag seq

01111110 | Data | 01111110

- After each 5 1's stuff one 0 to distinguish the delimeter from the data.

Access Control

DLL ┬ LLC
    └ MAC - MAC protocol

- Intention is to handled collisions

① Random Access
② controlled Access
③ channelized Access

(1) Random Access: Any station have data then that station is allowed to transfer the data at any time.
- peer to peer Rel^n.

* ALOHA: Based on 2 things < RA/MA, Acknowledgement.

Disadv: Capture effect: The prob of joining of a station's frame after successful tx is increased.

* Back off Algo: (Exponential)

Whenever there is a collision before making a retransmission of the frame, the station has to wait K × 51.2 μs

contension window. (n) contension slots {0 - 2^n - 1}
meaning → competition.
n → no of collisions

K_max → Backoff limit
K → Backoff variable

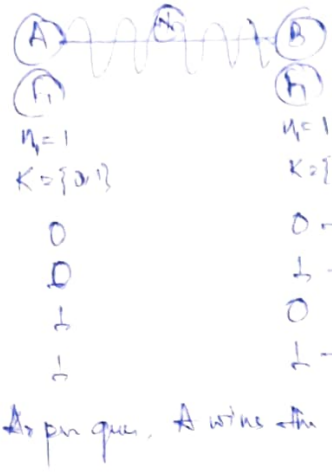
Backoff waiting time = K × 51.2 μs
where, K is a random variable which can be chosen in the range of {0 to 2^n - 1}
where n is no of collisions.

* This exponential growth in CWS is allowed till n=10 K={0,1,2,...1023}
n=11,12,13,...15 K={0,1,2,...1023}
once, n=10 then CWS is freezed

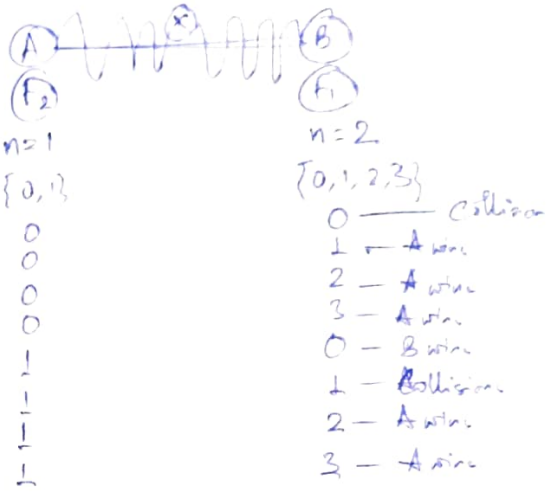
* freezing behaviour in CWS, is continued till n=15

once n=16 → Abort.

(GATE 2004) A and B are the only two stations on an Ethernet. Each has a steady queue of frames to send. Both A and B attempt to transmit a frame, collide, and A wins the first backoff race. At the end of this successful transmission by A, both A and B attempt to transmit and collide. The probability that A wins the second backoff is (0.5/0.625/0.75/1)



As per que, A wins the 1st Backoff



$P(A) = 5/8$ $P(B) = 1/8$

Classification of ALOHA

Slotted ALOHA / discrete ALOHA

- Vulnerable Time :
  $V_T = T_x$ (frame tx time)
- Throughput: $G_2 \times e^{-2G_2}$
  - max" throughput = 0.368 when $G_2 = 1$
  - Efficiency = 36.8 ≅ 37%
  - 37 frames successfully transmitted out of 100 no of frames.

Pure ALOHA / Continuous / Classical ALOHA

- Works based on Continuous timing
- Vulnerable timing:
  $V_T = 2 * T_x$ (frame tx time)
- Throughput: $G_2 \times e^{-2G_2}$

G_2 → avg. no of frames that are transmitted by a station during the frame tx time

- Max" throughput = 0.184 for $G_2 = 1/2$
- Efficiency = 18.4%

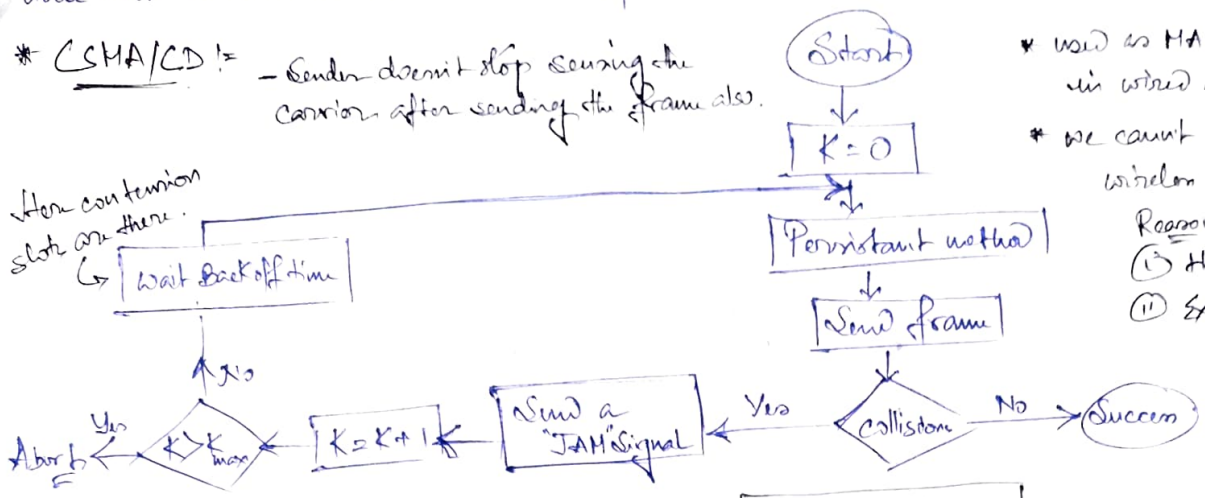
Some important points regarding Access Control.

1. Minimum size of frame to detect the collision in Ethernet (CSMA/CD)
   $T_d \geq 2 * P_d + T_d$(jam signal)
2. Backoff Algorithm
   waiting time = K * slot duration
   = K * RTT
   = K * 2 * P_d
   $K * 51.2\,\mu s$ / standard RTT
   {0 to $2^n - 1$} where n is the collision number
3. Efficiency in Ethernet (CSMA/CD)
   $\eta = \frac{1}{1 + 6.44a}$ or $\eta = \frac{\text{useful time}}{\text{total time}} = \frac{T_d}{\text{collision time} + T_d + P_d}$
   contention slot * slot duration
4. $p(1-p)^{N-1}$ → Probability of success for single station
   $N.p(1-p)^{N-1}$ → Probability of success for any station
   Assuming all station [Throughput of channel]
5. Ethernet [Packet size]

Min size	Max size
46	1500 [Data]
64	1518 [Frame]

* CSMA :- Before making the data tx the sender check for the idleness of the channel if it is idle then it will send otherwise wait for some time until the channel becomes idle.

* CSMA/CD :- - Sender doesn't stop sensing the carrier after sending the frame also.



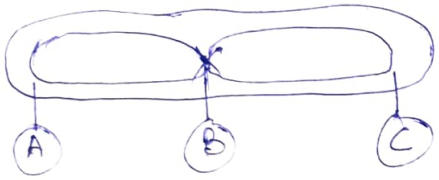
* used as MAC protocol in wired ethernet
* we cannot use it on wireless ethernet

Reason -
(i) Hidden terminal
(ii) Exposed terminal

Wired Ethernet in CSMA/CD → $T_{fr} = 2T_p$

c: Min frame size Calculation: $L \geq 2 \times \frac{d}{v} \times B$

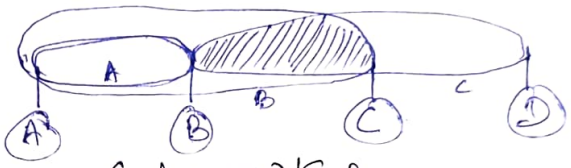
* Hidden terminal Problem :- (HTP)



When A and C sense the carrier it is idle so when they will send the data there is a chance of collision at B. It is not detected by CSMA/CD A is hidden to C and C is hidden to A.

So!! : 3-way Handshaking Signals < RTS (Req to send), CTS (Clear to send)

* Exposed Terminal Problem (ETP):



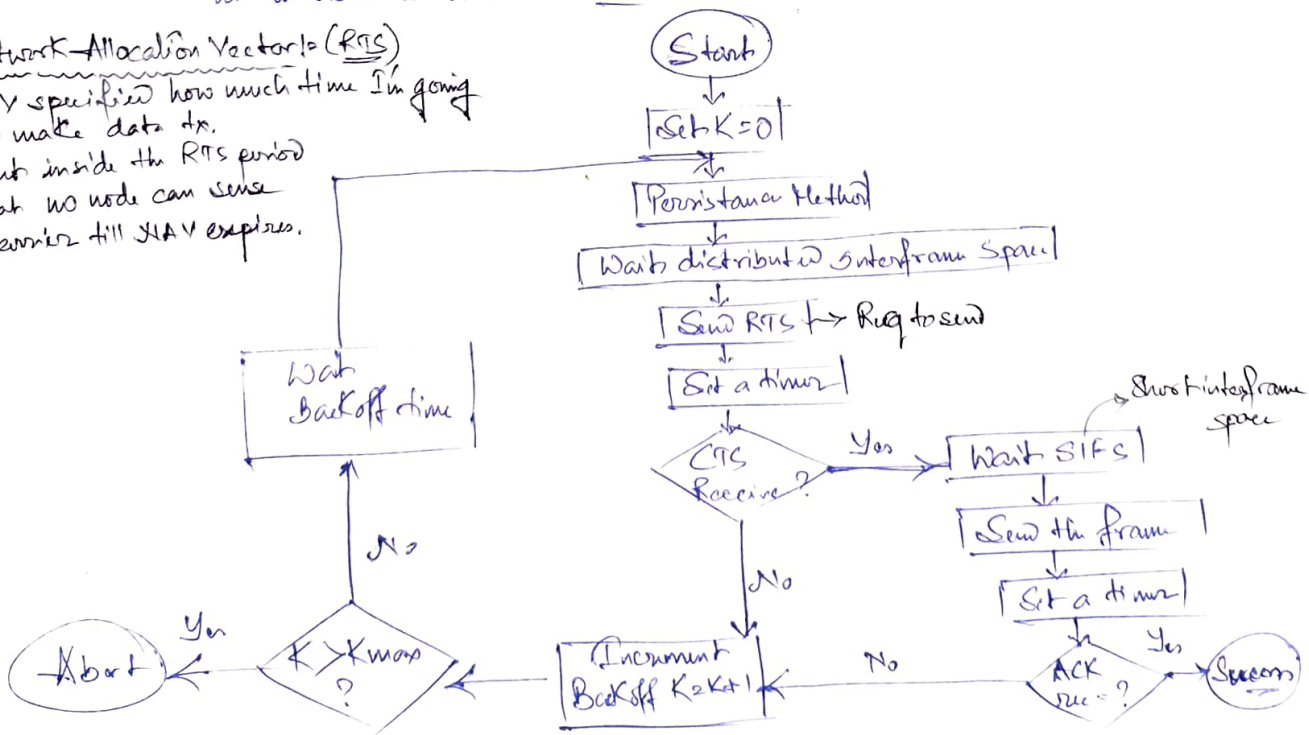
C is exposed to B.

When B send data to A if C wants to send data to D. it will sense and there are some interesting part there so C D is using the freq. so it will not transmit the data and wait unnecessarily

So!! : "MACA", "ISMA".

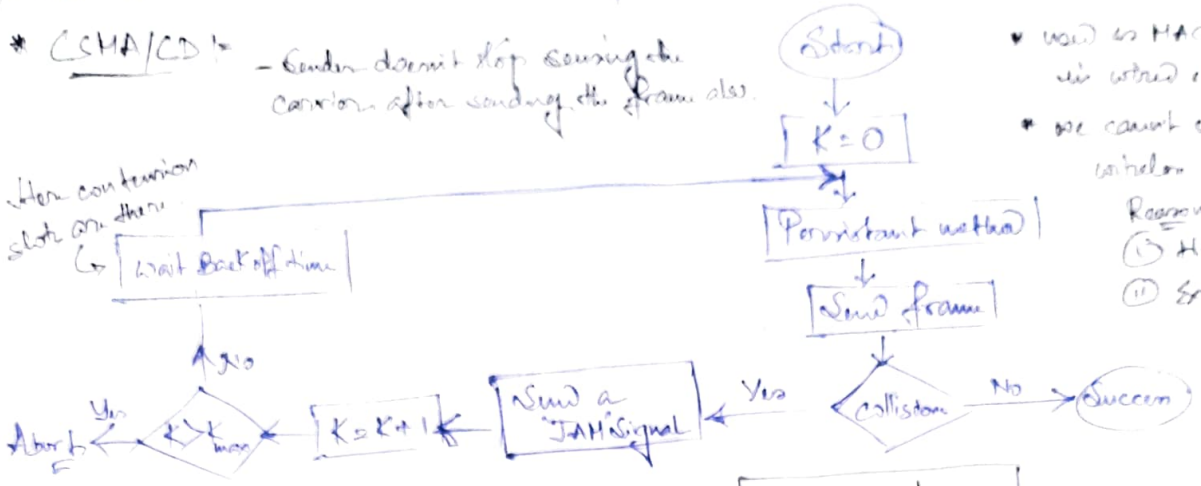
* CSMA/CA :- - using handshaking method by using RTS, CTS signal.
- it is used to remove HTP but uses ISMA to remove ETP

* Network Allocation Vector :- (RTS)
→ NAV specified how much time I'm going to make data tx.
→ Sent inside the RTS period so that no node can sense the carrier till NAV expires.



* CSMA : Before making the data tx the sender check for the silence of the channel if it is idle then it will send otherwise wait for some time until the channel becomes idle.

* CSMA/CD :- Sender doesn't stop sensing the carrier after sending the frame also.

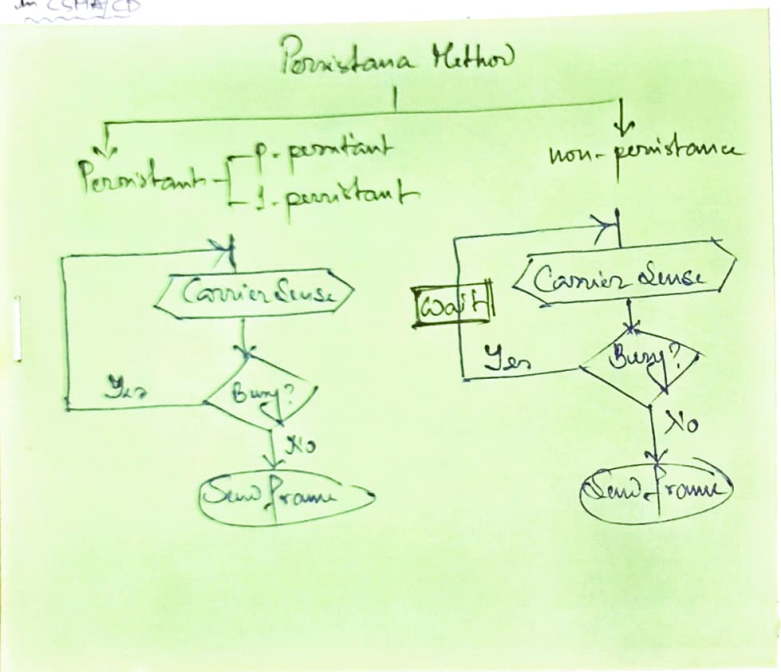


* used as MAC protocol in wired ethernet
* we cannot use it on wireless ethernet

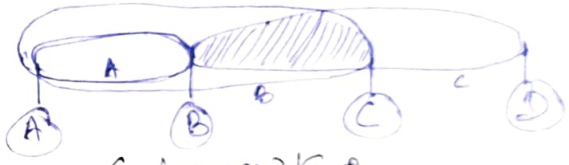
Reasons -
(i) Hidden terminal
(ii) Exposed terminal

wired ethernet in CSMA/CD $\rightarrow$ $T_p = 2T_p$

c : Min frame size calculation: $L \geq 2 \times \frac{d}{v} \times B$



* Exposed Terminal Problem (ETP):



C is exposed to B.

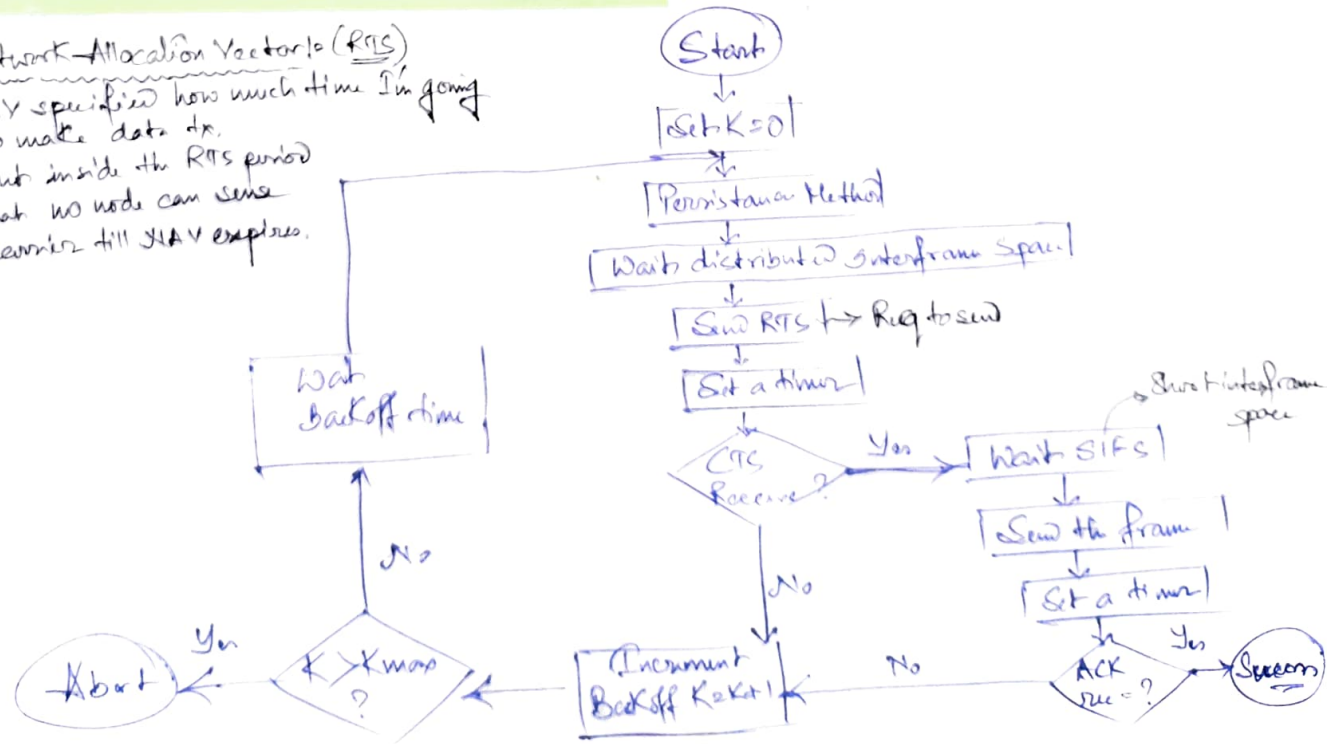
When B send data to A if C wants to send data to D. it will sense and then as same interested apart then as C D is using the freq. so it will not transmit the data and wait unnecessarily.

Sol": "HACA", "ISHA".

using RTS, CTS signal

but uses ISHA to remove CTP

* Network Allocation Vector (NAV) (RTS)
→ NAV specified how much time I'm going to make data tx.
→ Sent inside the RTS period so that no node can sense the carrier till NAV expires.



IEEE 802.4 → ~~Ethernet~~ Token Bus
IEEE 802.5 → Token Ring
IEEE 802.6 → DQDB [Distributed Queue Dual Bus]
IEEE 802.11 → WIFI
IEEE 802.15 → Bluetooth

LAN TECHNOLOGIES (ETHERNET)

↳ IEEE 802.3

Ethernet evolution

- Standard Ethernet 10 Mbps
- Fast Ethernet 100 Mbps
- Gigabit Ethernet 1 Gbps
- Ten-Gigabit Ethernet 10 Gbps

1. Datarate - 10 Mbps
2. MAC protocol - CSMA/CD
3. Expo Backoff Algo used
4. Encoding - Manchester
5. min frame size - 512 bits or 64B
6. Contention period - $2T_p = 51.2\mu s$

(i) 10Base5 (ii) 10Base2
(iii) 10BaseT (iv) 10BaseF.

Ethernet Packet Size := (Bytes)

	Min size	Max size
Data -	46	1500
frame -	64	1518

* IEEE 802.3 frame format:

Bytes →	7	1	6	6	2	46-1500	4
	PREAMBLE	SFD	DA	SA	LENGTH OF DATA	DATA	CRC or FCS

Alternate 1s and 0s

Physical layer header

10101011

Not added in frame format bcz it got added by Physical layer

DA: ↳ LSB = 0 - Unicast
1 - multicast
All 1 - then Broadcast

* Efficiency of Ethernet := $\eta = \frac{1}{1+6.44a}$ where $a = \frac{T_p}{T_x}$

NETWORK LAYER

- * PDU: Packets
- * Routing
- * Congestion Control
- * Fragmentation & Reassembly
- * Logical Addressing
- * End to End Delivery.

CO Protocol - TCP - Packet - CO PDU
CL Protocol - UDP - Datagram - CL PDU

DLL: Frames, EC, FC, AC, Physical Add, Node to Node

The process of dividing a large data unit into unequal sized small data units is said to be "fragmentation".
Each unequal sized small data unit is said to be a "fragment".

150 B → 30B | 30B | 30B | 10B

Maximum transfer unit ← "MTU" → use for "fragmentation".

"Identification" → IPv4 datagram header format
↳ Reassembly"

Conclusion: NL can do fragmentation → MTU of n/w or Router
Reassembly → Identification of IPv4 datagram header format.

VHTTIFF TPH SDO

20-65536B
20-60B | Header | Data

VER (4b)	HLEN (4b)	Type of Service (8b)	Total length (16b)
Identification (16b)		Flags (3b) 0 DF MF	Fragmentation offset (13b)
Time to Live TTL (8b)	Protocol (8b)	Header Checksum (16b)	
Source IP Address (32b)			
Destination IP Address (32b)			
Option (0-40B)			

32 bit

- IPv4 - 0100
- IPv6 - 0110

Defines HLEN in terms of 4B (5 - 15) min max

TTL value in dec by each router by 1 visited

Total length: 20 to $(2^{16}-1)$B; TL = HL + DL

Fragmentation offset: Defines offset in terms of 8B; first fragment offset value = 0

Header Checksum: Not 100% reliable (Best effort delivery) [detect only no corrects]
↳ Check for header only [TCP, UDP for total data + header]

* Flag:
1) 2^{nd} bit = MF bit (more fragment)
MF = 1 → that particular fragment is not last one
MF = 0 → last fragment
2) 3^{rd} bit = DF = Don't frag.
DF = 0 → can be fragmented.
DF = 1 → should not frag the datagram.

* Reassembly = Id no.
* Rearrangement = offset (mostly) + flag

Ipv4 "Datagram" Header format.

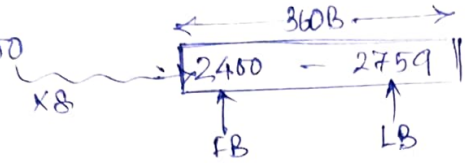
Some Problems Regarding Fragmentation & Reassembly:

Model-I: In an IPv4 datagram, the M bits is 0, the value of HLEN is 10, the value of total length is 400 and the fragment offset value is 300. The position of the datagram, the sequence numbers of the first and the last bytes of the payload, respectively are ⟶

M bit ⟶ 0
Last fragment

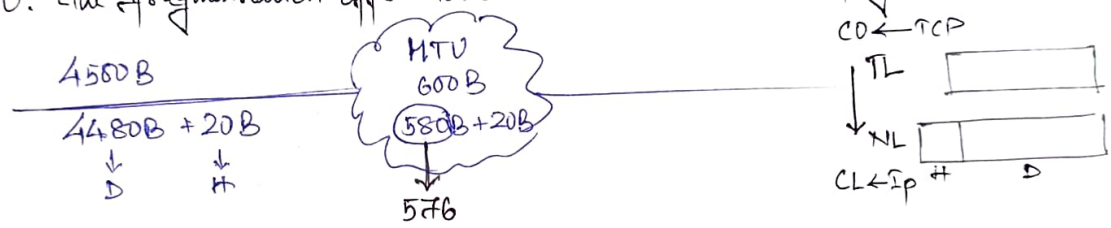
HLEN = 10 × 4 = 40B

DL = TL − HL = (400 − 40) = 360B



Last fragment: 1st Byte = 2400
Last Byte = 2759

Model-II: Consider an IP packet with a length of 4500 Bytes that includes a 20B IPv4 Header and 40B TCP header. The packet is forwarded to an IPv4 router that supports a MTU of 600 bytes. Assume that the length of the IP header in all the outgoing fragments of this packet is 20B. Assume that the fragmentation offset value stored in the first fragment is 0. The fragmentation offset value stored in the third fragment is ⟶



①	②	③
576	576	576
0 – 575	576 – 1151	1152 –
offset value = 0/8 = 0	576/8 = 72	1152/8 = 144
MF = 1	MF = 1	MF = 1

∴ offset value of third fragment ⟶ 144 (Ans)

Model-III: An IPv4 datagram is arrived at a router with the 1st four hexadecimal digits as shown below: 4 5 00 0100 0000 0000 1002 . . . .

ⓐ How many hops are being visited by this packet before being "dropped"?

4 5 00 0100 0000 0000 10 02 . . .

0100 0101 — VER HLEN | SERVICE | Total Length | Identification | Flags(3b) | Fragmentation offset | TTL

TTL = 1 × 16¹ + 0 × 16⁰ = 16

ⓑ How many bytes of data is being carried by this datagram?

TL = $(0100)_{16}$ = 1 × 16² = 256B

HL = $(5)_{10}$ × 4 = 20B

DL = (256 − 20) = 236B

* **Routing**: Process of finding the shortest path among multiple paths b/w S & R.

→ Routing is done by Router based on Routing table.

→ Routing algo based on **Optimality Principle**.

* States that if router 'j' involved in the shortest path of router 'i' and 'K' then, the shortest path router 'j' and router 'K' is also existed in the shortest path of router 'i' and router 'K'.

Connecting Devices

- N/w devices (ex: Repeater, Hub)
- Inter-n/w devices (ex: switch, router)
  - * Router - Layer-3 switch (NL) Bridge, Gateway
  - * Bridge - Layer-2 switch (DLL)
  - * Gateway - from TL to AL.

Classification of Routing Algo

Static Routing Algo:

- Never responds to the topological changes [Non-adaptive Algo]
- Admin perform routing path calculation

Dynamic Routing Algo:

- Always responds to topological changes (adaptive)
- Routing path calculation done by router multiple times.

Shortest path Routing

- Apply Dijkstra Algo to find MST

Flood Routing

- The process of forwarding the pkt to all connected node except the node from where the pkt is arrived.
- Flooding generates vast no of duplicates pkts. Different measures have taken to damp these pkts.
  1. Hopcount.
  2. Assigning seq no. to diff. pkt the router maintain a record of which seq. no. pkt get previously forwarded.
  3. Selective flooding.

Dynamic Routing Algo

Distance vector Routing (DVR)

- Bellman ford routing Algo.
- Works based on 3 principle:
  1. Know about whole n/w
  2. info sharing to only neighbour nodes.
  3. Irrespective of topological changes, updation takes place periodically

Link State Routing (LSR)

- Works on 3 principle:
  1. Know about neighbours.
  2. info sharing to all the nodes.
  3. Updation when there is a topological change occur. (Triggered update)

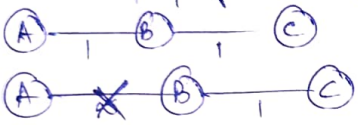
DVR

- Convergence process slower. [Construct Stabilized RT]
- Cycle formation - Sol": Split Horizon.
- RIP uses DVR
  routing information protocol
- Routing table also known as "Distance Vector"
- Format of RT or Distance Vector:

Dest"	Cost	Next Hop

- Count to ∞ prob:

Any decrease in cost (good news) propagates quickly, but any increase in cost (bad news) propagates slowly



Sol":

- Split Horizon: Don't send routes learnt from a neighbour back to it.

* within an autonomous system (like a subnet), both DVR and LSR can be used. [Intradomain Routing]
* Interdomain routing - BGP.

LSR

- Convergence process is faster.
- No cycle.
- OSPF uses RIP.
  Open shortest path first
- Routing table also known as "Link state Packet" or "Link state Advertisement".
- Format of Link state PKT or RT:

- Initial →

Advertiser	Dest"	Cost	Next Hop

(The node itself)

- final →

Dest"	Cost	Next Hop

* To construct the final LSP we need to
  (i) construct initial LSP
  (ii) construct Link state Database
  (iii) construct shortest path tree (for each node) [Dijkstra's Algo].
  (iv) final LSP.

Some definitions Regarding Routing ∈.

* Stabilized RT: If any router gets all the neighbours RT, then, which ever RT is maintained by the particular router. is called "Stabilized RT" or "final RT"
* Convergence: The process of constructing the final RT by a router is called "Convergence".
* Convergence Time: Time taken by router to construct its final (or stabilized RT is called as "Convergence Time".
* Stabilized n/w: In any n/w, if all the routers are ready with their final (or stabilized RT such n/w is "Stabilized n/w".

Congestion Control (NL)

* Congestion is the situation in n/w where the incoming pkt rate is higher than the processing rate of the n/w, Major Drawback due to congestion - "Packet loss".

At i/p Buffer: PAR ↑ high, PDR ↓ low

At o/p Buffer: PDR ↑↓ low, PPR ↑ high.

Congestion in n/w: Throughput ↓, Delay ↑

PAR - Pkt Arrival rate
PDR - Pkt Departure rate
PPR - Pkt processing rate

Congestion Control

Congestion Prevention / Open loop Congestion Control

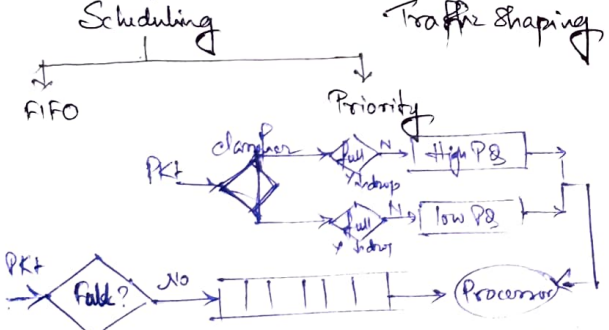
* Window policy - Selective Repeat.
* Retransmission policy - Only the errored pkt must be retransmit.
* ACK policy - Cummalative ACK.
* Admission policy - Case 2 is better.

Case 1: Is it better to give the ack after the entire receiver buffer size is full?

Case 2: Is it better to give the ack after half the receiver buffer size is full?

* Quality of Service: After performing congestion control policies to know the performance of congested n/w we use QoS.

QoS
- Scheduling
  - FIFO
  - Priority
- Traffic Shaping



Traffic Shaping: It is a QoS technique which can convert the bursty traffic into avg (or) fixed traffic

- Leaky Bucket (LBA)
- Token Bucket (TBA)

Leaky Bucket (LBA)

- Even though the i/p rate to the bucket is variable, the o/p rate from the bucket is always constant
- LBA can convert the bursty traffic into avg (or) fixed traffic by averaging all the i/p rates of the bucket.
- Drawback: Possibility from bucket full, which leads to packet loss.

Token Bucket (TBA)

- If the i/p rate to the bucket is variable then the o/p rate from bucket also variable

C = Bucket capacity
M = o/p rate from the bucket / departure rate
ρ = arrival rate to the bucket

then,

Burst length (S) or Burst duration = $\frac{C}{M-\rho}$

Congestion Avoidance / Closed loop Congestion Control

(1) Back Pressure: Sender knows the congestion from intermediate node.

Capacity: 100 100 100 10

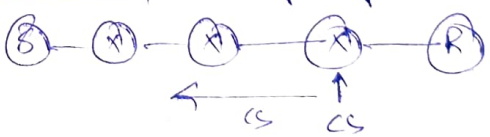
(2) Choke Packet: Sender knows the congestion directly from the node where congestion occurs.

(3) Implicit Signaling: If the sender doesn't get the ack in the time period it set after the tx then implicitly the sender thinks that there is a congestion in n/w.

(4) Explicit signalling:

(i) Forward explicit signaling:

(ii) Backward Explicit Signaling:



LBA vs TBA

(i) TBA calculates bucket capacity first.
(ii) TBA discards the tokens if the bucket is full but not packets.
(iii) The TBA is also a LBA but tokens are regulating the packet forwarding
(iv) While the packets are leaking from the bucket the tokens count in the bucket gets decreased.

Network Addressing (Ipv4)

* 32 bit (4B)/4 octet addr.
* Ipv4 provides 2^{32} no of addr.
* Ipv4 Addr space Length = 2^{32}

Ipv4
- Classful Addressing
- Classless Addressing (CIDR)

No of n/w	No of Host/n/w		Class	First Byte
$2^7 - 2$	$2^{24} - 2$	0-127	Class A	0
2^{14}	$2^{16} - 2$	128-191	Class B	10
2^{21}	$2^8 - 2$	192-223	Class C	110
		224-239	Class D	1110
		240-255	Class E	1111

(2 for Loopback Addr, Booting Addr.)

(2 for NA & BA)

1st Byte Dotted Decimal Representation

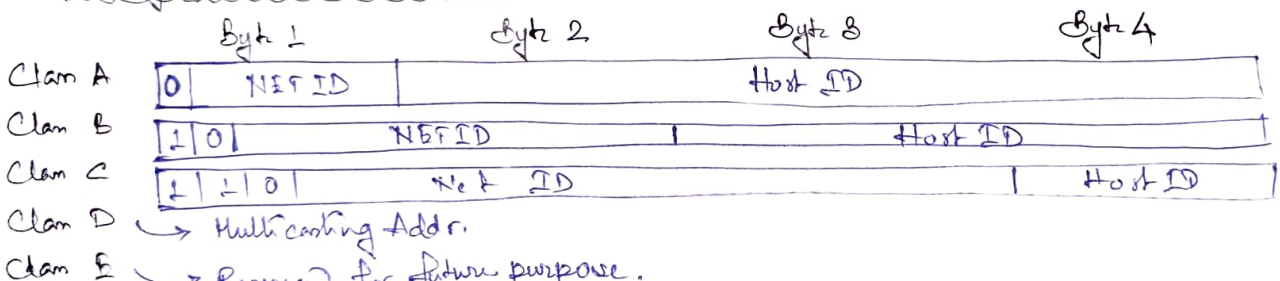
1B (8 bits)

* Relationship b/w Iso model Addr.:

IpAddr
- Host Addr. (LL) : It is an addr which is assigned to system in a n/w.
- Net Addr. (NL) : It is an addr which is assigned to a n/w in an internet.
- Port Addr. (TL) : Defines the location where the sender/system is connected to the internet.

Socket Addr.

* Finding the NETID & HOSTID :



Class D → Multicasting Addr.

Class E → Reserved for future purpose.

* Class A: (0-127) → $2^7 = 128$ - no of n/w's

(1-126) NID → 0 — 0.0.0.0 — 0.255.255.255 — NID → 127 — 127.0.0.0 — 127.255.255.255

Assignable range of 1st n/w of class A: 0.0.0.1 to 0.255.255.254

" " Last n/w of class A: 127.0.0.1 to 127.255.255.254

* Class B (128-191) — 128.0.0.0 — 128.0.255.255 — NID - 128.0 — NID → 191.255 — 191.255.0.0 — 191.255.255.255

Assignable range of 1st n/w of class B: 128.0.0.1 to 128.0.255.254

" " Last n/w of class B: 191.255.0.1 to 191.255.255.254

* 2nd n/w of class B: 128.1.0.0 to 128.1.255.255

* Class C - (192 - 223)

NID - 192.0.0 — 192.0.0.0 — 192.0.0.255 — NID - 223.255.255 — 223.255.255.0 — 223.2

Arrange the range of 1st n/w of class C: 192.0.0.1 to 192.0.0.254

" " of last n/w of class C: 223.255.255.1 to 223.255.255.254

* 2nd n/w of class C :- 192.0.1.0 to 192.0.1.255.

* Special Address

1) NA — first addr of n/w — NID: Specific | HID: All 0's → (1) class (2) range of addr of that n/w (3) NID portion.

2) DBA — Last addr of n/w — NID: Specific | HID: All 1's → * used to transfer the same msg to all the system of the other n/w.

3) LBA — NID: All | HID: 1's → * used to transfer the same msg to all the system of the n/w

4) Booting 0.0.0.0 — NID: All 0's | HID → * used to be initiate the n/w

5) Loop Back — NID: 127 | HID: Specific → 127.0.0.0 to 127.255.255.255

↳ Private Addr. ↳ * It is used to chk the internet

* How to calculate NA &

① Masking :- The process of finding the NA from the given IP Addr of a range is referred as Masking

* To find out the NA from the given IP Addr in a range, we consider,

1. Default Mask Addr. (DMA)

Class	DMA
A	255.0.0.0
B	255.255.0.0
C	255.255.255.0

2. Masking Rules

(i) mask byte → 255 then set the corresponding byte of an IP addr to the result

(ii) mask byte → 0 then set "0" to the result.

* Subnetting / Subnetworking :- Def^1 : process of dividing large n/w into small-small n/w. Small n/w is called as "subnetwork".

Def^2 : The process of borrowing the no of bits from the HID portion in order to create the SID portion.

Before Subnetting level 2 ip | NID | HID |

After Subnetting level 3 ip | NID | SID | HID |

NA — NID: Specific | HID: All 0's | — first address of n/w

SNA — NID: Specific | SID: Specific | HID: All 0's | — first address of subnet.

How to calculate Subnetwork Addr.?

(1) Subnet mask Addr. (SNMA)

(2) Along with masking rules,

"If the mask byte neither 255 nor 0 then perform "logical AND" op b/w the mask byte and the corresponding byte of an IP addr.

	DMA	Class	SNMA
	255.0.0.0 = 8 1's	(A)	> 8 1's (9,10,11...)
	255.255.0.0 = 16 1's	(B)	> 16 1's (17,18,19,...)
	255.255.255.0 = 24 1's	(C)	> 24 1's (25,26,27...)

NA
- DMA
- Masking rules

SNA
- SNMA
- Masking + Add rule

Q) Given IP Addr. - 201.14.78.66 and the subnet mask 255.255.255.224.

Model 1 : What is the Subnet Addr.

IP: 201.14.78.66

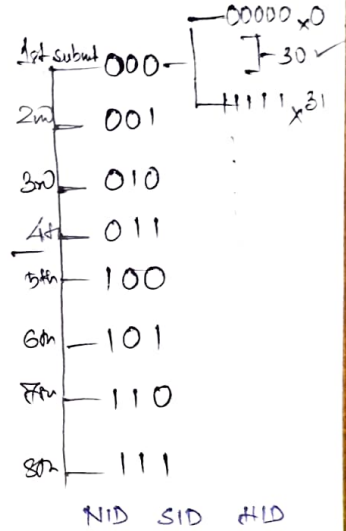
SNMA: 255.255.255.224

SNA: 201.14.78.64

NID: 201.14.78 | SID: 010 | HID: 00000

01000010
11100000
01000000

3rd subnet



Model 2 : Finding the Broadcast Addr.

IP: 201.14.78.66

SNMA: 255.255.255.224

SNA: 201.14.78.64

	NID	SID	HID			
SNA:	201.14.78	010	00000	spc	spc	All 0's
BA:	201.14.78	010	11111	spc	spc	All 1's

201.14.78.95 ← This is BA of 3rd subnet

Ex2: SID = 010 (3rd subnet)

Model 3 : Finding first and Last subnet addr

SNA = 201.14.78.64

201.14.78.010 00000 (SID | HID)

1st subnet: 201.14.78.000 00000
201.14.78.0

Last subnet: 201.14.78.111 00000
201.14.78.224

201.14.78 — 000 ... 111

Model 4 : Finding no of subnets/network

1) To find out the no of subnets/n/w, we consider the extra 1's that are added to the DMA

2) If the extra 1's that are added to the DMA is "n", no of subnets per n/w is 2^n

3) If the no of subnets is "n" then extra 1's that should be added is $\log_2 n$.

Model 5: Finding the no of hosts/subnet.

1) To find out no of hosts/subnet, we consider the total no of 0's (zeros) as part of given SNMA.

2) If the total no of 0's (zeros) in 'n' then the no of hosts/subnet is $2^n - 2$

* OBSERVATION: (1) n^{th} subnet means in SID (n-1) value in decimal

Like, 10^{th} subnet → 9 = 1001 in SID

(2) Last but one SNA

1110 = 14 ✓

1111 = 15 ✗

(3) n^{th} host → means n value in decimal

Like, 5^{th} host → 0101 in HID

Model 6: Finding the address of X host of Y subnet.

eg: 1) Find the addr of 5^{th} host of 5^{th} subnet - 192.168.14. 0100 0101 (SID | HID)

5^{th} subnet → 4 = 0100

5^{th} host → 5 = 0101

2) Find the Last but one assignable addr of 10^{th} subnet?

192.168.14. 1001 1101 (SID | HID)

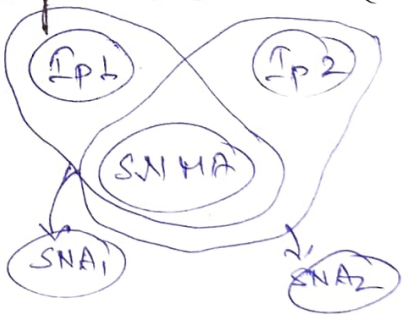
192.168.14.157

192.168.14.1001 —
- 0000
- 1101 ✓
- 1110
- 1111 ✗

* How to calculate whether two Ip addr. belongs to the same subnet(or) not.

* $SNA_1 = SNA_2$ → Ip1 & Ip2 belongs to same subnet

* $SNA_1 \neq SNA_2$ → Ip1 & Ip2 belongs to diff. subnet



* Model 8:

1) A router uses the following routing table:

Destination	Subnet mask	Interface
144.16.0.0	255.255.0.0	Eth 0
144.16.64.0	255.255.224.0	Eth 1
⋮	⋮	R2
		R3
Default		Interface 3
		Interface 4

Eth 3 ✗

matched: Eth 0, Eth 1, Eth 2 ✗

"logical mask" → more subnets → Configuration is very easy

In SNMA, more no of 1's

On which router/interface with the router forward packets addressed to dest^n 144.16.68.117 respectively?

* If Nothing matches then go to "Default".

CIDR (Classless Inter Domain Routing)

* VLSM : Variable length subnet mask.
* features : No classes
  2 classification — Public, Private

Address format → x.y.z.t /n → "mask bit"/prefix
(Ip Addr/suffix)

* Range of Private Ip Addr :

Class	Range
1-126 A	10.0.0.0 to 10.255.255.255
2^{14} (128-191) B	127.16.0.0 to 127.31.255.255
2^{21} (192-223) C	192.168.0.0 to 192.168.255.255.

Note! :

* finding the first addr of a Block/Nw :

Technique (1) : First Addr = (Any Addr) AND (Network MASK)

Technique (2) : We can keep the n left most bits of any addr in the block and set the (32-n) bits to 0's to find the first Addr.
→ Set (32-n) no of right most bits to 0's.

* finding the Last Addr/Boardcast Addr of a Block/Nw :

Technique (1) : Last Addr = (any Addr) OR (NOT (Network Addr))

Technique (2) : We can keep the n left most bits of any Addr in the Block and set the (32-n) bits to 1's to find the last Addr.
→ Set (32-n) no of right most bits to 1's.

* finding the no of addresses in Block/Nw : 2^{32-n}.

---

* Rules for forming CIDR Blocks :

1) All Ip addresses must be contiguous
2) Block size must be the power of 2 (2^n).
3) First Ip addr of the Block must be evenly divisible by the size of the block.

Model 2: An ISP has the following chunk of CIDR based IP addresses available. 245.248.128.0/20. The ISP wants to give half of that chunk of addresses to Org. A and a quarter to Organization B, while retaining the remaining with itself. Which of the following is a valid addr of addresses to A and B?

245.248.128.0/20

$2^{32-n} = 2^{32-20} = 2^{12}$

Org A: no of addrs $= 2^{32-n_1} = 2^{11} = 2^{12}/2$

$\Rightarrow n_1 = 21$

Org B: $2^{32-n_2} = 2^{10} = 2^{12}/4$

$\Rightarrow n_2 = 22$

FA: 245.248.128.0/22

= 245.248.1000 0000.00000000/22

LA: 245.248.1000 0011.1111 1111/22

↳ 245.248.131.255/22

FA: 245.248.1000 0000.00000000/21

245.248.136.0/21

LA: 245.248.1000 1111.1111 1111/21

245.248.143.255/21

Model 3: A router has the following (CIDR) entries in its routing table:

Address/Mask	Next Hop
135.46.56.0/22	Interface 0
135.46.60.0/22	Interface 1
135.53.40.0/23	Router 1
Default	Router 2

* What is the next hop the router routes if a packet with the IP address 135.46.63.9 arrives at it?

TL: Segments
* Congestion Control.
* EC * Connection Management
* FC * Segmentation & Reassembly
* Port Addressing
* Process to Process Delivery

Transport Layer:

DLL
frames
EC
FC
AC
Physical Addr.
Node to Node

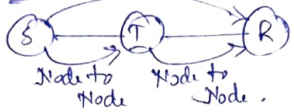
NL
packet
Routing
C.C.
Frag & Reassembly
Logical Addr.
End to End.

Delivery in OSI Reference Model

DLL → Node to Node
* Hop to Hop
* 

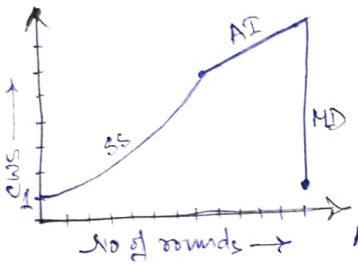
* 3 types: Sender - Intermidiate, Int. - Int, Int - Recv.

NL → End to End
* Host to Host
* Case-1: End to End


* Case-2: End to End


* To perform End to End delivery, Node to Node is optimal (Depends on distance b/w S & R)

TL → Process to Process
* End to End b/w 2 process
* End to End is mandatory to perform process to process delivery.



Congestion Control in TL

Slow Start (SS) phase.
1) The sender cal^n the CW to the receiver with the size of one segment initially.
2) After 1st tx of 1 seg the CW = 2.
3) After tx of 2 seg the CW = 4 and so on...
4) It seems there is an exponential growth in CWS.
5) This exp^n growth in CW is continued till reaching the congestion threshold.

CW = 32KB (timeout) CT = CW/2 = 16KB

S.S
- 0th round → 1 MSS (1×2KB) = 2KB
- 1st round → 2 MSS (2×2KB) = 4KB
- 2nd round → 4 MSS (4×2KB) = 8KB
- 3rd round → 8 MSS (8×2KB) = 16KB ← CT is reached
- 4th round → 9 MSS = 16KB + 2KB = 18KB
- 5th round → 10 MSS = 18 + 2KB = 20KB
- 6th round → 11 MSS (20 + 2) = 22KB
- 7th round → 12 MSS (22 + 2) = 24KB
- 8th round → 13 MSS (24 + 2) = 26KB
- 9th round → 14 MSS = 28KB
- 10th " → 15 MSS = 30KB
- 11th " → 16 MSS = 32KB

11 × 100ms = 1100ms (or) 12 × 100ms = 1200ms

Additive Increase (AI) phase (Avoidance)
i) Once the CWS reaches to the CT, from that moment onwards there is a linear growth in CWS.
ii) This linear growth in CW is continued till "timeout" occurs.
↓ Retransmission timer.

Multiplicative Decrease (MD) phase (Detection)
* Once the timeout is occured, the CT for the next round of the Slow start phase is set to half to the CW and at the same time, the CWS is reset to initial size (one seg).

[GATE 2014]
Let the size of congestion window of TCP connection be 32KB when a timeout occurs. The RTT of the conn. in 100ms and max. segment size (MSS) used is 2KB. The time taken (in ms) by the TCP conn. to get back to 32KB CW is
Ans → ([1100 - 1200]) ms

* TCP packet Header format :- (20-60) Bytes.

Header	Data

Source Port Address (16b)	Destination Port Address (16b)
Sequence Number (32b)	
Acknowledgement Number (32b)	
HLEN 4bit	Reserved 6bit	URG ACK PSH RST SYN FIN	Window size (16b)
Checksum (16b)	Urgent pointer (16b)
Options and Padding (40B)	

Defines HL in 4B words [5-15] [20-60]B

(*) used for error checking purpose.
TCP calculates the checksum for both header and data.
As TCP is reliable, inclusion of checksum is mandatory.

(*) TCP is a stream transport Protocol & Reliable, Byte oriented

TCP works

* defines the next expected byte seq. no.

As TCP is a byte oriented protocol

(*) defines the no assigned to the first byte of data contained in the segment

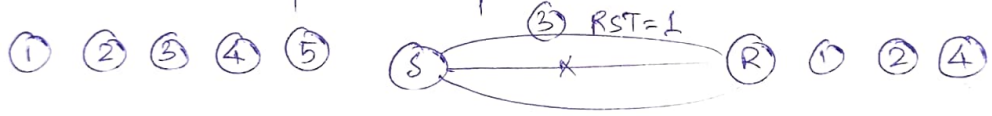
(*) TCP assigns the seq no. to each & every Bytes of the segment

→ It defines how many bytes of data that recev. can receive. [0 to $2^{16}-1$]B

→ It points to urgent Bytes of a segment for which urg flag is set to 1.

* Sender window size = min(cwnd, rwnd)

* Flags :- ACK + SYN — connection Estd.
ACK + FIN — connection Released.
PSH = 1 — Then that seg. should be acknowledged immediately after receiving.
URG = 1 — Then that seg should reach to recev. immediately before Others through any media.
RST = 1 — Then that pkt. have ✗ travelled in new conn.



* UDP Datagram Header format :-

Header	Data

UDP datagram: [H | D] TL
IP: [H | H | D] NL

Source Port Addr. (16b)	Destination Port Addr. (16b)
Total length (16b)	Checksum (16b)

Same as TCP

Same as TCP

UDP calculates the checksum for both header and data. As UDP is unreliable, inclusion of checksum is not mandatory.

IPTL = IPHL + IPDL
⇒ IPDL = IPTL − IPHL = TL UDP datagram

* Total length of UDP datagram = IPTL − IP's header length.

IP → 20-60B
TCP → 20-60B
UDP → 8B
} Header's length.

Flow Control (TL)

* All TCP connections are → Full Duplex (Sliding window)
* TCP uses sliding window as flow control protocol.
* Sliding window can suffer from a problem called "Silly Window Syndrome"

Silly window Syndrome - Occurs at sliding window when the sender generates only 1B at a time and when the recv. receives only 1B at a time.

Sol": Nagle's Algo: Sender will not transmit the pkt till the sender accumulates 1 MSS.

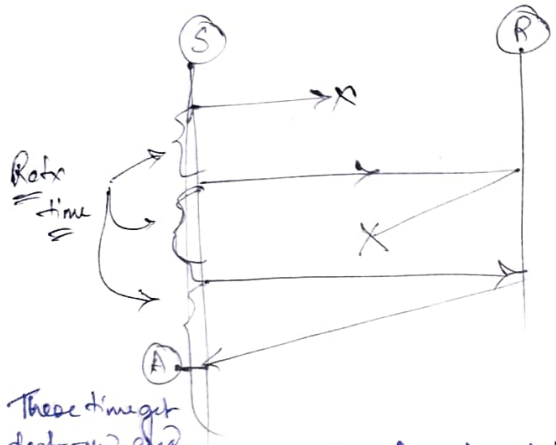
Clark's Algo: Recv. will advt. the window till it reaches 1 MSS.

∑ Transport layer – Packet level Errors
Data link layer – Bit level Errors

Error Control (TL)

* Errors control in transport layer means "packet loss".
* To make retransmission TCP sender needs "timeout" value, To calculate timeout value, TCP uses following 4 types of timers.

(1) ACK (Retransmission) Timer or Timeout Timer :- used by TCP

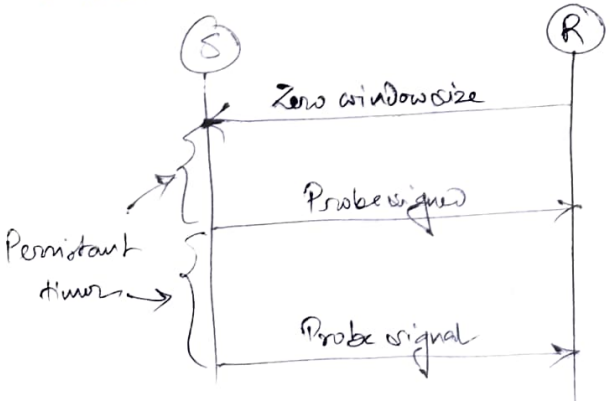


These timer get destroyed and sender transmit new packet after A point only

* Keep Alive timer → long idle connections
* Persistent time → zero window size advertisement (Deadlock)
* Timed out timer → Connections Termination phase
* ACK (Retransmission timer)
  ↳ ACK loss, Packet loss, ...

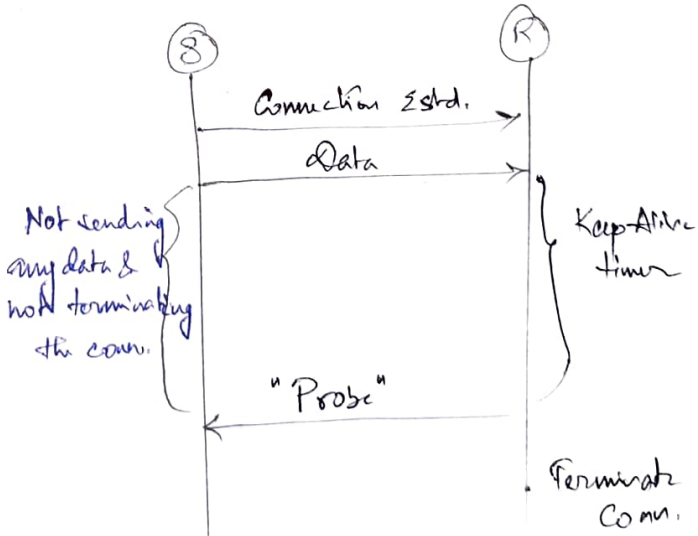
(3) Persistant Timer

* used to deal 0 window size advt



* If probe signal will not sent by sender and the advt. window sent by recv. after got some part empty is lost in b/w then deadlock may arrive.

(4) Keep Alive timer



"Probe" → is a control signal to checking the client's status

Flow Control (TL)

* All TCP connections are → Full Duplex (Sliding window)
* TCP uses sliding window as flow control protocol.
* Sliding window can suffer from a problem called "Silly Window Syndrome"

Silly window Syndrome - Occurs at sliding window when the sender generates only 1B at a time and when the recv. receives only 1B at a time.

Sol": Nagle's Algo :- Sender will not transmit the pkt till the sender accumulates 1 MSS.

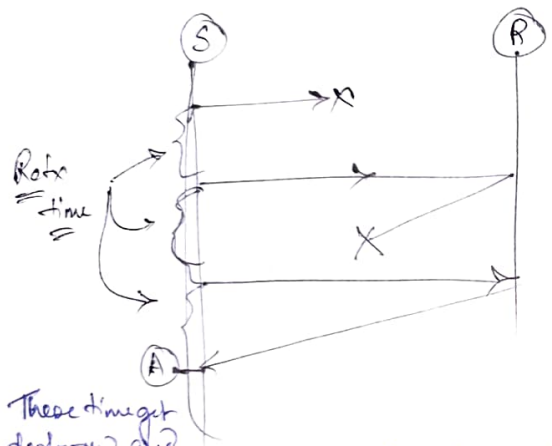
Clark's Algo :- Recv. will advt. the window till it reaches 1 MSS.

Transport layer – Packet level Errors
Data link layer – Bit level Errors.

Error Control (TL)

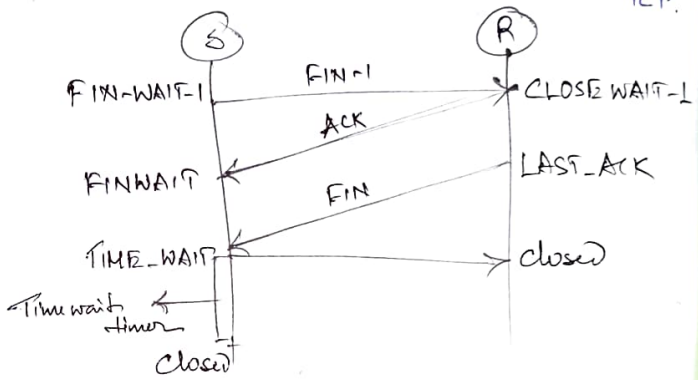
* Errors control in transport layer means "packet loss".
* To make retransmission TCP sender needs "timeout" value, To calculate timeout value, TCP uses following 4 types of timers.

(1) ACK (Retransmission) Timer or Timeout Timer :- used by TCP



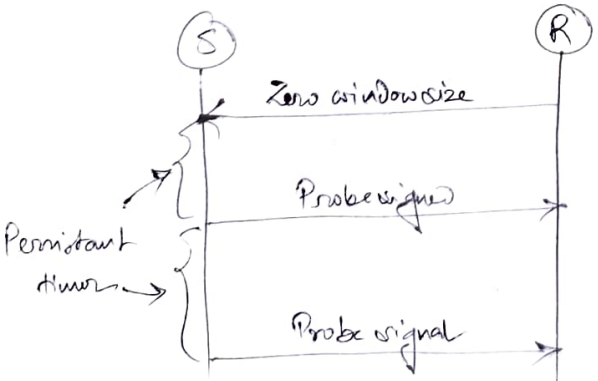
(2) Time wait Timer :-

* used in connection termination phase of TCP.



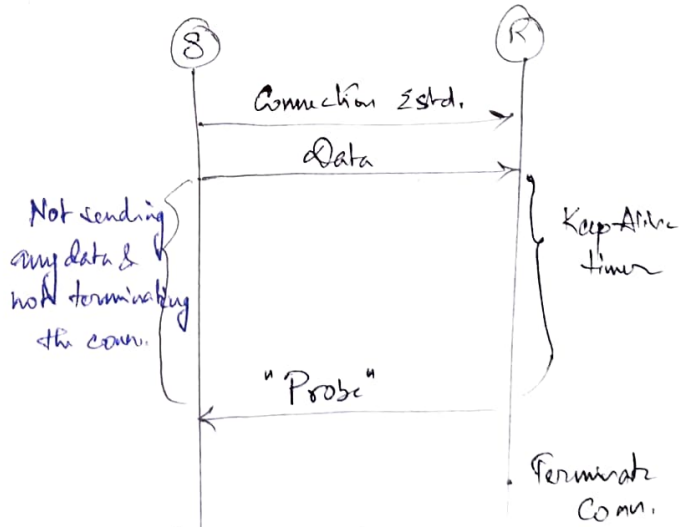
(3) Persistant Timer

* used to deal 0 window size advt



* SP probe signal will not sent by sender and the advt. window sent by recv. after got some part empty is lost in between b/w then deadlock may arrive.

(4) Keep Alive timer



"Probe" → is a control signal to checking the client's status

TIMEOUT VALUE CALCULATION [To find out Retx time value in TCP]

Basic Algo

$E_{RTT} = \alpha * I_{RTT} + (1-\alpha) * N_{RTT}$

$Timeout = \beta * E_{RTT}$

where, E_{RTT} = Estimated RTT [Retransmission timer]

I_{RTT} = Initial RTT

N_{RTT} = New RTT

α = smoothing factor b/w I_{RTT} & N_{RTT} usually $0 \le \alpha \le 1$

β = 2 usually (constant)

Jacobson's Algo.

$D_N = |I_{RTT} - N_{RTT}|$

$D_E = \alpha * D_I + (1-\alpha) * D_N$

$Timeout = 4 * D_E + E_{RTT}$

where, D_N = New Deviation

D_I = Initial Deviation

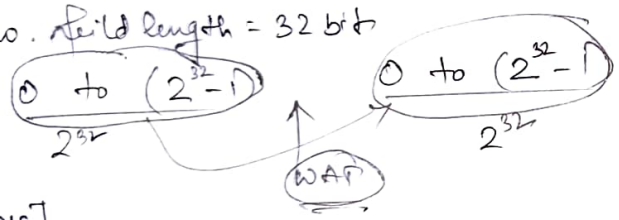
D_E = Estimated Deviation.

* Calculation of WRAP Around Time :- [After generation of one set of sequence no., to generate the same set of sequence no.]

BW given. From there we can find out the no of Bytes per second consume then divide that value with 2^{32} we can get the WAT.

* Window Scale option :- If bandwidth delay product $(BW \times RTT) > (2^{16}-1)$ [window size] then window scale option can be used. The window scale can be expanded for $(2^{16}-1)$ to $(2^{16}-1) \times 2^{14}$ B.

* Seq no. field length = 32 bit



* Wrap Around Time (WAT) = $\frac{\text{Total Seq. No.}}{\text{Bandwidth } \left[\frac{\text{Bytes}}{\text{sec}}\right]}$

8) [GATE 2018]
Consider a long-lived TCP session with an end-to-end bandwidth of 1 Gbps ($=10^9$ bps). The session started with a sequence number of 1134. The minimum time (in sec, rounded to the closest integer) before this sequence number can be used again is ⟶

1 Gbps = 2^{30} bps ✗

= 10^9 bps ✓ [given]

1 sec ⟶ 10^9 bits

1 sec ⟶ $\frac{10^9}{8}$ Bytes

1 sec ⟶ $\frac{10^9}{8}$ seq. no.

2^{32} seq no. ⟶ $\frac{2^{32} \times 8}{10^9}$ sec.

∴ WAT = 34.35 sec (Ans)

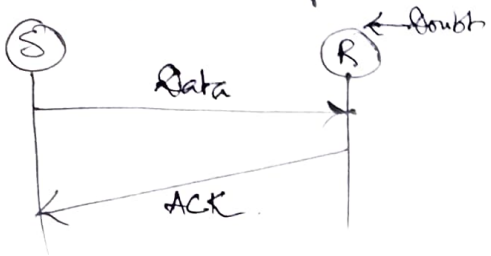
* Min seq. no. required to Avoid Wrap Around Time with in life time = B × LT

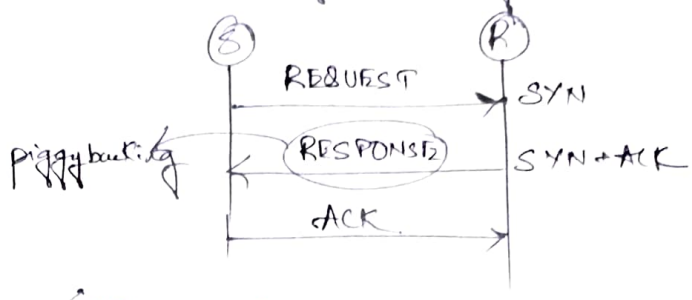
* Min no of bits required to Avoid Wrap Around Time with in LT = $\lceil \log_2 B \times LT \rceil$

Connection Management (TL) → Connection Estd
→ Connection Release.

TCP - CO Protocol → CE, DT, CR.

2-way Handshaking ⟷ CE ⟷ 3-way Handshaking





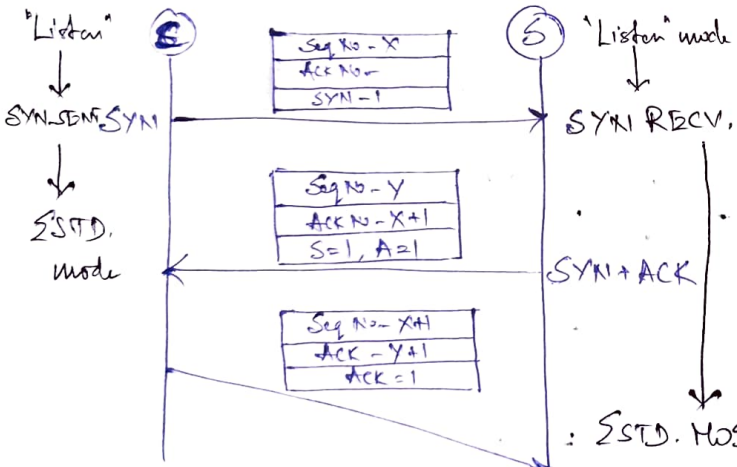
Drawback of 2WH :- 2 Army Problem.

2 flags:	SYN	ACK	Meaning	
	1	0	REQUEST	Confirmation
	1	1	RESPONSE	Confirmation
	0	1	ACK	Confirmation
	0	0	DT	← Indication of Data tx.

- SYN = 1 → 1 seq No.
- ACK = 1 → 0 seq. No.
- FIN = 1 → 1 seq. No.
- 1 Data type → 1 seq. No.

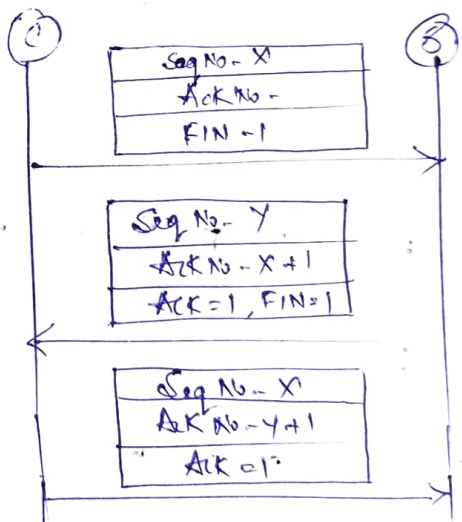
Port No.	Name.
0 - 1023	Well known Port No.
1024 - 49151	Registered Port No.
49152 - 65535	Dynamic.

Connection Estd. Phase.



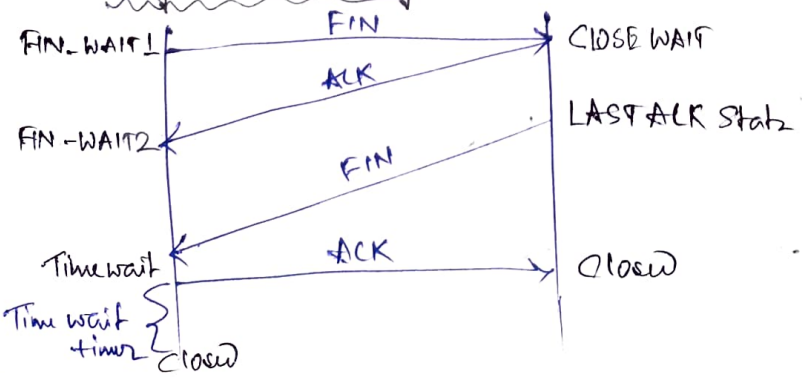
* ACK segment not carrying any data - no seq no.
* ACK segment ~~not~~ carrying any data - seq no. can be consumed.

Connection Termination Phase



* TCP uses 4 way handshaking for Connection termination
* FIN segment carrying any data consumes one seq no. otherwise not.

4 way Handshaking



* SESSION LAYER

* PDU - message
* Session management
* Password validation
* Synch.
* Dialog control

* APPLICATION LAYER

* PDU - message
* Email service
* Directory service
* Remote Login
* File transfer

* PRESENTATION LAYER

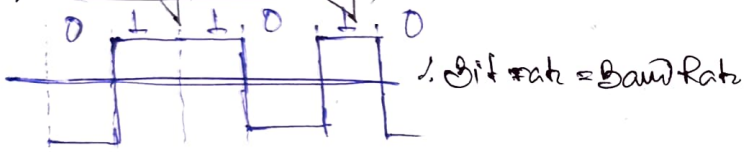
* PDU - message
* Translation
* Compression :- process of reducing the size of data w/o any data loss.
  - Lossy
  - Lossless
* Encryption :- plain text → Cyphertext
* Syntax & semantics of message
* Password Verification :- Rules are there to set the password.

* PHYSICAL LAYER

* PDU - bits
* Tx rate of the channel, BW, channel capacity
* Tx media, betw" sender & Rec.
* Tx mode - Full Duplex, Half Duplex, Simplex,
* Topology of the n/w.
* Encoding

* Bit Rate : No of bits that are transferred per unit time.
* Baud Rate :- No of signals that are transferred per unit time.

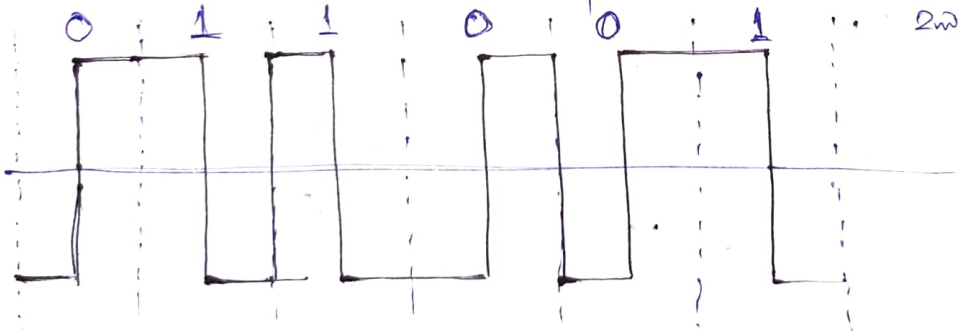
1) Binary Encoding



∴ Bit rate = Baud Rate

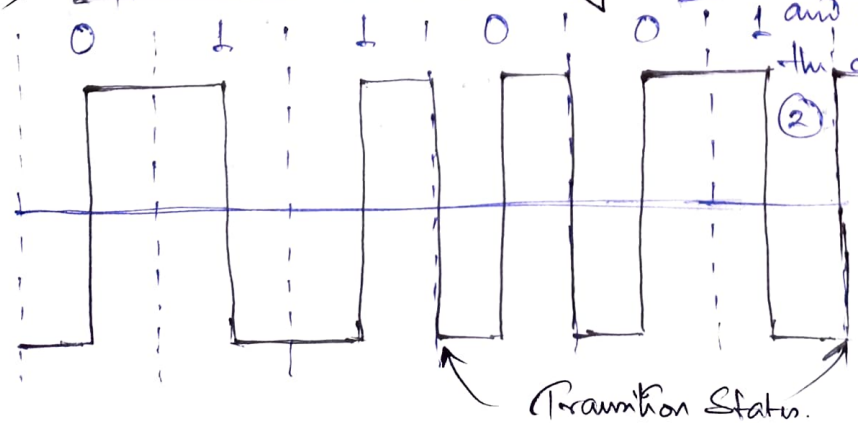
2) Manchester Encoding :- Rules: ① Allocate each clk tick to each and every digital bit further subdivide the clk tick into 2 parts
② If the encountered bit as 1 then 1st half is high and second half in low.
③ If the bit is 0 then 1st half in low and 2nd Half in High.

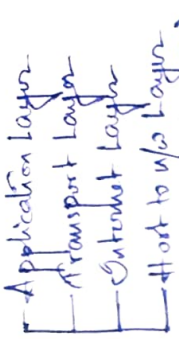
* Baud Rate = 2 * Bit Rate.
* used in Standard Ethernet.



3) Differential Manchester Encoding :- Rules: ① Allocate each clk tick to each and every digital bit further subdivide the clk tick into 2 parts
② 0 - 1 - continue 1 - 0 - complement
1 - 1 - continue 0 - 0 - copy

* Baud rate = 2 * Bit rate
* used in token ring





TCP/IP Reference Model → 4 layer

* DLL individual users for it's internet application.
PPP (point-to-point)
SLIP (Serial Line IP)

* Protocols:

1) Host to n/w Layer:
* It supports all the protocols which are existed in above layer.

2) Application Layer Protocols:

* SHOT TRICK TABLE *

	Port No.	TL Protocol	Stateful?	Persistent / Non-Persistant	Push/Pull	Inband/Out-of-band	Connection Oriented	Function	Extra
FTP	20 (Data conn.) 21 (Control conn.)	TCP	✓	21 (ctrl) - persistant 20 (Data) - Non-persistant	Can't	Out-of-band	Connection Oriented	* used to tx files from one system to another system	
TFTP (Trivial FTP)	69	UDP						* used to tx files in connectionless domain.	
HTTP	80	TCP	X	HTTP 1.0 is non-persistant HTTP 1.1 is persistant	-	Inband	Connection less	* checking the given URL addr. is existed in websource or not	
SMTP	25	TCP	X	Persistant	Push (Server side)	Inband	Connection Oriented	* transfer mail from C to S & vice versa	* Also called as Email
POP	110	TCP	✓	Persistant	Pull (client side)	Inband	Connection Oriented	* used to download the content from mail server to permanent mem of local system	* once downloaded we don't req. internet connection to access the content.
IMAP [Internet mail Access Protocol]	143	TCP	✓	Persistant	Pull	Inband	Connection Oriented	* used to download the content from mail server to virtual mem.	* Req. internet connection to access the content.
DNS [Domain name system]	53	UDP	X	Non-persistant	-	Inband	Connection less	* convert URL to IP	
NNTP [Network news transfer protocol]	119	TCP						* transporting the news group msg across the n/w.	* Advert. about the searched content.

* All protocols used TCP as TL protocol are connection Oriented protocol.

* Connection Oriented :- Three phases exist — (1) Conn" estd. (2) Data tx (3) Conn" release.
* Stateful :- Buffer or records.
* Persistant :- For Data tx (each pkt) no need for conn", each time in one connected we can tx, n' no of pkts
* Inband :- In one connection both data and control can be sent, eg:- SMTP
* Out band :- Diff connection for data and ctrl. eg :- FTP.

Application	Port No.	Transport Protocol
DNS	53	UDP
HTTP	80	TCP
FTP	20 (Data conn.) 21 (Control conn.)	TCP.
SMTP	25	TCP
POP	110	TCP
SNMP	161 - command 162 - msg.	UDP
TFTP	69	UDP
IMAP	143	TCP
Telnet	23	TCP
DHCP	67 - client 68 - Server	UDP

* Transport Layer Protocols :- (TCP)
  * Reliable & connect^n oriented Protocol.
  * Heavy weight compared to UDP header.
  * TCP is slower than UDP
  * 100% error control (Both detect^n and correct^n)
  * PDU - packet
  * TCP conn^n are FD conn^n. (Full duplex)
  * Byte Oriented.
  * Var len header (20B to 60B)
  * Uses sliding window protocol (combination of GBN & SR)
  * Inorder & stream based delivery
    ↳ The same order it sent, they will received that order.

* User Datagram Protocol (UDP)
  * Unreliable, connectionless protocol.
  * Best effort delivery [No error checking & tracking]
  * message oriented Protocol.
  * Can't do flow ctrl. and congestion control.
  * Doesn't ensure inorder delivery [Every pkt has header appended, they can take any path]
  * Realtime multimedia data tx.
  * 50% error control (Error detect only)
  * used in multicast & broadcast appl^n.

* SCTP :-
  * It is combination of TCP & UDP (Best features)
  * Connect^n oriented
  * Reliable
  * message oriented protocol.
  * can do error ctrl. & congestion ctrl.
  * Inorder delivery
  * Used in VOIP application.

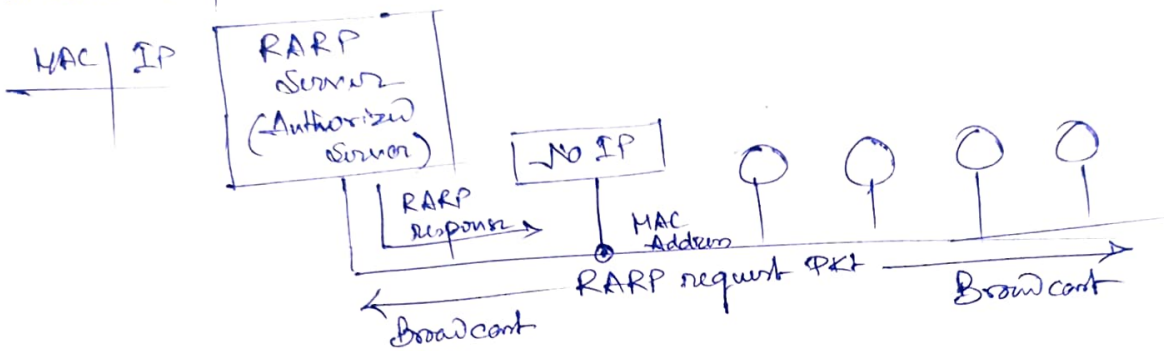
Internet layer Protocol:

(1) ARP Protocol: (Address Resolution Protocol)

* ARP accepts a LA from the IP Protocol, maps the address to the corresponding PA and pass it to the DLL. (IP → MAC)

(2) RARP Protocol (Reverse Address Resolution Protocol)

* used to find the IP Address that corresponds to MAC Address.



(3) ICMP (Internet ctrl. message Protocol)

Need: (1) No error tracking facility in IP
(2) N/w Admin. wants to know something from a specific host

ICMP Message

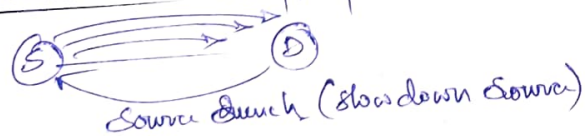
- Error Reporting
- Query Message

Error Reporting

* report problem that a router may encounter
* Always report errors to the original source.

(1) Dest unreachable

(2) Source Quench: Buffer prob. at recv.



(3) Time Exceed: TTL exceed

(4) Parameter Problem: Header issue

(5) Redirect: if the path is not available or otherwise is shortest.

Query Message

* helps a host or n/w manager get specific info from a router or another host

(1) Echo Request & Reply.

(2) Time stamp req/reply.
Time synch. in diff time zone.

④ Internet Protocol :
* unreliable connectⁿ less protocol.
* Provides best effort delivery (No error tracking & checking)
* In association with ICMP, IP can do error reporting.

⑤ IGMP (Internet Group message protocol)
* used to transfer the same msg to the group of receivers.
* used for multicasting

⑥ BGP (Border Gateway Protocol)
* In between 2 or more autonomous system routing process is possible which is called Interdomain routing.

⑦ DHCP (Dynamic Host configuration Protocol)
* Allowing a host to interact with the internet.
* connectⁿ less.
* BOOTP (Static) in nature.
* Application layer protocol.
* uses UDP as TL protocol. Port no — 67, 68.
* DHCP operations in 4 steps [DORA]
  1) DHCP Discovery Request — Broadcast req.
  2) DHCP Offer message —
  3) DHCP Request — plz find the MAC & address with the IP add.
  4) DHCP ACK — plz start your internet service.
* DHCP server in same n/w then no issue, diff n/w then relay agent come into play. It unicast the request to the DHCP server.

SWITCHING

* Process of interchanging the switches is called as switching.

① Circuit Switching : - Connection Oriented Switching → Connⁿ Estd. → Data tx → Connⁿ termination.

(i) Connectⁿ Estd Phase :
* Dedicated Resources are reserved for only data tx b/w S & R.
* Addressing is req. in only connectⁿ set up phase but not data tx phase.

(ii) Data transfer phase :
* Continuous flow of data
* Inorder flow of data.
* No addressing is req.

(iii) Connectⁿ termination phase :

- Performance of Circuit Switching :
- Efficiency is less
- Delay = Connectⁿ Setup phase (Req t_t + Req t_p + Ack t_t + Ack t_p) + Data tx ($t_t + t_p$) + Delay in teardown phase (Rec tx + t_p)

Packet Switching :

Datagram

* Connectionless
* message broken down into small parts. Each small part is called packet.
* packet carries control info (header) and original data (data)
* packets of same message travel in diff paths.
* Out of order arrival at the destn
* No resource Reservation.
* Pkt switching is implemented at the n/w layer.
* Delay = $(n+1)(t_x + t_p)$ + waiting time at router.
  n = no of router betn S & R.

Virtual Circuit

* Connection Oriented.
* combinations of best features of circuit & packet switching.
* Guarded delivery
* All the packets are travelled in the same path.
* Reliable
* Implemented at DLL
* Once Virtual circuit is estd, if there is no data tx betn sender & recev. those resources can be used by some other sender & recev.
* Only for the 1st packet the global header can be appended. The 1st pkt. ~~resources~~ reserves resource at each server along the path. Subsequently the pkt will follow the same path as the 1st sent packet for the connect^n time.

Connecting Devices

Networking Device

Device which can be used within a n/w

eg. Repeater, Hub, Switch.

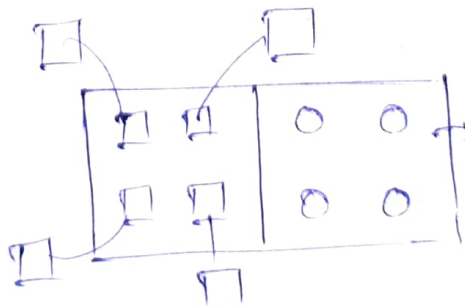
Inter networking Device

The device which is used to connect 2 or more n/ws.

eg. - Bridge, Router, gateway, Brouter.

* **Repeater :**
  * n/w device which has just 2 ports
  * physical layer device
  * pure n/w device
  * It just regenerate but not amplifies.
  * forward the data
  * filtering is not there (No intelligence)
  * No collision & Broadcast domain.

* **HUB :** Multiport Repeater
  - Forwarding the data
  - Not a store & forward device
  - No ~~filtering~~ filtering capability (No intelligence)



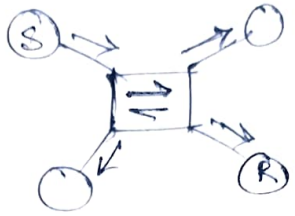
  - Broadcast & collision domain exist
  - PL Device.

* **Router:**
  * Network layer Device
  * No collision domain
  * Broadcast domain exist
  * Intelligent device.
  * Visible to all the systems of both n/w.

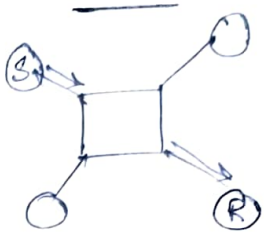
* **Gateway**
  - Application layer
  - Protocol converter (of 2 n/w working with diff protocol)

* **Brouter:** Bridge + Router
  * Either in DLL or in NL.
  * Both MAC & IP address.

HUB



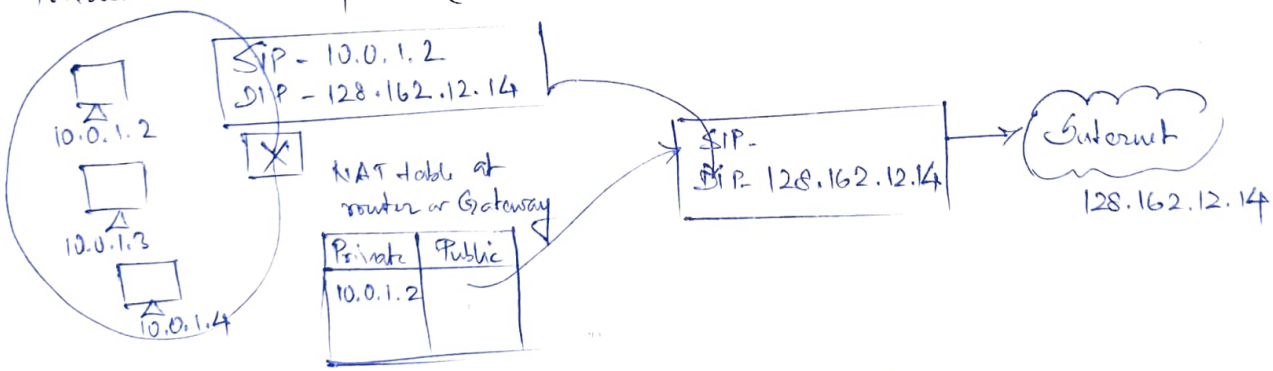
SWITCH



* **Bridge:**
  * It is a layer 2 device.
  * Works with MAC address
  * Filtering capability (Based on bridge table) → MAC | port no.
  * It is used to connect 2 diff/similar n/w.
  * Break collision domain, not broadcast domain.
  * Having store & forward capability.
  * **Principle:**
    - Same n/w, Sender and Dest^n then discard.
    - If the dest^n addr is known the bridge send. if don't know then flooding.

- **Drawback:** Allowing the pkt into loops — **Sol^n:** Spanning tree, LAN as node, Bridge as edge.

* **Private** —NAT→ **public** (Route table)

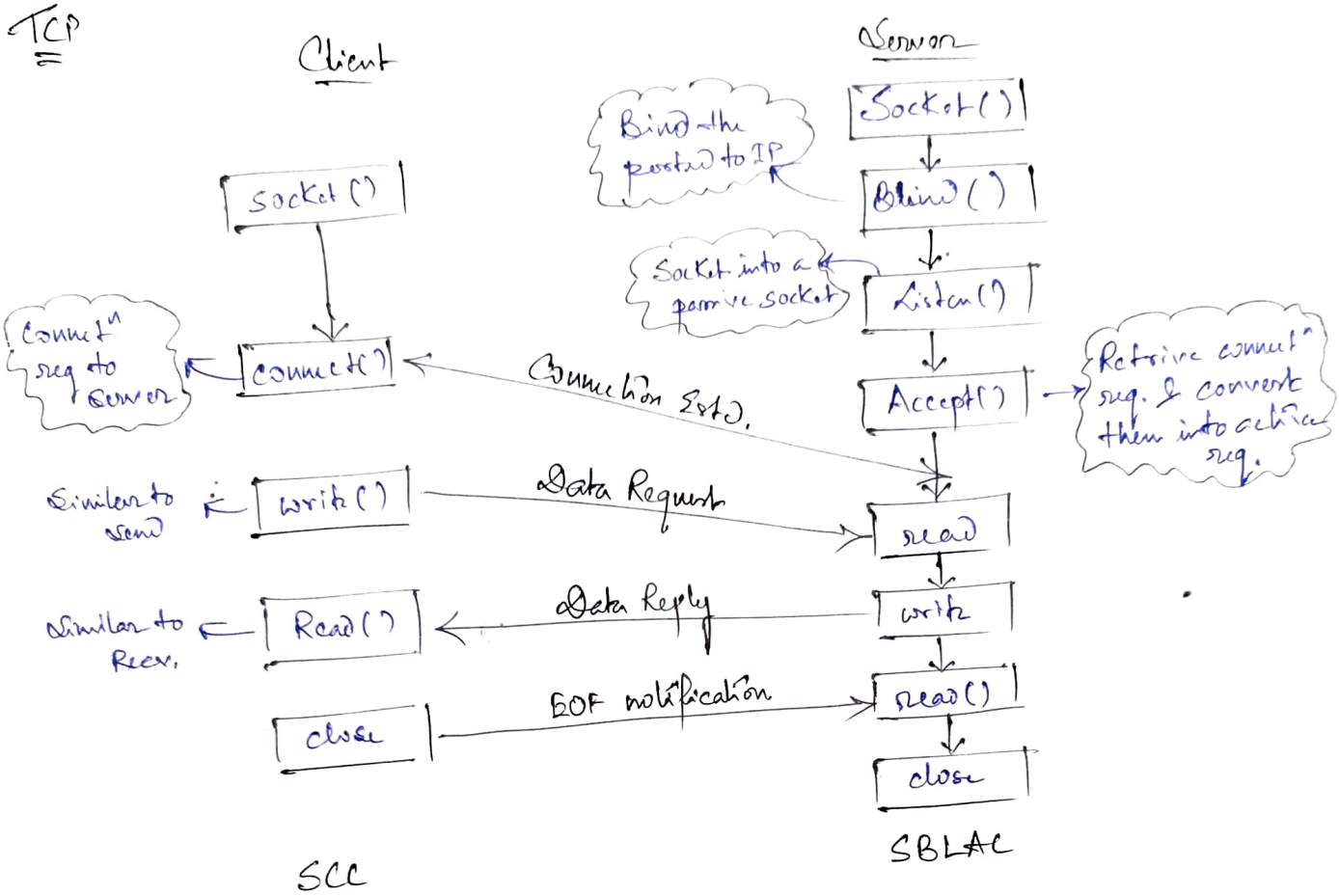


Static NAT	Dynamic NAT	PAT
* The public ip addr corresponds to one private ip is always same. * 1 to 1 correspondence.	* The public ip Addr. corresponds to one private ip (may be diff.) - Not 1 to 1 correspondance - Limited public ip is used	* Also called NAT overflow * In that PAT we use just 1 public ip.

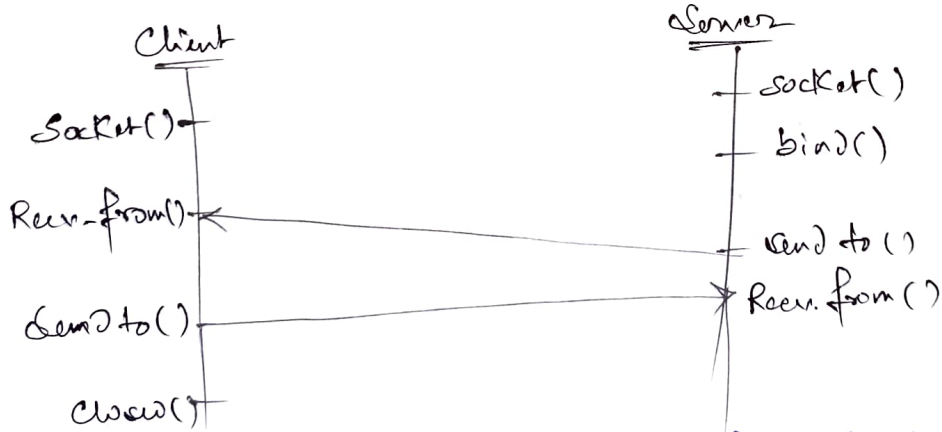
SOCKET

* Used to allow 2 or more process to communicate together.
* Socket = Ip Addr + Port Addr.

TCP



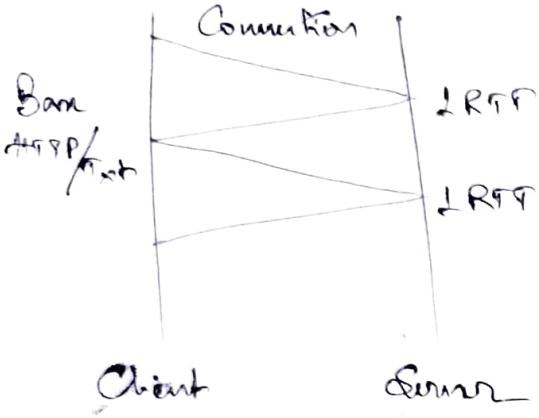
UDP



- A process can successfully call connect func" again for an already connected UDP socket. (As it is already known that a connect func" can be called again to setup new process and to disconnect the already connected sockets. So this statement is correct.
- Iterative in nature (one at a time connect")
- The process with a connected UDP socket can call connect" again for that socket for one at 2 reasons. 1) To specify a new Ip addr. and port
  2) To unconnect the socket.

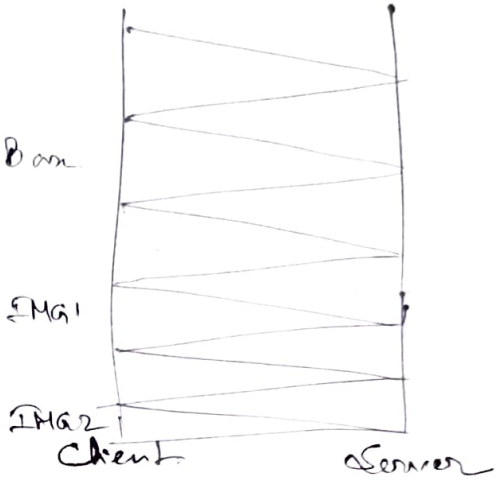
Persistant HTTP Connection (HTTP/1.1)

1) Server leaves connection open after sending response
   1 RTT for all referenced object.
2) Less overhead.



Non-Persistant HTTP (HTTP/1.0)

1) It requires 2 RTTs per object
2) More overhead.



Q) How much time is required to transmit the HTML base file and 10 objects using the non-persistent HTTP connection. The given round trip time is 20 ms.
[Note: TCP connection take one round trip time, Ignore the processing & other delays.]

(a) 400 ms (b) 420 ms (c) 440 ms (d) 380 ms

We require 2 RTT for transmitting every objects in the non persistent HTTP.

So base file + ten objects = 11

So, $11 \times 2\text{RTT} = 22$ RTTs required.

$\therefore 22 \times 20\text{ ms} = 440$ ms