CN

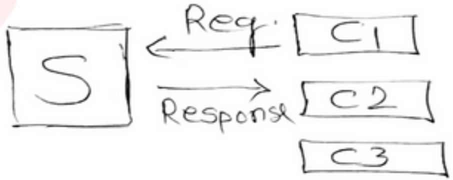
Computer — "Nodes" "Digital devices"

Networks [Connect : wired / wireless] — "Communication" " Data exchange / transfer / Sharing" " Resource Sharing"

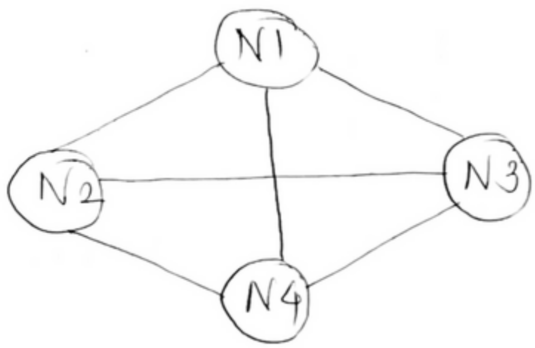
- H/W
- S/W

• Basic Terminologies

* Nodes : Computer, printer, server, Router, etc.
* Protocol : Set of Rules. (TCP, IP, UDP, FTP, etc)
* IP address : A unique identifier / Number sequence Assigned to every device on n/w.
* DNS : Domain Name system is a protocol To translate Domain Names into IP addresses.
* Client - Server :



* Peer-to peer :

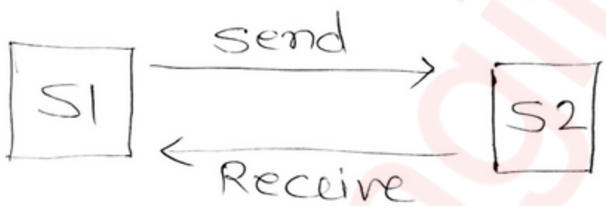


(No central Server unit)

Transmission Modes

- Simplex (one way)

- Half duplex (Both direction) But one at a time



(Not at same time)

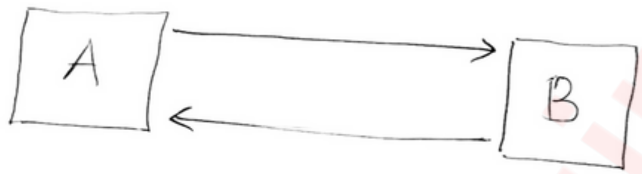
- full duplex (Both direction) At same time

Connection Types

- Point-to-Point
  (Direct dedicated link betn devices is provided)
- Multipoint
  [Shared link betn 2 or more devices]
  "Shared channel"

- Topology
  - Defines the structure.
    "how the nodes are connected to each other in a network".

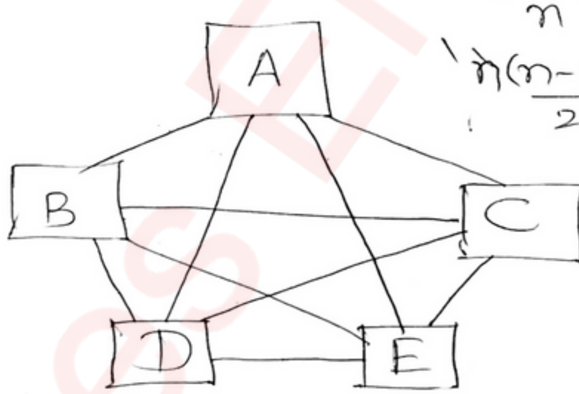
① Point-to Point Topology



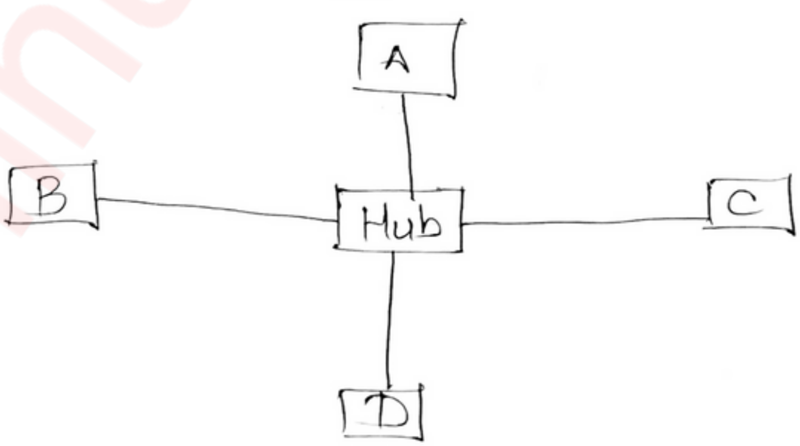
② Mesh Topology

'n' devices

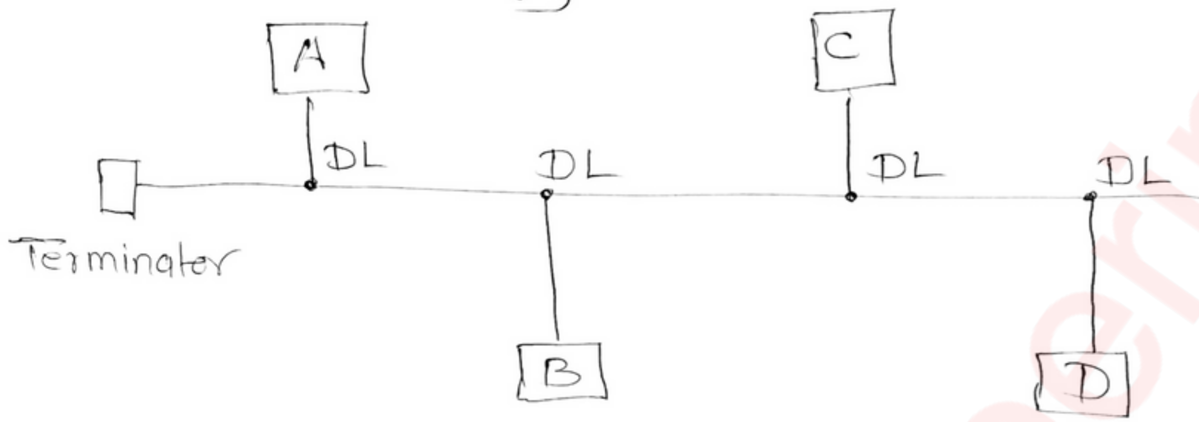
$\frac{n(n-1)}{2}$ Links



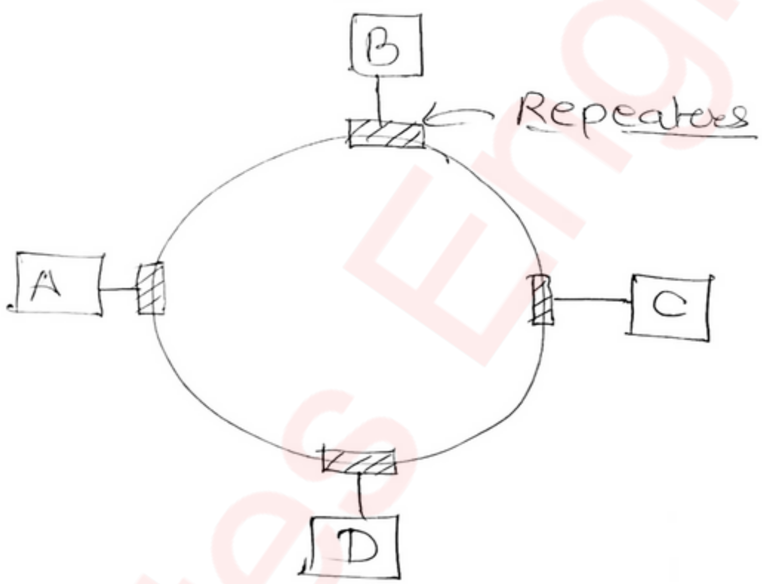
③ Star Topology



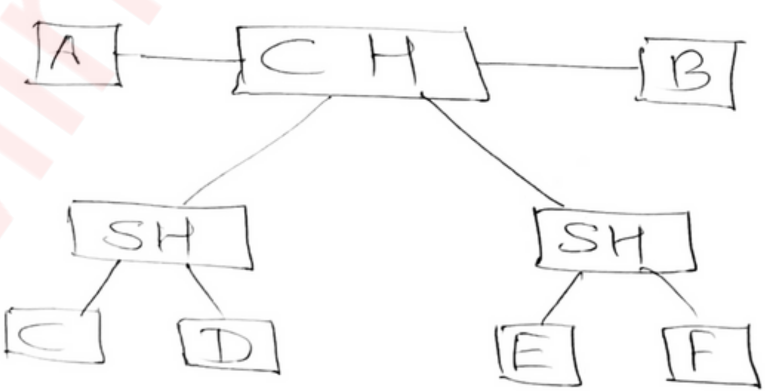
④ Bus Topology



⑤ Ring Topology



⑥ Tree Topology



Types of CN

PAN	LAN	CAN	MAN	WAN
Personal	Local	Campus	Metro-politan	Wide
100 m	2 km	5 km	5 - 50 km	>50 km
Home/Rome	office	College/University	City	Counties/States
EH Speed	VH Speed	High Speed	Avg Speed	Low speed
Private	Private	Private	Private/Public	Private/Public

OSI Model

Sending

application layer ka header original data mae embed kiya jata h and then it as a packet for next layer and so on

AL	AH | Data
PL	PH | AH | Data
SL	SH | PH | AH | Data
TL	TH | SH | PH | AH | Data
NL	NH | TH | SH | PH | AH | Data
DLL	DH | NH | TH | SH | PH | AH | Data | DT
PL	Data bits

in DL Layer it adds header as well as trailer also

Receiving

① Physical Layer

- → Data is sequence of 0s & 1s
- → Encoding : Bits ⇄ signals
- → Data Transfer Rate.
- → Connection type. (Line configuration)
- → Topology
- → Transmission Mode

② DLL (Data Link Layer)

- → Framing (divide data into frames)
- → Physical Addressing (MAC Address)
- → Access Control
- → Flow Control
- → Error Control.

③ **Network Layer** (Source $\xrightarrow[\text{Data packet}]{}$ destination

- → Logical Addressing
  (IP address)
- → Routing
- → Packetizing.

④ **Transport Layer**

- → Port Addressing
  (Process to Process Delivery)
- → Segmentation (segments)
- → Congestion Control.
- → flow Control
- → Error Control

⑤ **Session Layer**

- → Session mgmt
- → Dialog Control
- → Synchronization
  (Checkpoints)

⑥ Presentation Layer

- Translation
- Encryption & Decryption

  PT → CT   CT → PT

  sending   Receiving
- Compression

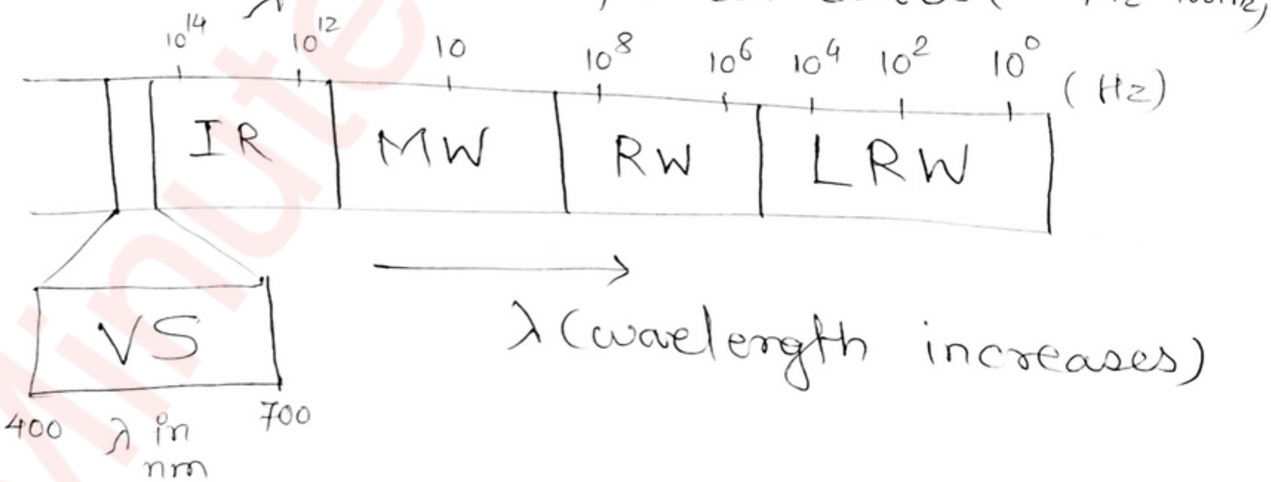
⑦ Application Layer

- Protocols
  - HTTP
  - FTP
  - POP
  - SMTP

Transmission Media

- Physical Path betⁿ (T) & (R) (Channel)
- Types
  - Guided Media (Wired) (Cables)
    - 10 Mbps (BW) Baseband T = 100 M wire length.
    - (10 Base T) (100 Base T) → Twisted pair cable
    - (10 Base 2) (10 Base 5) → Coaxial cable
    - Optical fiber cable (100 Base FX) 2 km Full duplex
  - Unguided Media (Wireless)
    - Radio waves (3 KHz – 1 GHz)
    - Microwaves (1 GHz – 300 GHz)
    - Infrared waves (300 GHz – 400 THz)

$$f \propto \frac{1}{\lambda}$$

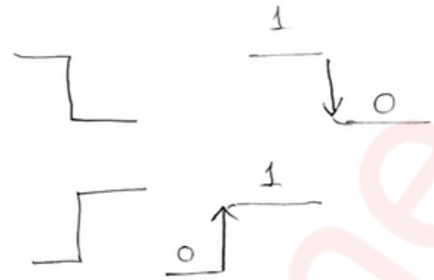


Encoding (Manchester Encoding) & (Differential Manchester Encoding)

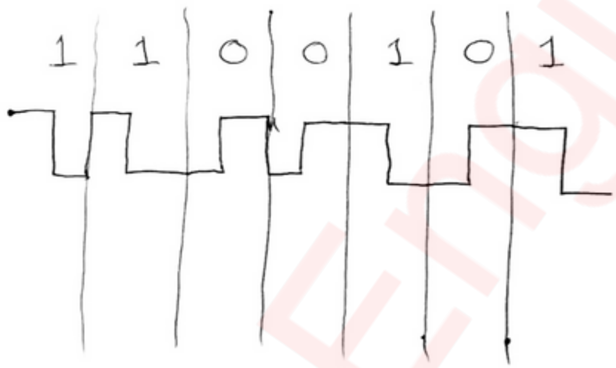
a) Manchester E

⇒ 1 →

0 →



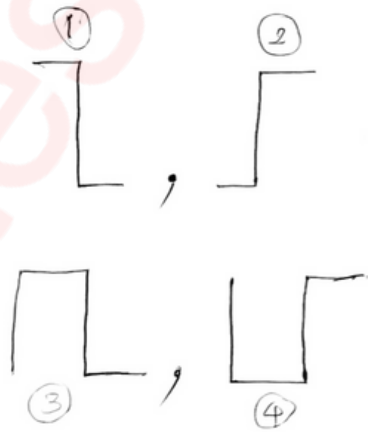
eg:

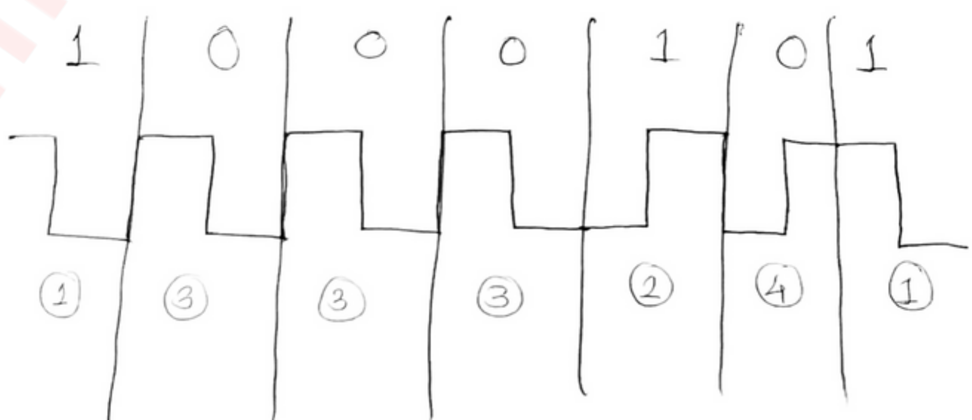


b) DME

⇒ 1 →

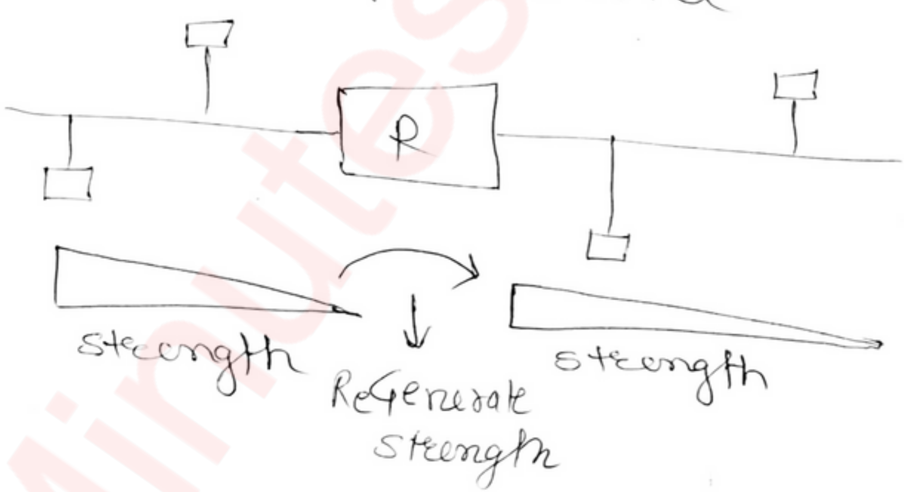
0 →





- Devices in CN
  - → Cables
  - → Repeaters
  - → Hubs
  - → Bridges
  - → Switches
  - → Routers

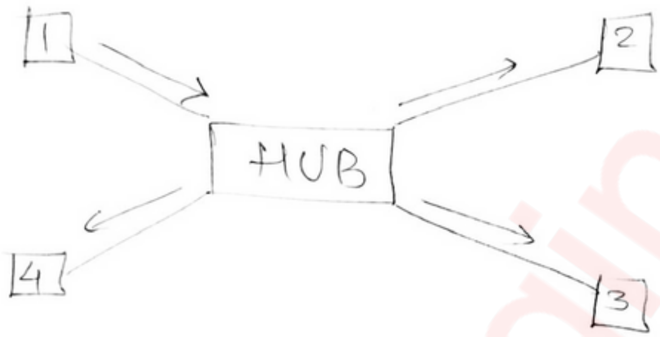
* Repeaters :
  - → Pure H/W
  - → NO filtering
  - → forwording
  - → 2 port Devicl



⇒ Collision Domain ⇒ 'n'
(no. of devices)

* Hub

- → Pure H/W
- → Multiport Repeater
- → Forwarding
- → No filtering (Broadcast)



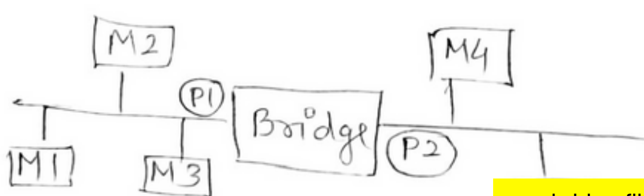
⇒ Collision Domain = 'n'

* Bridges

- → H/W, S/W [store & forward]
- → Connect 2 LANS
- → forwarding.
- → filtering
- → Physical & DLL (MAC address)

Static

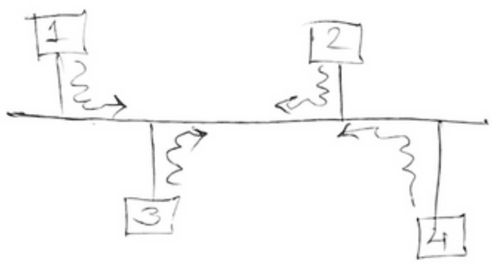
MAC	PORT
M1	P1
M2	P1
M3	P1
M4	P2
M5	P2



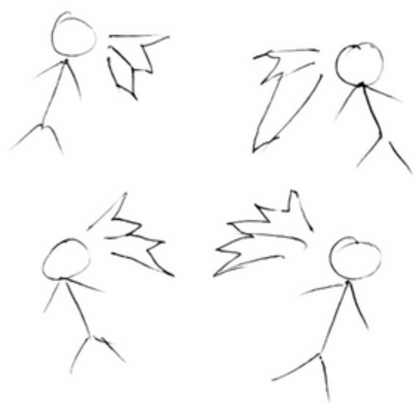
so bridge filter and do not forward msg to another side

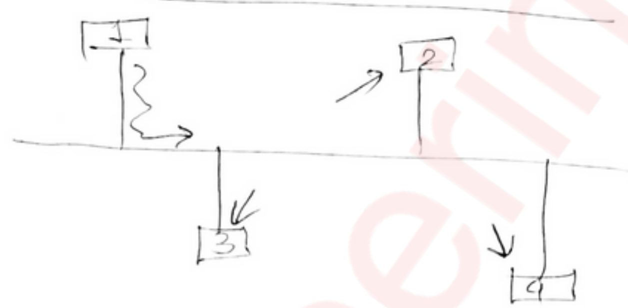
MAC(S)	MAC(D)
Source	Destination

Collision Domain Vs Broadcast Domain

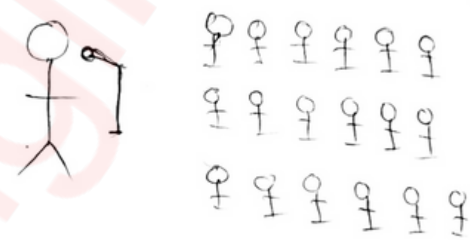


"Collision"



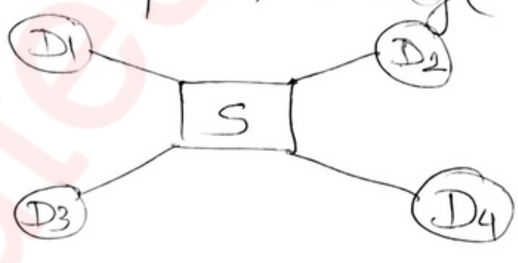


"Broadcast"
(Every m/c, device receive it)



* Switch:

→ Multiport Bridge



→ Collision Domain is '0'

→ Full duplex Links

→ Low Traffic.

* Routers
  - → PL, DLL, NL
  - → IP address
  - → Connect 2 N/ws.
  - → forwarding
  - → filtering
  - → Routing (Routing Table)
  - → Flooding

- **Switching**: Transfer of data packets from one device to other in n/w.

  Types
  - → Message switching
  - → Circuit switching
  - → Packet switching.

* Message switching
  - → Message / Data block is forwarded across entire n/w.
  - → Hop by Hop delivery
  - → High delay
  - → High Traffic

Circuit Switching (Physical Layer)

1. { Connection set up (Dedicated path)

2. { Transmission time + Propagation Delay

3. { Tear Down.

⇒ Inorder Transmission
⇒ Resources get reserved. (Cannot be used by other)
⇒ No store & forward
Low Delay

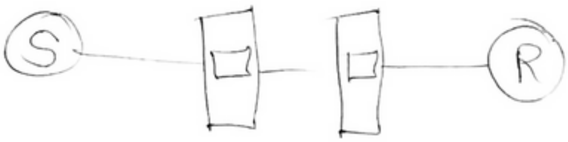
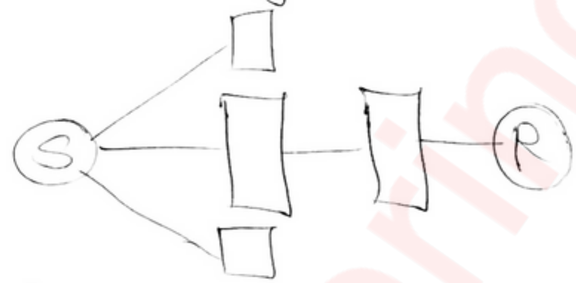
Packet switching (DLL & NL)

→ Data is divided into packets
→ store & forward
→ High Efficiency
→ Time = Transmission Time (T_t) + Propagation Time (PD)
→ No Reservation
→ High Delay

$T_t = \frac{Length}{BW}$

$T_p = \frac{d}{v}$

velocity

◦ Virtual Circuit	Datagram
	
1) $\underbrace{\text{BW, CPU, Buffer}}_{\text{Reservation}}$	1) NO Reservation
2) Path is same (for all packet)	2) May or May Not
3) Inorder	3) out of order
4) Connection oriented	4) Connection less
5) setup & teardown phase.	5) NO such phase
6) Costly	6) less Costly
7) Reliable ↑↑	7) Less Reliable ↓

Data Link Layer

- → NL↓
- → PL↑
- → Flow control
  - → Stop & wait
  - → Go Back N
  - → Selective Repeat
- → Error Control
  - CRC
  - checksum
- → Framing
- → Physical Addressing (MAC)
- → Access control
  - → CSMA/CD/CA
  - → Aloha
  - → Token Ring

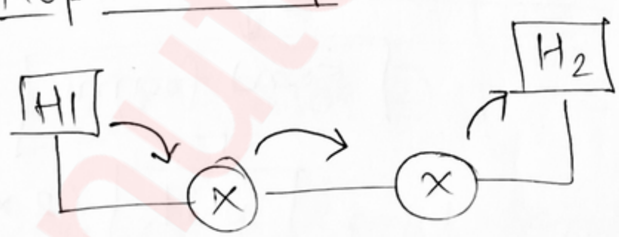
LLC	Logical Link Control
MAC	Media Access Control

Data

↓

Header	Payload	Trailer

- Hop-to-Hop (Node to Node)



MAC

- Access control
- Physical Addressing

LLC

- Flow control
- Error control

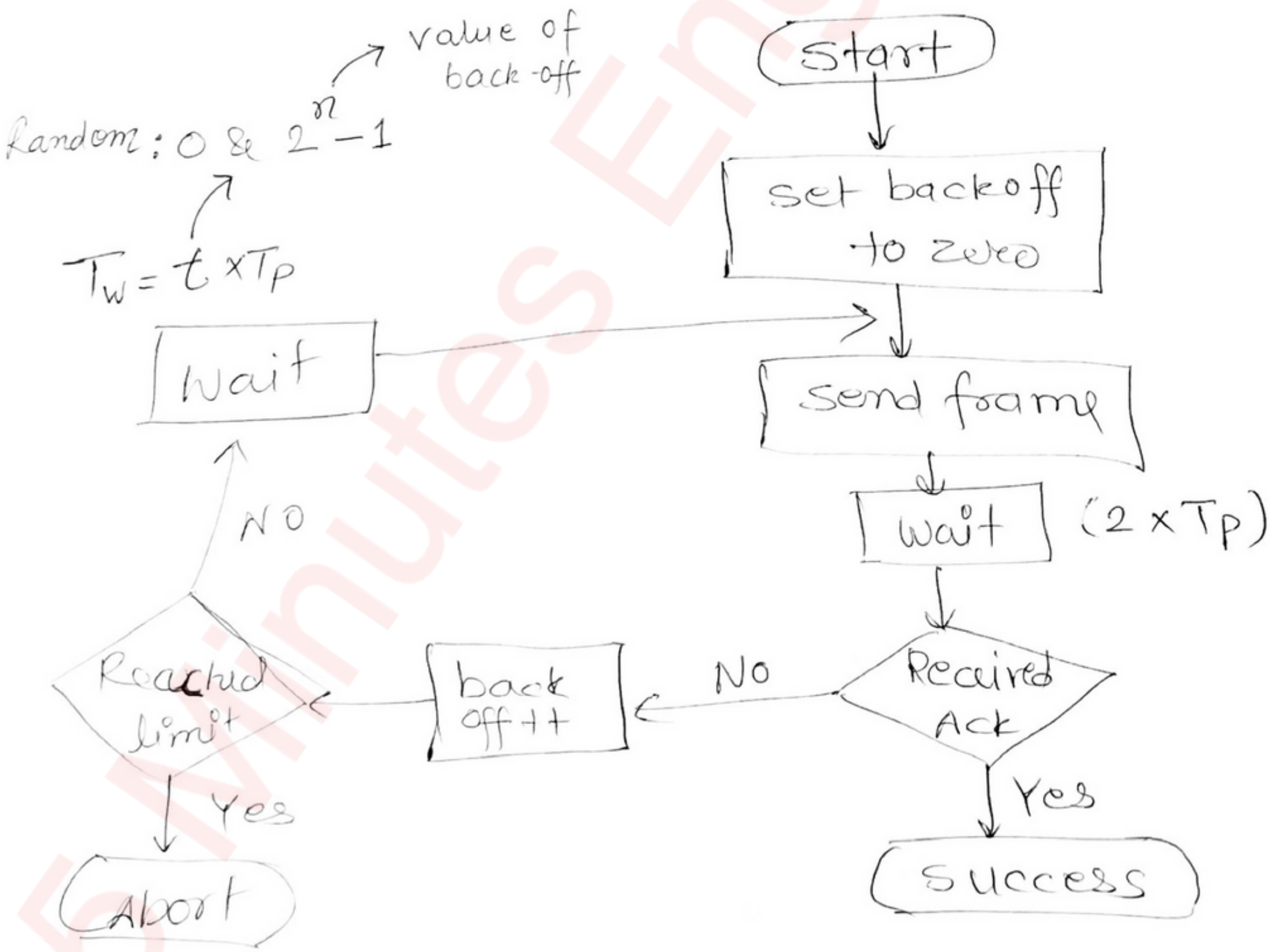
- Framing.

* Access Control

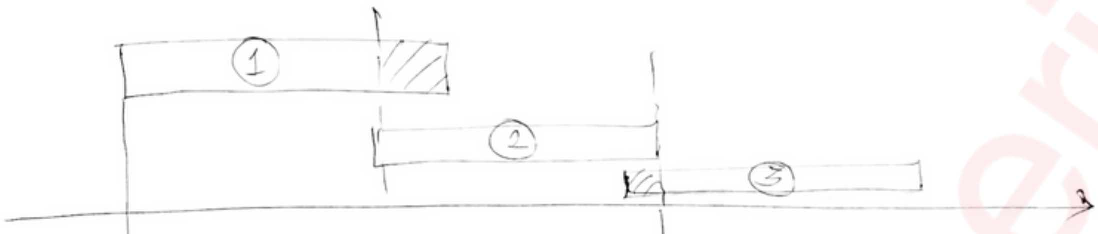
Random Access

- → ALOHA
  - → Pure
  - → Slotted
- → CSMA
- → CSMA/CD
- → CSMA/CA

⇒ ALOHA



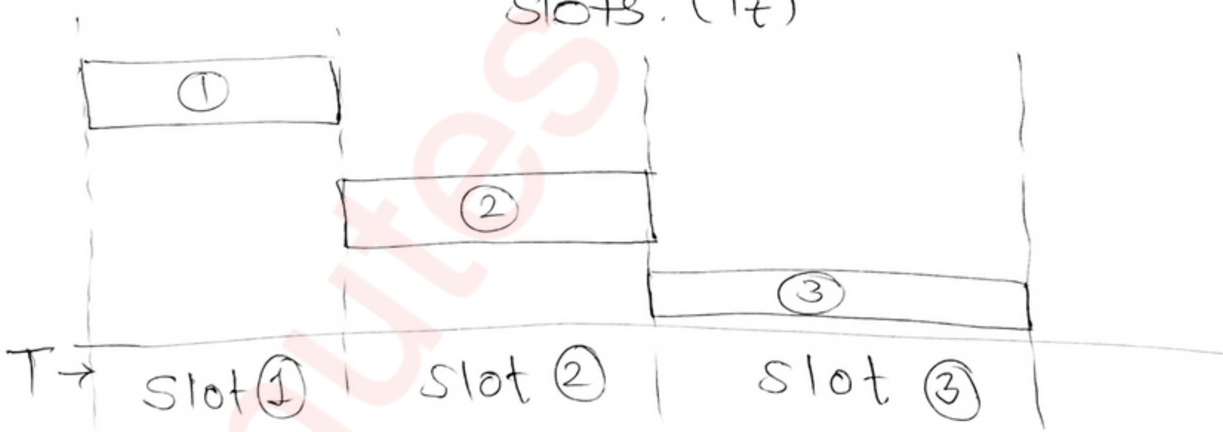
- Vulnerable Time
  - → Possibility of collision



$\leftarrow V_t = 2 \times T_t \rightarrow$

↑
Because of Randomness (18%) efficiency

"But"
→ Slotted ALOHA
→ Divide the time 'T' into slots. (T_t)



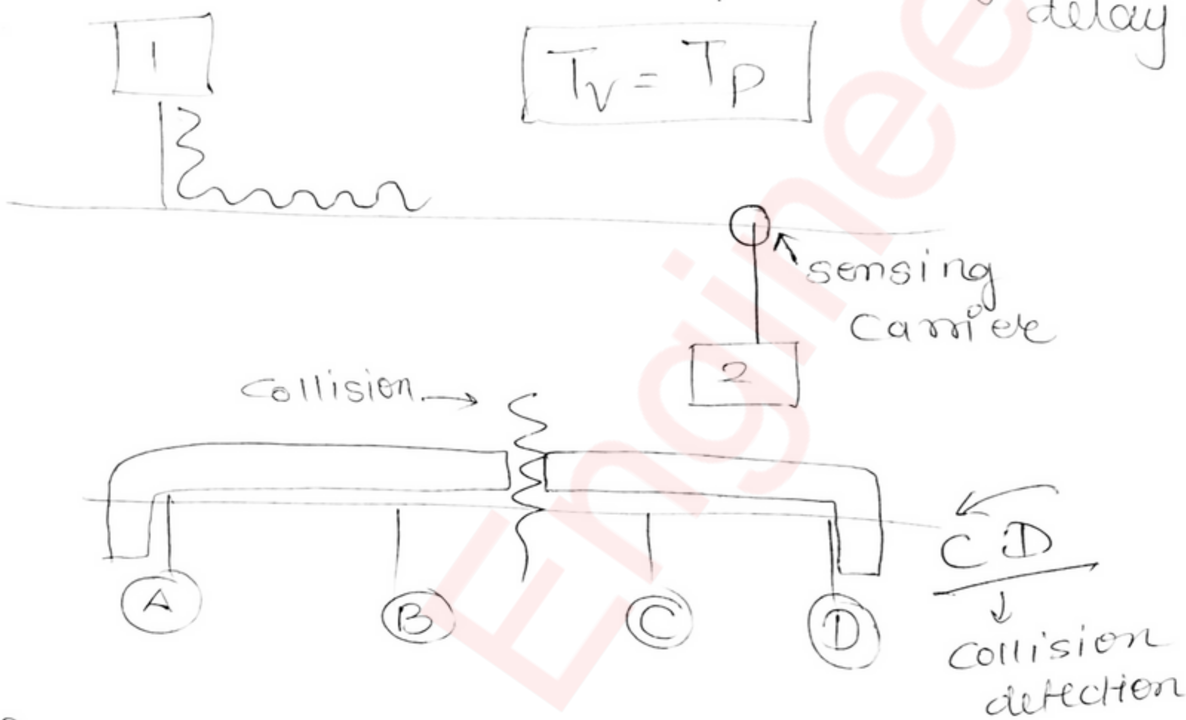
- Station can send only at beginning of slot.

(37%) efficiency.

⇒ CSMA

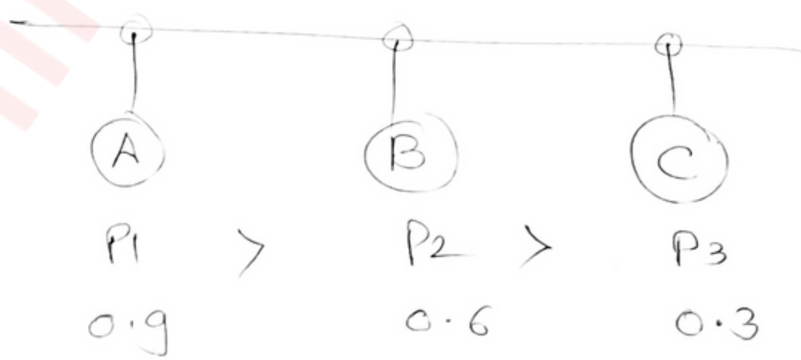
→ Carrier Sense Multiple Access
→ Collision may happen due to "T_p" propagation delay.

$\boxed{T_v = T_p}$



○ Persistence Methods

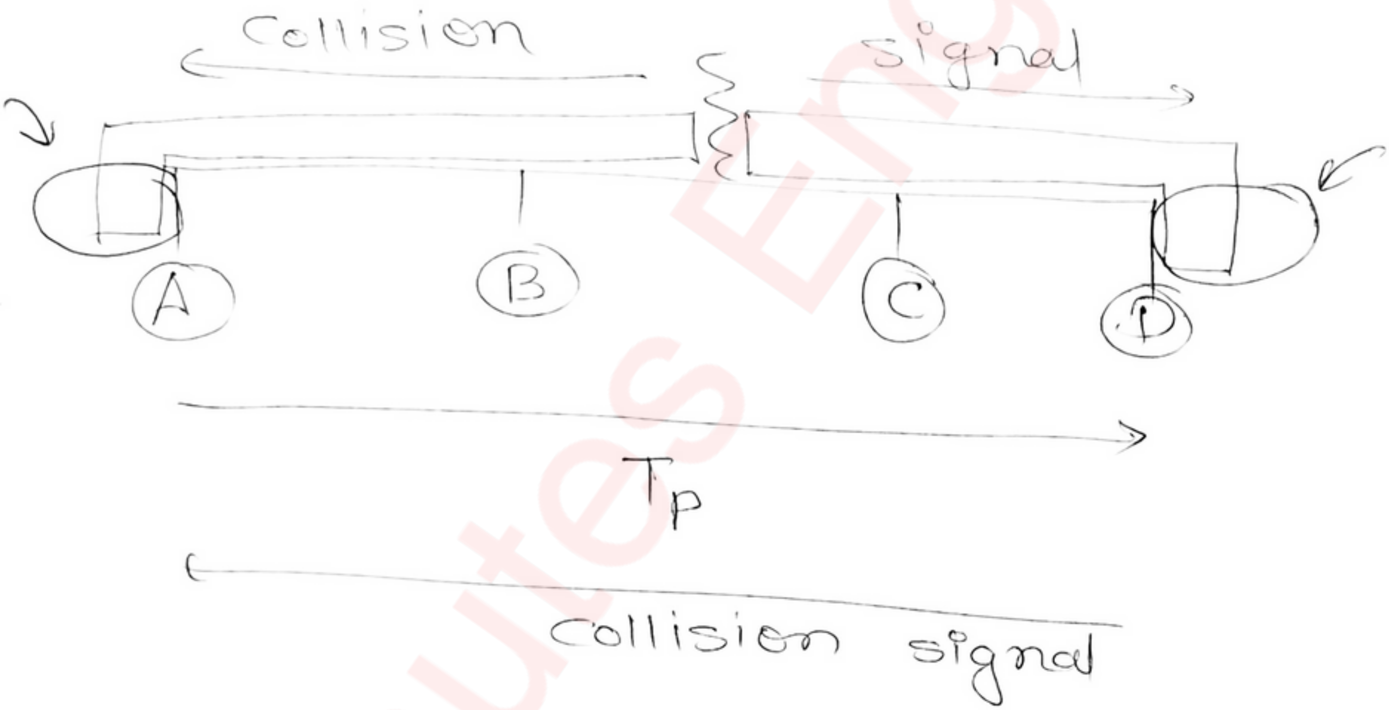
→ 1-persistent (Continuosly sensing)
→ Non-persistent (Random time wait then sense.)
→ P-persistent (Probability)



CSMA/CD (wired)

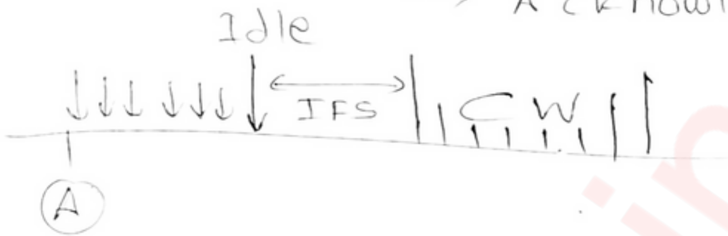
$$\hookrightarrow T_t \geq 2 \times T_p$$

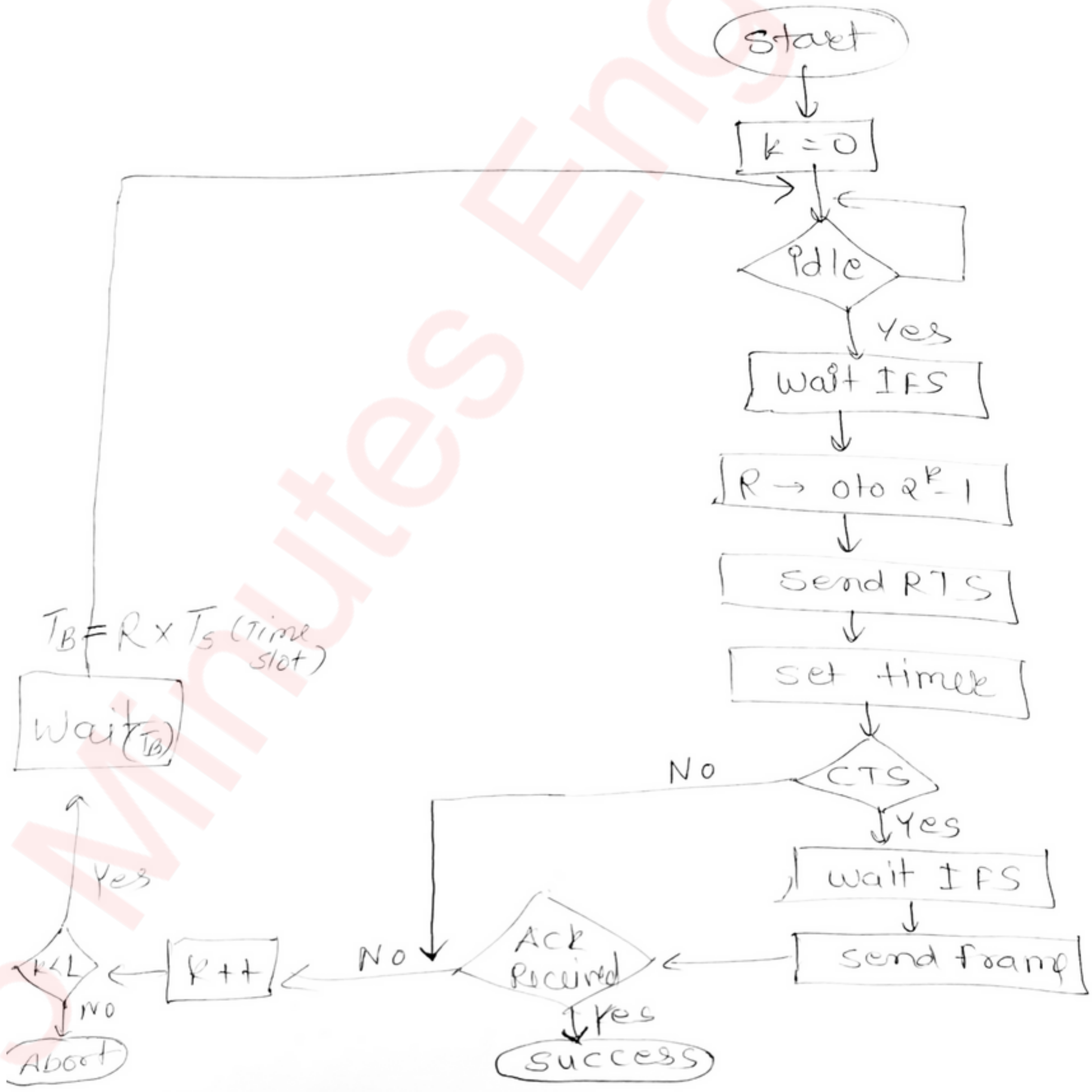
$$\frac{L}{B} \geq 2 \times \frac{D}{V} \quad \Big\} \text{ Imp.}$$



$$\boxed{L \geq 2 \times T_p \times B}$$ Length of packet.

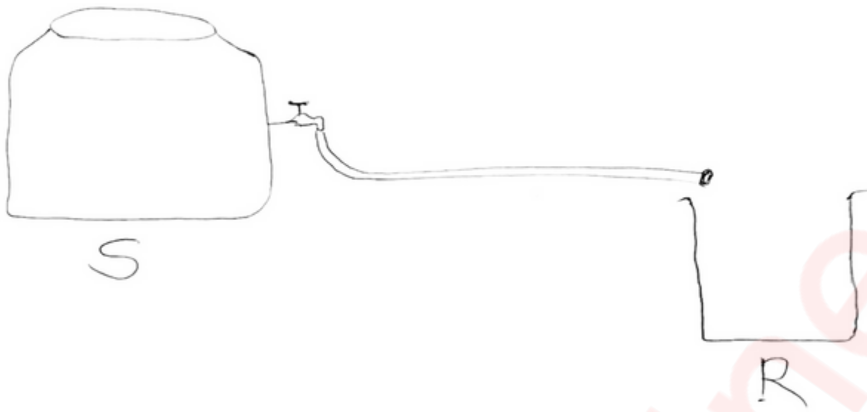
- CSMA/CA (Collision Avoidance) (wireless).
  - Collision may occur.
  - Strategies
    - Interframe space
    - Contention window
    - Acknowledgement.





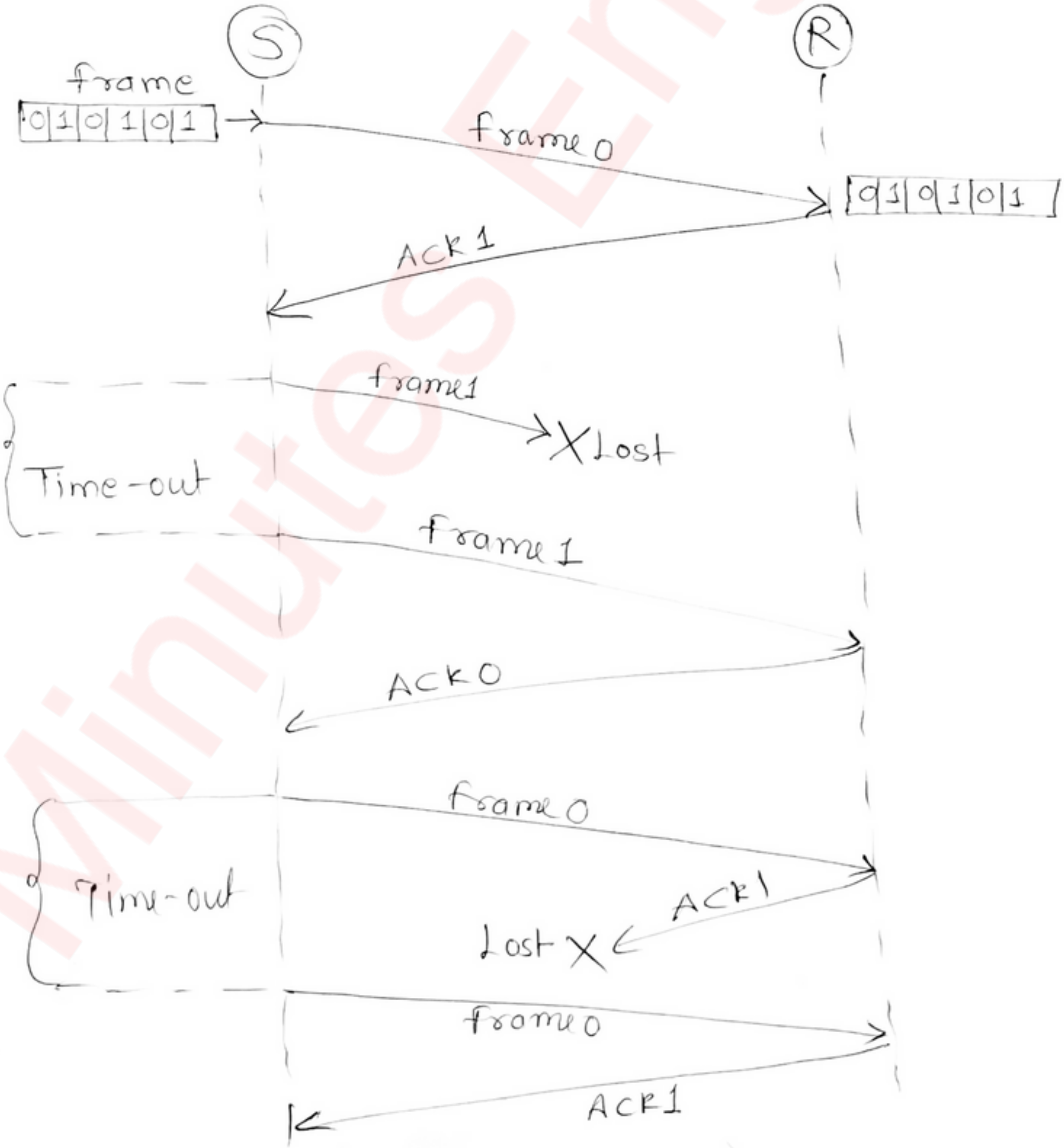
* Flow Control

↳ ① Data Overflow (x)



⇒ Sliding window Protocol

① Stop & wait



Total time (1 packet) $= T_{t_{data}} + T_p + T_q + T_{pro} + T_{t_{ack}} + T_{p_{ack}}$

T_q (Queuing delay)
$\hookrightarrow 0$ $\hookrightarrow$ waiting in I/p or o/p queue of Router

T_{pro} (Processing delay)
$\hookrightarrow 0$ $\hookrightarrow$ Time for processing packet at destination end.

$$TT = T_{t_{data}} + 2T_p + \underbrace{T_{t_{ack}}}_{\text{negligible}}$$

$$\boxed{TT = T_{t_{data}} + 2T_p}$$

In piggy backing $T_{t_{data}} = T_{t_{ack}}$

$$TT = 2T_{t_{data}} + 2T_p$$

$\Rightarrow 2T_p \rightarrow$ RTT (Round trip time).

$$\text{Efficiency}(\eta) = \frac{\text{Useful time}}{\text{Total cycle time}} = \frac{T_t}{T_t + 2T_p} = \frac{1}{1 + 2\frac{T_p}{T_t}}$$

$$= \frac{1}{1+2a}$$

- Troughput: no. of bits we are actually able to send per second

$$S = \frac{L}{T_t + 2T_p} = \frac{(L/B) \times B}{T_t + 2T_p} = \frac{T_t}{T_t + 2T_p} \times B$$

$$= \frac{1}{1 + \frac{2T_p}{T_t}} \times B$$

$$S = \frac{1}{1 + 2a} \times B$$

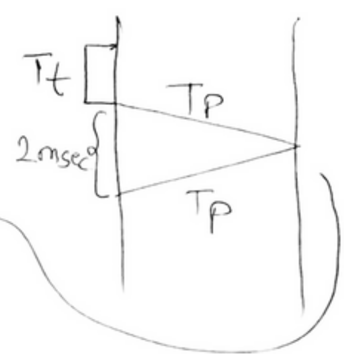
$$\downarrow$$

$$\boxed{S = \eta \times B}$$

<u>Q</u> $T_t = 1\,msec$

$T_p = 1\,msec$

$$\eta = \frac{1}{1 + 2\left(\frac{1}{1}\right)} = \frac{1}{3}$$



$$RTT = 2T_p = 2 \times 1\,ms$$
$$= \underline{2\,msec}$$

- If $\eta \geq 0.5$ then $\frac{T_t}{T_t + 2T_p} \geq \frac{1}{2}$

$\longrightarrow T_t \geq 2T_p$

$$2T_t \geq T_t + 2T_p$$

$$\boxed{T_t \geq 2T_p}$$

$$\frac{L}{B} \geq 2 \times T_p$$

$$\boxed{L \geq 2 \times T_p \times B}$$

If $L = \uparrow$ then 50% 'η'

Q $B = 4$ Mbps
$T_p = 1$ ms
$L = ?$ for $\eta = 0.5$

$L \geq 2 \times T_p \times B$

$L \geq 2 \times 10^{-3} \times 4 \times 10^{6}$

$L \geq 2 \times 10^{3} \times 4$

$L \geq 8 \times 10^{3}$ bits

$S = \eta \times B = 0.5 \times 4 \times 10^{6}$

$$\boxed{S = 2 \times 10^{6} = 2 \text{ Mbps}}$$

- factor affecting 'η'.

$$\eta = \frac{1}{1 + 2\left(\frac{T_p}{T_t}\right)} = \frac{1}{1 + 2\left(\frac{d}{v} \times \frac{B}{L}\right)}$$

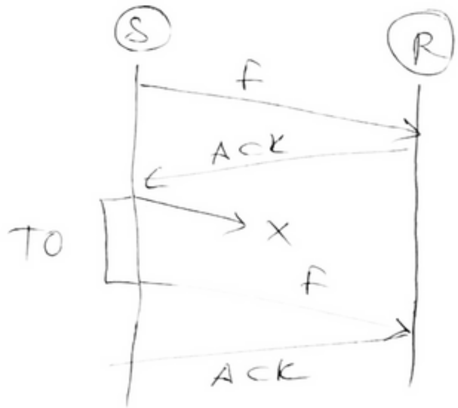
If $d \uparrow$ $\eta \downarrow$

If $L \uparrow$ $\eta \uparrow$

v & B constant as they are property of Link.

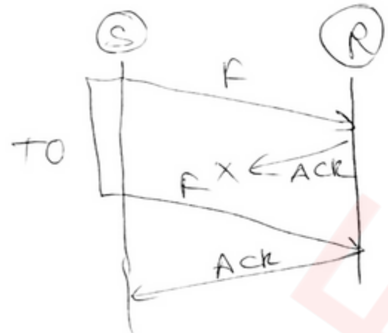
Problem in stop & wait

1) Data Packet lost



S&W + Time out = S&W ARQ
↓
(Automatic Repeat Request)

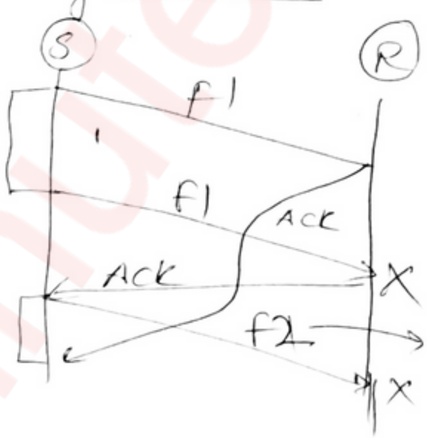
2) Ack lost.



Duplicate packet problem.

Solⁿ: S&W + TO + seq. no. (sequence of data/frame)

3) Delayed Ack



→ missed packet

solⁿ => sequence no. (ACK)

$\boxed{\text{S\&W + TO + Seq(Data) + Seq(Ack)}}$

② Go Back N (ARQ)

① WS: Sender window size (N)

GB10 $\longrightarrow$ $W_S = 10$ $(\underline{\underline{N > 1}})$

If $N = 1$ then it's simply Stop & wait.

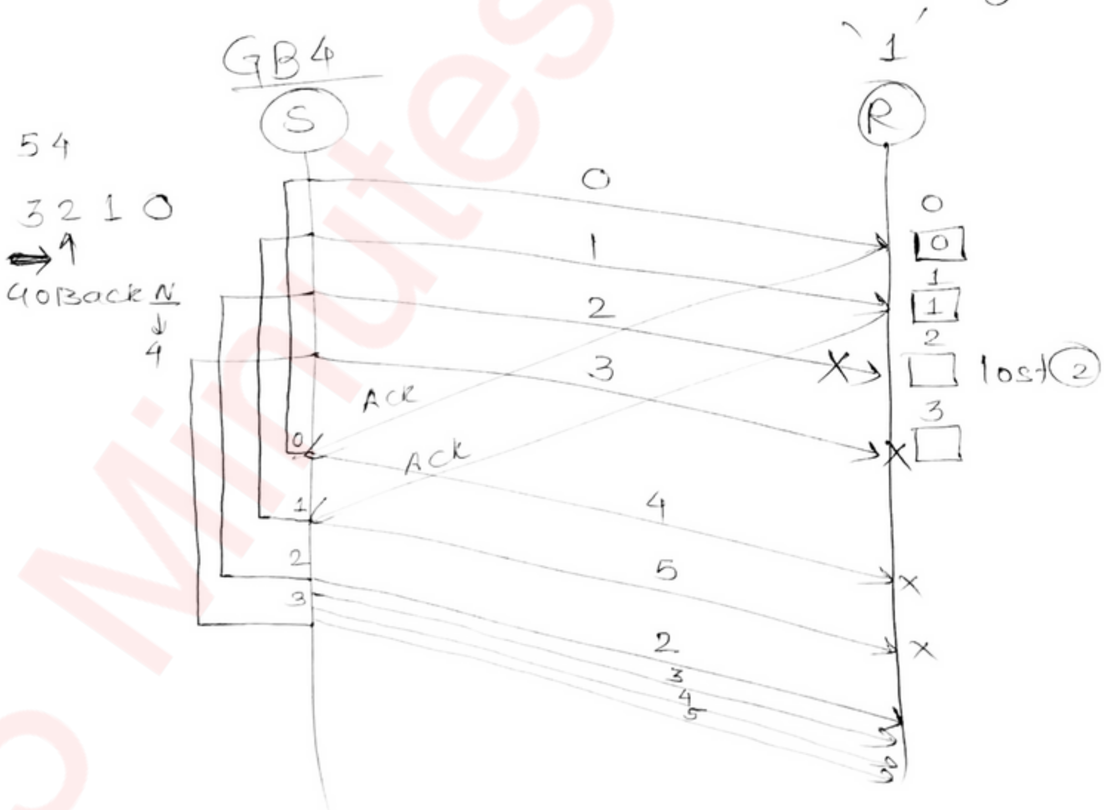
$$\eta = \frac{N}{1+2a} \qquad S\&W = \frac{1}{1+2a}$$

$$a = T_p / T_t$$

$$S = \eta \times B$$

$$S = \frac{NB}{(1+2a)}$$

② WR: Receiver window size is $\underline{'1'}$ always.

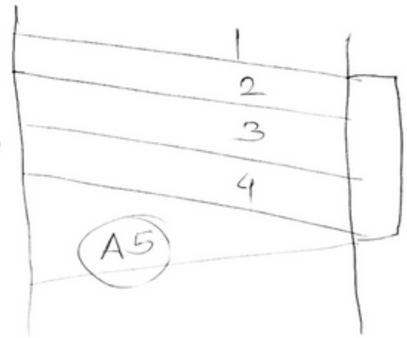


• Acknowledgements types

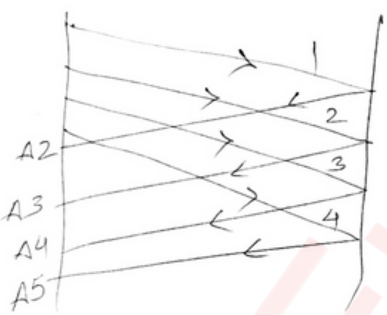
↳ Cumulative Ack

Adv: less Traffic

disadv: Reliability is less



→ Independent Ack



more { reliable, Traffic }

Timeout (TO) Timer > Ack Time

[Not too long, nor too small] → TO expire

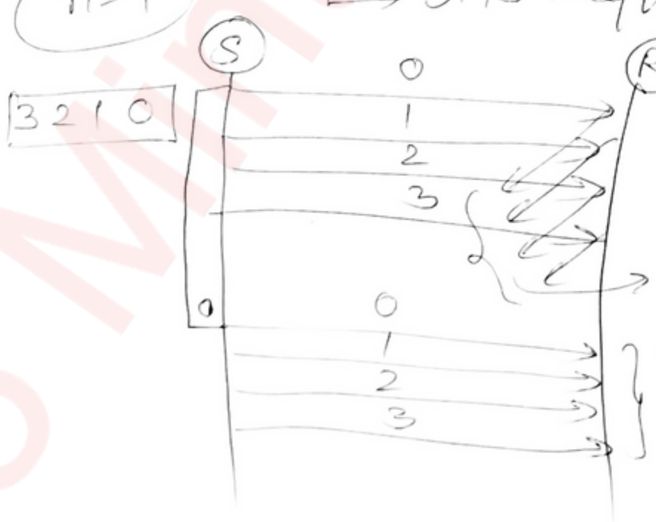
→ Independent Ack

⇒ Sequence Numbers (n)

→ N + 1

→ Bits Required = ceil$\left(\left[\log_2(N+1)\right]\right)$

n = 4



$\boxed{0}$ $\boxed{1}$ $\boxed{2}$ $\boxed{3}$

But for n = 5

i.e N+1

This problem get eliminated.

$\Rightarrow$ W_S	W_R	seq. no. required
N	1	$(N+1)$

$\uparrow$ at least this much is required.

Available sequence no. $\geq (W_S + W_R)$

$\Rightarrow$ Bits $= k$

then seqnos $= 2^k$

then $W_S = 2^k - 1$

$W_R = 1$

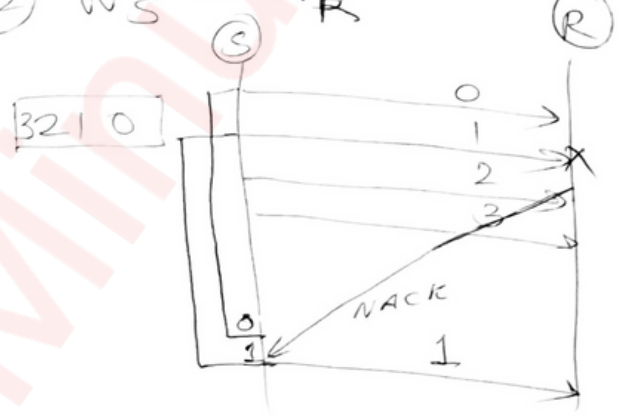
$$2^k \rightarrow 2^k - 1, \; 1$$

③ Selective Repeat

$$2^k \rightarrow \frac{2^k}{2} = 2^{k-1}, \; \frac{2^k}{2} = 2^{k-1}$$

① $W_S > 1$

② $W_S = W_R$



It will only Resend '0', & not the whole window as in GBN Approach.

SR

S&W (Re-trasmission) ← SR → GBN (efficiency)

	S&W	GBN	SR
'η'	$1/1+2a$	$N/1+2a$	$N/1+2a$
Buffer	$1+1=2$	$(N+1)$	$(N+N)=2N$
Seq. nos	$1+1$	$(N+1)$	$(N+N)$
Re-transmission (If 1 Packet lost)	1	N	1
ACK	Independent	Cummulative	Independent
Implementation	Easy	medium/ moderate	Complex/ difficult

- Piggybacking

$$S \xrightarrow{\text{Data}} R$$

$$S \xleftarrow{\text{Data + Ack}} R$$

3 Error Control

S ⟶ R

| 0 | 1 | 0 | 1 | 0 |

| 0 | 0 | 1 | 0 | 0 |

corrupted / modified / altered / changed / manipulated.

- 1 bit (single bit). [less common]
- Burst error. (2 or more bits) ✓ normal / common.

1 Hamming Distance

↳ Counting the no. of dissimilar bits of given 2 nos.

eg:

```
     1 0 1 0 1
XOR  1 1 0 1 0
     ---------
     0 1 1 1 1   -> 4 ✓
```

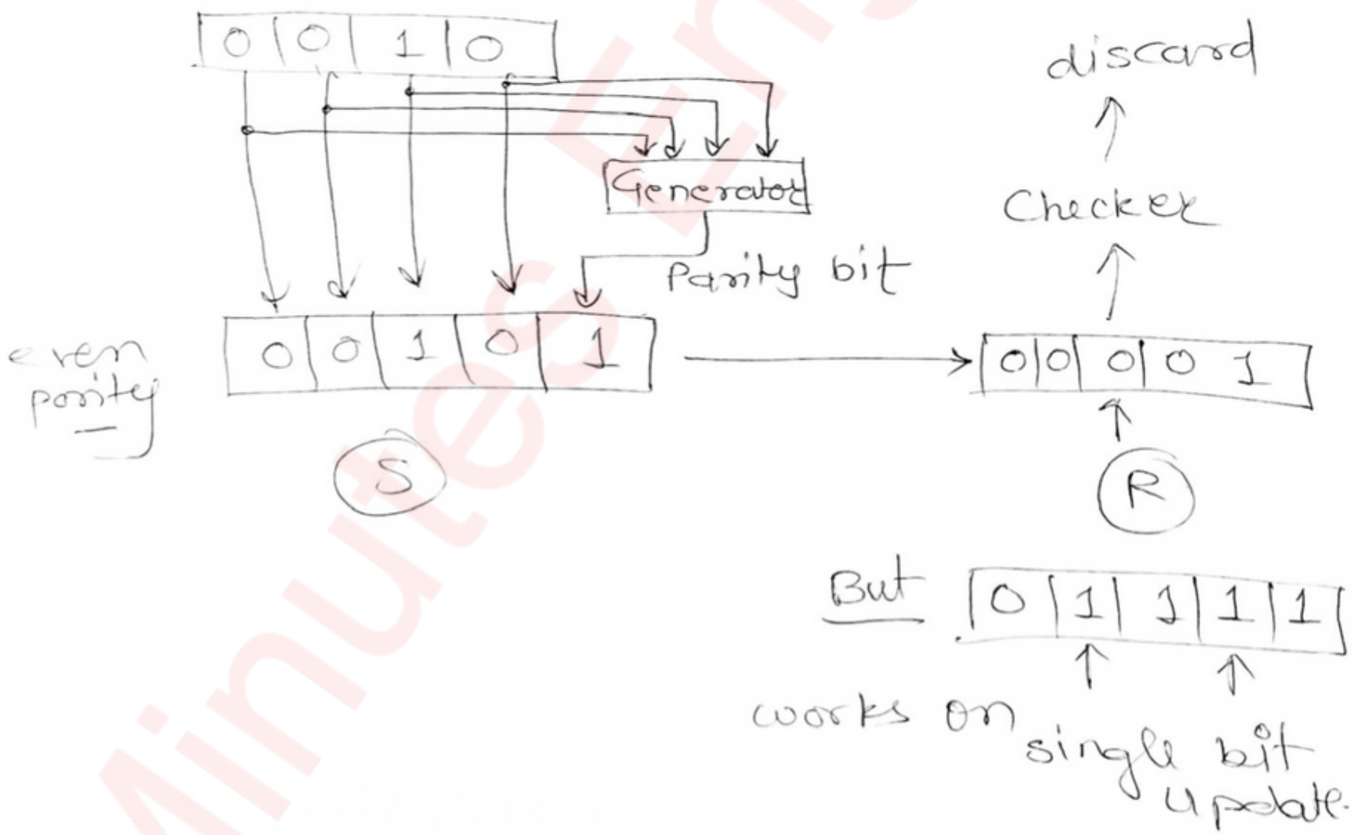
0	0	0
0	1	1
1	0	1
1	1	0

Detection of Error

- simple Parity check
- 2D Parity check
- checksum
- CRC

⇒ simple (single bit Parity).

- odd
- even.



⇒ 2D Parity check

⇒ 101, 010, 111

$$\begin{array}{ccc|c} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ \hline 0 & 0 & 0 & \end{array}$$

← Row parities

Column parities

1/single bit error detect & correct.

Error detection till '3bits'

>3 bit error → nor detect, neither correct.

- Checksum (check the sum)

eg:- | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | C.S. |

8 2 4 5 2 6 4 5

10
14
19
21
27
31
36

36 $\xrightarrow{\text{1's complement}}$ −36

If we get 5 bit no. then ?

(36)
↓
1 0 0 1 0 0
→ 1 0
0 1 1 0 (6)
$\boxed{1\ 0\ 0\ 1}$ (9) 1's complement

(R) end: $36 + 9 = 45$

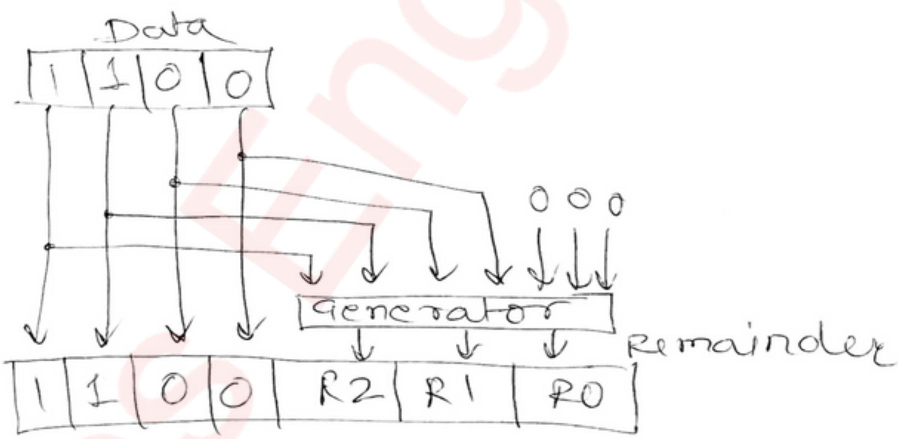
$$\begin{array}{r} 101101 \\ +\ 10 \\ \hline 1111 \\ 0000 \\ \hline \end{array}$$

1111 (15) → 1s complement → 0000

✓ valid / No error.

=> CRC Cyclic Redundancy Check.

eg:



Divisor : 1 0 1

$x^3 + x + 1$

↓ ↓ ↓ ↓

1 0 1 1

x^2

(4−1) no. of zeros

$$\begin{array}{r l} 1011 \overline{)\,1100\,000} & \\ \text{XOR}\quad 1011 & \\ \hline 0111000 & \\ 1011 & \\ \hline 010100 & \\ 1011 & \\ \hline 000010 & \end{array}$$

$R_2\ R_1\ R_0 \Rightarrow \boxed{1100010}$

At (R) side Checker ↓

```
1011 ) 1100010
       1011
       -------
       0111010
        1011
        ------
        010110
         1011
         -----
         00000   ← all zero.
```

* Ethernet

→ Topology [BUS]

→ Access control Method [CSMA/C.D]

→ NO Ack

→ Encoding Technique (Manchester)

→ Data Rates: 10 Mbps — Normal
100 Mbps — fast
1 Gbps — Gigabits

→ Real Time Appl^n (X)
→ Interactive Appl^n (X)
} low performanc

Ethernet frame format

Preamble	SFD	DA	SA	Length	Data	CRC
7 byte	1 byte	6	6	2		4
PL Header		DL Header				Tail

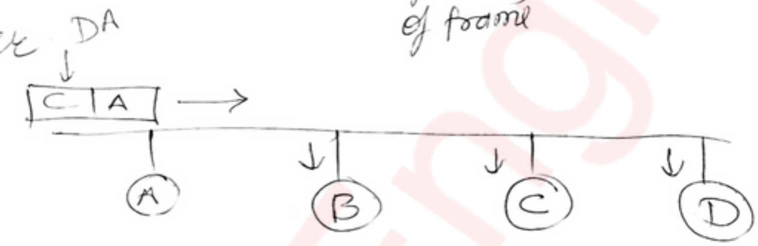
Preamble : 101010 - - 10

SFD : 10101011

(Start of frame) delimiter

beginning of frame

MAC addr (DA, SA)



only valid Receive Reads it

	Min	Max
Data	64-18 = 46 B	1500 B
frame	64B	1518 B

* MAC Address :

NIC

↓

ROM (MAC addr Hard coded)

A1 : 2B : C4 : D2 : 89 : A5

A1 - 2B - C4 - D2 - 89 - A5

Format (: , -)

$\underbrace{A1}_{1B}$ - $\underbrace{2B}_{1B}$ - $\underbrace{C4}_{1B}$ - $\underbrace{D2}_{1B}$ - $\underbrace{89}_{1B}$ - $\underbrace{A5}_{1B}$ = 6 B = 48 bit

• Types:

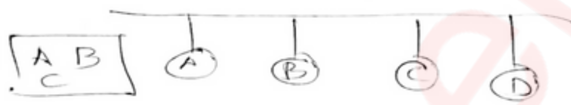
→ Unicast (one-to-one)

LS: 1 byte → '0' A2 1010001 0

LSB

→ Multicast (one-to-many)

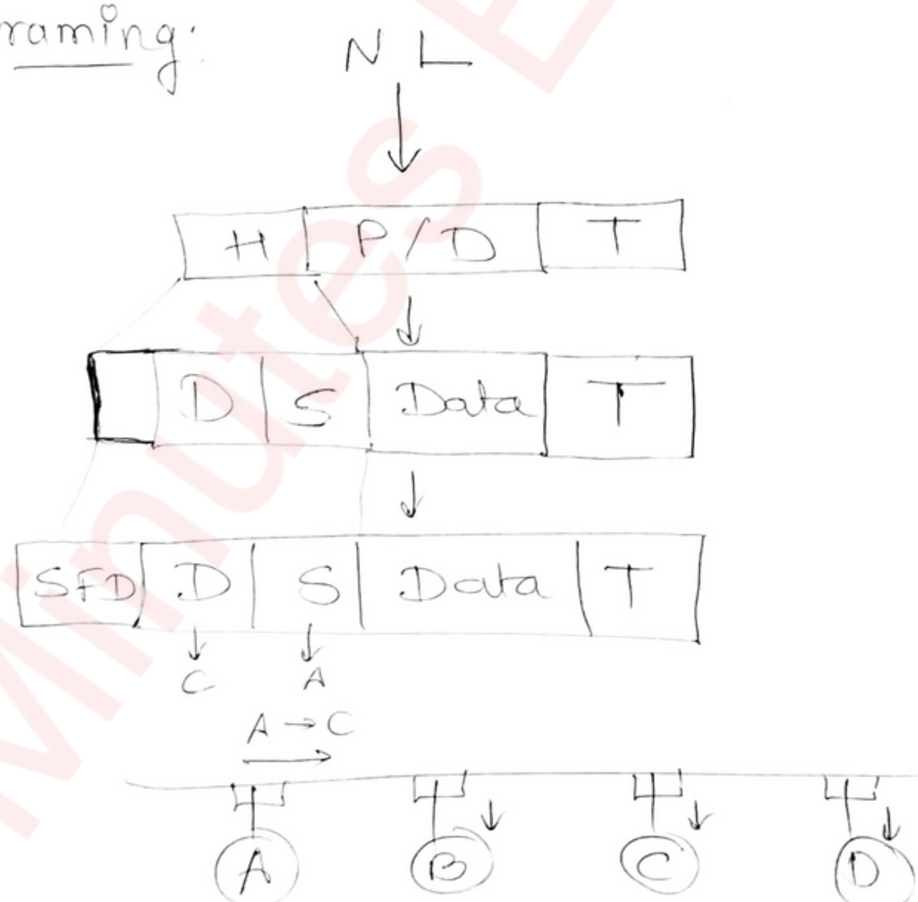
LSB 1 byte → '1' Group ID



→ Broadcast (one-to-all)

FF:FF:FF:FF:FF:FF

* framing:



Frame length

- Fixed (100 B)
  - [20B Data + 80 B padding]
- Variable (10 B, 100 B, 1000 B)
  - End of frame
    - Length field (Ethernet)
    - (Token Ring) End Delimiter
      - Character Stuffing
      - Bit Stuffing

H	$\0	$

↓

H	$ \0 \0\0	$

'\0' → null character

10000

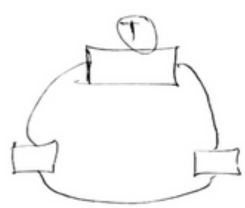
stuff the bit here.

1000(1)0
1000(1)1
(Data)

ED = 10000

* Token Ring

- → Topology [Ring]



- → Access control method [Token Passing]
- → Data flow : Unidirectional.
- → Ack - Piggybacking
- → Encoding : Differential Manchester
- → Data Rates : 4Mbps, 16 Mbps

Token Ring frame format

SD	AC	FC	DA	SA	Data	CRC	ED	FS
1	1	1	6	6	(0 to max) " ≥ 0 "	4	1	1

Token:

SD	AC	ED

SD →

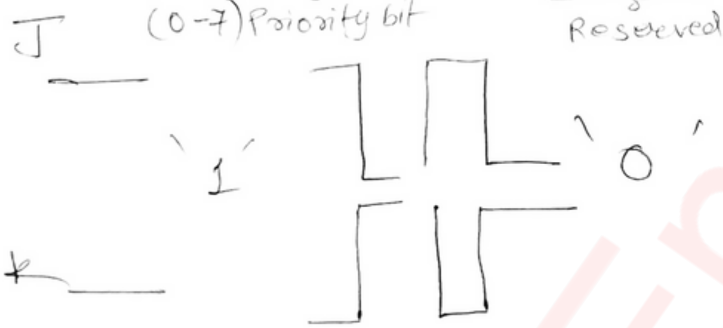
J	K	0	0	J	K	0	0

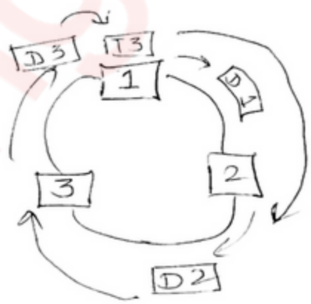
AC →

P	P	P	T	M	R	R	R

(0-7) Priority bit — Reserved.

T : Token (1) & T = 0 Not Token
M : Monitor bit





FC :

00 → data frame
11 → control frame
(AMP, Becon
[Alive Monitor Packet]

ED :

J	K	1	1	J	K	I	E

Info bit — Error bit.

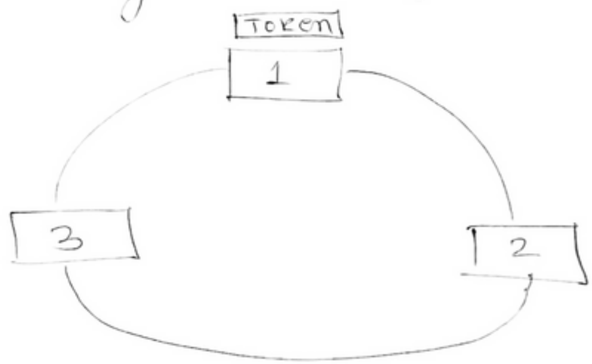
I = 1 [more data coming]
I = 0 [No more data.

FS :

A	C	0	0	A	C	0	0

Frame status — Availability (Destination Available or not) — Copy

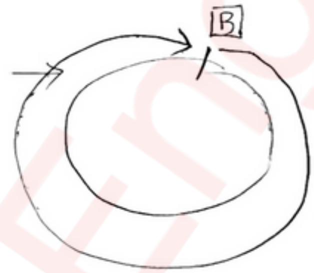
- Token Passing
  - → Ring Topology



⇒ hold the token the transmit the data

⇒ unidirection.

⇒ NO collision

⇒ Ring Latency → 

$$\Rightarrow \frac{d}{v} + N \times b$$

length (d), velocity (v), no. of station (N)

b: Time taken by each station to hold the bit before Trasmitting.

⇒ LAN

⇒ Token Holding Time (THT) [10msec] default

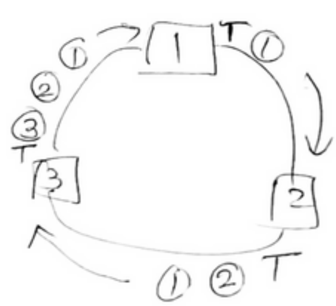
⇒ Delayed Token Reinsertion | Early Token Reinsertion

Delayed Token Reinsertion

- $T_t + RL$
- Transmit the data on link, then it takes a Round trip of ring & return to station, then token is released.
- Only one packet at a time in Ring.

Early Token Reinsertion

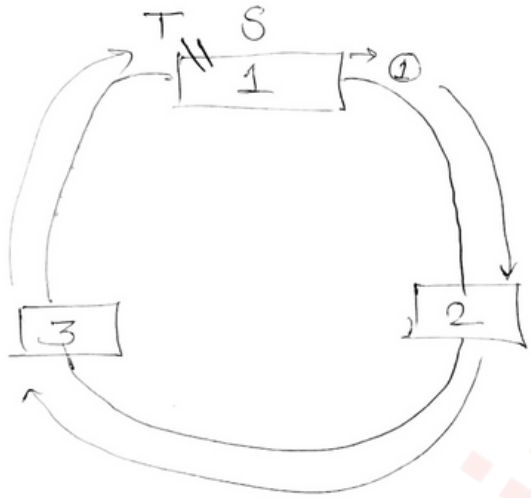
- T_t
- NO Round trip



- More than 1 packet at a time in Ring

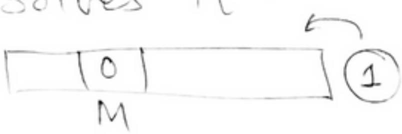
Token Ring Problems

(1) Source Related



"Orphan" packet as source is down

- Now monitor station solves it.



1st time M = 1
2nd time → monitor discards the packet.

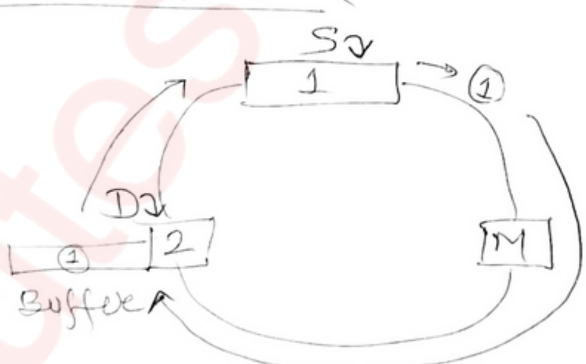
" "Stray" packet gets too much corrupt" so, station 1 doesn't Identify it & doesn't Remove it

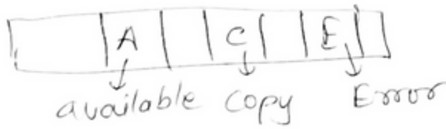
- Monitor station solves it using "CRC"

" CRC mismatch" so packet is discarded.

(2) Destination Related

- Down
- Busy (Buffer full)
- Copied



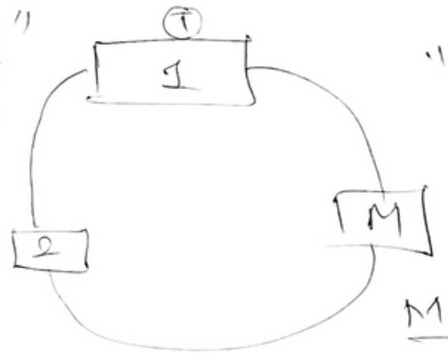


	available	copy	Error
Initially	0	0	0
Down	0	0	0
Busy	1	0	0
Error	1	0	1
Copied	1	1	0

③ Token Related

"Captured Token"
(Monopolisation)
[Injustice]

- Max THT
  → 10 msec



"Token lost"

- Station ① holding Token ① gets down

Min TRT : RL

Max TRT : RL + THT × N

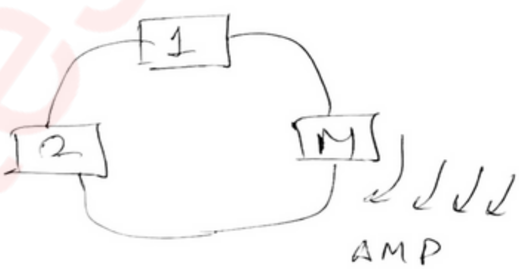
- Monitor regenerate Token after Max TRT

"Token corrupt"
(3 Byte Packet)

Monitor takes care of it
⇒ Its considered as Disturbance / Noise in Ring.

④ Monitor Related

"Monitor Down"
(Heart beat Signal)
↓
AMP
(Alive Monitor Packet)

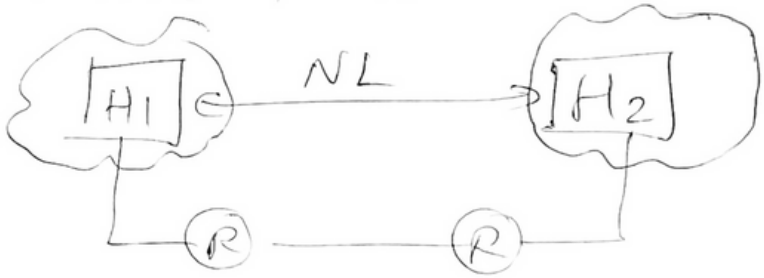


Sol^n: Polling (Make/elect a station as monitor)

"Monitor Malfunction" (Chacked).
Sol^n: Human intervention".

Network Layer

→ Host-to-Host

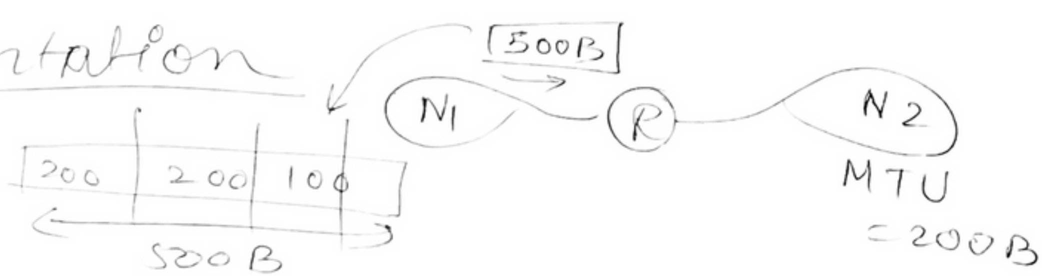


(Source to destination) delivery

- (Logical Address)
  IP Address
  → Network
  → Host

- Routing (Path, where next to send packet).
  best Route
  → Intra domain
    → RIP (Distance vector)
    → OSPF (Link state)

- Fragmentation



- IP address Vs Mac Address

IP address	Mac Address
→ Internet Protocol	→ Media Access Control
→ 4 Byte IPV4 6 Byte IPV6	→ 6 Byte hexadecimal address.
→ ISP provides it (Internet Service Provider)	→ NIC (N/w Interface card) manufacturer provides it.
→ Logical address	→ Physical address
→ Operates on NL	→ Operates on DLL
→ Identify the connection of device on n/w	→ Identify the device
→ S/w oriented	→ H/w oriented.
→ Can be changed	→ can't be changed.

- Notation.
  - → Binary Notation.

    [00000000 00000000 00000000 00000000]
  - → Dotted-Decimal Notation.

    [172 . 16 . 254 . 2]

* Classfull Addressing

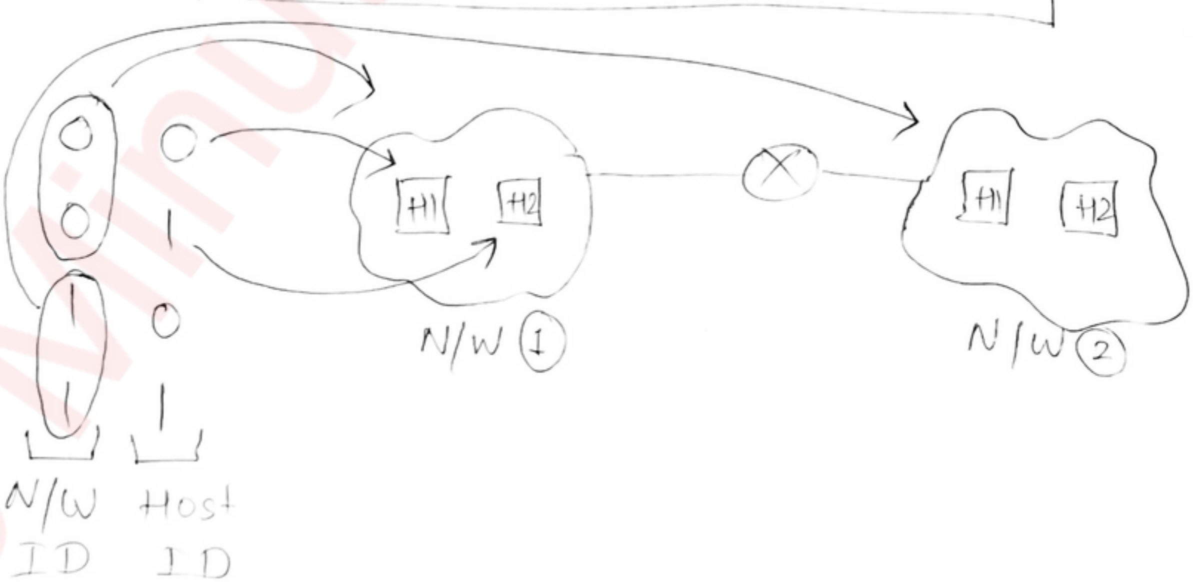
→ 32 bit / 4 Byte / IPV4 address

- → Class A
- → Class B
- → Class C
- → Class D
- → Class E

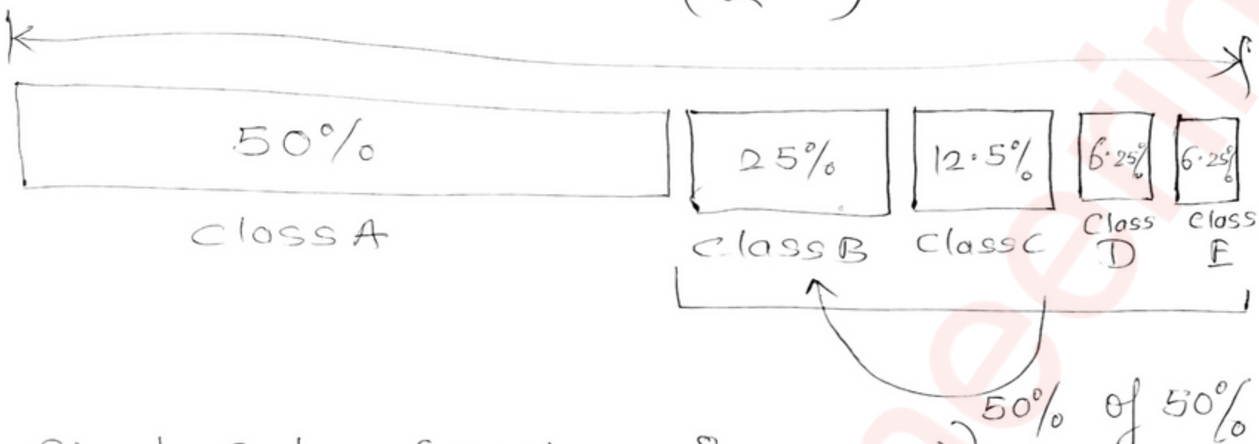
	|← B1 →|	|← B2 →|← B3 →|← B4 →|
Class A :	N/W ID	HOST ID

Class B :	N/W ID	HOSID
Class C :	N/W ID	HOST ID

Class D :	MULTICAST Address
Class E :	Reserved.



Address Space : 4,294,967,296 Addresses
(2^{32})



First Byte (8 bit $\rightarrow 2^8 (0 - 255)$)

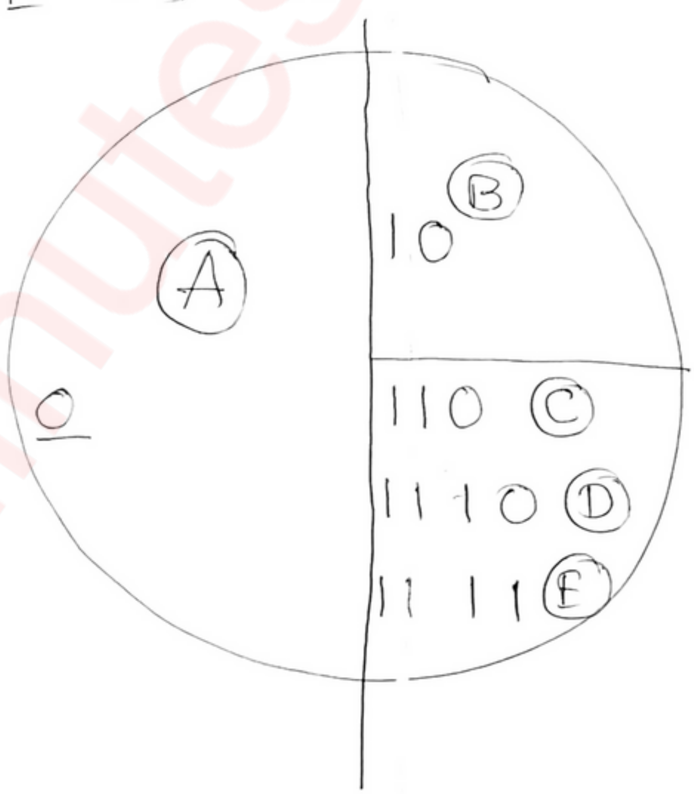
Class A $\longrightarrow$ 0 to 127 [50% of 256]

Class B $\longrightarrow$ 128 to 191 [50% of 128]

Class C $\longrightarrow$ 192 to 223 [50% of 64]

Class D $\longrightarrow$ 224 to 239 [50% of 32]

Class E $\longrightarrow$ 240 to 255 [50% of 32]



- Class A

0 _ _ _ _ _ _ _

0 0 0 0 0 0 0 → 0 × → Broadcasting

0 1 → 1

1 0 → 2

$\underbrace{1\ 1\ 1\ 1\ 1\ 1\ 1}_{\text{7 bits.}} \rightarrow 127 \times$ self connectivity (127.0.0.1)

$2^7 = 128$ n/w of class A.

$2^{24} = 16M$ IP addr in class A

$[0, 127] \Longrightarrow [1, 126]$

$[1, 126] \rightarrow 2^7 - 2$

$\cancel{0}$ ——— $\cancel{127}$

1 ——— 126

- $2^{24} - 2$ [1, 67, 77, 214] Host IDs. in class A.

$2^{24} - 2$ → 000 - - - 0 (all zeros) n/w ID

$2^{24} - 2$ → 1111 - - - 1 (all ones) Broadcasting address.

- ## Class B

$$\underbrace{\boxed{10 \qquad\qquad}}_{16\,\text{bit}}\underbrace{\boxed{\text{Host ID}}}_{16\,\text{bit}}$$

$2^{14} = 16k$ n/w of Class B

2^{16} = IP addr in class B

10 _ _ _ _ _ _

0 0 0 0 0 0 → 128

1 1 1 1 1 1 → 191

$\left.\begin{matrix} 128 \\ \vdots \\ 191 \end{matrix}\right\} \underline{64}$

Range of class B [128 → 191]

→ N/W ID

$2^{16} - 2$ → 65k Host ID

→ Broadcast address

- ## Class C

$$\underbrace{\boxed{110 \qquad\qquad}}_{24\,\text{bit}}\underbrace{\boxed{\text{HOST ID}}}_{8\,\text{bit}}$$

2^{21} n/w of class C [20,97,152]

$2^{8} = 256$ IP addr.

1 1 0 _ _ _ _ _

0 0 0 0 0 → 192

1 1 1 1 1 → 223

} 32

Range [192, 223]

- <u>Class D</u>

1 1 1 0	

NO N/W ID & HOST ID
same for class E.

1 1 1 0 _ _ _ _

0 0 0 0 → 224

1 1 1 1 → 239

} 16

Range [224, 239]

- <u>Class E</u>

Range [240, 255]

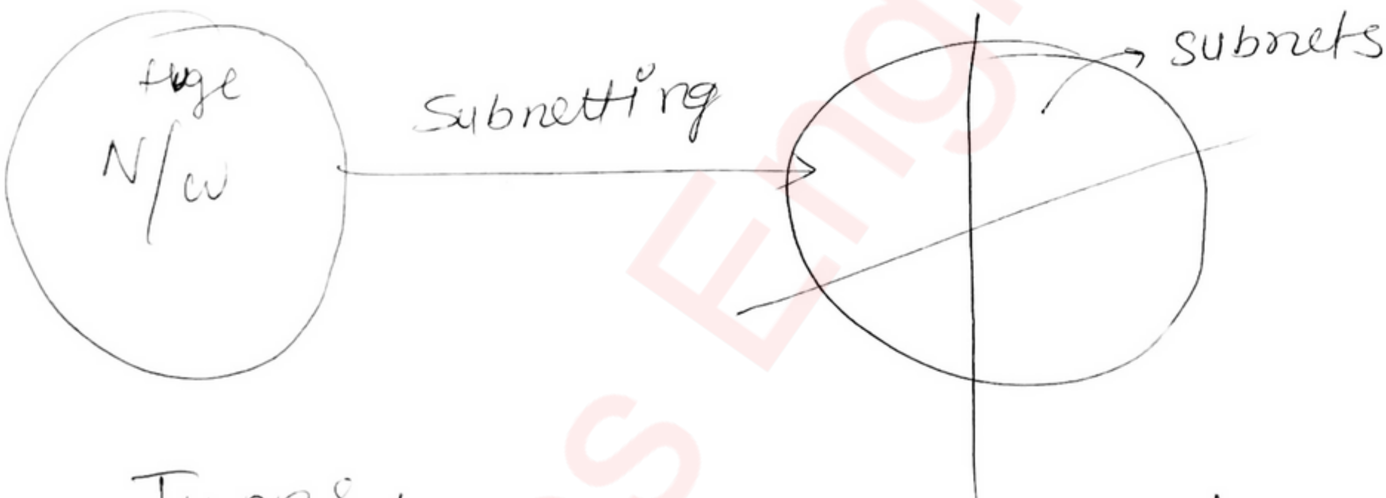
Reserved 1 1 1 1 0 0 0 0 → 240

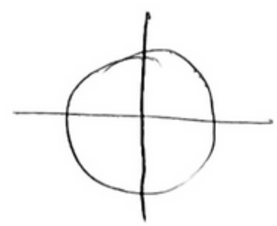
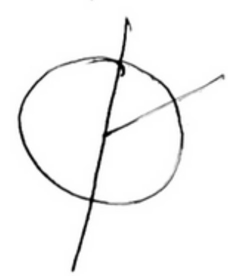
1 1 1 1 → 255

} 16.

- Subnetting
  - → Dividing a huge n/w into smaller ones
    (Class A & B) huge n/w.

Smaller n/ws (subnets).



Types:
- → fixed length 
- → Variable length 

- Fixed length Subnetting
  - ↳ equal no. of addresses.

Eg: $\underbrace{201.10.20.}_{NID}$ _ _ _ _ _ _ _ _

0 _ _ _ _ _ _ _
0 0 0 0 0 0 0 → 0
1 1 1 1 1 1 1 → 127

1 _ _ _ _ _ _ _
128: 0 0 0 0 0 0 0
255: 1 1 1 1 1 1 1

(201.10.20.0 → subnet ID
to
201.10.20.127) → Broadcast addr

⇒ 128 − 2
⇒ 126 HOST ID

(201.10.20.128 to
201.10.20.255)

⇒ 128 − 2
⇒ 126 HOST ID.

Subnet Mask

255.255.255.128

10000000

"ANDing" with Mask

201.10.20.0

NID

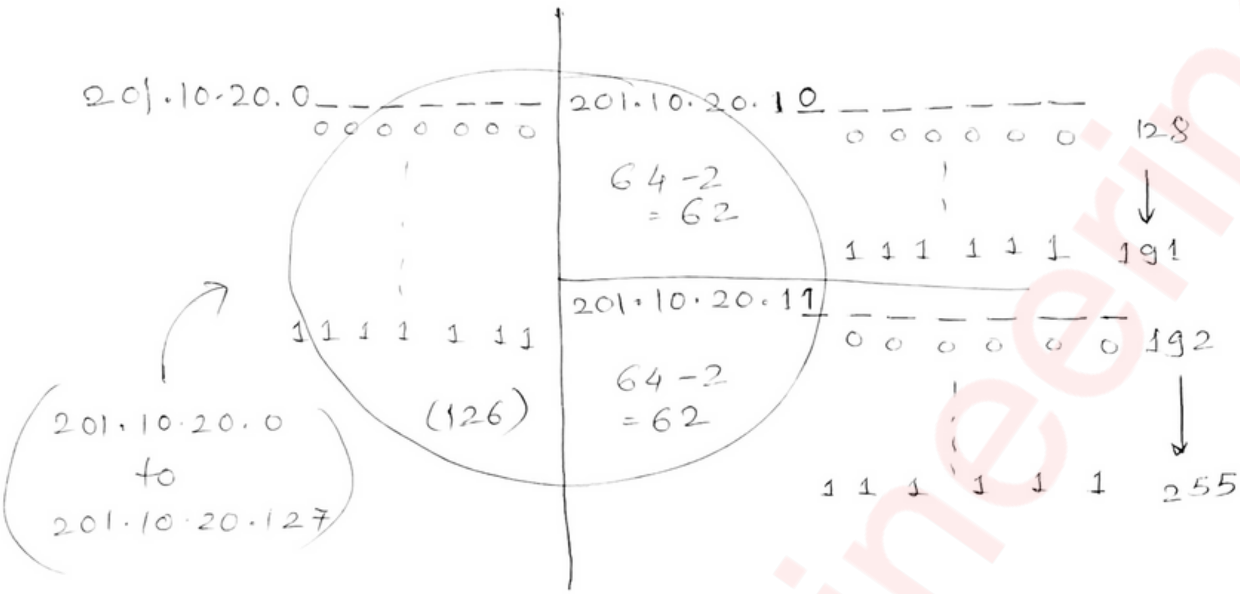
201.10.20.255

N/W Broadcast addr

Mask Class ©

255.255.255.0

Variable length subnetting



128 − 2
= 126

201.10.20.0 (NID)

Subnet Mask ⇒ 255.255.255.128 (A or B)

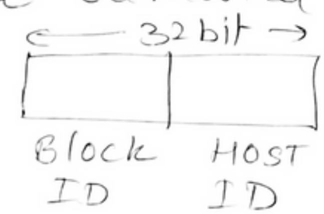
255.255.255.192 (In B)

B1 or B2

* Classless Addressing (CIDR)

→ NO classes, Blocks (✓)

→ As per demand → supply/provide



Notation

→ a.b.c.d / m → no. of bits Represent block/NID ID.

→ 201.10.20.5/28 → 28 1's (NID)

$\longleftarrow$ 32 bit $\longrightarrow$

28bit	4bit

$\therefore 2^4 = \underline{16}$ Host IDs

$$\underbrace{\underbrace{11111111}_{255} . \underbrace{11111111}_{255} . \underbrace{11111111}_{255} . \underbrace{11110000}_{240}}_{\text{Mask}}$$

for NID

IP address

201 . 10 . 20 . 5 / 28

$\Downarrow \quad \Downarrow \quad \Downarrow \quad \Downarrow$

$\underbrace{201 . 10 . 20 . 0000}\ \underbrace{0101}_{\downarrow\ 0000}$

201 . 10 . 20 . 0/28 (NID)

OR

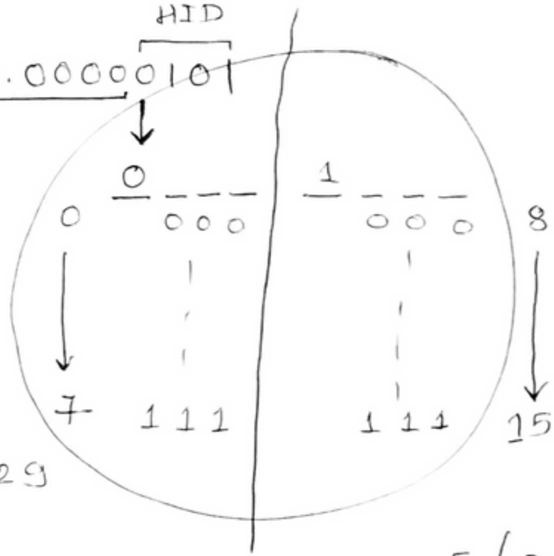
$$\begin{array}{rl} & 201 . 10 . 20 . 5 \\ \text{AND} & 255 . 255 . 255 . 240 \\ \hline & 201 . 10 . 20 . 0 \end{array}$$

Subnetting in CIDR

HID

201.10.20.0000 0101

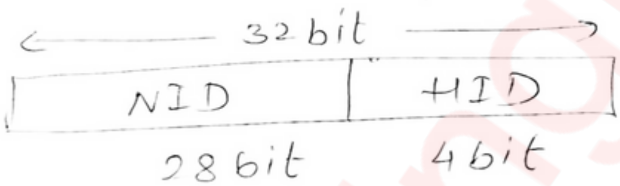
NID



201.10.20.8/29
to
201.10.20.15/29

201.10.20.0/29
to
201.10.20.7/29

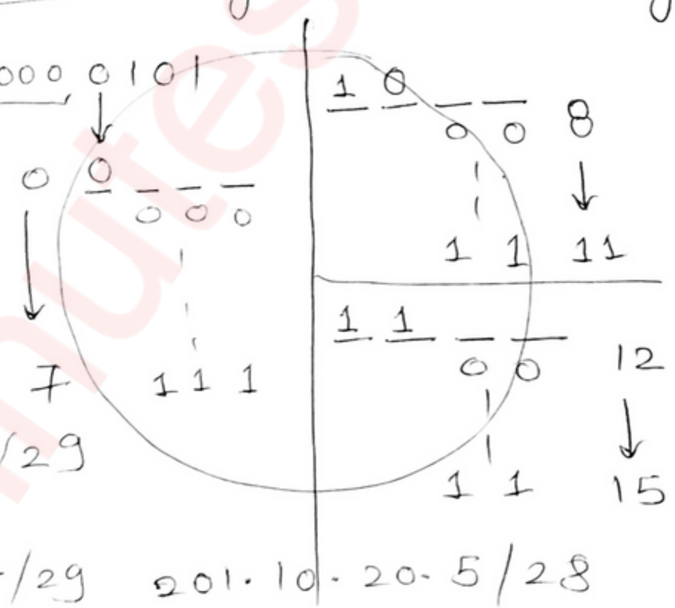
201.10.20.5/28



→ Variable length subnetting in CIDR

201.10.20.0000 0101

NID



201.10.20.8/30
to
201.10.20.11/30

201.10.20.12/30
to
201.10.20.15/30

201.10.20.0/29
to
201.10.20.7/29

201.10.20.5/28

IPV4 Header

	0	4	8	16	31
4B	VER	HLEN	Service type	Total length	
4B	Identification			Flags (3 bits)	Fragmentation offset
4B	Time to live (TTL)		Protocol	Header checksum	
4B	Source IP address				
4B	Destination IP address				
20B	Options + padding (0 to 40 Bytes)				

Header	Payload
(20 - 60 Bytes)	(0 - 65515 Bytes)

Total length (Datagram = 20 + 65515 = 65535 = 2^{16}

* VER (4bit) ⇒ 0100 → IPV4   0101 → IPV6

* HLEN (4bit) 0000 0 (0-4) ×
  5 → 5×4 ⇒ 20
  1111 15 → 15×4 ⇒ 60
  $\frac{60}{15} = 4$ ← scaling factor

* TOS (8 bit)

P	P	P	D	T	R	C	0

$\underbrace{P\ P\ P}_{\text{precedence priority}}$ $\underbrace{D\ T\ R\ C}_{\text{TOS bits}}$

Precedence priority
LP → 0 0 0 0
HP → 1 1 1 1

TOS bits
0000 → Default
0001 → Minimize cost
0010 → Max Reliability
0100 → Max Throughput
1000 → Min delay.

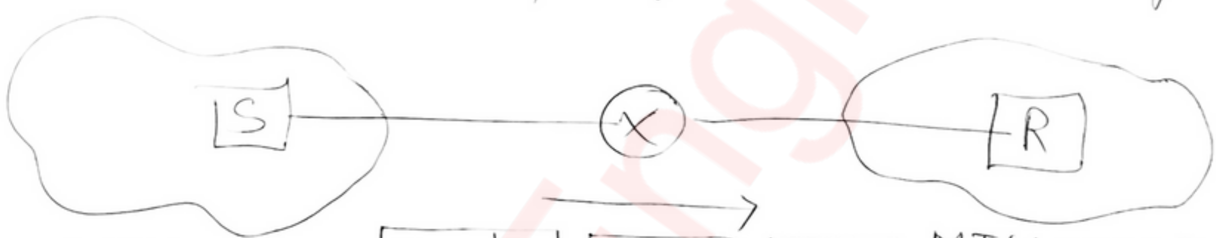
* Total length

Total length (16 bits) → 65535

65535 → HL + DL

$$\underline{DL = TL - (HL) \times 4}$$ ← scaling factor

* Identification, fragment offset & Flags



MTU = 520B

MTU = 200B

[140 | 20]	[180 | 20]	[180 | 20]	
10	10	10	← Identification
360	180	0	← FO (13 bit)
$\frac{360}{8}$	$\frac{180}{8}$ → 22.5	$\frac{0}{8} = 0$	
[148 | 20]	[176 | 20]	[176 | 20]	
$\frac{352}{8} = 44$	$\frac{176}{8} = 22$	$\frac{0}{8} = 0$	
0	1	1	← MF

Donot fragment ← DF

65535

[65515 | 20]

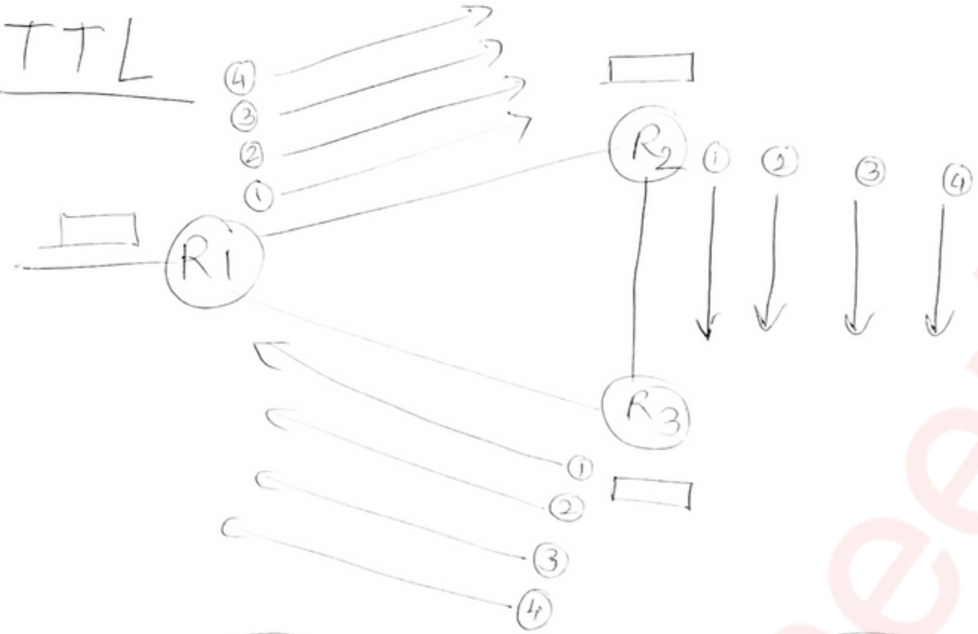
[1 | 20] [ ] [ ] [ ] [ ]

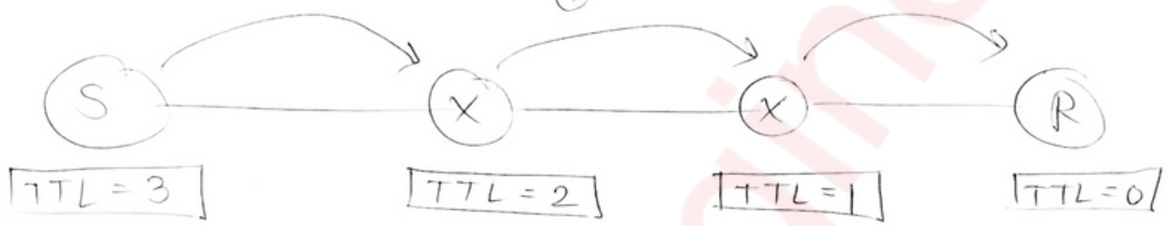
FO = 65514

$\approx 2^{16}$

$$\Rightarrow \frac{2^{16}}{2^{13}} = 2^3 = 8$$ ← Scaling factor

* TTL





At Routers the TTL > 0

at destination = 0

* Protocol (8 bits)

→ ICMP : 1

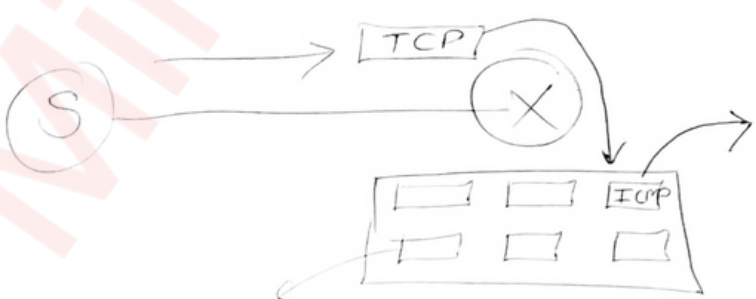
→ IGMP : 2

→ UDP : 17

→ TCP : 6

ICMP < IGMP < UDP < TCP

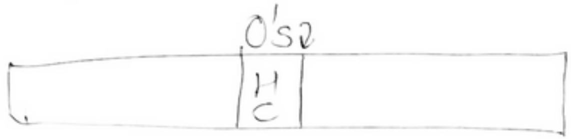
Imp.



Datagrams Buffer

* Header checksum (16 bit)

Divide it in 16 bit parts (2B)



- TTL
- option
- TL
- HL

} can change

so, Recompute 'HC' at each point/Router/node.

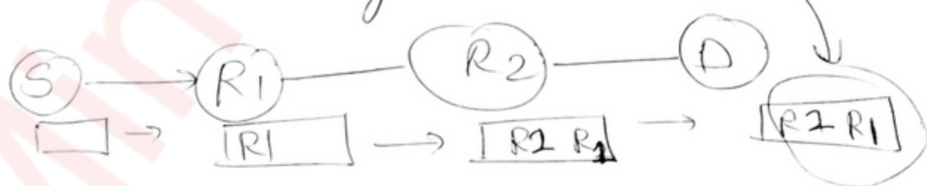
* Option (0 - 40B)

→ Record Route

→ Source Routing

→ Padding.

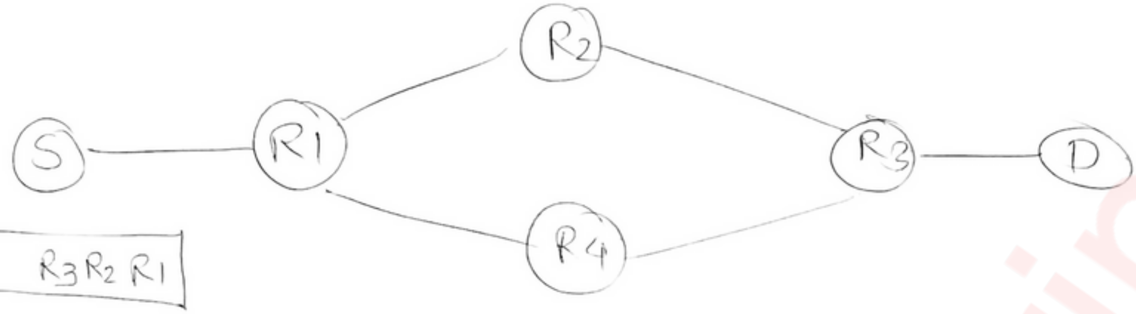
Record Route (IP addr of Router)



1 IP addr = 32 bit = 4B so

In total or max 10 Router addr

but we use max 9



$\boxed{R_3 R_2 R_1}$

↑ strict source Routing

$\boxed{R_3 R_1}$

↑ loose source Routind.

- HLEN (4 bit)

$$\frac{30}{4} = 7.5 \rightarrow \underset{\text{padding}}{+2} = \frac{32}{4} = 8$$

IPV6

|← 32 bits →|

fixed header	BASE	Version (4)	Priority / Traffic class (8)	flow Label (20 bits)	
		Payload length (16)		Next header (8 bit)	Hop-limit (8)
		Source IP addr (128)			
		Destination IP addr (128)			
		Extension headers (128)			
		Data / Payload.			

* Priority :- similar to service field in IPV4.

0 → NO specific traffic
1 → Background data
2 → Unattended data traffic
3 → Reserved
4 → Attended bulk data traffic
5 → Reserved
6 → Interactive traffic
7 → control traffic

* Flow label
  ↳ Real time processing (RTS)

* Hop limit : TTL

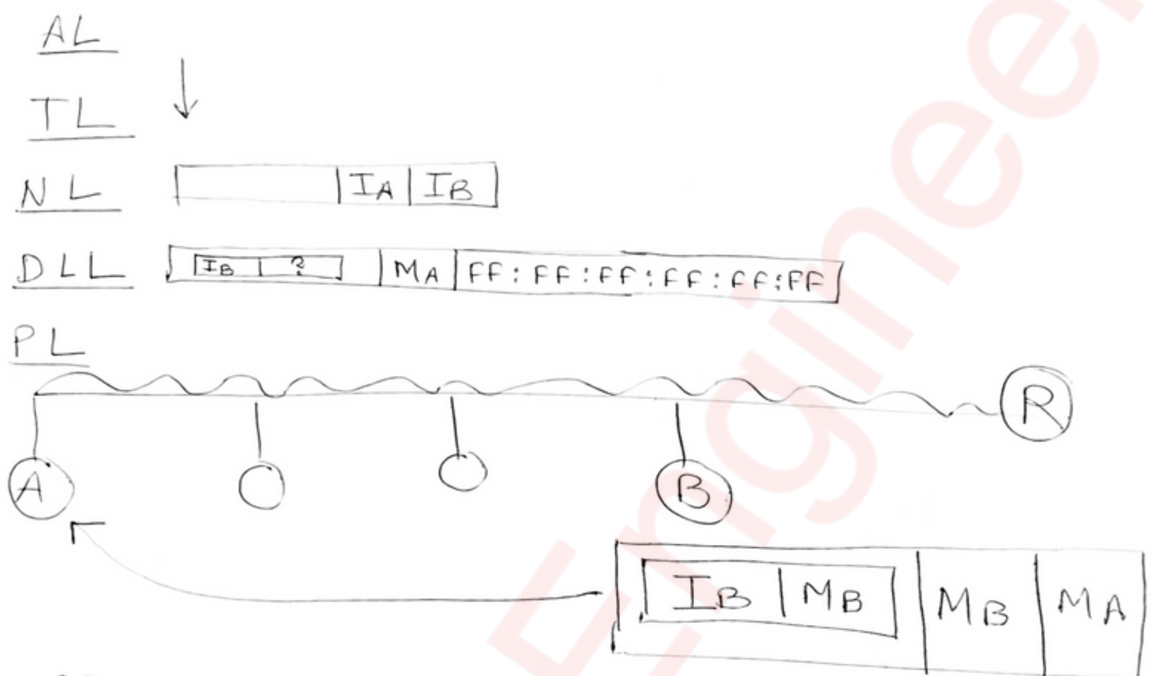
* Next header : Mention of Extension header

- Routing header (43)
- Hop by Hop option (0)
- Fragment Header (44)
- Authentication Header (51)
- Destination option (60)
- Encapsulating security payload (50)

NO next header (59)

- Address Resolution Protocol (ARP)

$\Rightarrow$ (IP $\xrightarrow[\text{locate}]{\text{find}}$ MAC)

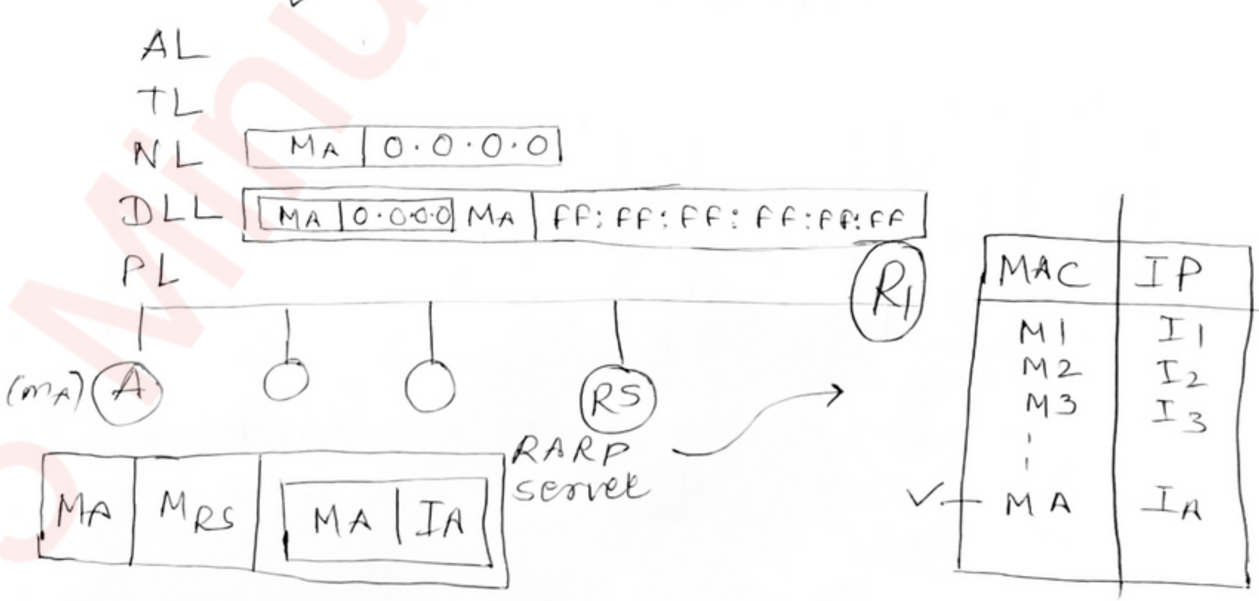


ARP Request → Broadcast

ARP Response → Unicast

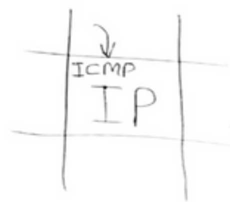
- Reverse Address Resolution Protocol (RARP)

$\Rightarrow$ (MAC $\xrightarrow[\text{locate}]{\text{find}}$ IP)



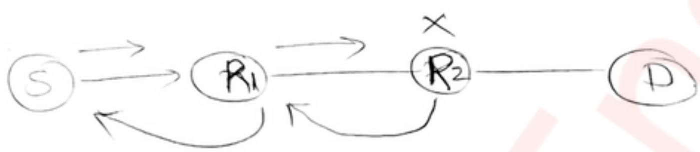
Internet Control Message Protocol (ICMP)

- In n/w layer

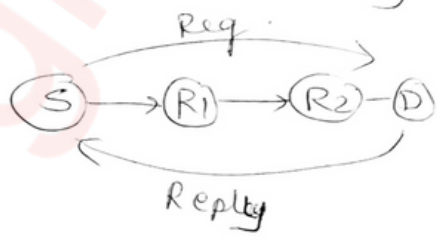


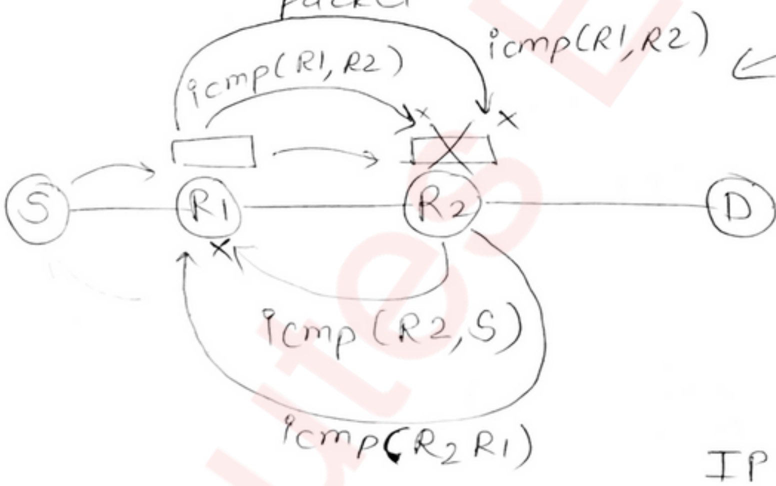
- Types
  - Error handling
  - Request & Reply
- ICMP < IGMP < UDP < TCP

Error handling (feedback)



Request & Reply

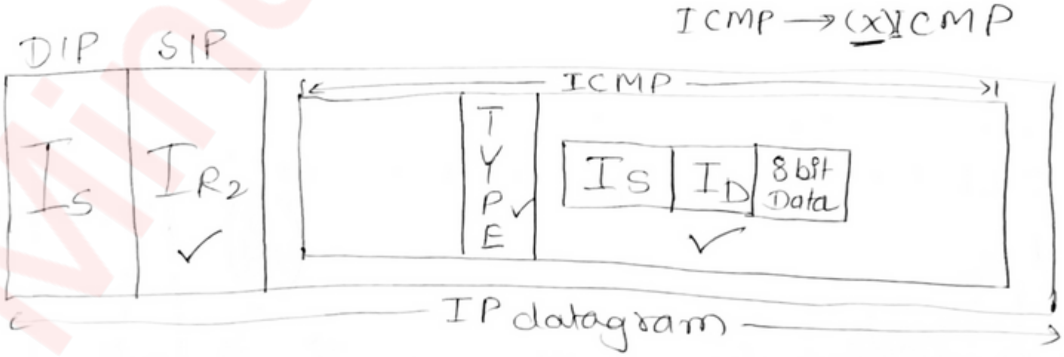




Not not generate icmp packet if a icmp packet discards

IP → ICMP

ICMP → (x) ICMP



- Destination unreachable (3)
- Time exceeded (11)
- Redirection (5)
- source quence (4)
- Parameter problem (12)

- IGMP (Internet Group mgmt Protocol)
  → one-to-one & one to many communication

VER (1) (bits 0–4)	TYPE (1-2) (bits 4–8)	Unused (bits 8–16)	16-bit checksum (bits 16–32)
32 bit Group addr			

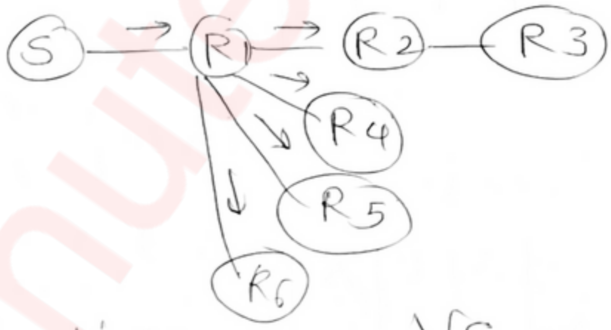
} 8B

IGMPv1 Packet format

* **Routing**: The process of designing a routing table.

* **Switching**: Sending a packet to some path.

* **Flooding**: sending packet to all possible paths. (NO routing table)



Routing	VS	Flooding
1) ✓RT		1) NO RT
2) less Reliable		2) More Reliable
3) Duplicate packets(✗)		3) DP(✓)
4) Traffic less		4) high traffic

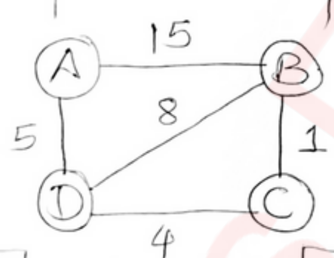
Routing Algorithm

- static (Manually)
- Dynamic
  - DVR (Distance vector)
  - LSR (Link state

① DVR

A	0	A
B	15	B
C	∞	–
D	5	D

A	15	A
B	0	B
C	1	C
D	8	D



A	5	A
B	8	B
C	4	C
D	0	D

A	∞	–
B	1	B
C	0	C
D	4	D

At A

B	D
15	5
0	8
1	4
8	0

A

A	0	A
B		
C		
D		

$(A-B) \begin{cases} (A,B) \\ (A,D)+(D-B) \end{cases}$

$(A-C) \begin{cases} (A,B)+(B-C) \\ (A,D)+(D-C) \end{cases}$

$(A-D) \begin{cases} (A,D) \\ (A,B)+(B-D) \end{cases}$

◦ short cut

AB = 15 AD = 5

	B		D	
A	15	(15+0)	5	(8+5)
B	0	15 ×	8	13 ✓
C	1		4	
D	8		0	

A	0	A
B	13	D
C	9	D
D	5	D

◦ $\frac{15+1}{16}$ ◦ $\frac{4+5}{9}$

◦ $\frac{8+15}{23}$ ◦ $\frac{0+5}{5}$

At C

	B	D
A	15	5
B	0	8
C	1	4
D	8	0

A	9	D
B	1	B
C	0	C
D	4	D

CB = 1 CD = 4

⇒ 'n' routers → (n−1) times/Rounds.

R1 : A → B (15)

R2 : A → D → B (13)

R3 : A → D → C → B (10)

(4−1) 3 times Rounds

A

A	0	A
B	10	D
C	9	D
D	5	D

B

A	10	C
B	0	B
C	1	C
D	5	C

C

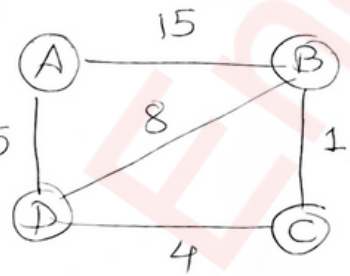
A	9	D
B	1	B
C	0	C
D	4	D

D

A	5	A
B	5	C
C	4	C
D	0	D

* LSR (Link state Routing)

A	
B	15
D	5



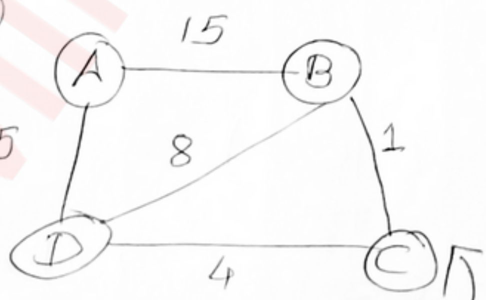
B	
A	15
D	8
C	1

D	
A	5
B	8
C	4

C	
D	4
B	1

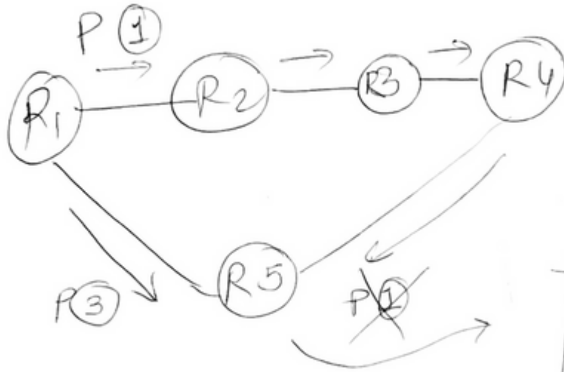
⇒ * Flooding of packets (Global knowledge) At each router

At A



(Dijkistra algo)

A	0	A
B	10	D
C	9	D
D	5	D

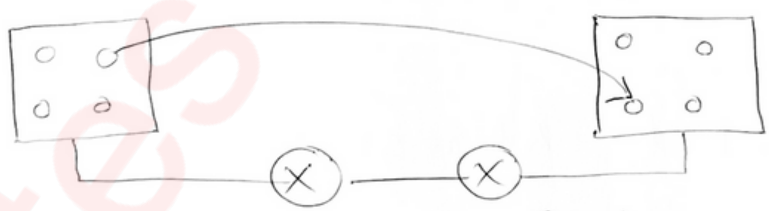


R	Latest Packet
R1	3
R2	5
R3	10
R4	9

→ sequence no
↑ → latest info
↓ → old info

P(4)
10 corrupt
↓
Timer solution.
"hold it for a period of time"

* Transport Layer (4th layer ↑)
  ↳ End-to-End Delivery
  (port-to-port)



(process-to-process)

⇒ TCP & UDP at TL
UDP → connectionless & unreliable
TCP ↳ more Reliable
↳ Inorder

⇒ Error control (checksum)
⇒ flow control
⇒ congestion control.

- Port Number
  - → 16 bit number

Internet Assigned Numbers Authority

$(0 \rightarrow 2^{16} - 1)$

↓ 65535

0 WK 1023 || 1024 Registered 49151 || 49152 Dynamic 65535

(Applications)

eg:-

DNS → 53

POP3 → 110

FTP
- → data → 20
- → control → 21

SMTP → 25

HTTPS (SSL) → 443

- Socket Addresses
  - → IP address + Port Number

(201.10.20.5 20)

* Transmission Control Protocol

Header

SP (16)					DP (16)				4B
Sequence Number (32)									4B
Acknowledgement Number (32)									4B
HL(4)	R(6)	URG	ACK	PSH	PST	SYN	FIN	Window size (16)	4B
Checksum(16)					Urgent Pointer(16)				4B
Options(0 - 40B)									
Data									

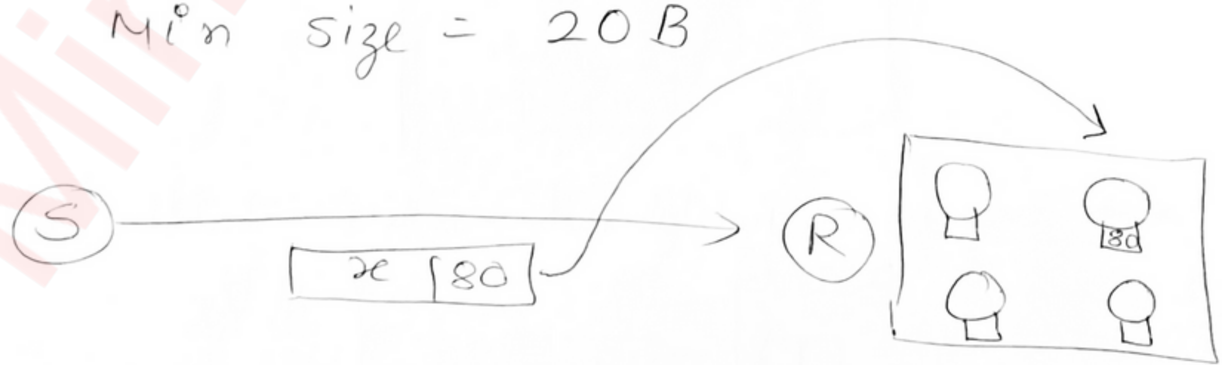
TL → Segment = Header + Data/Payload

NL → Datagram

DLL → Frame

Max size = 20 + 40 = 60B

Min size = 20B

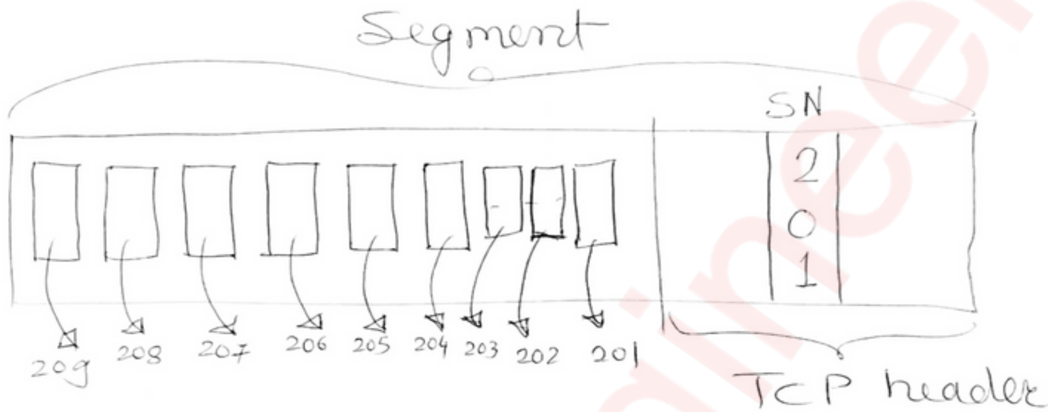


- Sequence Numbers (32 bit)

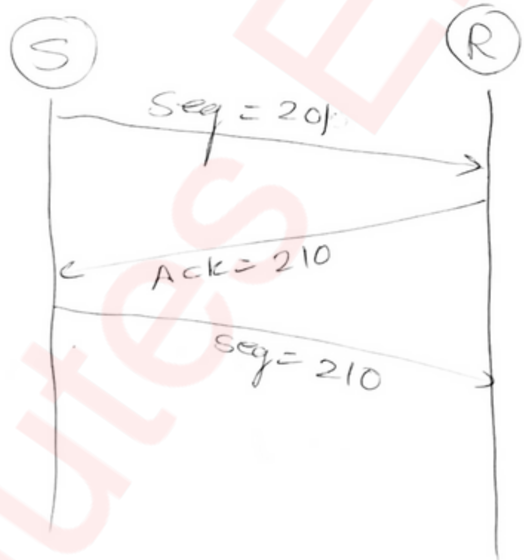
TCP is Byte stream protocol

IP is packet stream protocol.

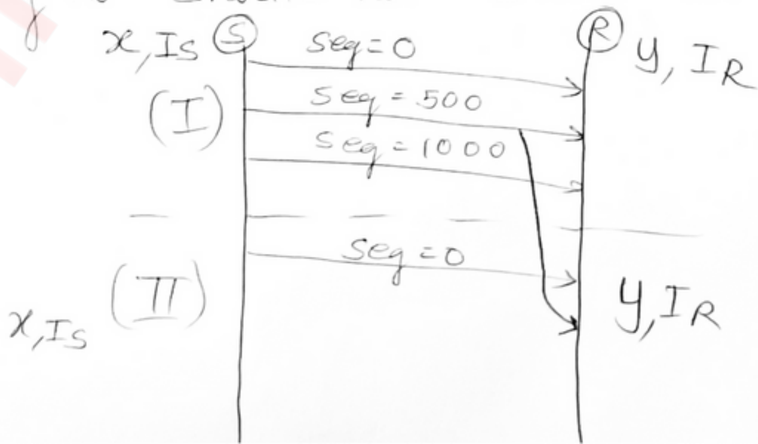
"Assigning a number to every byte"



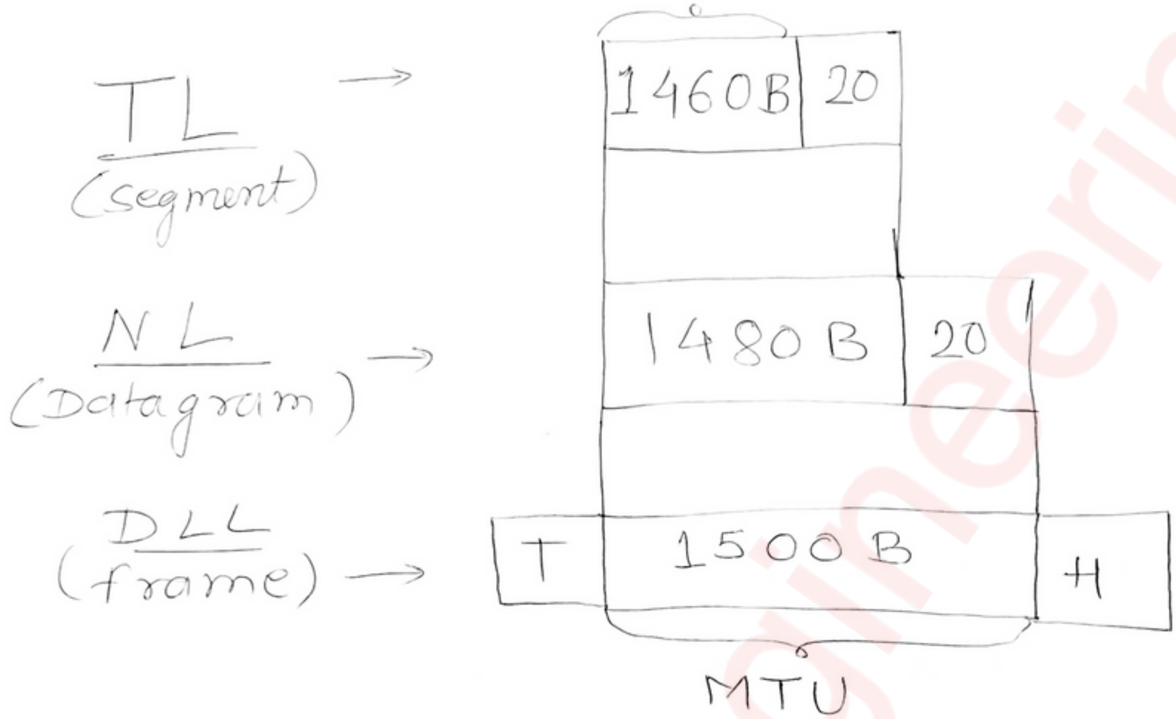
- Acknowlegment Number (32 bit)



*→ seq no. should not start with '0'



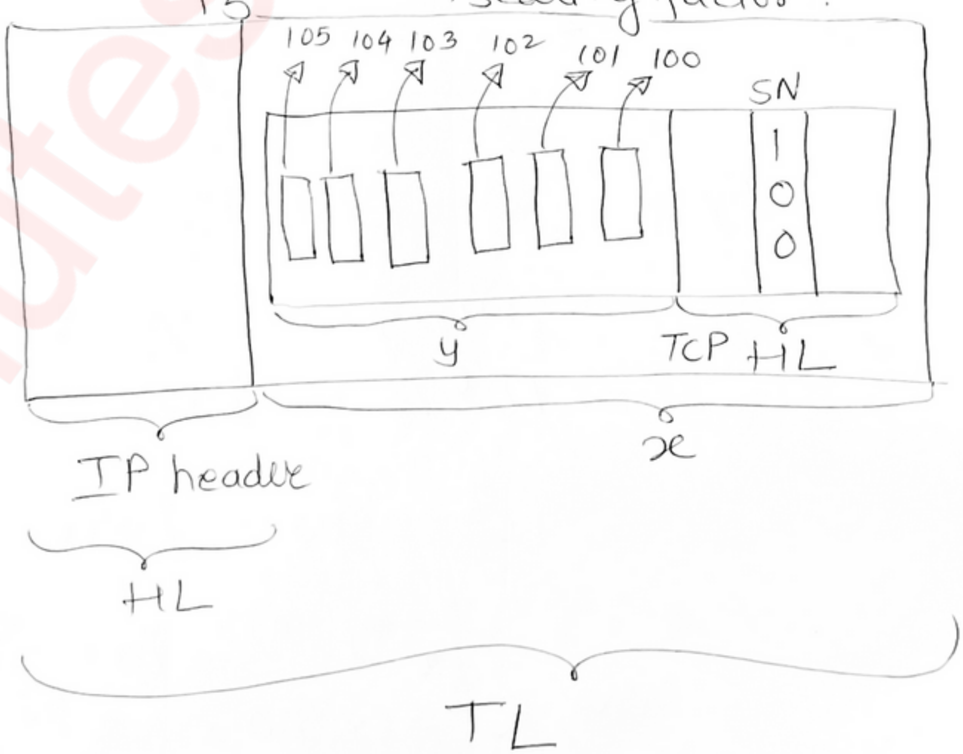
- Maximum segment size



- Header length (4 bit) → 0 0 0 0 → 0

(20B – 60B) 1 1 1 1 → 15

$\frac{60}{15} \rightarrow 4$ ← scaling factor.



① TL − HL = x

② x − TCP HL

③ 100 + y − 1

→ 100 + 6 − 1

→ 105

\+ 1

→ Ack num (106)

- Wrap Around Time

Seq no: $2^{32} \rightarrow 2^{2} \cdot 2^{30}$

$\downarrow \quad \downarrow$

4 G Seq.

$\downarrow$ 1 seq → 1 Byte

4 GB data

$\downarrow$

But for 4 GB + 1 B

'?'

start with '0'

Wrap around

Not start with '0'

WAT

$\downarrow$

Time taken to repeat/reuse the same seq no. again.

$WAT \propto B/W$

$B/W = 1\,Mbps$

$= 10^{6}\,B - 1\,sec$

$\downarrow$

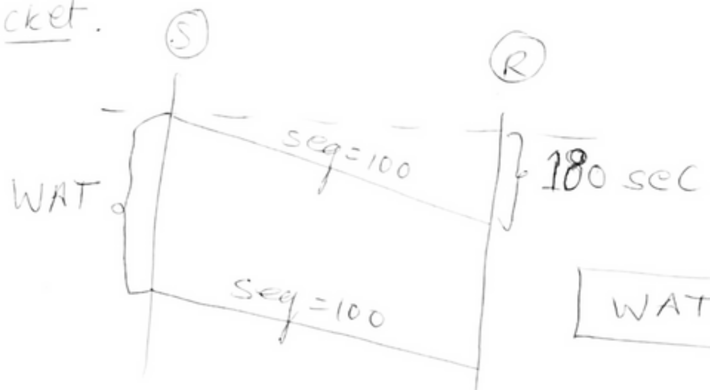
10^{6} seq no $-$ 1 sec

$1 \text{ seq no} = \frac{1}{10^{6}} \text{ sec}$

So, $2^{32} \text{ seq no} = \frac{2^{32}}{10^{6}} \text{ sec}$

WAT = 4294.96729 sec

Lifetime : 3 min (180 sec)
of packet.
(LT)



$\boxed{WAT > LT}$

But if $B/w = 1\,Gbps$

$\downarrow$

$10^9\ B \rightarrow 1\ sec$

10^9 seqno $\rightarrow 1\ sec$

2^{32} seq no $= \dfrac{2^{32}}{10^9}$ sec

$\boxed{WAT = 4.29496\ sec}$

$\boxed{LT > WAT}$ ✗

so,

$B/w = 1\,Gbps$

$LT = 180$ sec

$\boxed{\lceil \log_2 B/w \times LT \rceil}$

1 seq $\rightarrow 10^9$ seq nos

180 sec $\rightarrow x$

$x = \lceil 10^9 \times 180 \rceil$ seq no.

$\Downarrow$

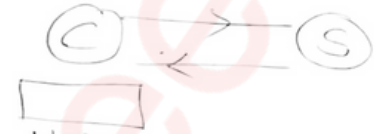
but seq no (32 bits) $\xrightarrow{+6}$ 38 bits

$\downarrow$

option field (6 bits).

I Connection Establishment

(3-way handshak)



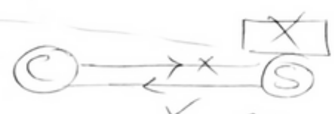
$\frac{14600}{1460}$ } → 10 packets on client capacity

Flowcontrol (WS)

Don't send me more than what I can receive

SYN, FIN, 1B consumes 1 seq. no.

But Ack flag don't consume so use it again.

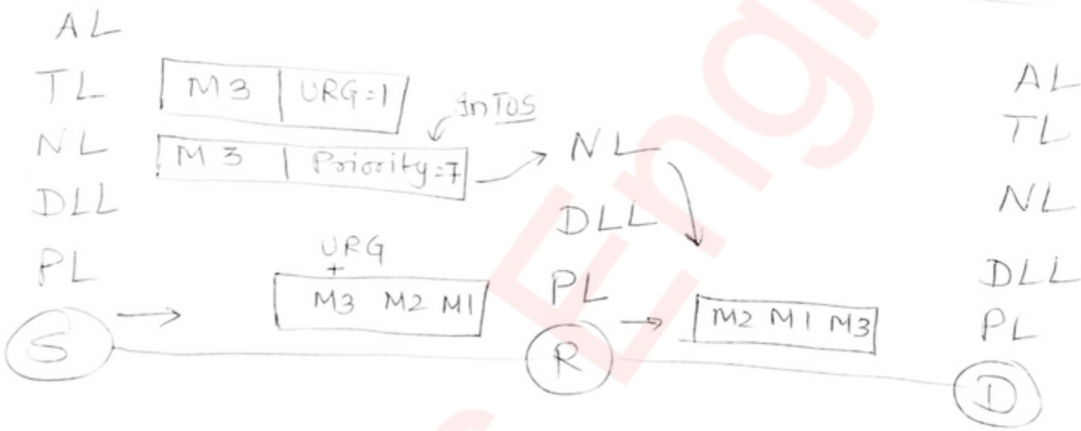


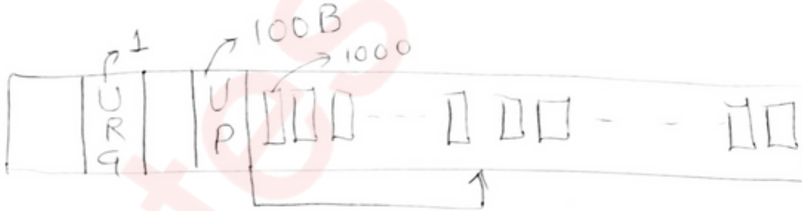
(C→S) Pure Ack ✓ Data X

(S→C) data ✓ PGAck ✓

SYN	ACK	
1	0	→ Request Packet (I)
1	1	→ Reply Packet (II)
0	0	→ × Not possible
0	1	→ Transmission or Termination

* URG Flag & Urgent Pointer





first 100B

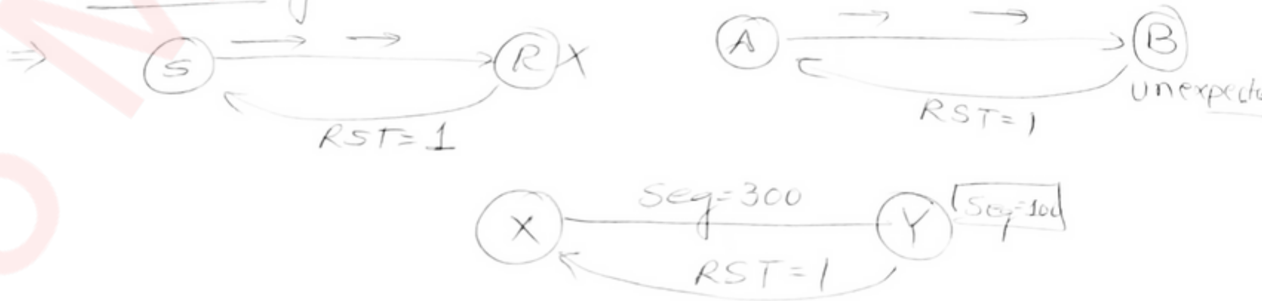
1000 → (1000+100)

↓

1100

urgent data

* RST flag



* User Datagram Protocol (UDP)

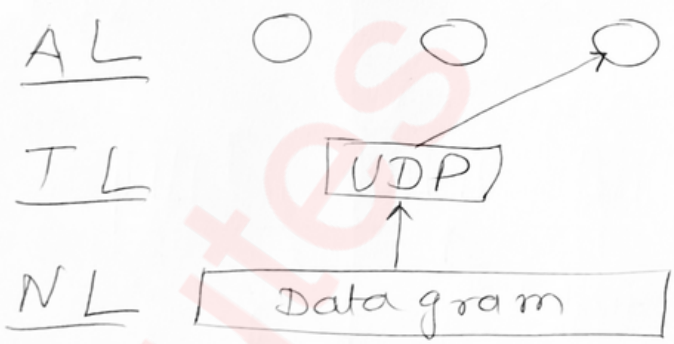
- → Connectionless
- → Unreliable

TCP → CE + DT + CT (x) not required in UDP

Eg:- Pen required in examhall
UDP works (Yes/NO).

Eg: Broadcast (TCP would be expensive)

Eg: Games, multimedia (speed required)



SYN, FIN Flag & ACK
seq no. field
not required.

UDP header →

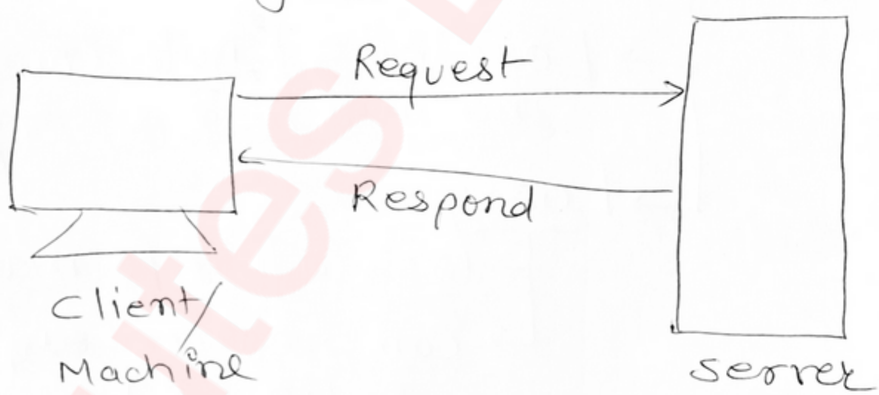
SP (16)	DP (16)
(16) Length (H+D)	checksum (16)

8 Byte header size

TCP VS UDP

TCP	UDP
1. Connection oriented	1. Connectionless
2. Reliable (↑)	2. Low/Less Reliable
3. Inorder	3. No order
4. Slow	4. fast
5. EC is mandatory	5. EC is optional.
6. HL = (20-60)B variable	6. HL = 8 B fixed
7. Byte stream	7. Message stream
8. high overhead	8. Low overhead

* Session Layer

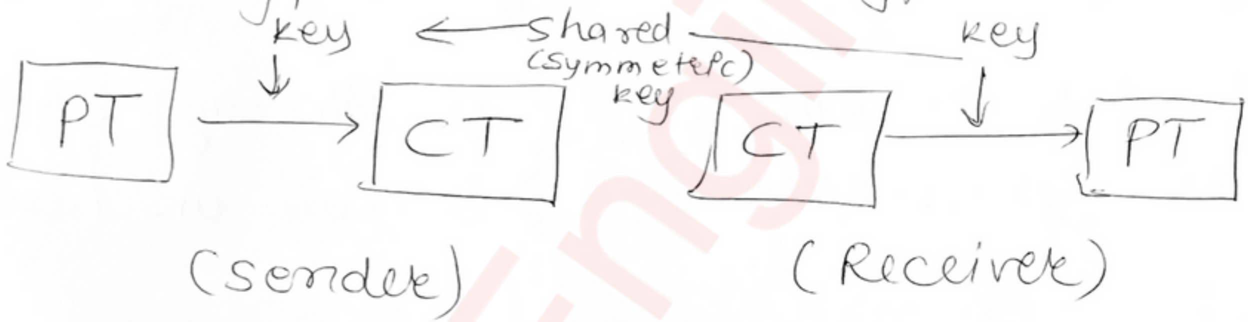


- Authentication
- Authorization
- Session Mgmt (Synchronization)
- Checkpoint [Session Restoration]
- Session establishment
- Dialog mgmt (log data of connection)

* Presentation Layer

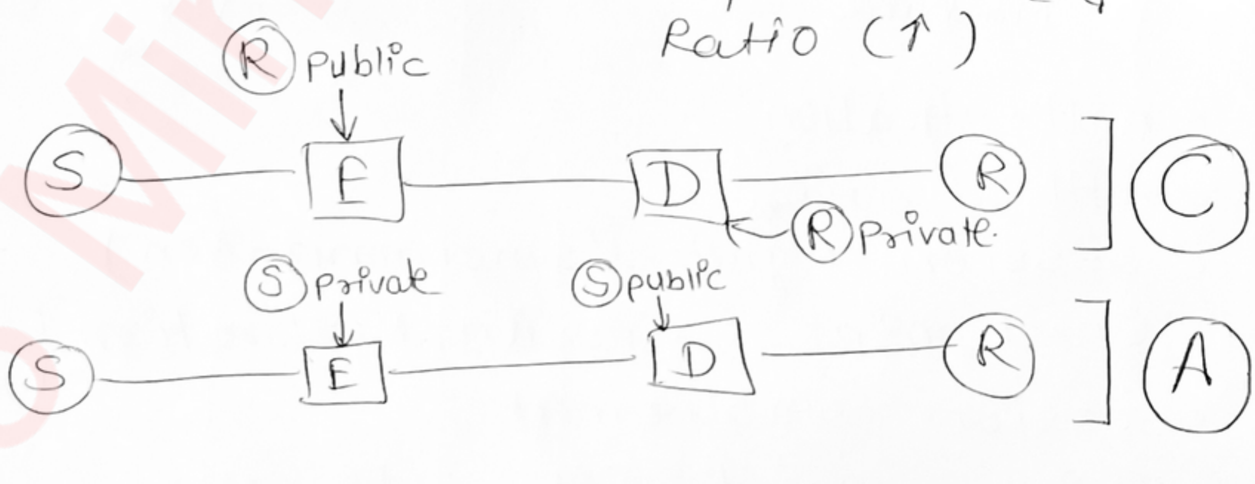
→ Translation [Accepted / understandable format]

→ Encryption Decryption



→ Compression (zip file)

- → Lossless [Data Compaction]
  "no info loss" & Compression Ratio is (↓)
- → Lossy
  - → loss of information
  - → Compression Ratio (↑)

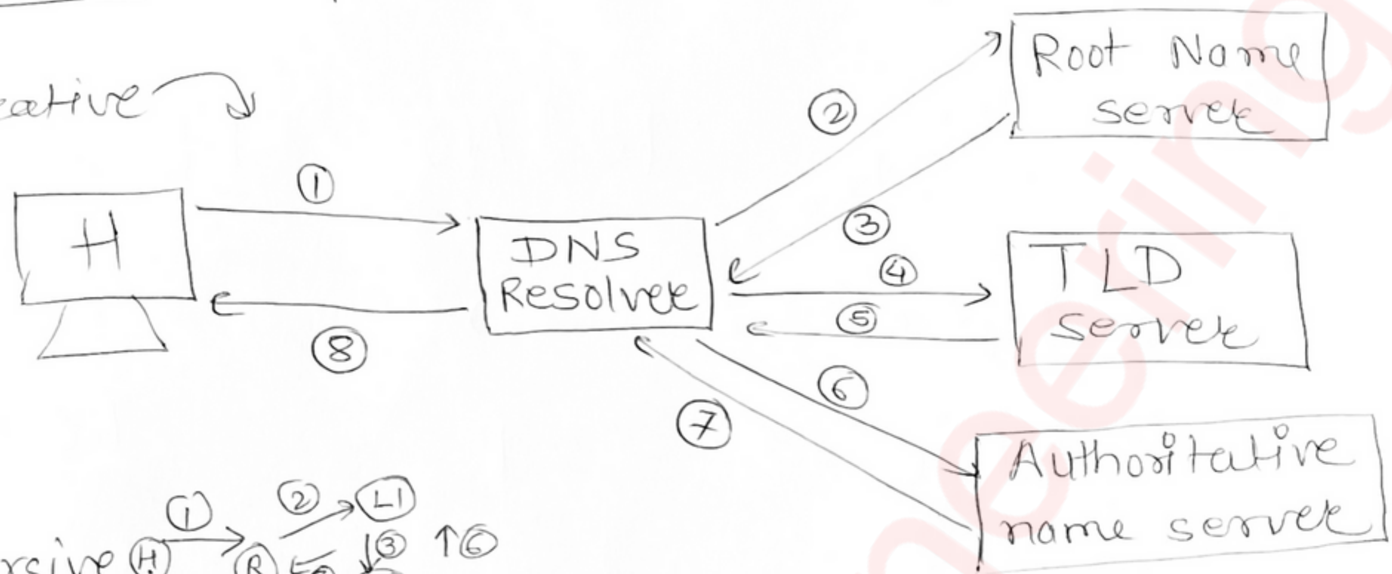


* Application Layer

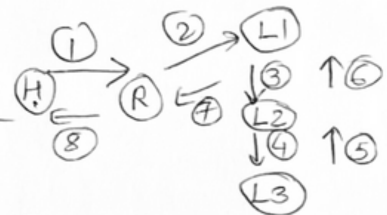
Application Protocol	Port No	TL protocol
FTP D	20	TCP
FTP C	21	TCP
Telnet	23	TCP
SMTP	25	TCP
HTTPS (SSL)	443	TCP
HTTP	80	TCP
POP	110	TCP
DNS	53	UDP
DHCP	67/68	UDP
TFTP	69	UDP
NTP	123	UDP
RIP	520	UDP
Echo	7	TCP/UDP

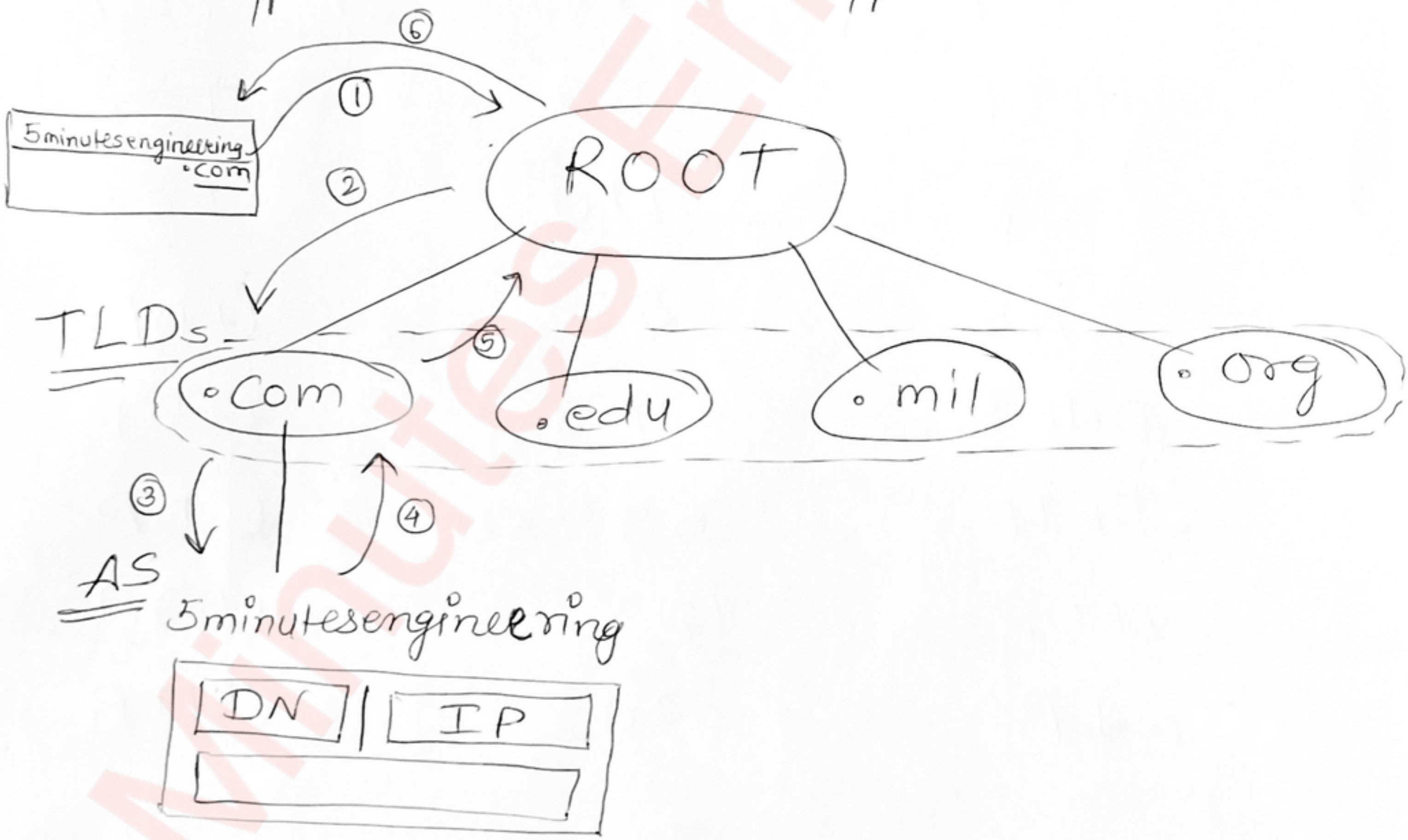
• DNS (operate on AL)

• Iterative



• Recursive





* HTTP

- → Port no : 80
- → TCP ✓ (To get Reliability)
- → stateless [NO info saving/storing]
- → Inband protocol → 1 port no. (80) → Data as well as commands.
- → HTTP 1.0 (Non-Persistent)
  HTTP 1.1 (Persistent) } Connection
- → NO security [HTTP run over SSL] → HTTPS (CIA)
- → Combination of FTP & SMTP

Non-persistent	Persistent
One TCP connection is made for each Req / Reply. "slow"	Server leaves the connection open for more than 1 request → It is closed on → Req. of client → Time-out reached.

* FTP

- → file transfer
- → NOt Inband
  - → Data : Port no 20
  - → Control : Port no. 21
- → Data connection is "non-persistent"
- → Control connection is "persistent"
- → stateful
- → Reliability ↑
- → Performance ↑

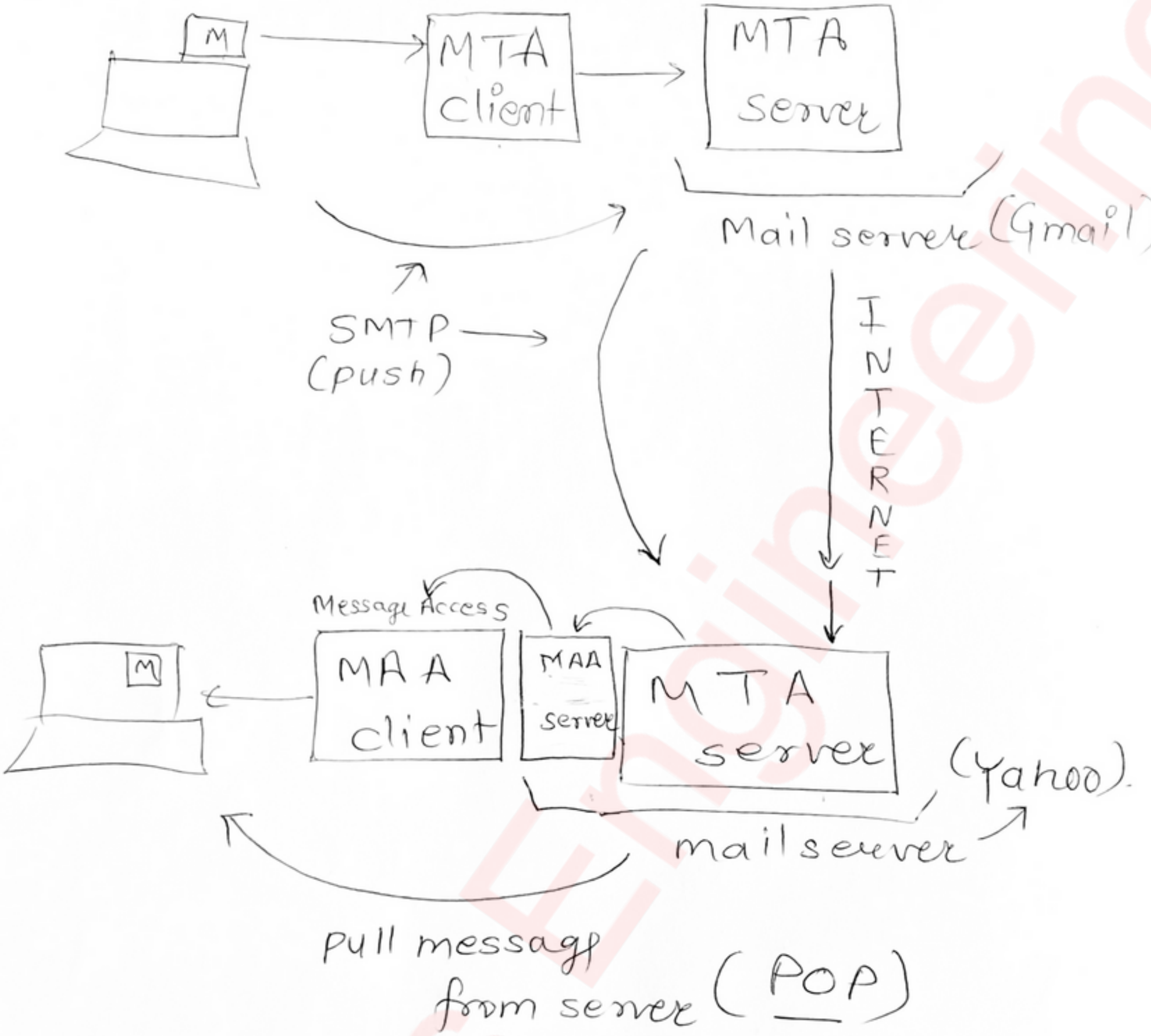
⇒ Trivial FTP (TFTP)

- → Trivial / simple (Read/write operation)
- → UDP
- → Port 69
- → Demand : security, long-distance transfer (X)

* SMTP (Simple mail Transfer Protocol)

- → Asynchronous nature support.
- → Port no: 25 for mail pushing
- → MTA (message transfer agent)

Mail send



post office Protocol version 3 (POP3)

→ TCP

→ Port no : 110

- **MIME** (Multipurpose Internet mail Extension)
  - → Audio
  - → video
  - → Image

  } Non text data