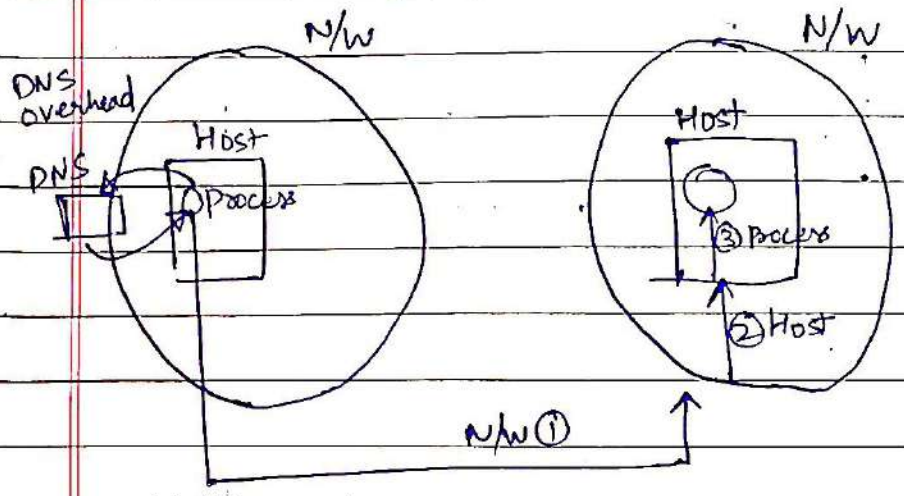
1 IP Address Subnetting Supernetting



www.google.com

⇓

IP Address

NID	HID

+ Port Number
80 (web service)

n bits

k	n−k

2^k parts each part : 2^{n-k}

Initially back in 1980's

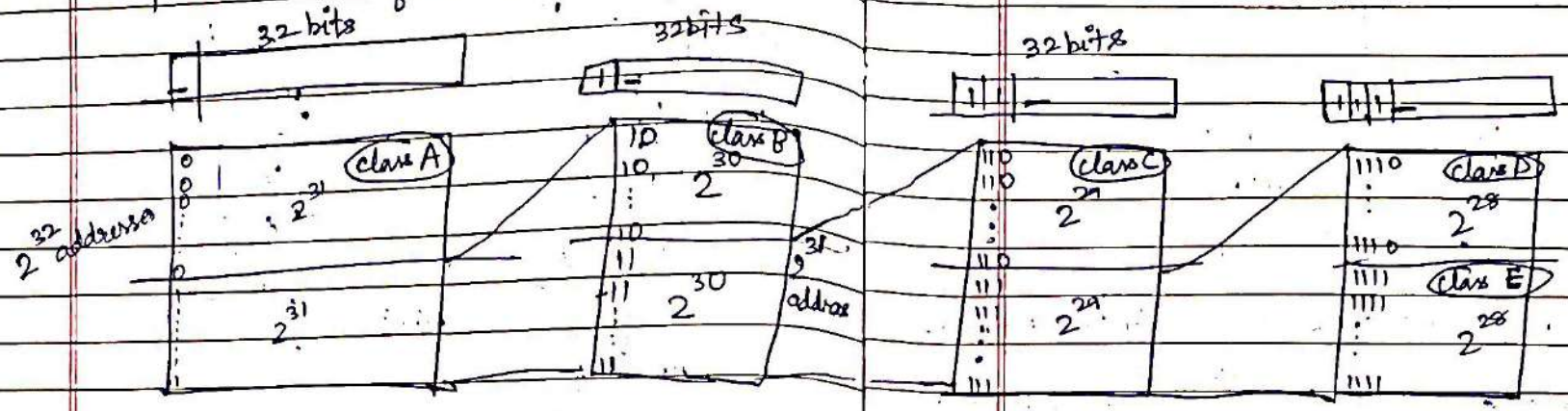
32 bits

8	24
NID	HID

256 16 M Hosts

✗ NOT feasible Now ↓ Too big Number

classful manner of IP addressing :-



(IP address in class A) = 2^{31} 0

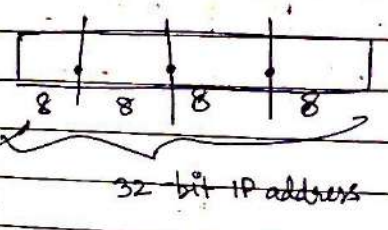
(——— B) = 2^{30} 10

(——— C) = 2^{29} 110

(——— D) = 2^{28} 1110

(——— E) = 2^{28} 1111

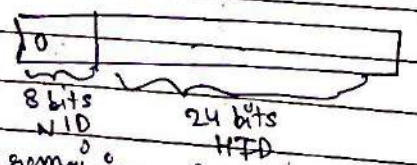
Prefix Codes



Dotted Decimal Representation

Class A :- 2^{31} IP addresses

(1-126)



(1 bit fixed as 0)

→ 2^7 N/w of class A are present

★ 128 N/w of class A are there in which each N/w consists of 2^{24} i.e. 16 million IP addresses are present per network of class A.

0 - - - - - - -

00000000 → 0 ×

00000001 → 1

00000010 → 2

⋮

11111111 → 127 ×

· We practically do NOT use 0 or 127. So practically, # Networks (1 to 126 (First octet).) is 126.

Date
Page

Class B :- 2^{30} IP addresses

(128 - 191) | 10 - - - - - - - - | - - - - - - - - | 32 bits

16 bits — 16 bits

NID — HID

Networks = $2^{14} \simeq 16K$

IP per Network = 2^{16} ~~hosts~~ IP / Network

$= 2^{16} - 2$ hosts / Network

First Octet :-

10 - - - - - -

0 0 0 0 0 0 0 = 128

0 0 0 0 0 1 = 129

⋮

1 1 1 1 1 1 = 191

Class C :- 2^{29} IP addresses

(192-223) | 100 - - - - - - - - - - - | - - - - - | 32 bits

24 bits — 8 bits

NID — HID

Networks = $2^{21} \simeq$ 2Million

IP per Network = $2^8 = 256$ ~~hosts~~ IP / Network

$256 - 2 = 254$ hosts/Net

First octet :-

110 - - - - -

0 0 0 0 0 = 192

0 0 0 0 1 = 193

⋮

1 1 1 1 1 = 223

192
31
223

Class D :- 2^{28} IP addresses
(224-239) (No such division of NID and HID)
↓ First Octet :-
used for multi-casting

1110 - - - -
~~0000~~ - 224
0001 - 225
⋮
1111 - 239

224
15
239

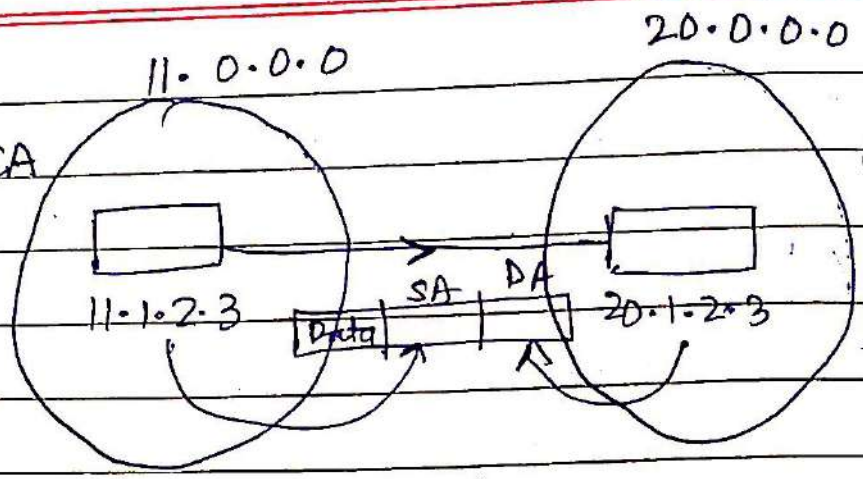
Class E :- 2^{28} IP addresses
(240-255) (No such division of NID and HID)
↓ First octet :-
reserved

1111 - - - -
0000 - 240
0001 - 241
⋮
1111 - 255

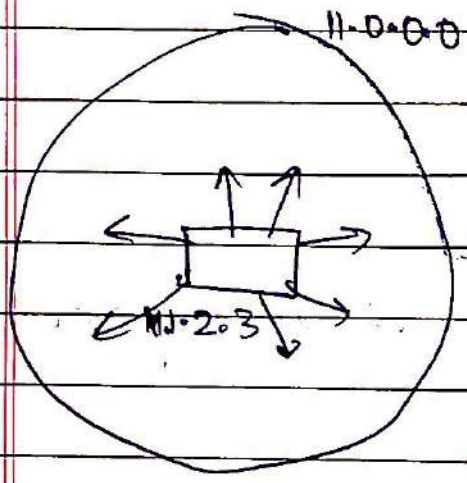
Casting

Unicast :- one host $\xrightarrow{\text{Packet}}$ one host

Broadcast :- one host $\xrightarrow{\text{Packet}}$ multi host



Broadcast → Limited Broadcast
Broadcast → Directed Broadcast

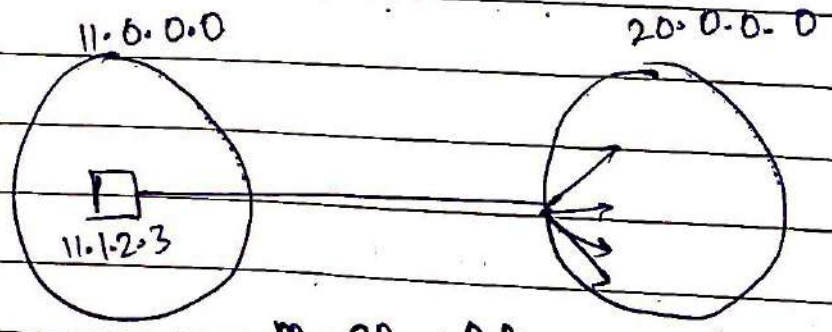


$2^{24} = 16\,m$ hosts

m	SA	DA
data	11.1.2.3	255.255.255.255

Limited Broadcast Address :- 255.255.255.255
Directed ——— (NID).255.255.255

Directed Broadcast :-



m	SA	DA
data	11.1.2.3	20.255.255.255

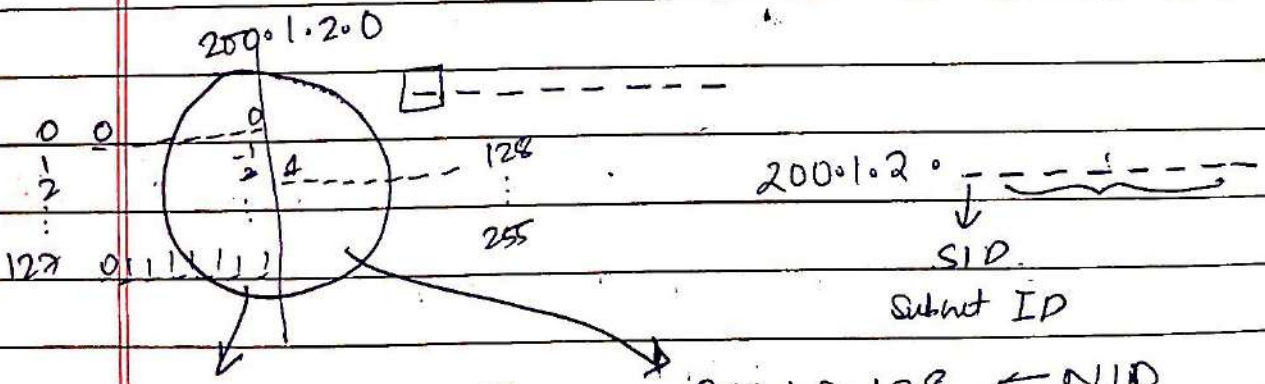
class A - 1-126, class B - 128-191, class C - 192-223
class D - 224-240, class E - 241-255

eg.

class	IP	NID	DBA	LBA
A	1.2.3.4	1.0.0.0	1.255.255.255	255.255.255.255
A	10.15.20.60	10.0.0.0	10.255.255.255	255.255.255.255
B	130.1.2.3	130.1.0.0	130.1.255.255	255.255.255.255
B	150.0.150.150	150.0.0.0	150.0.255.255	255.255.255.255
C	200.1.10.100	200.1.10.0	200.1.10.255	255.255.255.255
C	220.15.1.10	220.15.1.0	220.15.1.255	255.255.255.255
E	250.0.1.2	–	–	–
X	300.1.2.3	–	–	–

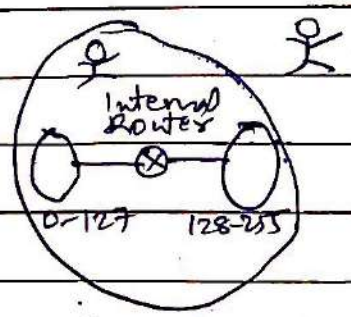
NOT a valid IP address

Subnets



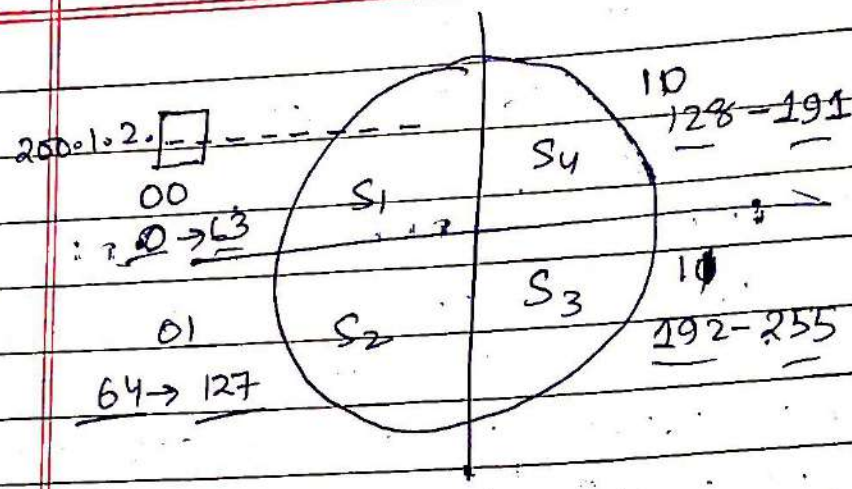
200.1.2.0 ← NID
200.1.2.127 ← DBA

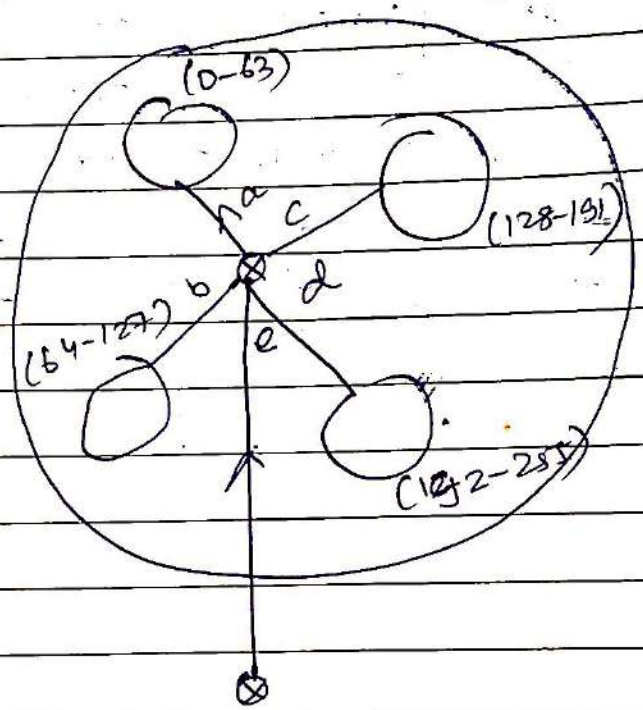
200.1.2.128 ← NID
200.1.2.255 ← DBA



Total # hosts = $(128-2) \times 2$
$= 252$

200.1.2.0





Subnet mask :- 32-bit numbers in which

1's → NID and SID

0's → H.I.D

11111111 . 11111111 . 11111111 . 11000000

255 . 255 . 255 . 192

eg: 200.1.2.130

11001000 . 00000001 . 00000010 . 10000010

bit wise &

11001000 . 00000001 . 00000010 . 10000000

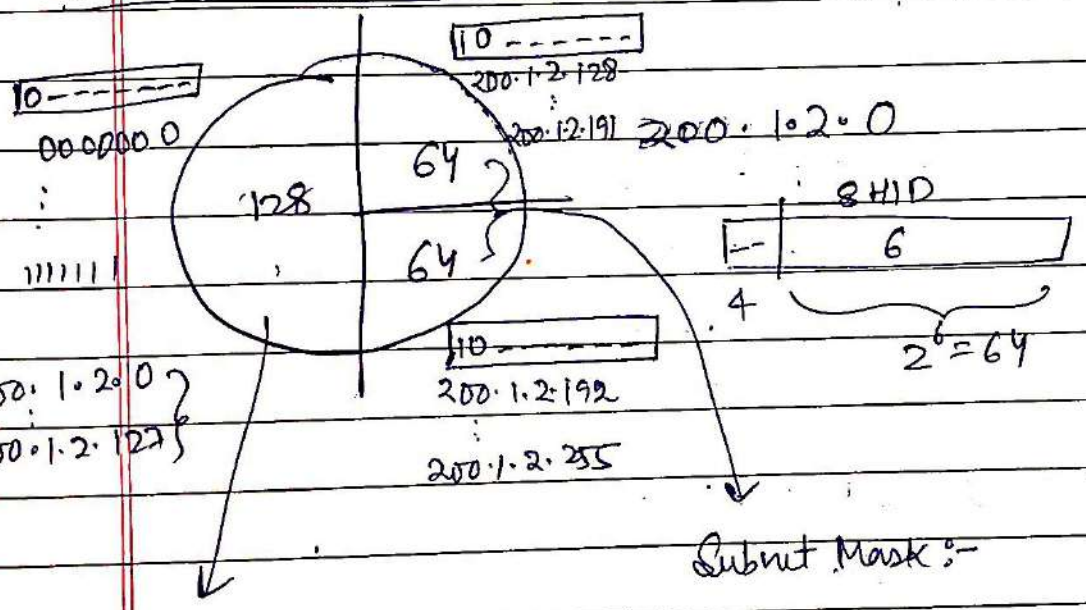
200 . 1 . 2 . 128 ← NID of subnet

Routing Table

NID	Subnet Mask SM	Interface
200.1.2.0	255.255.255.192	a
200.1.2.64	255.255.255.192	b
200.1.2.128	255.255.255.192	c
200.1.2.192	255.255.255.192	d
0.0.0.0	0.0.0.0	e

default entry

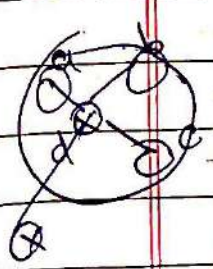
Variable length Subnet Masking (VLSM)



Subnet Mask :-
255.255.255.192

Subnet Mask :-
255.255.255.128

NID	SM	Interface
200.1.2.0	255.255.255.128	a
200.1.2.128	255.255.255.192	b
200.1.2.192	255.255.255.192	c
0.0.0.0	0.0.0.0	d



~~SID~~ :

SM :- 255.255.255.192

11111111.11111111.11111111.11000000

26 1's

~~HID~~ + SID

6 0's

HO

$2^6 = 64$ IP/SN

Class A :- 8 + SID = 26

SID = 18

(SN) = 2^{18}

class B :- 16 + SID = 26

SID = 10

(SN) = 1024

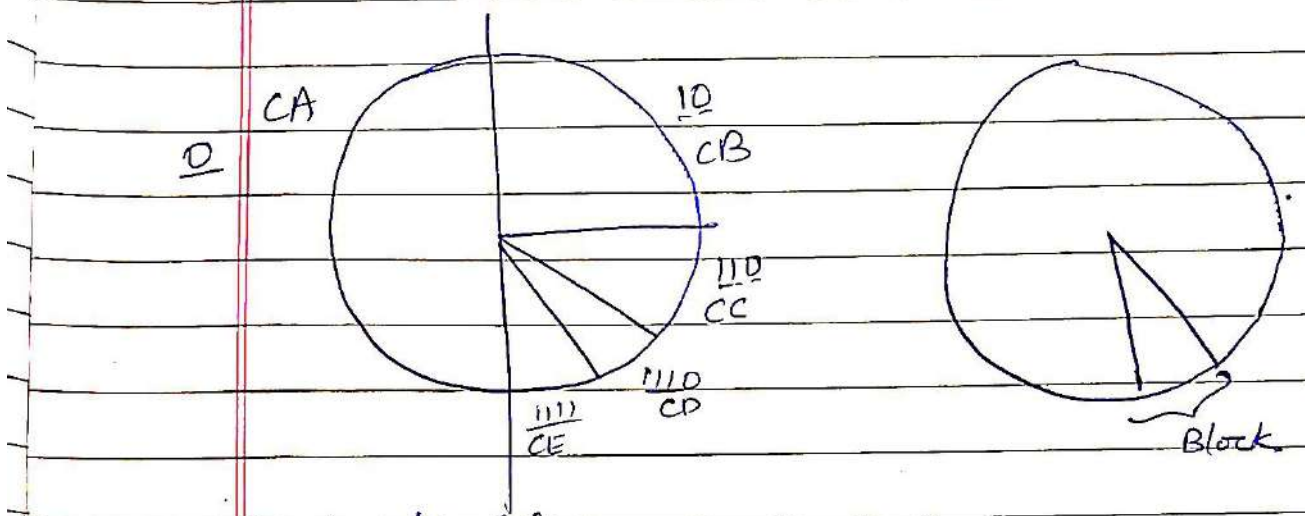
class C :- 24 + SID = 26

SID = 2

(SN) = 4.

SM	# Hosts	Subnets in Class A	Class B	class C
255.0.0.0	$2^{24}-2$	1	–	–
255.128.0.0	$2^{23}-2$	2^1	–	–
255.192.0.0	$2^{22}-2$	2^2	–	–
255.240.0.0	$2^{20}-2$	2^4	–	–
255.255.0.0	$2^{16}-2$	2^8	1	–
255.255.254.0	2^9-2	2^{15}	2^7	–
255.255.255.0	2^8-2	2^{16}	2^8	1

SM	# Hosts	subnets in Class A	Class B	Class C.
255.255.255.224	$2^5 - 2$	2^{19}	2^{11}	2^3
255.255.255.240	$2^4 - 2$	2^{20}	2^{12}	2^4



classful classification

CIDR (classless)
Classless Inter Domain Routing

a.b.c.d / n → #(bits) of NID

eg. 20.10.50.100/20

NID = 20

HID = 12

#(IP addresses) = 2^{12}

Rules for forming CIDR blocks:-

- All IP addresses should be contigous.
- Always ask for (power of 2) IP addresses
- First IP addr in the block should be evenly divisible by size of the block

eg. 100.1.2.32
100.1.2.33
⋮
100.1.2.47

CIDR Block or NOT ?

IP = 16 = 2^4 ✓ $\boxed{\text{Yes}}$

100.1.2.32
↓
00100000
(X)

eg. 20.10.30.32
20.10.30.33
⋮
20.10.30.63

CIDR Block or NOT

IP = 32 = 2^5 $\boxed{\text{Yes}}$

00100000
(X)

eg. 150.10.20.64
150.10.20.65
⋮
150.10.20.127

CIDR block or NOT

IP = 64 = 2^6 $\boxed{\text{Yes}}$

01000000

Representation of CIDR blocks

eg. 100.1.2.32
⋮
100.1.2.47

HID = 4
BID = 28

100.1.2.32/28
100.1.2.40/28 } any...

any IP/~~block~~

eg. 150.10.20.64
⋮
150.10.20.127

150.10.20.64/26

First IP addresses :- Block ID
Last IP addresses :- Directed Broadcast

(Hosts) = # (IP) − 2

Subnetting in CIDR

20.30.40.10/25

20.30.40.0 / 0001010
BID HID

0000000

20.30.40.00-----
20.30.40.0
↓
20.30.40.63

20.30.40.01-----
20.30.40.64
↓
20.30.40.127.

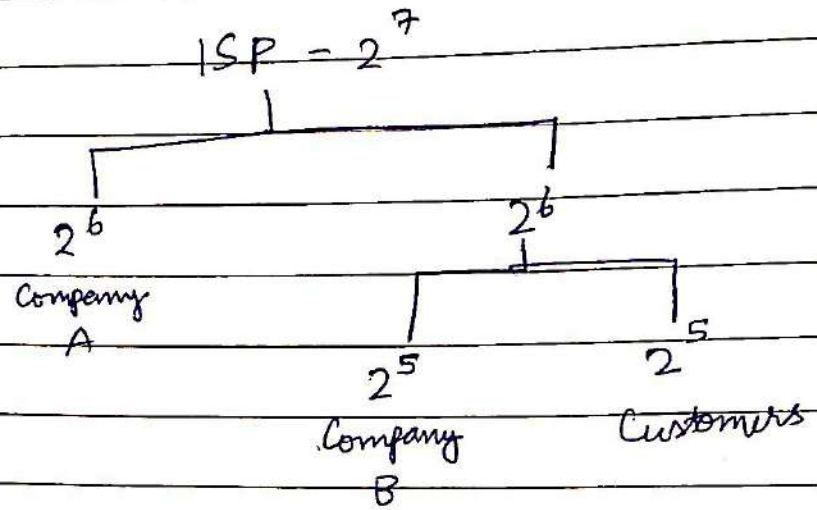
20.30.40.0/26

20.30.40.64/26

Date
Page

VLSM.. in CIDR blocks

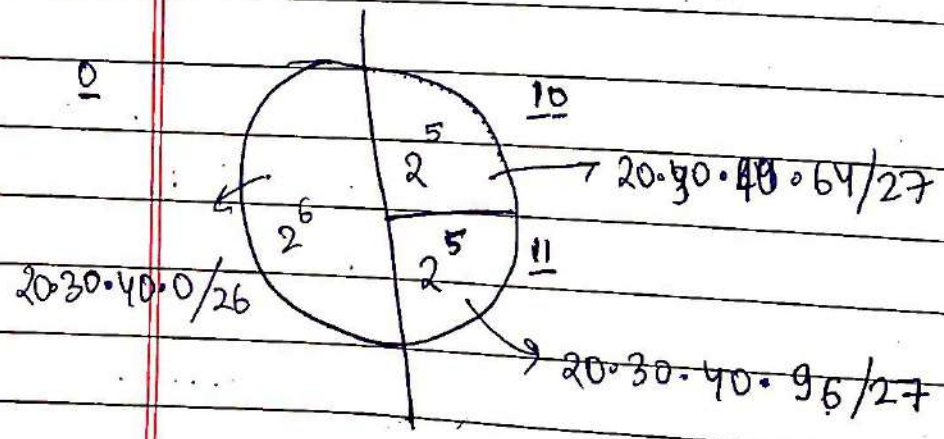
eg. 20.30.40.10/25



20.30.40.0 | 0000000 - 0

⋮

111 111 1 - 127



eg. 40.30.10.10/20 2^{12} IP — 2^{12} — 2^{10} — 2^{10}

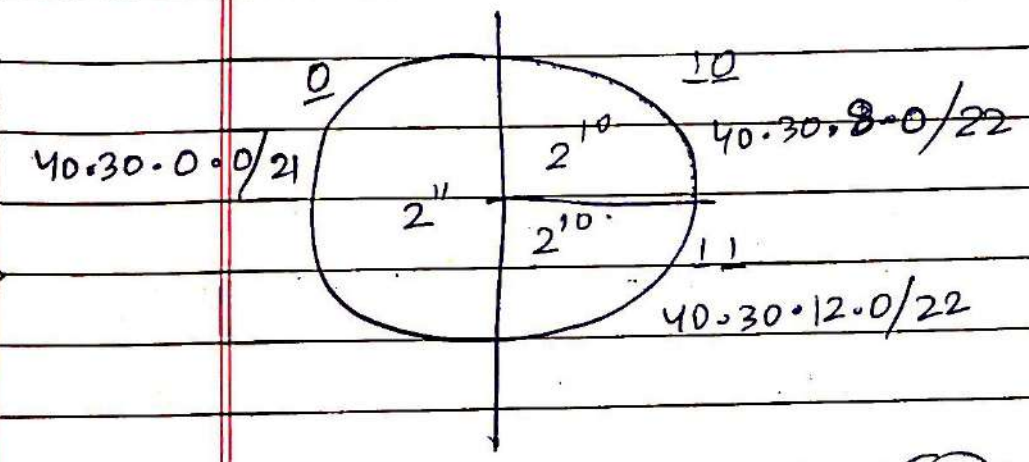
⇓

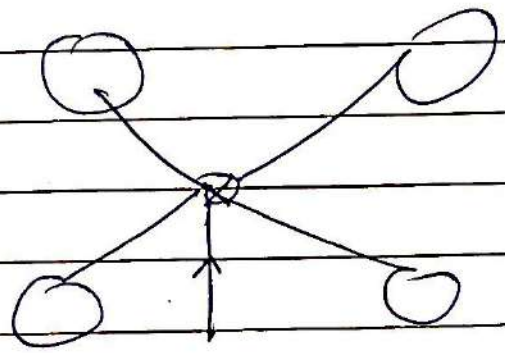
40.30. 0000 1010 . 00001010

BID 0000 . 00000000 — 40.30.0.0

⋮

1111 . 11111111 — 40.30.15.255

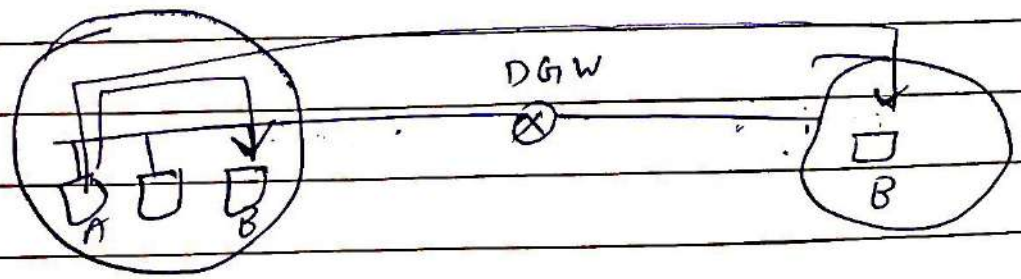




Subnet Mask.

\& IP address

Subnet ID



A : I_A

SA

NID$_A$A

⟷ compare

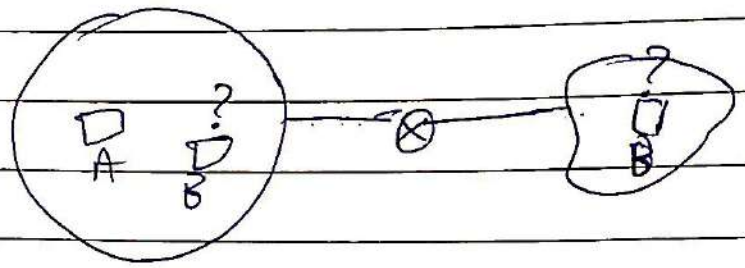
B : I_B

SA

NID$_B$A

eg. A :- I_A : 200.1.2.10
S_A : 255.255.255.128

B :- I_B : 200.1.2.130



I_A :- 11001000.00000001.00000010.00001010
& 11111111.11111111.11111111.10000000
11001000.00000001.00000010.00000000

$NID_A A$ = 200.1.2.0

I_B :- 11001000.00000001.00000010.10000010
& 11111111.11111111.11111111.100000000
11001000.00000001.00000010.10000000

$NID_A B$ = 200.1.2.130

$NID_A A \neq NID_A B$ so Not in Same network as seen by A

eg. I_A : 200.1.2.10
S_A : 255.255.255.128

I_B : 200.1.2.69
S_B : 255.255.255.192

I_A : 11001000.00000001.00000010.00001010
& 11111111.11111111.11111111.10000000
200.1.2.0

I_B &
S_A
200.1.2.0

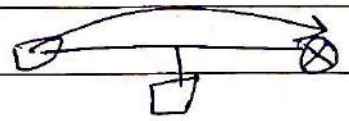
According to A,
$NID_A A = NID_A B$

I_B &
S_B
200.1.2.64

I_A &
S_B
200.1.2.0

But acc. to B,
$NID_B A \neq NID_B B$

SM: 255.255.255.255



only communicate through router.

Not sent data to any host (even though in same network) directly.

e.g. D.B.A :- 200.1.15.255

11001000.00000001.0000 1111.11111111

HID

/20 /21 /22 ... /32

any one is possible

Supernetting OR Aggregation

Rules of Supernetting

1) Contigous
2) Same Size (2^n)
3) I NID | (2^m)

200.1.0.0/24

200.1.1.0/24

200.1.2.0/24

200.1.3.0/24

Total size = 4×2^8

$= 2^{10}$

200.1.0.0

div by 2^{10} ✓

Supernet Mask :- 1's :- Fixed,

0's :- Variable

200.1. 0000 0000.0000 0000

200.1. 0000 0001.0000 0000

200.1. 0000 0010.0000 0000

200.1. 0000 0011.0000 0000

255.255.252.0 → Super-net mask

Supernet ID :-

200 . 1 . 0 . 0
& 255 . 255 . 252 . 0
200 . 1 . 0 . 0 ✓ Network ID

eg. 200.1.32.0/24
200.1.33.0/24
⋮
200.1.47.0/24

$16 \times 2^8 = 2^{12}$

200.1.0001 0000.00000000

Supernet mask :-

255.255.240.0

Supernet ID :- 200.1.32.0/20

eg. 100.1.2.0/25
100.1.2.128/26
100.1.2.192/26

100.1.2.128/25 →

100.1.2.0/24

Private IP Addresses

* 10.0.0.0 to 10.255.255.255 (Class A)
* 172.16.0.0 to 172.31.255.255 (Class B)
* 192.168.~~255.255~~ 0.0 to 192.168.255.255 (Class C)

$2^{24} > 16\ M$

16 × classes $= 16 \times 2^{16} > 1\ M$

256 classes $= 2^8 \times 2^8 = 2^{16} > 64K$

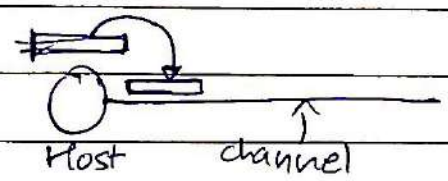
Gate 12 ~~254~~ 245.248.128.0/20 (A)

2. Flow Control Methods

Delays in CN

① Transmission Delay

T_t



eg. BW = 1 bps

Data size = 10 bits

$T_t = 10\,s$

eg. L = 1000 bits, BW = 1 Kbps

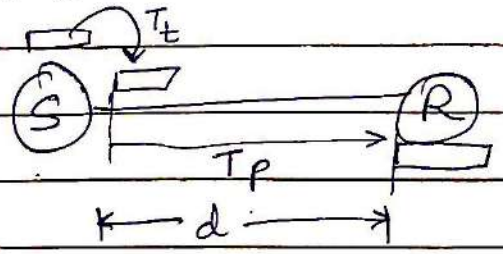
$T_t = \frac{L}{B} = 1\,s.$

eg: L = 1 Kb BW = 1 Kbps

$T_t = \frac{1024\ b}{1000\ bps} = 1.024$

Data	BW
K - 1024	K - 1000
M - 1024×1024	M - 10^6
G - $(1024)^3$	G - 10^9

② Propogation Delay



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

$v \sim 3\times10^8 \times 0.7$ (optical fibre)

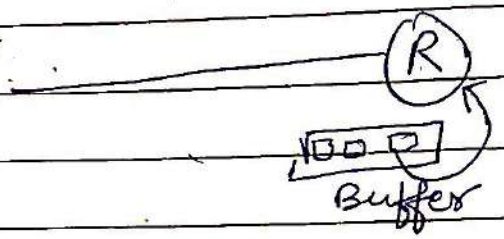
$= 2.1\times10^8\ m/s$

eg. $d = 2.1\ Km$, $v = 2.1 \times 10^8\ m/s$

$$T_p = \frac{2.1 \times 10^2}{2.1 \times 10^8} = 10^{-5}\ s$$

$$= 10\mu s$$

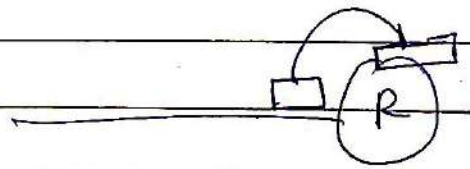
Queuing Delay



$(\simeq 0)$

(No formula to calculate Queuing Delay)

Processing Delay



$(\simeq 0)$

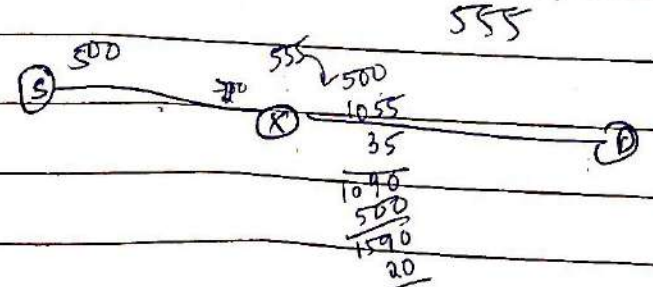
(No formula to calculate Processing delay)

Gate 15. $T_b = \frac{5000}{10^7}\ s = 500\mu s$

1000
555

520 + 35

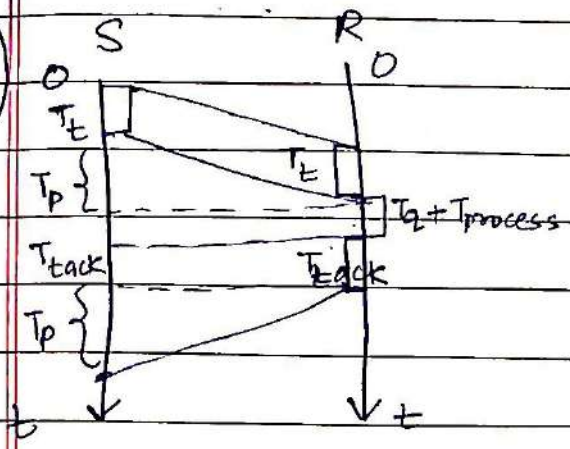
555



1055
500
1555

Flow Control Methods :-

STOP AND WAIT



Total time =

$$T_{tdata} + T_p + \cancel{T_q}^{\simeq 0} + \cancel{T_{process}}^{\simeq 0} + T_{tack} + T_p$$

$$= T_{tdata} + 2T_p + T_{tack}$$

Generally, $T_{tack} << T_{tdata}$

$$\text{Total time} \simeq T_{tdata} + 2T_p$$

Efficiency,

$$\eta = \frac{\text{Useful time}}{\text{Total Cycle time}} = \frac{T_t}{T_t + 2 * T_p}$$

$$= \frac{1}{1 + 2 * \frac{T_p}{T_t}}$$

$$= \frac{1}{1 + 2a}, \quad a = \frac{T_p}{T_t}$$

Throughput (Effective Bandwidth or Bandwidth utilization) $= \dfrac{L \text{ bits (1 packet)}}{T_t + 2 * T_p}$

$$= \frac{\left(\frac{L}{B}\right) \times B}{T_t + 2 * T_p} = \frac{T_t}{T_t + 2 * T_p} \times B$$

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

eg. $T_t = 1ms$, $T_p = 1ms$

$RTT = 2ms \; (2 \times T_p)$

(Round Trip time)

$\eta = ?$

$$\eta = \frac{1}{1 + 2 \times \frac{1}{1}} = \frac{1}{3} = 33.3\%$$

eg. $T_t = 2ms$, $T_p = 1ms$

$$\eta = \frac{1}{1 + 2 \times \frac{1}{2}} = \frac{1}{2} = 50\%$$

eg. If $\eta \geq 0.5$, $T_t \overset{\text{relation}}{\Leftrightarrow} T_p$?

$$\frac{1}{1 + 2\frac{T_p}{T_t}} \geq 0.5$$

$$\Rightarrow 1 \geq 0.5 + \frac{T_p}{T_t}$$

$$\Rightarrow \frac{T_p}{T_t} \leq 0.5 \quad \Rightarrow T_p \leq \frac{T_t}{2}$$

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

or $\frac{L}{B} \geq 2 * T_p \Rightarrow \boxed{L \geq 2 * T_p * B}$

eg. $B = 4\ mbps$

$T_p = 1ms$

$L = ?$ so that $\eta \geq 0.5$

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

$$= 2 \times 1ms \times \frac{4 \times 10^6}{s}$$

$$= 8 \times 10^3 \text{ bits}$$

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

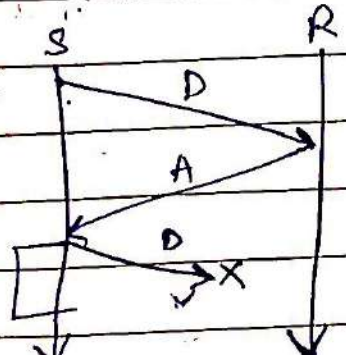
- $$\eta = \frac{1}{1 + 2 \times \frac{T_p}{T_t}}$$

$$= \frac{1}{1 + 2 \times \frac{d}{v} \times \frac{B}{L}}$$

- $d \uparrow \quad \eta \downarrow$ stop & wait is (Better for LANs) (Not good for WANs)

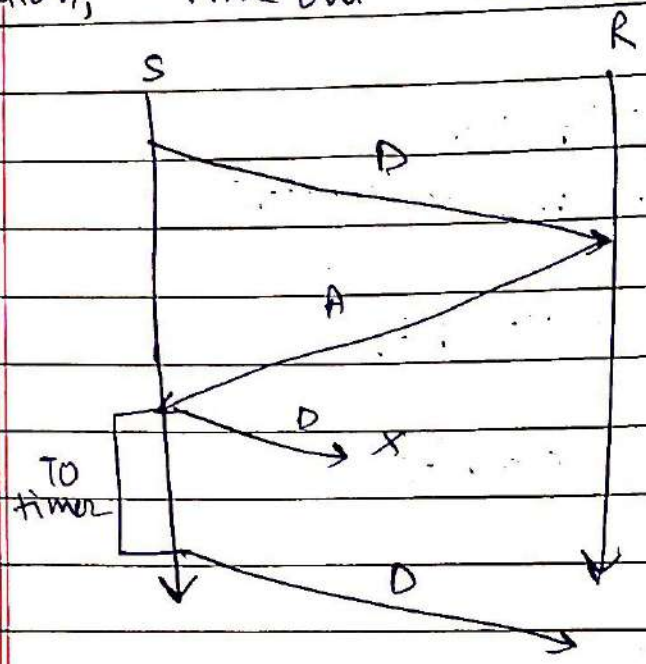
- $L \uparrow \quad \eta \uparrow$ (Better for big packets)

Problem within Stop and wait



Deadlock

Solution- Have a time out timer. If no ack. signal comes within TO timer duration, time out occurs



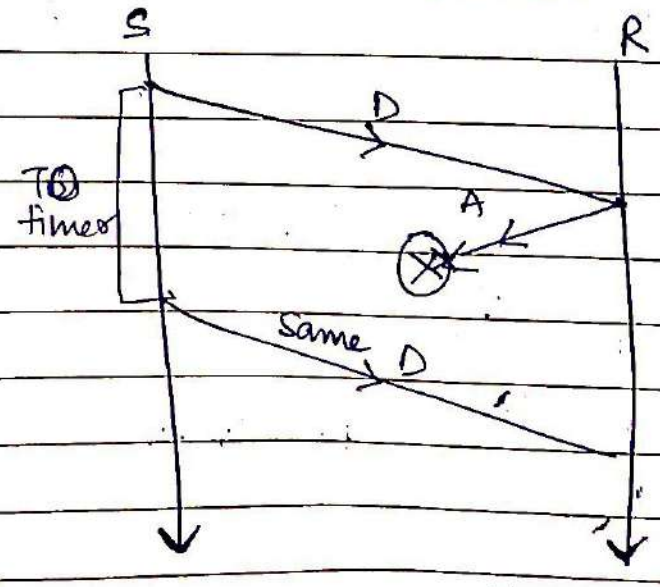
Problem Name:-

Data Problem Lost ⚠

S&W + TO timer = S&W ARQ

Automatic repeat request

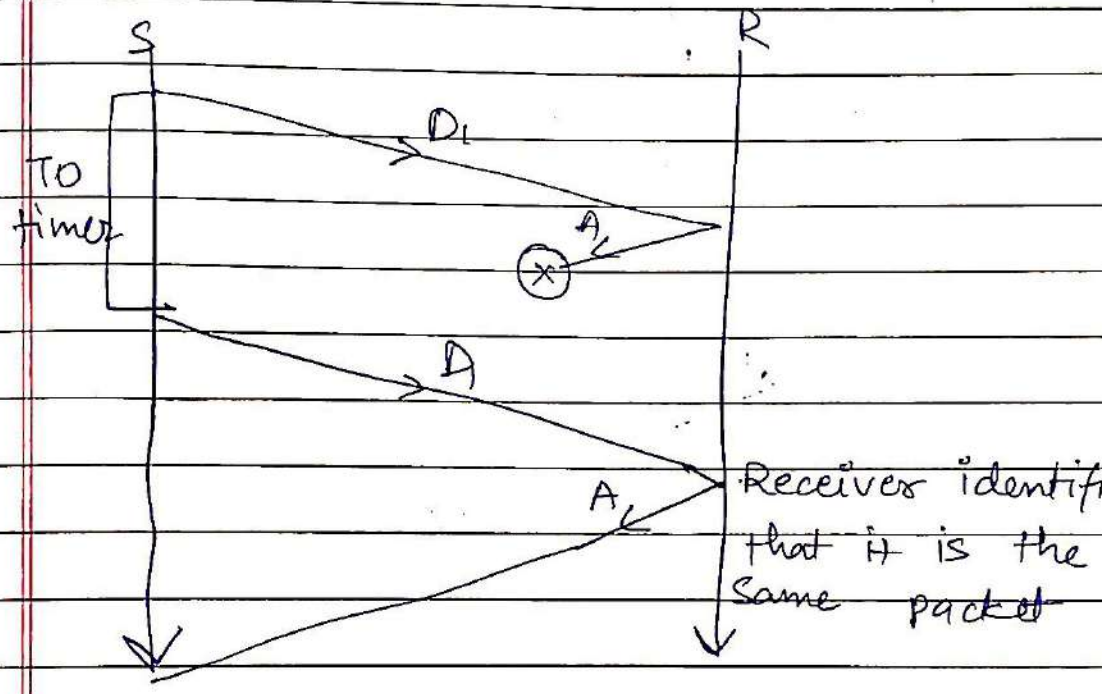
Next problem :- Acknowledgement lost



Problem solved with TO timer in reference to sender

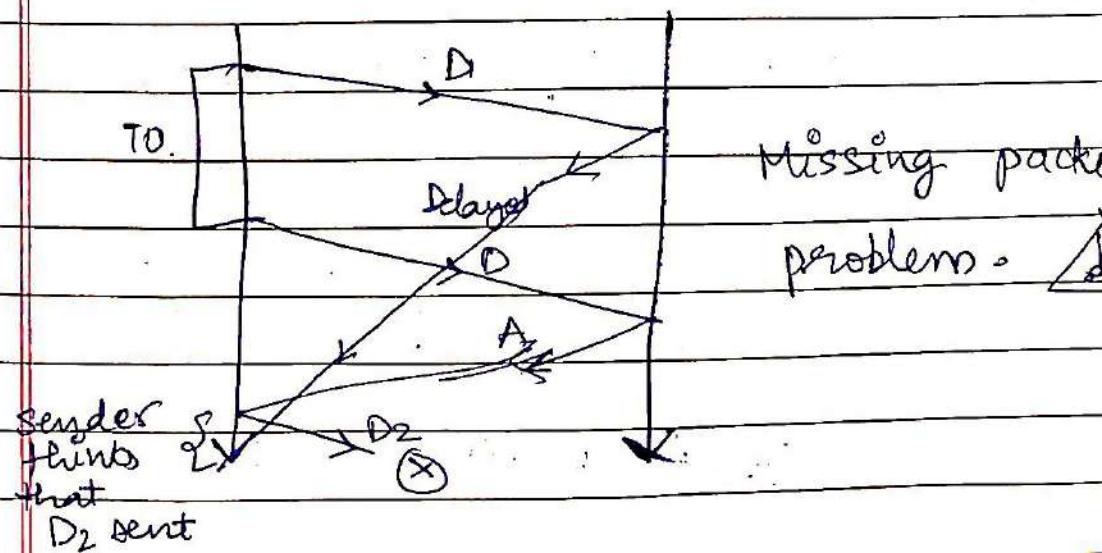
But receiver re-processes the same data thinking it is different data.
→ Duplicate Packet Problem. ⚠

Solution- Number on packets.



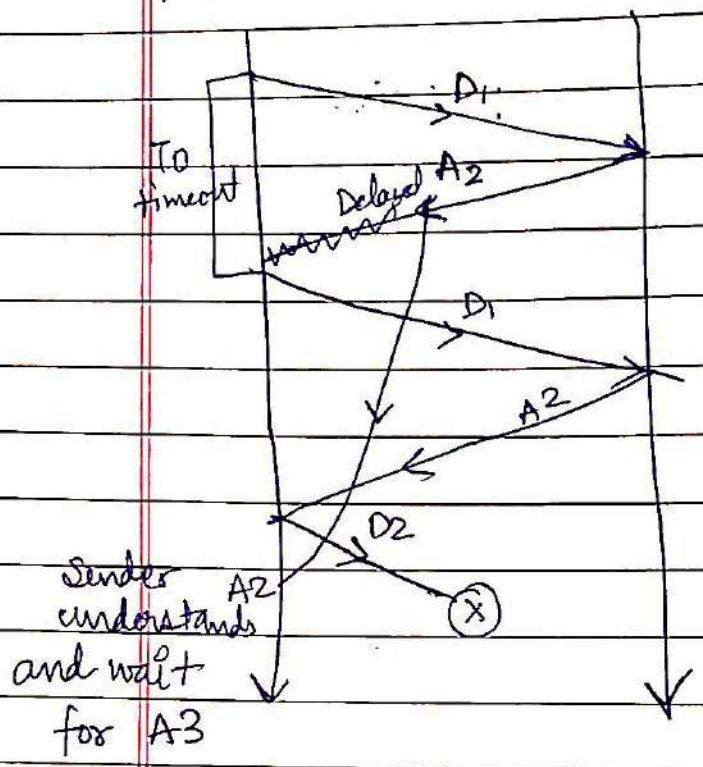
∴ S&W + TO timer + Sequence Numbers to data

Next Problem :- Acknowledgment Delayed.



Missing packets problem. ⚠

Solution - Numbers on acknowledgment also.



S&W + TO + Seq. Number on data + Seq. Number on Acknowledg.

= All problems solved

eg: S&W 10 packets S→R

Every 4th packet is lost

1 2 3 4 4 5 6 7 7 8 9 10 10

#(packets) transmitted = 13

eg. S ——→ R

Error probability = 0.2

400 packets sent. #(packets) trans. = ?

$400 \times 0.2 = 80$

$400 + 80$

↓

Out of these 80 again 20% lost.

Thus #(packets)

$= 400 + 400 \times 0.2 + 400 \times 0.2 \times 0.2 + \dots$

$= \frac{400}{1-0.2} = \frac{\cancel{400}^{50}}{0.\cancel{8}} \times 10 = 500$

• for n packets, p – error probability

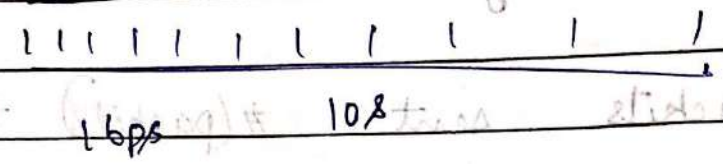
$\#(\text{packets}) = n + np + np^2 + np^3 + \dots$

$= \frac{n}{1-p}$

Gate: 64 Kb/s. $\eta \geq 0.5$ $\frac{1}{1+2\times\frac{T_p}{t}\times B_w}$

$L \geq 40 \times 64 \times 8 = 320\ b$

Capacity of channel / pipe



capacity = $Bw * T_p$ (Half Duplex)

= $2 * Bw * T_p$ (Full Duplex)

Problem with stop & wait :-

However if the channel capacity is too high, we use the channel for a single packet only.

For thick pipe (High capacity)
↳ stop & wait ✗ NOT useful

For thin pipe
↳ stop & wait ✓ useful

Pipelining

T_t sec – 1 packet

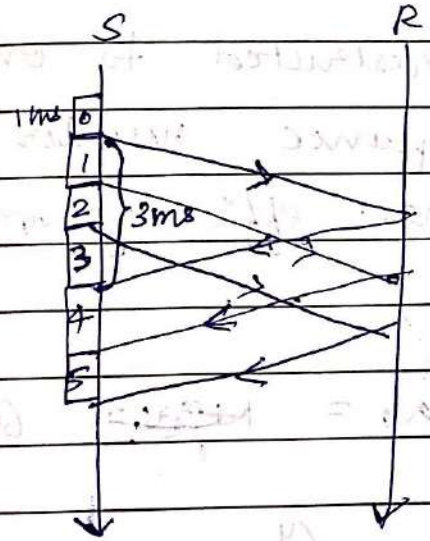
1 sec – $\frac{1}{T_t}$ packets

$$T_t + 2 * T_p - \frac{T_t + 2 * T_p}{T_t} \text{ packets}$$

$$= (1+2a) \text{ packets}$$

eg. $T_t = 1ms$, $T_p = 1.5ms$

$$\eta = \frac{1}{1+2a} = \frac{1}{4} = 25\%$$



Sliding window protocol

At any point, window contains atleast 4 packets.

Packets present to right of window - transmitted

left - yet to be transmitted

3	2	1	0

window min. size = 4 packets

$$W_s \geq (1+2a) \text{ packets}$$

Min. no. of sequence numbers needed to transmit :- $\boxed{0, 1, 2, 3}$ ✓

2 bits only.

$W_{s\,min} = 1+2a$, min. seq $= 1+2a$

$\#(\text{bits})_{min} = \lceil \log_2(1+2a) \rceil$

eg. $T_t = 1ms$, $T_p = 49.5\ ms$

$W.S. = 1 + 2a = 100$

Seq. number $= 100$

$\#\ bits = \lceil \log_2 100 \rceil = 7$

eg: For $T_t = 1ms$, $T_p = 49.5\ ms$

If we are restricted to only 6 bits in the sequence number field. what is the max. efficiency we could get ?

Sol: Seq. number max = 64

$\therefore \eta_{max} = \frac{64}{100} = 64\%$

$W.S. = \min(1+2a, 2^N)$

eg: $T_t = 1ms$, $T_p = 99.5\ ms$

$WS = 1+2a = 200$

$\#\ bits = \lceil \log_2 200 \rceil = 8$

If Max bits = 7

$\eta_{max} = \frac{128}{200} = 64\%$

Sliding window protocol

GBN — Go Back N

SR — Selective Repeat

1)
2) Three parameters —
3)

1)
2)
3)

Go Back N

1) Sender window size in GBN is N.

Note :- $N > 1$; if $N = 1$, it is simply stop and wait and no pipelining is there.

eg. $T_t = 1\,ms$, $T_p = 49.5\,ms$, $\eta = ?$ for GB10

$N = 10 = (1+2a)_{max}$

$$\eta = \frac{10}{1+99} = 10\%$$

eg. If $BW = 40\,mbps$

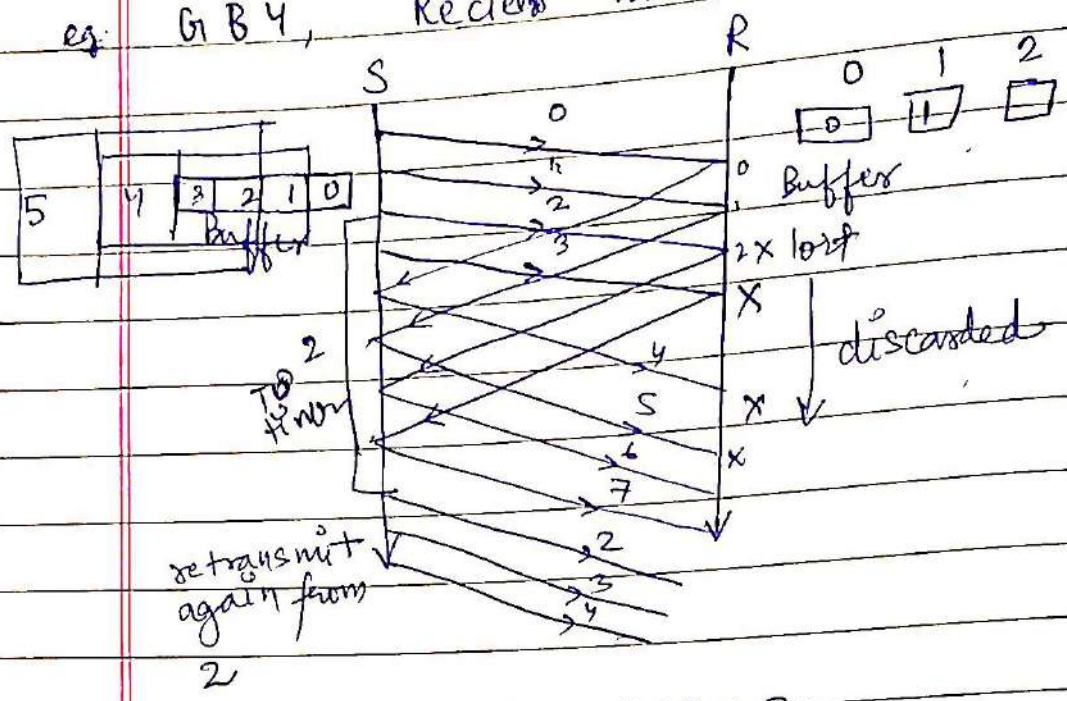
Throughput $= ? = \eta \times BW$

$= 0.1 \times 40\,mbps$

$= 4\,mbps$

2) Receiver window size $= 1$ in any GBN

eg. GBN, Reciever window size = 1



(Go back 4) 5 → 4 → 3 → 2

eg. In GB3, if every 5th packet that is being transmitted is lost and if we have to send 10 packets, then how many transmissions are required?

Sol. 1 2 3 4 5 6 7 5 6 7 8 9 7 8 9

8 10 9 10

18 packets

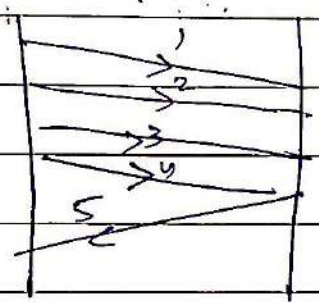
eg. In GB4, if every 6th packet that is being transmitted is lost and if we have to send 10 packets, how many transmissions are required?

Sol:- 1 2 3 4 5 6 7 8 9 6 7 8 9 10

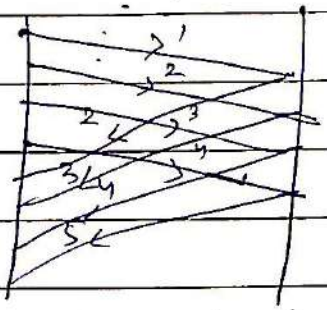
8 9 10

17 packets

3) Acknowledgement :-



Cummulative acknowledgment

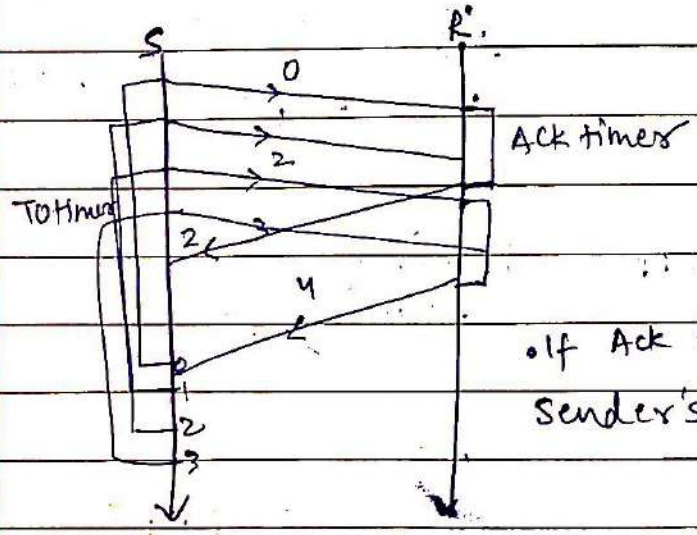


Independent acknowledgement

adv :- less traffic
disadv :- less reliable

adv :- high reliable
disadv :- high traffic

- Go Back N uses Cummulative acknowledgement

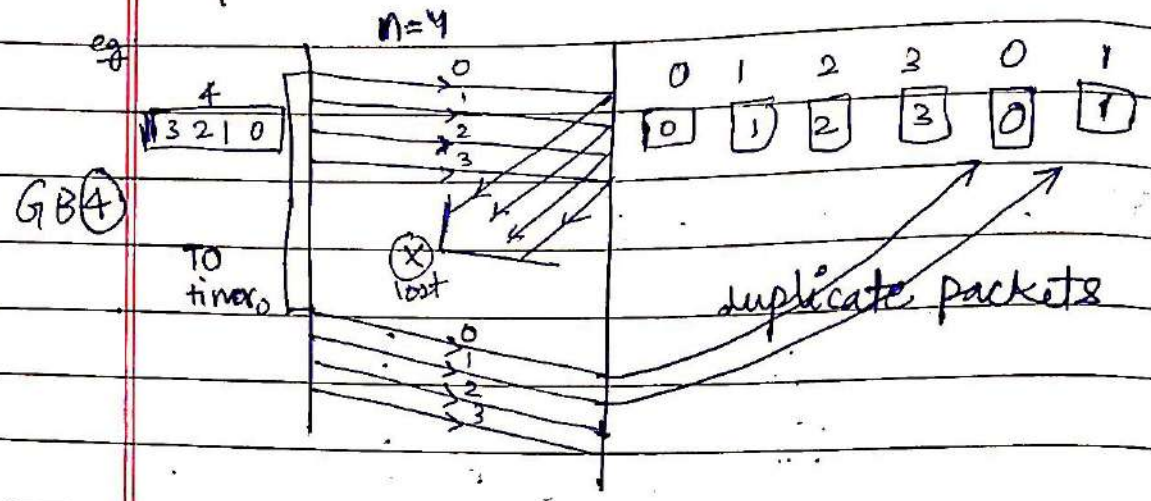


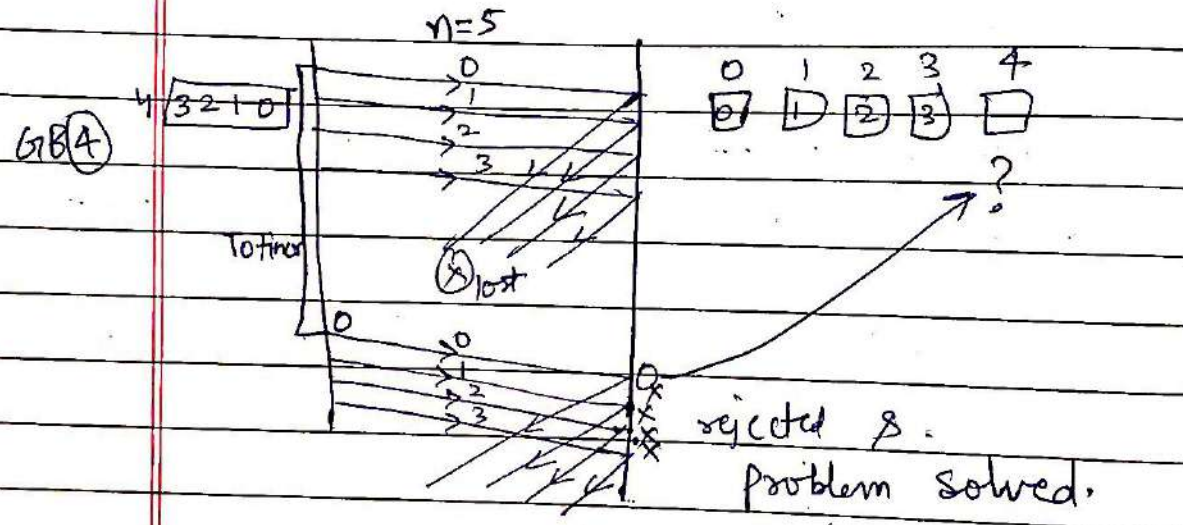
- If Ack timer is too large, sender's TO may occur

- If Ack timer is too small, every packet there will be acknowledgement.

- Relationship b/w window sizes and sequence numbers in Go back N

eg





For GBN, receiving window size = 1

(Sequence Numbers) required to detect duplicate no's = N+1

also; #(Seq. No.) ≥ (W_S + W_R)

Sender ; Receiver

(1) $w_S = N \quad w_R = 1, \quad seq = N+1$

bits $= \lceil \log_2 (N+1) \rceil$

(2) $seq = N, \quad w_S = N-1 \quad w_R = 1$

for $seq = 4, \quad w_S = 3, \quad w_R = 1$

(3) bits $= k, \quad seq = 2^k, \quad w_S = 2^k - 1,$

$w_R = 1$

Selective Repeat

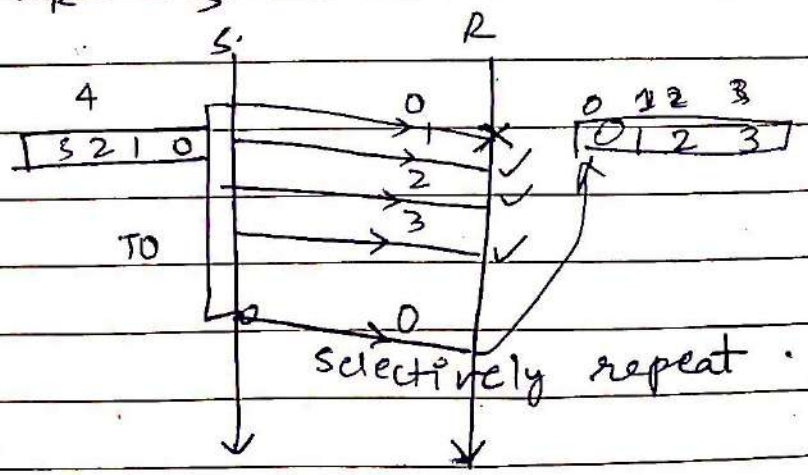
1) $w_S > 1, \quad w_S \in Z^+$

eg. $T_t = 1ms, \quad T_p = 49.5\,ms, \quad w_S = 50, \quad SR, \eta = ?$

$$\eta = \frac{50}{1 + 2 \times 49.5}$$

$$= \frac{1}{2} = 50\%$$

2) $w_R = w_S$



selectively repeat.

eg. $W_S = 3$, 10 packets to be sent, every 5th packet is lost. #(packets sent) = ?

1 2 3 4 5 6 5 7 8 9 9 10

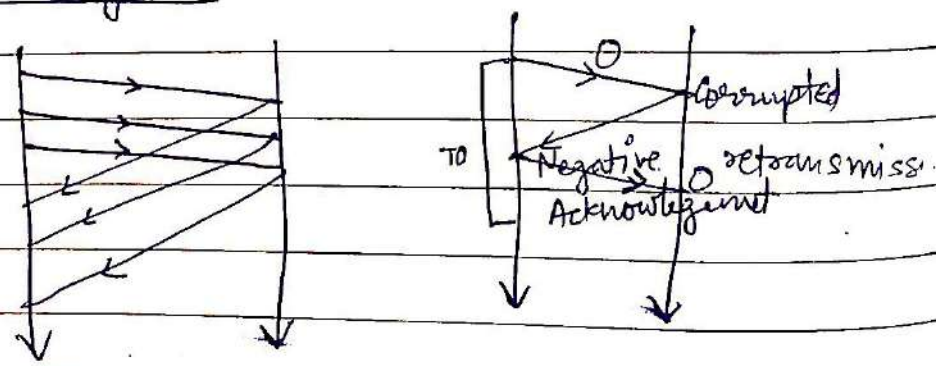
$\boxed{12}$ packets transmitted

eg. 10 packets → every 4th lost

1 2 3 4 5 6 4 7 8 9 7 10 10

$\boxed{13}$ packets transmitted

3) Acknowledgments :-

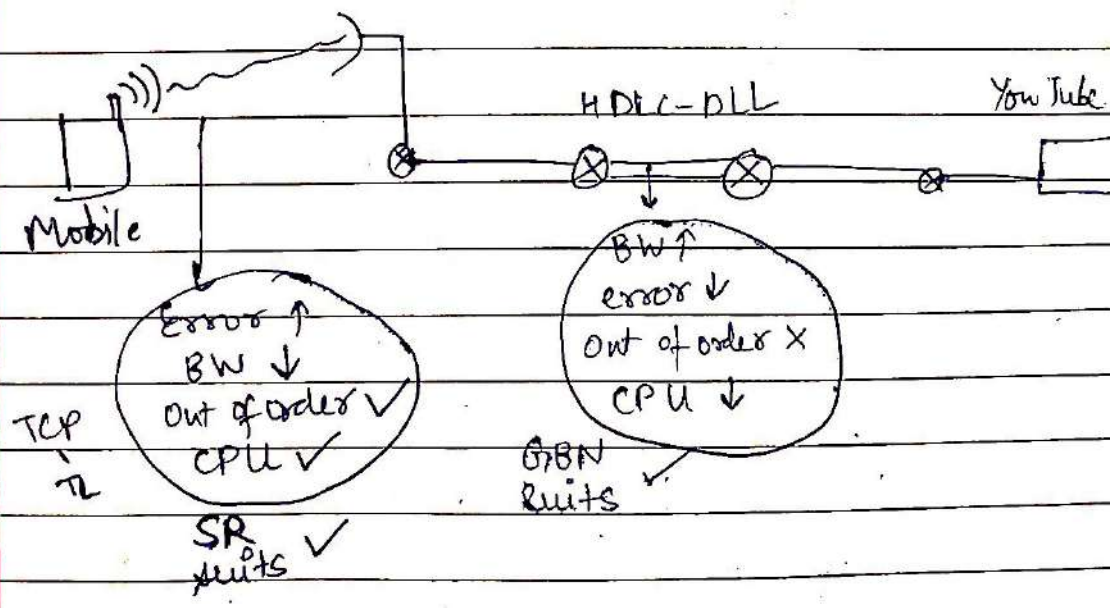


Independent acknowledgment

For missing packet, both GBN & SR discard silently
For corrupted packet, GBN discard silently while, SR sends neg. ack.

	Stop & wait	GBN	SR
η	$\frac{1}{1+2a}$	$N/1+2a$	$N/1+2a$
Buffers	$1+1=2$	$N+1$	$N+N=2N$
Sequence Numbers	$1+1=2$	$N+1$	$N+N=2N$
Retransmission (1 packet lost)	1	N	1
Bandwidth	low	high	moderate
CPU	low	moderate	high
Implement-ation	low	moderate	complex

eg



link to link — GBN protocol
end to end — SR protocol

Gate 16, ~~1024 × 1024~~

64

$T_t = \frac{8 \times 1024 \text{ bit}}{128 \times 10^3 \text{ bit/s}}$

$= 64 \times 10^{-3}$ s $= 64$ ms

$W_s = 1 + 2 \times \frac{150}{64}$

$= 1 + \frac{150}{32}$ 75 16

$= \frac{91}{16}$ 5.

W_s 5

$5 \times 2 = 10$

bits $= 4$ B

Gate05

2^n seq. No's

$\frac{2^n}{2} \leq W_s$ 2^{n-1} B

Gate 03:

1000 bytes

$\frac{1000 \times 5 \text{ bytes}}{100 \text{ s}}$

~~= 12.5 b/s~~ 25×4

B

Gate 06

$$1+2a = 1+2\times\frac{40\times10^{-3}}{32\times8}\times128\times1024$$

$$= 1+2\times40\times10^{-3}\times4\times128$$

$$= 1+40.96$$

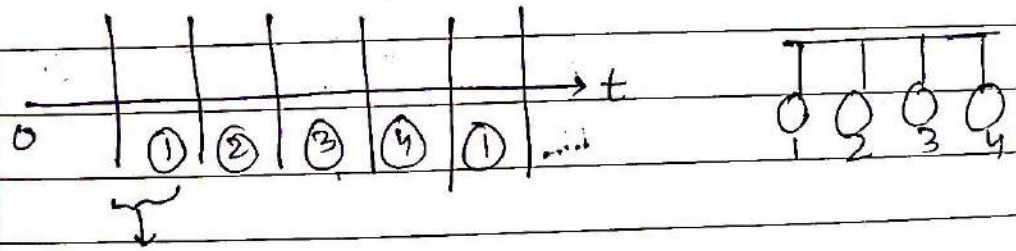
$$= 41.096 \quad (41)$$

Access Control Methods

links

- BC links (Broadcast)
- PP links (Point-to-point).

TDM (Time Division Multiplexing)



Time Division $= T_t + T_p$

$$\eta = \frac{\text{Useful time}}{\text{cycle time}} = \frac{T_t}{T_t+T_p} = \frac{1}{1+a}$$

Eg: $T_t = 1\,ms$, $T_p = 1\,ms$, $\eta = \frac{1}{2}$ (50%)

eg. BW = 4 mbps,

E.B.W (Thr.) = 0.5 × 4 = 2 mbps

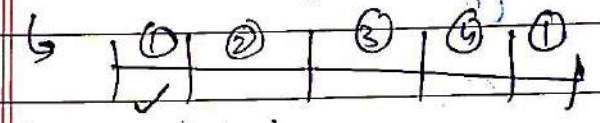
eg. N station with each station requiring 2 k.bps bandwidth. Max. no. of stations connected?

Sol. BW = 4 mbps we use = 2 mbps

2 kbps

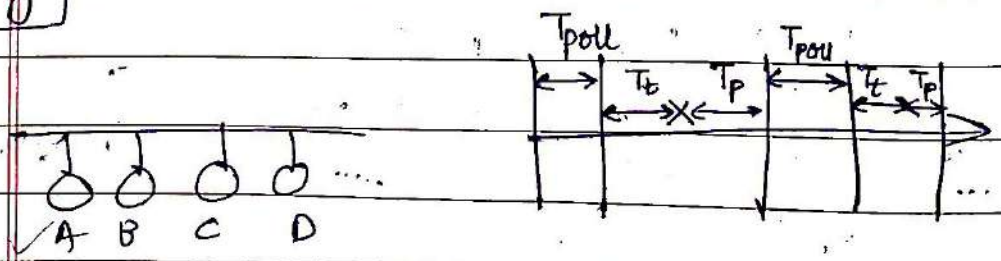
N = 1000 stations

Disadvantage of TDM



slots wasted iff ① has no data to transmit.

Polling



Disadvantage

↓

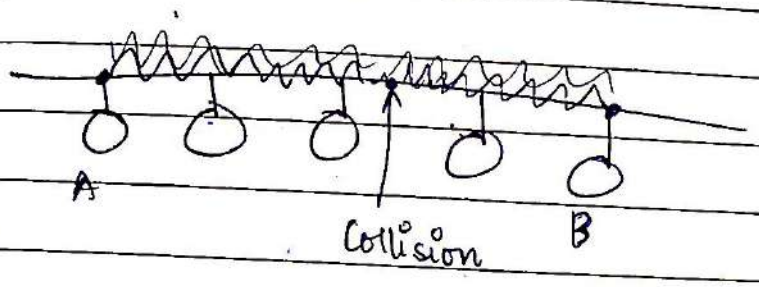
Some stations may starve

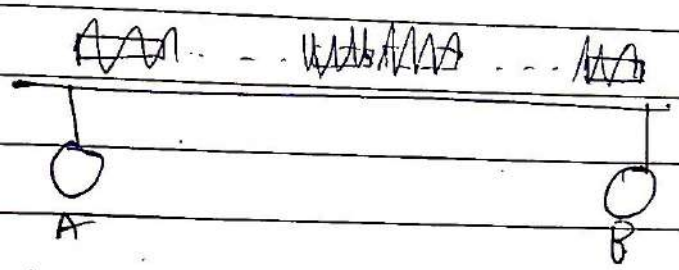
$$\eta = \frac{\text{useful time}}{\text{cycle time}} = \frac{T_t}{T_{poll} + T_t + T_p}$$

CSMA / CD

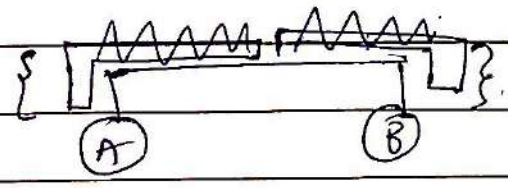
Carrier Sensing — Multiple Access — Collision Detection

(No Acknowledgement)

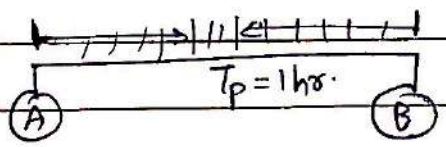




If the packet is large enough that by the time collision signal comes back to A, A still has data to transmit; then A can be sure that it's its own data that was lost in collision else NOT



eg.



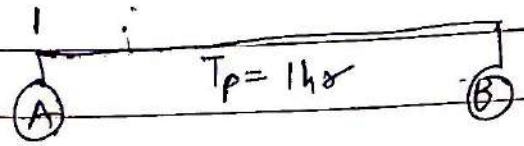
t=10.00AM, A & B starts transmitting

t=10:30AM, Collision

t=11:00 AM, A & B. see collision signal

For this eg. $T_t > T_p$ then only this could be sure

In worst case,



$t = 10:00$ AM A started transmission

$t = 10:59:59$ AM 1st bit of A's data is about to reach B and B 1st bit started

$t = 11:00$ AM Collision

$t = 12:00$ A will see collision signal

∴ In worst case, to A to ~~get~~ detect collision data,

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

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

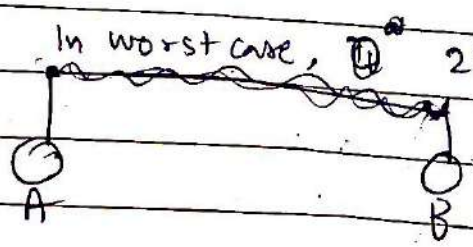
$$\Rightarrow \boxed{L \geq 2 \times T_p \times B}$$

minimum length of packet so as to detect collision.

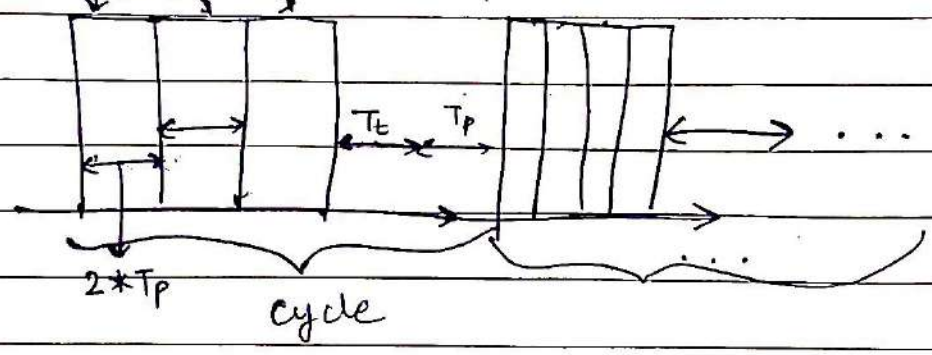
Efficiency of CSMA/CD :-

or efficiency of Ethernet (uses)

In worst case, 2 T_p time wasted



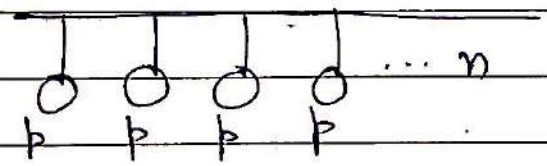
(Collision slot) → we don't know



$$\eta = \frac{T_t}{c \times 2 * T_p + T_t + T_p}$$

Probabilistic analysis to find 'c' (# (collisions))

1) n stations, p ← every station wants to send a data with probability 'p'



2) Prob. for success is when only one station transmits the data

$$P_{success} = {}^nC_1 * p * (1-p)^{n-1}$$

To find :-
max ($P_{success}$)

$$\frac{dP_{succ}}{dp} = 0 \Rightarrow p = 1/n, \quad P_{success} = \max.$$

$$\Rightarrow -(n-1)(1-p)^{n-2}\, {}^nC_1\, p + {}^nC_1 (1-p)^{n-1} = 0$$

$p - np + 1 - p = 0$ $\Rightarrow (1-p)^{n-2}\, {}^nC_1 \left[(1-n)p + (1-p)\right] = 0$

$p = 1/n$ $\Rightarrow p = 1/n$

$$P_{max} = n \times 1/n \times (1 - 1/n)^{n-1}$$

$$= (1 - 1/n)^{n-1}$$

$$\lim_{n \to \infty} P_{max} = 1/e$$

3) No. of times we should try before getting the first success

$$1/P_{max} = 1/(1/e) = e$$

∴ # (collision) before successful transmission according to poisson's distⁿ = e

$$\therefore \eta = \frac{T_t}{e \times 2 \times T_p + T_t + T_p}$$

$$= \frac{T_t}{e \times 2 \times T_p + T_t + T_p}$$

$$\simeq \frac{1}{1 + 6.44a}$$

$$\eta = \frac{1}{1 + 6.44\left(\frac{d}{v}\right)\left(\frac{B}{L}\right)}$$

→ $d \uparrow$ $\eta \downarrow$ (useful for LANs NOT WANs)

→ $L \uparrow$ $\eta \uparrow$ (Bigger packets suitable)

Gate'15:

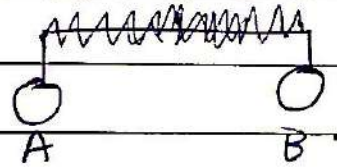
10^8 bps $\qquad L = 1\,km = 10^3\,m.$

1250 bytes $= 10000$ bits $= 10^4$ bits

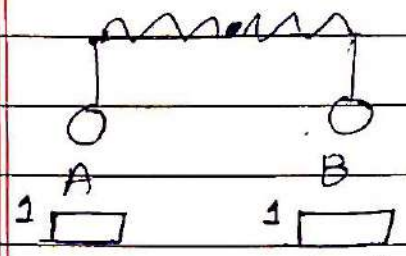
$$T_t \geq 2 * T_p$$

$$\frac{L}{B} \geq 2 * \frac{d}{v} \qquad v \geq 20000\ km/s$$

Back-off algorithm for CSMA/CD



→ this will keep on happening (they re-transmit at same time again & again)



$n = 1 \qquad n = 1$ (collision number)

$(0, 2^{n-1})$

$(0,1)$ $\qquad$ $(0,1)$

A $\qquad$ B

choose Rand. No. from this range

0 $\qquad$ 0

0 $\qquad$ 1 → A's waiting time $= 0 * T_{slot}$,

1 $\qquad$ 0 $\qquad$ B's waiting time $= 1 * T_{slot}$

1 $\qquad$ 1

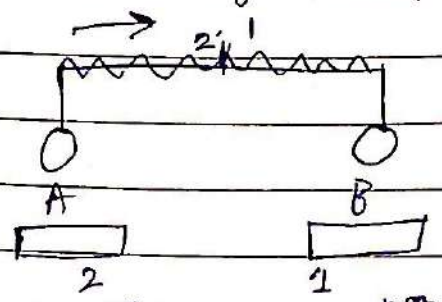
$P(A \text{ winning}) = 1/4$

$P(B \text{ winning}) = 1/4$

$P(\text{Collision again}) = 1/2$

let

Case I (A won)



Now this first collision for 'A' and second collision for 'B'

$(0, 2^{n=1})$ $\quad$ $(0, 2^{n=2})$

~~(0,1,2,3)~~ $\quad$ ~~(0,1)~~

$n=1$ $\quad$ $n=2$

(0,1) $\quad$ (0,1,2,3)

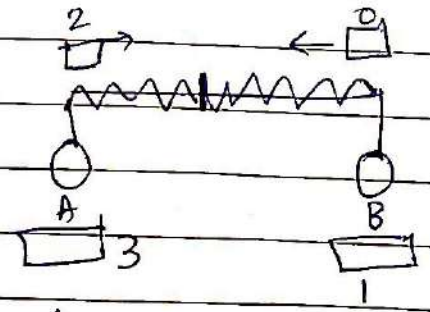
0	0	→ Collision (C)
0	1	→ A
0	2	→ A
0	3	→ A
1	0	→ B
1	1	→ C
1	2	→ A
1	3	→ A

$P(A) = 5/8, \quad P(B) = 1/8, \quad P(C) = 1/4$

let

Again

(A won)



Now this is first collision for 'A' & third for 'B'

$n = 1$ $\quad$ $n = 3$

(0,1) $\quad$ (0,1,2,3,4,5,6,7)

A	B		
0	0	→	C
0	1	→	A.
0	2	→	A
0	3	→	A
0	4	→	A.
0	5	→	A.
0	6	→	A
0	7	→	A.
1	0	→	B
1	1	→	C
1	2	→	A.
1	3	→	A.
1	4	→	A.
1	5	→	A.
1	6	→	A.
1	7	→	A.

$P(C) = \frac{1}{8}$

$P(A) = \frac{13}{16}$

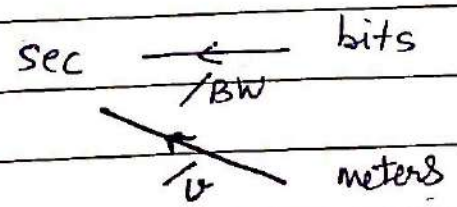
$P(B) = \frac{1}{16}$

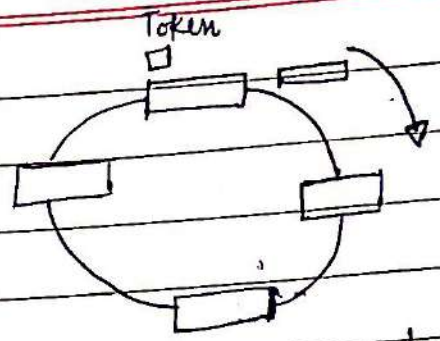
∴ collision prob. is ↓ continously.

- If A's won then again prob of 'A' winning again ↑ and this continues

↳ "Capture Effect"

Token Passing Access Control Method





Ring latency :-

Time taken by a single bit to start from a location in token ring and turn around to come back at same location.

$$\text{Ring latency} = \left(\frac{d}{v} + \frac{(N \times b)}{B}\right) \text{ sec.}$$

d :- length of ring

v :- velocity of packet

N :- No. of stations

b :- bit delay by which every station delays a packet

Cycle time :- Time taken by a token to start at a location & come back at same location.

THT (Token holding time) :- Time that every station holds a packet

$$\text{Cycle time} = \left(\frac{d}{v} + N \times THT\right)$$

$$= (T_p + N \times THT)$$

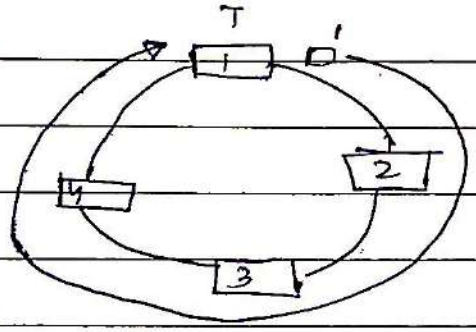
$$\eta = \frac{\text{Useful time}}{\text{cycle time}}$$

$$= \frac{N \times T_t}{(T_p + N \times THT)}$$

↓ Delayed Token reinsertion (DTR)

↓ Early Token reinsertion (ETR)

1) DTR :-



$$THT = T_t + R.L.$$

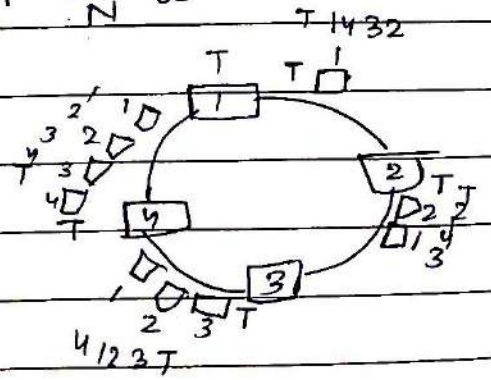
$$= T_t + T_p + \cancel{N \times b} \quad \rightarrow 0 \text{ (Assume)}$$

$$= T_t + T_p$$

$$\eta = \frac{N \times T_t}{T_p + N \times (T_t + T_p)}$$

$$= \frac{1}{1 + \frac{N+1}{N} a}, \quad a = \frac{T_p}{T_t}$$

2) ETR :-



$$THT = T_t$$

$$\eta = \frac{N \times T_t}{T_p + N \times T_t}$$

$$= \frac{1}{1+\left(\frac{1}{N}\right)a} \quad , \quad a = \frac{T_p}{T_t}$$

$$\therefore \boxed{\eta_{DTR} = \frac{1}{1+\frac{N+1}{N}a} \qquad \eta_{ETR} = \frac{1}{1+\frac{a}{N}}}$$

η ✗ $\qquad\qquad$ η ✓

Reliable ✓ $\qquad\qquad$ Reliable ✗

ALOHA

Additive Links On-line Hawaii Area

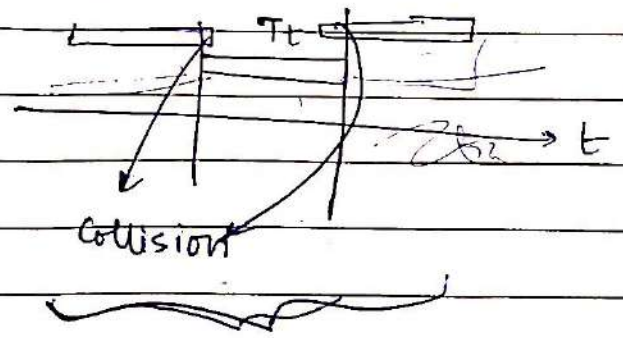
- Anytime send packets - No carrier sensing
- Collisions possible
- Acknowledgement ✓ (No need of collision detection)
  ↓
- Retransmission takes place

ALOHA

↓ Pure $\qquad$ ↓ Slotted

Pure Aloha :-

Vulnerable Time



Vulnerable time $= 2T_t$

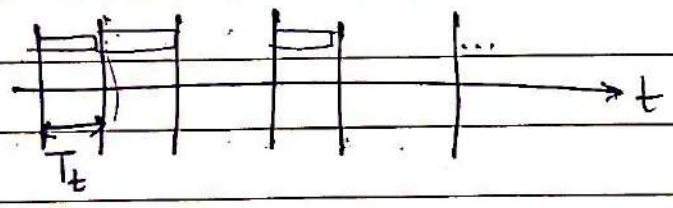
$$\eta_{\text{Pure Aloha}} = G \times e^{-2G}$$

G :- No. of stations who wants to transmit in T_t slot

$$\eta_{max} :- \frac{d\eta}{dG} = 0 \Rightarrow G = 1/2 \quad \left(\begin{array}{l}\text{In 2 time slots} \\ \text{1 station transmit}\end{array}\right)$$

$$\eta_{max} = \frac{e^{-1}}{2} = \frac{1}{2e} = 18.4\%$$

Slotted Aloha :-



Vulnerable time $= T_t$

$$\eta_{\text{Sloted Aloha}} = G \times e^{-G} \qquad \eta_{max} :- G = 1$$

$$\eta_{max} = \frac{1}{e} = 36.8\%$$

Date
Page

Flow control

Stop & wait :- $\eta = \frac{1}{1+2a}$

Sliding window :- $\eta = \frac{N}{1+2a}$

(both GBN, SR)

Access Control :-

TDM :- $\eta = \frac{1}{1+a}$

Polling :- $\eta = \frac{T_t}{T_{poll} + T_t + T_p}$

Ethernet CSMA/CD :- $\eta = \frac{1}{1+6.44a}$

Token ring

- DTR :- $\eta = \frac{1}{1+\frac{a(N+1)}{N}}$
- ETR :- $\eta = \frac{1}{1+\frac{a}{N}}$

Aloha

- Pure :- $\eta = G \times e^{-2G}$ max 18.4%
- Slotted :- $\eta = G \times e^{-G}$ 36.8%

Gate 1:

$0.1 \times 0.8 \times 0.7 \times 0.6 + 0.2 \times 0.9 \times 0.7 \times 0.6 +$

$0.3 \times 0.9 \times 0.8 \times 0.6 + 0.4 \times 0.9 \times 0.8 \times 0.7$

$= 0.44$ $\boxed{A}$

3. Error Control Methods

Error Handling

Error Detection	Error Correction
1) Send data twice, if different, ask for retransmission	1) Hamming Code
2) Parity checking	
3) CRC (cyclic Redundancy checking)	
4) check Sum	

CRC (Cyclic Redundancy Check)

CRC Generator

S $\boxed{1101}$ R

1011011 $\boxed{000}$

1011011 001 ⟶

Data CRC

⊕ Ex-OR → mod 2 sum

```
1101 ) 1011011000 (
       ⊕ 1101
       ---------
       0110011000
       ⊕ 1101
       ---------
       0001110000
       ⊕  1101
       ---------
```

```
  1100
⊕ 1101
  0001  ← CRC
```

Receiver Side :- (correct data)

```
1101 ) 1011011001 (
       ⊕ 1101
       0110011001
       ⊕ 1101
       0001110001
       ⊕ 1101
        001101
       ⊕ 1101
        0000  ✓ correct
```

zero.

Receiver Side :- (Incorrect data)

```
1101 ) 1001011001 (
       1101
       0100011001
        1101
       0100011001
        1101
       0110100 1
        1101
       0000 001  × Incorrect
```

NOT zero

eg. $x^3 + x + 1 \rightarrow$ CRC

$1 \cdot x^3 + 0 \cdot x^2 + 1 \cdot x + 1 \cdot x^0$

$\boxed{1011} \rightarrow$ CRC

eg. At sender :-

D:- 11010

CRC G.Poly :- x^3+x+1 (1011)

```
1011 ) 11010000 (
       1011
       --------
       01100000
       1011
       --------
       0111000
       1011
       --------
       0101000
       1011
       --------
       00 [010] <- CRC
```

At receiver:- (If correct)

```
1011 ) 11010010 (
       1011
       --------
       01100010
       1011
       --------
       0111010
       1011
       --------
       0101010
       1011
       --------
       00 [000] ✓ correct
```

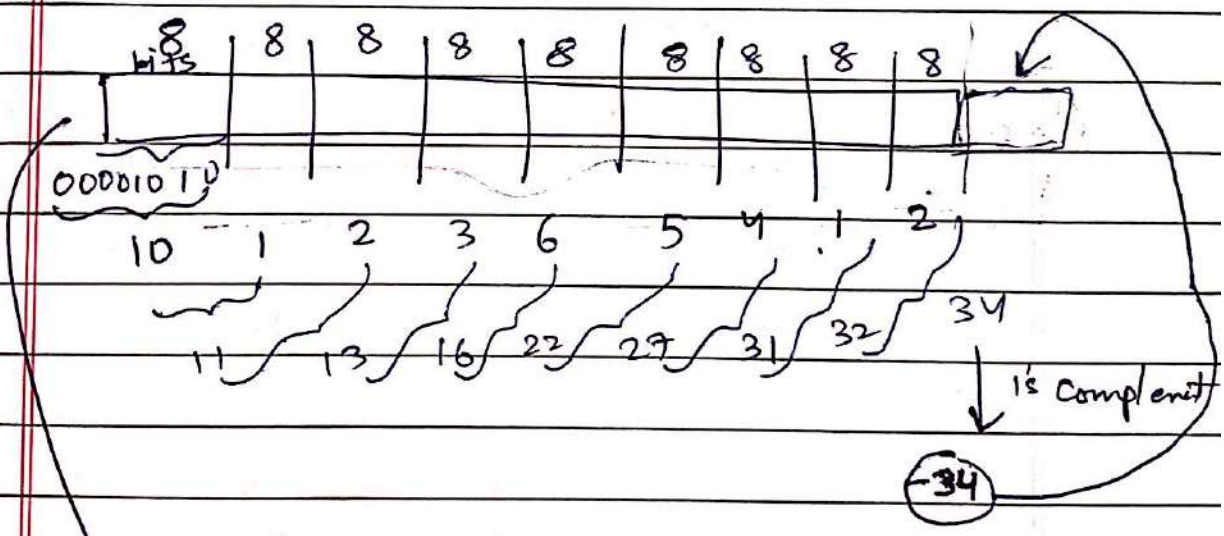
(If Incorrect)

At receiver:

```
1011 ) 1101 0110 (
       1011
       ----
       0110 0110
       1011
       ----
       0111 1110
       1011
       ----
       0100 10
       1011
       ----
       00 100
```

Incorrect

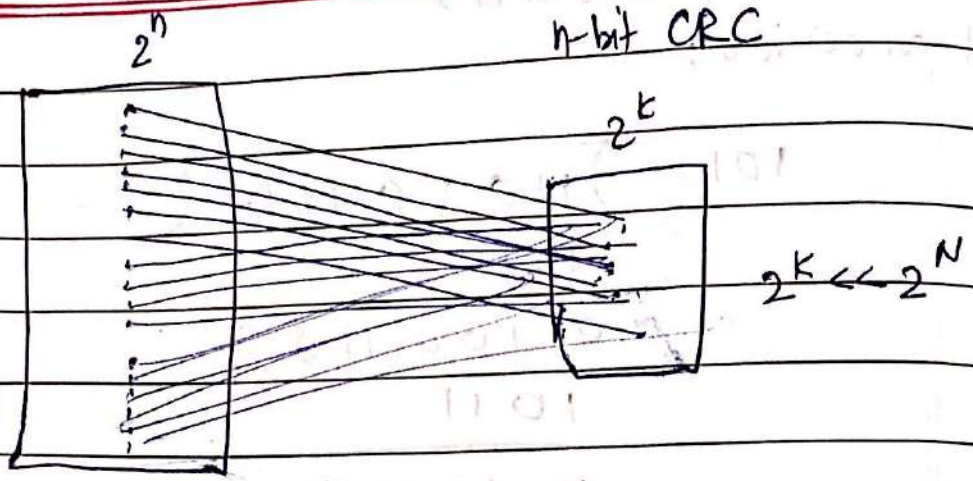
Check Sum



Receiver

checks sum and it should be zero

n-bit data



~~many~~ many-to-one f^n.

Thatswhy, 100% error detection not possible.

4. ISO/OSI Stack

Functions

Mandatory	Optional
1) Error Control	1) Encryption & Decryption
2) Flow Control	2) Checkpointing
3) Access Control	3) Routing (Option avail - Flooding)
4) Multiplexing & Demultiplexing	⋮ etc
5) Addressing	
⋮ etc	

about 70 such f^n

Grouping is done by :-

1. ISO-OSI } In Syllabus
2. TCP/IP
3. ATM
4. X.25
5. IEEE

⋮ etc

ISO-OSI Model

International Standards Organization - Open Systems Interconnection Model

Benefits of layers:-
- Divide & Conquer
- Encapsulation
- Abstraction
- Testing

Application Layer
Presentation Layer } User Convinience
Session Layer
Transport Layer - Thick
Network Layer - Complex
Data Link Layer - H/W & S/W
Physical Layer - H/W

Physical Layer

↓

Electrical, Mechanical, Functional, Procedural characteristics of physical links

Physical link	Signal
Copper	Electrical
optical	light
wireless	EM Waves

→ Transmission Modes:-

Simplex

(Simplex)

Half Duplex

(Half Duplex)

(one at a time)

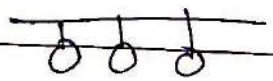
Full Duplex

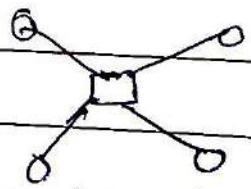
(Full Duplex)

(Both at same time)

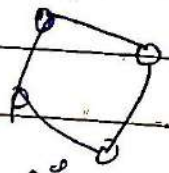
→ Topologies:-

Bus

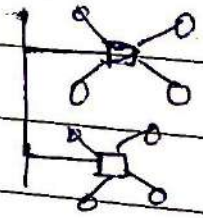




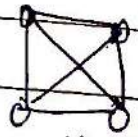
Star



Ring



Hybrid

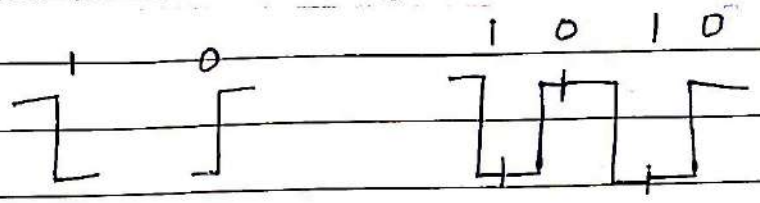


Mesh

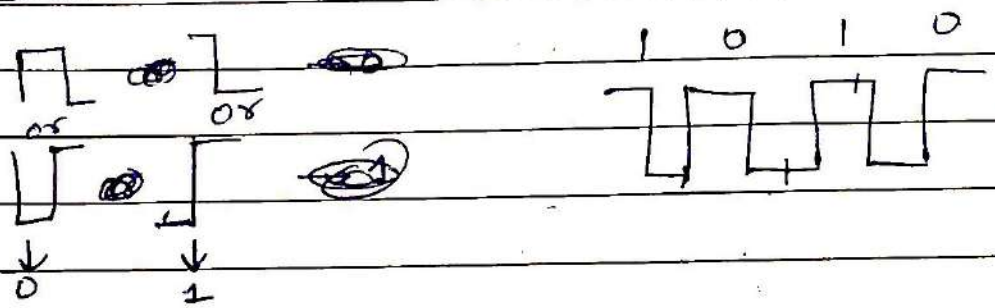
→ Encoding :-

1 0 1 0

Manchester Encoding

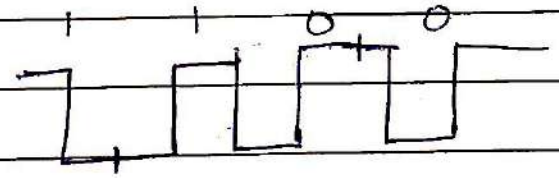


Diff. Manchester Encoding



eg. 1 1 0 0

Diff. Manchester :-



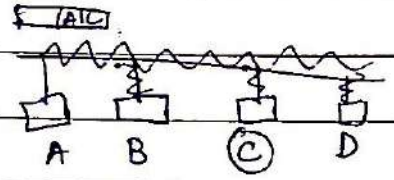
• Both in Manchester & Diff. Manchester Encoding, Band Rate = 2 × Bit-Rate

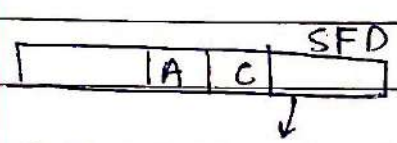
↓ voltages/s. ↓ bits/sec.

Data link layer

1) Flow Control (SPW, GBN, SR)
2) Error Control (CRC, Check Sum)
3) Framing
4) Physical Addressing
5) Access Control (CSMA/CD, Token Passing)

Framing at DLL

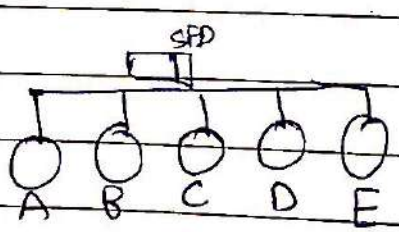




SFD → Start of Frame Delimiter

$(101010\cdots11)\ (1+0)^*$

RE → NFA → DFA → Seq. circuit recognises this pattern



Frames

- Fixed length
  - eg. L = 1000 B
  - (only SFD required)
- Variable length
  - EFD also required

Variable length Frame

length field (add length to frame)

(used in Ethernets)

Problem: If length bit gets corrupted

add ED (End delimiter)

ED		SFD

(used in token rings)

Problem:- ED matches with a data segment

End delimiter

Character stuffing

Bit Stuffing

Character Stuffing:

ED	Data	SFD

$

Problem:- Data contained '$'

$	...10$|01...	SFD

$\0

$\0\0\0

Bit Stuffing :-

SFD	Head ER	01111	ED

add a zero to differ data from ED 01111

01111

Sender :-

D: 01111

⇓

D: 011101

Receiver

011101

⇓

01111 ✓

Problem :-

D: 011101 → 011101

⇓

01111 error

New rule

D:- 011101

⇓

0111001

0111001

⇓

01110 1 ✓

eg. ED:- 01111

Sender :-

011110

⇓

0111100

0111111

⇓

0111101

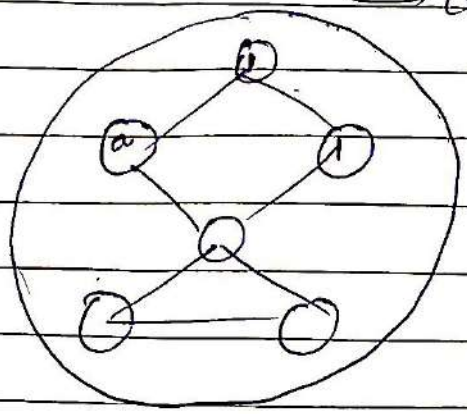
eg. ED:- 01111

Data :- 0111000 11110

Data after stuffing :- 01110000 11010

Physical Addressing

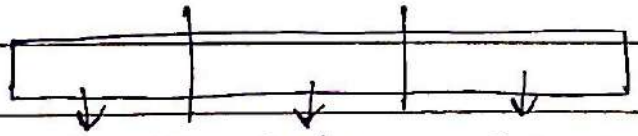
Address → Physical → unique in one network
Address → Logical → unique in entire world



Address → IP address → 32 bit S/W Number (Globally unique)
Address → Mac address → 48 bit H/W Number

NIC → ROM → unique globally
(Network Interface Card)

NIC (Mac address)
↓



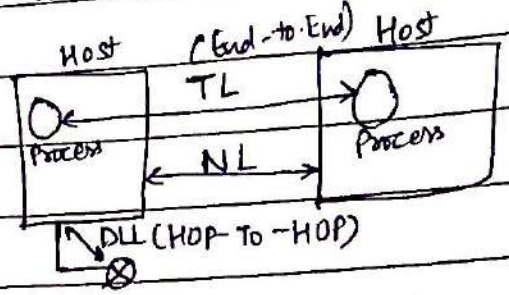
IP is used as logical address &
MAC is used as physical address (in Ethernet & Token Ring)

DLL
- LLC → Flow control, Error Control → logical link control
- MAC → Framing, Access Cont, Addressing → Medium Access Control

Date
Page

Network Layer (L:-3) :-

1) Host to Host Connectivity



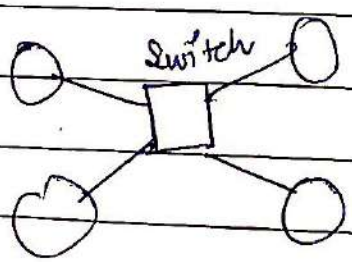
DLL: Used in one network

Many networks connect :- NL

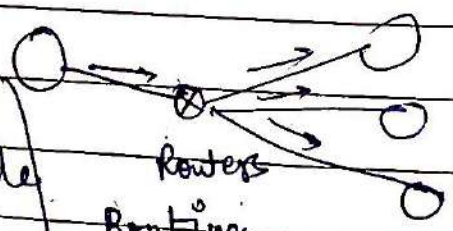
DLL :- Hop-to-Hop connectivity
NL :- Host-to-Host connectivity
TL :- End-to-End connectivity

2) Logical Addressing

3) Switching (Connecting various N/w together)

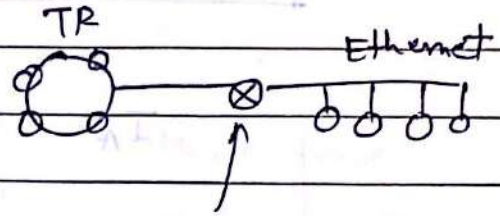


4) Routing (Using Routing Table is called switching)



5) Congestion Control

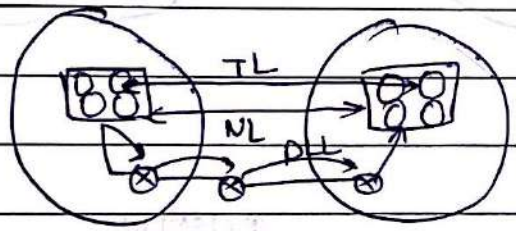
6) Fragmentation



→ large packets get divided into smaller for compactibility

Transport Layer

1) End-to-End Connectivity

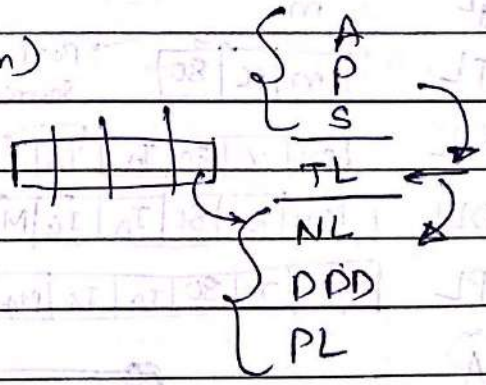


SPA
(Service, Point Addressing)

2) Flow Control (SR)

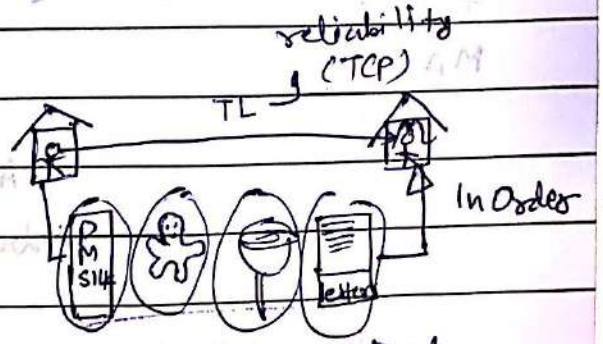
3) Error Control (Checksum)

4) Segmentation



5) Multiplexing & Demultiplexing

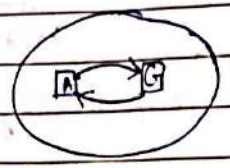
6) Congestion Control



How all the layers work together

I_A ✓ Google I_G ✓

Case I:-

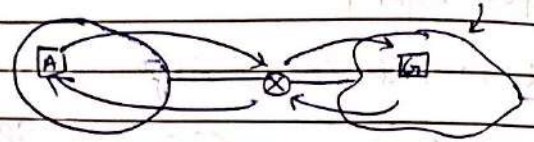


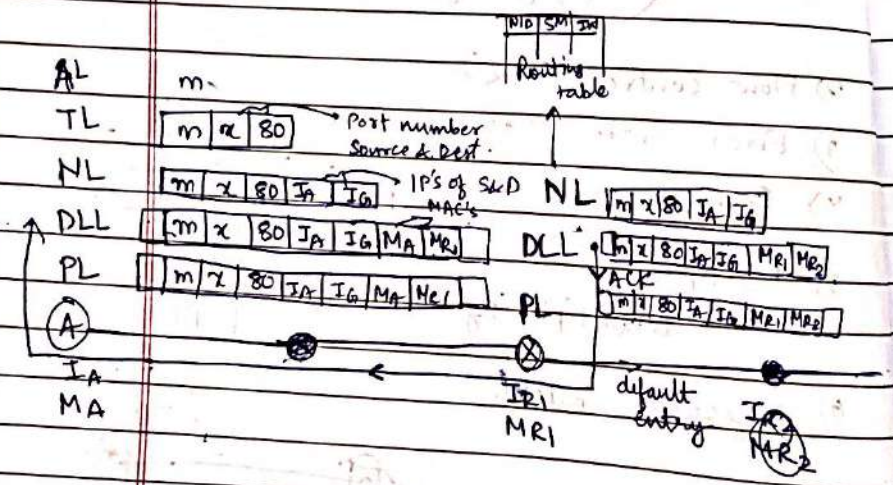
I_A & S_A — I_G & S_A

$NID_A A = NID_A G$

$\neq$

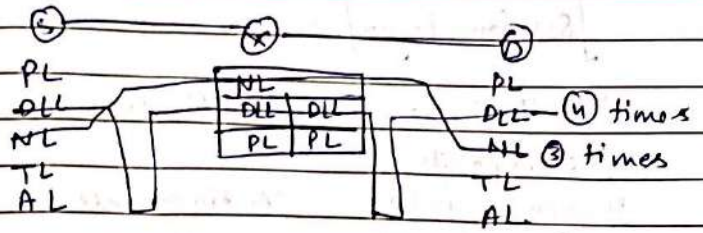
Case-II:-

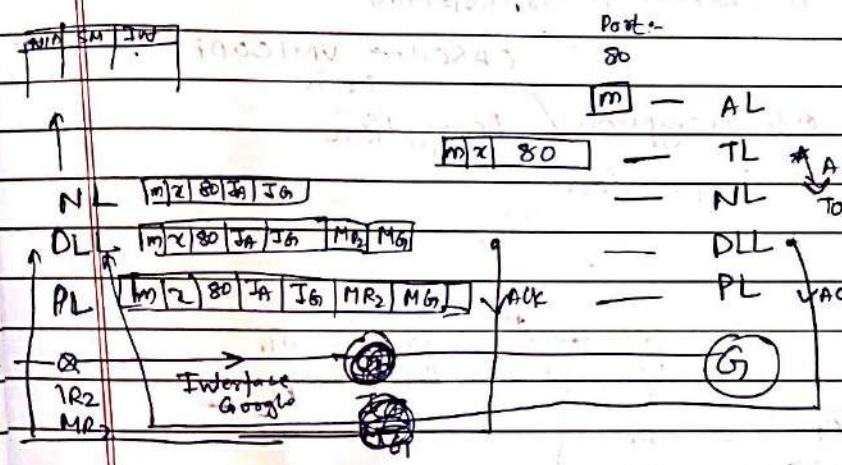




ARP (IR_1) $\longrightarrow$ MR_1

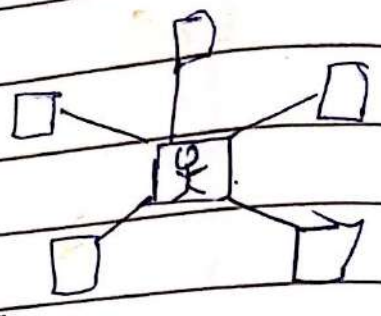
Address Resolution Protocol





Session layer :-

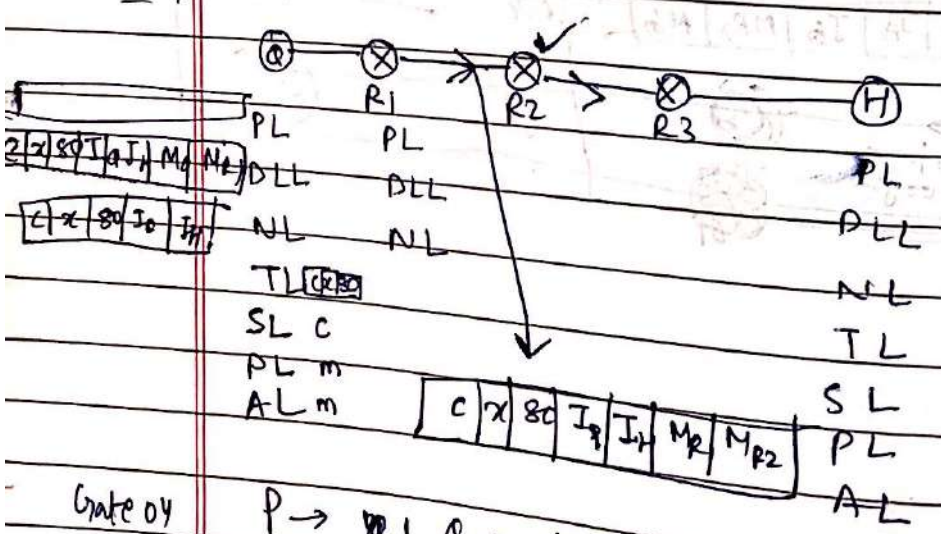
1) Authorization & Authenthication
2) checkpoints
3) Synchronization (Audio & Video)
4) Dialog Control
5) logical grouping (Atomicity)



Presentation Layer :-

1) Character translation (ASCII ↔ UNICODE) or so...
2) Encryption / Decryption
3) Compression

Gate 14

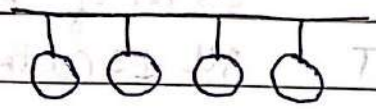


Gate 04 P → 1 q → 3 R → 4

5. LAN Technologies

ETHERNET (IEEE 802.3)

1) Topology - BUS



2) Access Control Method :- CSMA/CD.

3) No Acknowledgement

4) Band width → 10 Mbps → 100 Mbps → 1 Gbps
↓ Fast Ethernet ↓ Gigabit Ethernet

5) Encoding Technique :- Manchester Encoding

AL - message
TL - segment
NL - Datagram
DLL - Frame
PL - Single protocol Data Unit (IPDU)

Ethernet Frame Format (802.3)

7 bytes	1 byte	6 byte	6 byte	2 bytes		4 bytes
PREAMBLE	SFD	DA	SA	LENGTH	DATA	CRC

← Physical layer header → ← DLL Header → ← DLL tail →
← Frame →

Preamble :- $\underbrace{10\ 10 \ldots 10}_{7\text{bytes}}$ } used for -
- Start of Frame
- Synchronization

SFD :- 10101011

MAC address :- Physical address (6 bytes)

Date
Page

Types of MAC address

- UNICAST — L.S.Bit of I byte is 0
- MUTICAST — L.S.Bit of I byte is 1
- BROAD CAST — All 1's (in the MAC add).

eg. of MAC address :- 1A : 2B : 34 : 48 : 56

↓

$\boxed{00011010}$ (Unicast)

- Source Address must always be UNICAST address

eg. A3 : B4 : C5 : 67 : 89 : 10

↓

$\boxed{10100011}$ (Multicast)

eg. FF : FF : FF : FF : FF : FF

↓

$\boxed{\text{all 1's}}$ (Broadcast)

- In CSMA/CD :-

$L \geq 2 * T_p * BW$

$\Rightarrow L \geq 64\,B$ (Frame size) (For collision Detection)

- Max. size of Data :- 1500 B (data)

	min	max
Data	46 B	1500 B
Frame	64 B	1518 B

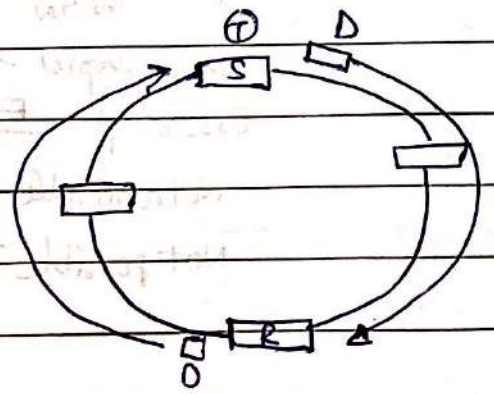
Disadvantages of Ethernet :-

- Not useful for Real time application (Due to collisions)
- Send at least 46 B of data (wastage)
  ↓ (padding of useless data)
  (NOT useful for Interactive applications)
- No priorities (NOT useful for client server app)

Gate 16. A ✗ B ✗ C ✗ D ✓

TOKEN RING (IEEE 802.5)

1) Topology :- Ring
2) Access Control Method :- Token Passing
3) Unidirectional (Simplex wires used)
4) Data Rates :- 4 mbps → 16 mbps
5) Piggy backing Acknowledge.
6) Encoding :- Differential Manchester



If after sending data, sender gets down, the packet will be rotating forever
⇓
called "orphan packet"

• Problems due to Source :- (Sender is dead)

(i) Orphan packet problem

(ii) Stray packet problem

(packet got corrupted)

Solution to these problems :-

(i) Inserting a special "monitor" station

↳ packet contains a monitor bit

m	CRC

(ii) If CRC changes, monitor removes the data

• Problems due to Destination :-

(i) Destination Down

(ii) Busy Destination (Buffers full)

(iii) ERROR in data

(iv) Copied (all correct)

↳ 4 circumstances that might happen.

⇓

Solution

↳ Header contains 2 bits :-

	A (ack)	C (copied)
Initial	0	0
Dest Busy	1	1
Error	E	1
Not available	0	0
Not possible	0	1

- Problem due to Token :-

1) Captured token :-

(Someone is using token and is not releasing it bcoz it has large data to send) – Results in Monopolization

↓ Solution → max. THT (10 ms by default)

eg. $THT_{max} = 10ms$, $BW = 4 mbps$

Max. frame size in ETR = ?

$= 10 ms \times 4 mbps$

$= 40,000$ bits

eg. Max. frame size in DTR = ?

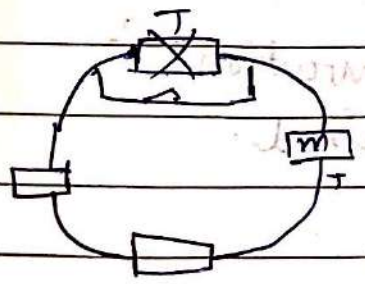
$THT = 10 ms$, $RL = 9 ms$, $BW = 4 mbps$

Max. fr size $= 1 ms \times 4 mbps$

$= 4,000$ bits

2) Token lost :-

After aquising token, station gets down (Token lost with station)



Token Return Time

min. TRT = RL

max. TRT = RL + N * THT

If within this time, token does not get to monitor, token is lost

eg. $RL = 10ms$, $THT_{max} = 10ms$

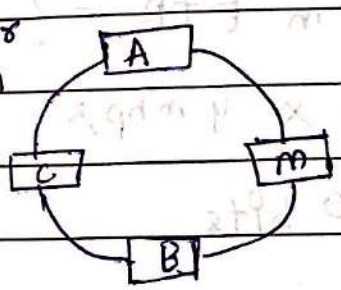
$N = 10$

$Max\ TRT = 10 + 10 \times 10 = 110\ ms$

3) Token gets corrupted
↓ Problem solved by Monitor
(by waiting Max TRT and regenerate it)

- Problem due to Monitor :-

1) Monitor down



Solution :-
Heart beat message
↓
Periodically m sends
AMP frame
↑
Active monitor presence

Rest will do polling → New monitor formed

2) Monitor hacked (Malfunctioning) :- Just sending AMP frame but NOT doing any work

↓ NO solution
→ Human Intervention is required.

This has so many problems but still is used. Why?

- Real time applications
- Interactive applications (any no. of bytes allowed)
- Client Server Applications (Priorities allowed).

Token Ring Frame Format :-

Data Frame

SD	AC	FC	DA	SA	Data	CRC	ED	FS
1	1	1	6	6		4	1	1

← size in bytes

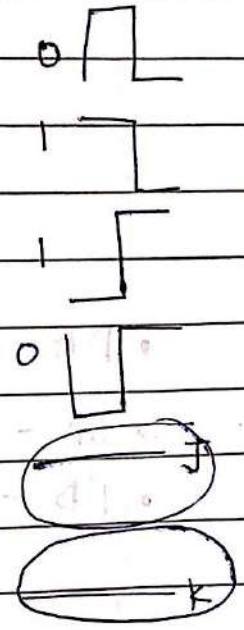
Token.

SD	AC	ED
1	1	1

← size in bytes

- SD – Start Delimiter

J	K	0	0	J	K	0	0



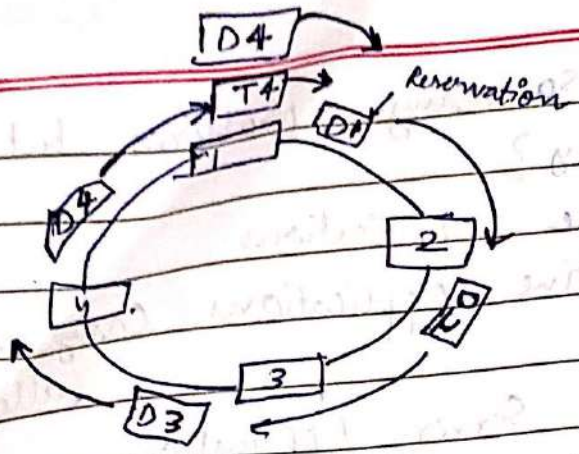
- AC – Access Control

P	P	P	T	M	R	R	R

Priority — Token bit — Reservation — Monitor bit

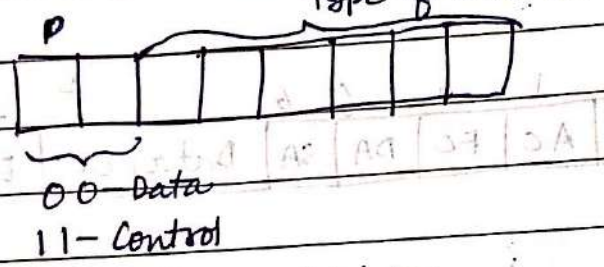
Token bit = 1 (Token)
= 0 (Not Token)

Monitor bit = 1 (stamped)



Numbers in the boxes are their respective priority values

- Frame Control :- (FC)



eg. of Types of Control :-

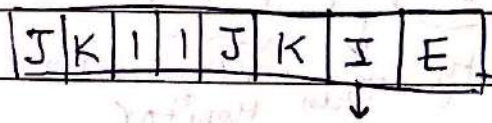
AMP (Active Monitor)

Polling (New monitor)

Purge (Everyone Stop)

Beckon (Cut in ring)

- DA - Destination Mac Address (48-bit)
- SA - Source Mac Address (48-bit)
- ED - End delimiter



I → Information → more inf. is following or NOT

E → Error → Data NOT accepted

- Frame Status -(FS)

A	C	0	0	A	C	0	0

Available Copied.

- why 2 copies ?

  Because CRC is NOT taking care of FS.

- why CRC is not Computed on FS?

  So that ~~Monitor~~ Receiver need not change CRC when destination A/C bit changed.

Minimum length of token ring :-

Atleast hold the token completely

∴ at least 24 bits

Min. length of token ring in order to put 24 bits :-

Capacity = $T_p \times BW$

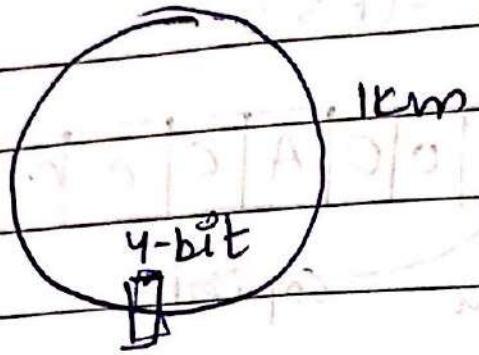
$T_p * BW \geq 24$

$\Rightarrow \frac{d}{v} * B_W \geq 24 \Rightarrow \boxed{d \geq \frac{24 \times v}{B_W}}$

Standard BW = 4mbps, $v = 2 \times 10^8$ m/s.

$\Rightarrow d \geq \frac{24 \times V}{B} = \frac{24 \times 2 \times 10^8}{4 \times 10^6}$

$\Rightarrow \boxed{d \geq 1200 \text{ m}}$



$$\frac{200}{2\times10^{8}} \times 4\times10^{6}$$

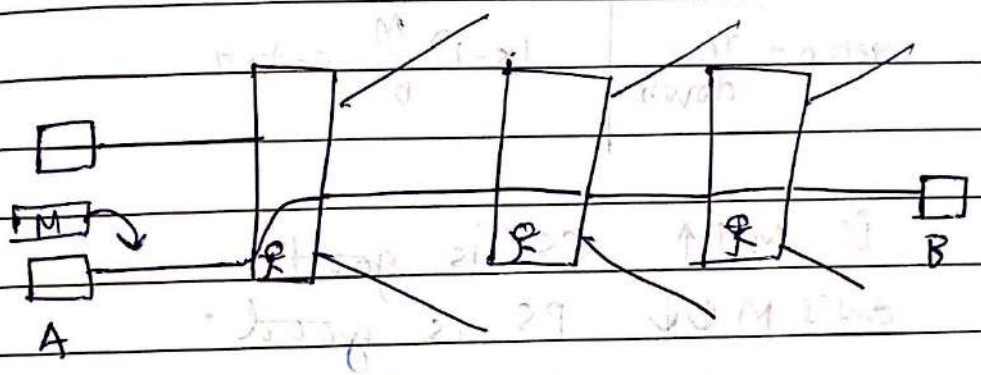
$= 4$ bit time

So 4 bit delay is required for 1km wire.

6. Switching

- Switching
  - Circuit Switching
  - Packet Switching
    - Virtual circuits
    - Datagram
  - Message Switching

Circuit Switching



eg: M : size of message
B : Bandwidth
x : # (Hops)
d :- length of a hop
v :- velocity

$$\underset{\text{Total time}}{TT} = \text{Setup Time} + \underbrace{M/B}_{T_t} + \underbrace{x\,d/v}_{T_p} + \text{Tear down Time}$$

i) No Headers Required
ii) No Reordering done

- It is applied at physical layer.
- It is obsolete.

Packet switching

No human intervention. Multiplexing done at all stations.

$$TT = \underbrace{\frac{xm}{B}}_{T_t} + \underbrace{\frac{xd}{v}}_{T_p}$$

CS	PS
setup + Tear down	$(x-1)\frac{m}{B}$ extra

if M↑↑ CS is good,
and if M↓↓ PS is good.

Pipelining (Packetisation)

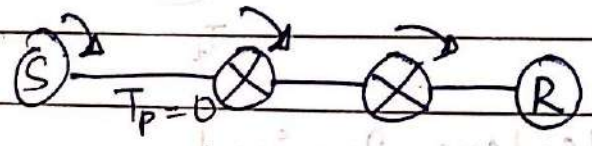
eg. Data = 1000 Bytes

B/w = 1 mBps

= 10^6 Bps

Header = 100 Byte

Case I: | D: 1000 | H: 100 | Packet size = 1100 B



$$T_t = \frac{L}{B} = \frac{1100}{10^6} = 1.1\ ms$$

$$TT = 3 * T_t = \boxed{3.3\ ms}$$

Case II (pipelining)

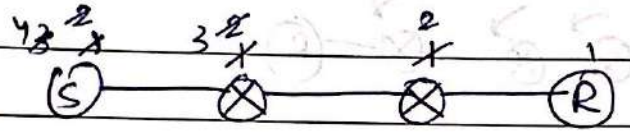
Packets = 5

Data = $\frac{1000}{5}$ = 200 Bytes

B/W = 1 mBps, Header = 100 Bytes

Packet size = 300 Bytes

$T_t = \frac{L}{B} = \frac{300}{10^6} = 0.3$ ms



1st packet = $3 \times T_t = 0.9$ ms

Remaining 4 packets :- $4 \times T_t = 1.2$ ms.

$\boxed{T_t = 2.1 \text{ ms.}}$

Case III (pipeling)

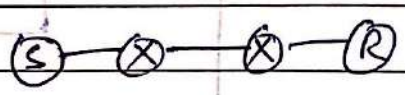
Packets = 10

Data = $\frac{1000}{10}$ = 100 Bytes

B/W = 1 mBps, Header = 100 Bytes

P.S = 200 B

$T_t = \frac{L}{B} = \frac{200}{10^6} = 0.2$ ms



1st packet = $3 \times T_t = 0.6$ ms

9 packets = $9 \times T_t = 1.8$ ms

$\boxed{T_t = 2.4 \text{ ms}}$

Case IV
(Pipeling)

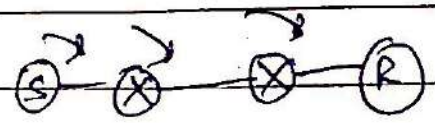
packets = 20

~~Data = 50 Bytes~~

B/W = 1mbps HS = 100 B

PS = 150 B

$T_t = \frac{150}{10^6} = 0.15 ms$

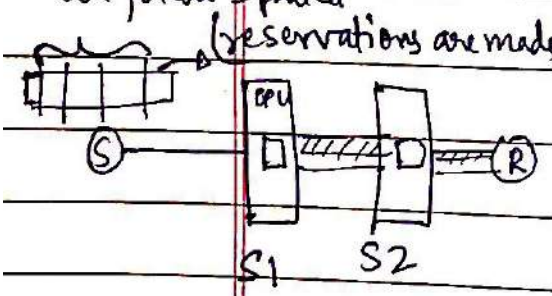
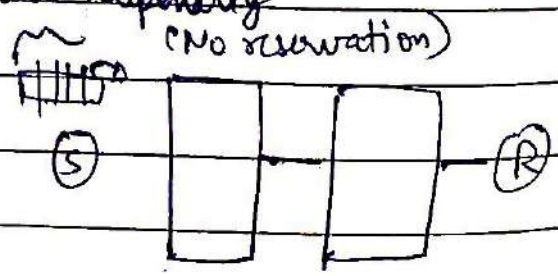


1st packet = 0.15 × 3 = 0.45 ms

19 packets = 19 × 0.15 = 1.85 ms

T_t = 3.3 ms.

Packet Switching

Virtual Circuits	Datagram
all follow 1 packet (reservations are made)	all independently (No reservation)
	
eg. Calling (Acc. to time)	eg. e-mail (Acc. to data)

• Header required only for I packet (Global header) For remaining we just have local header	• Everyone contains a header
• Connection oriented (Resource reservation) ↑ Buffers, CPU, BW	• Connection less ↑ No resource reservation
• ~~Data~~ Entire data follows the same path ↓ therefore inorder ensures	• Same path may NOT be followed ↓ OUT of order packets MAY arrive
• Highly reliable	• NOT reliable
• High Cost	• Less Cost
• eg. - ATM ↓ Asynchronous Transfer Mode	• eg. IP ↓ Internet Protocol

7. INTERNET PROTOCOL

IPV4 header

version(4)	HL(4)	Type of Service(8)	Total length (16)			
			0	DF	MF	Fragment Offset(13)
Identification (16)			Header checkSum (16)			
TTL(8)	Protcol (8)					
Source IP (32)						
Destination IP (32)						
options (0 to 40 Bytes)						
Data						

(4bytes) 32 bits

20 bytes

HL : Header Length

DF : Do not fragment

MF : More fragment

TTL : Time to Live

Header Length :-

- Min. Header length = 20 bytes + 0 B = 20 B.
  (static, options)
- Max. Header length = 20 bytes + 40 B = 60 B.
  (static, options)

- HL field = 4 bit = 15 B (Max)
  But we have to put 60 B (worst case)
- ∴ Scaling factor = $\frac{60}{15} = 4$
- Divide HL by 4 and then put the resultant no. in HL field (4 bits)
- $20 \le HL \le 60$, $5 \le HLF \le 15$

- If the Header length = 30, we cannot put either 7 or 7.5 or 8 as HLF, we add 2 bytes of dummy data so as to make it a multiple of 4.
  New HL = 32 bytes
  HLF = 8. "PADDING"

- In worst case, #(bytes that can be added to make option multiple of 4) = 3

Version :-

		Version bits
V1		
V2		
V3		
V4 ✓	—	0100
V5 ✗		
V6 ✓ (NOW in syllabus)	—	0110

Identification :-

Field of 16 bits. We use it to uniquely number a datagram coming out of a host.

- Useful for fragmentation
  All fragments get same identification number

DF: We don't want fragmentation at all (For this particular datagram)

MF: Indicates that more fragments are following this data packet

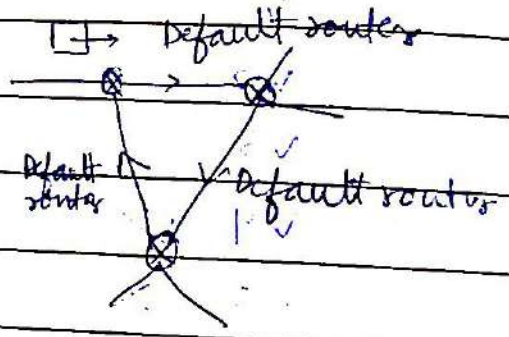
Fragment offset:- # (data bytes) ahead of this particular fragment after fragmentation.

TTL :-

(Time to live) :-

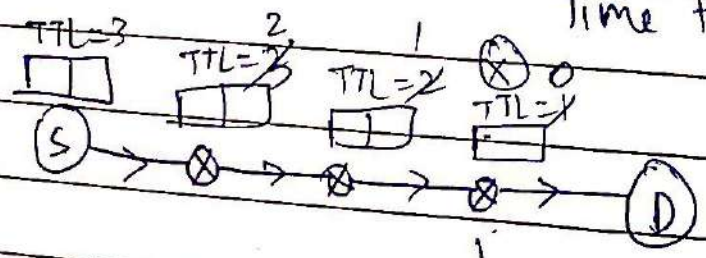
Routing table

NID	SID	Interface



Solution

Time to live



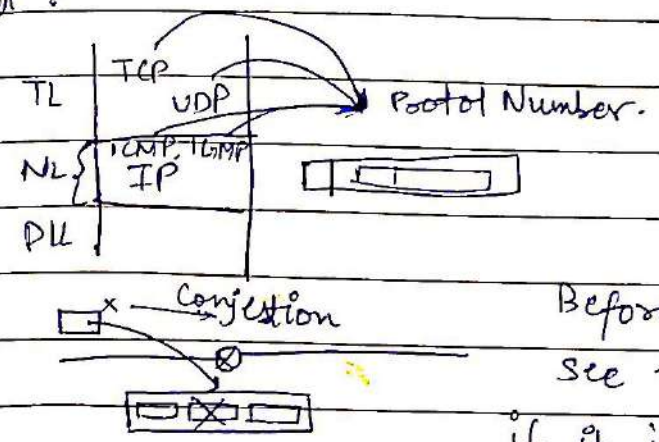
Discarded

4 ~~4~~ ~~3~~ ~~2~~ ~~1~~
3 2 1 0 accepted

5 ~~5~~ ~~4~~ ~~3~~ ~~2~~
4 3 2 1

- Purpose of TTL :- discard the packets that fall in infinite loop.

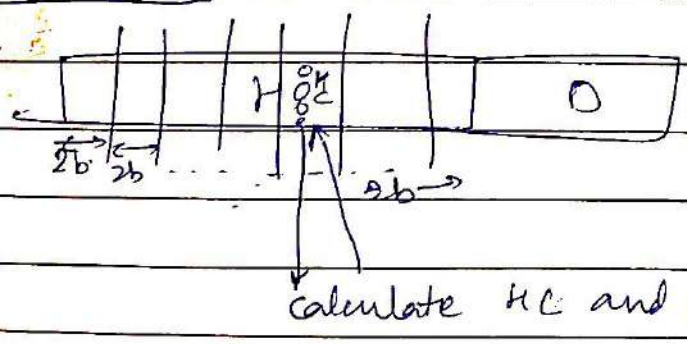
Protocol :-



Before discarding, see the protocol, if it is TCP, discard one of the packets which is not TCP and place it there.

Priority :- ICMP → IGMP → UDP → TCP

Header Checksum :-



Fields that can change in routers:-

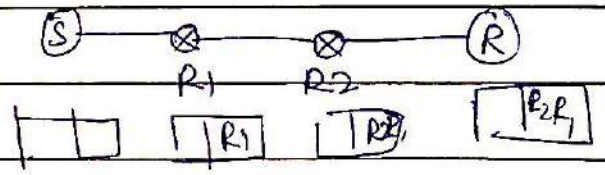
- TTL always
- FO, MF, TL (maybe)
- (Options) (maybe)

Source/Destination Address :-

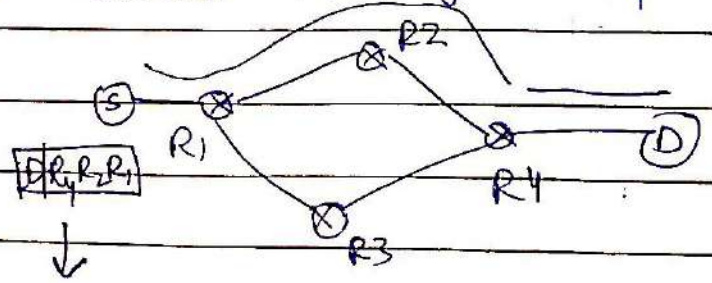
	NID	HID	
SIP ✓ DIP ✓	✓	✓	– Valid IP
SIP ✓ DIP ✗	✓	0's	– NID
SIP ✗ DIP ✓	✓	1's	– Directed Bro. Ad.
SIP ✗ DIP ✓	1's	1's	– Lim. Bro. Add
SIP ✗ DIP ✗	1's	0's	– Subnet Mask
SIP ✓ DIP ✓	0's	✓	– Host within a network
SIP ✓ DIP ✗	0's	0's	– "I don't have IP"
SIP ✗ DIP ✓	127	✓	– loop back address

Options :-

(i) Record route



(ii) Source Routing



↓

Strict Routing

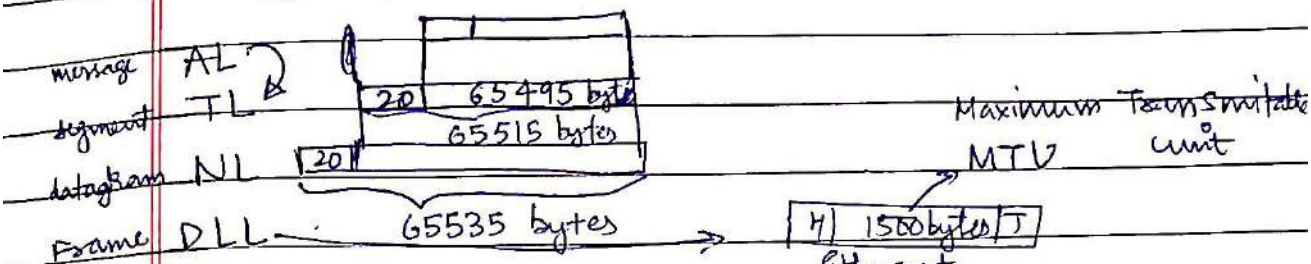
DR4R1 → loose source Routing

8. FRAGMENTATION

Total length :-

Total size of the datagram

16 bits — max $= 2^{16} - 1 = 65535$



- In general, AL can give any amount of data to transport layer, now it is the responsibilty of TL to divide that packet into parts so that each part is at most 65495 bytes.

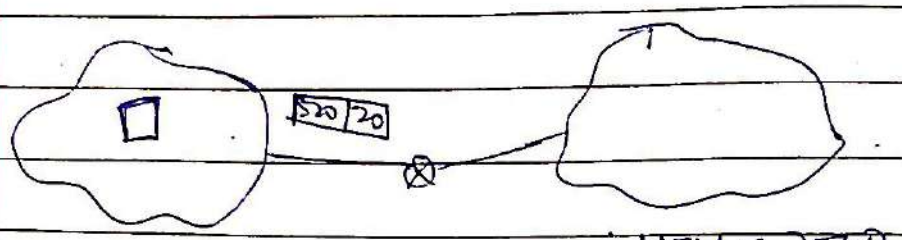
  ↳ "Segmentation" → Done at TL.

- Now if DLL contains Ethernet with restriction ~~to~~ of data to be at most 1500 bytes, than the NL has to divide the datagram into parts so that each part is at most 1500 bytes

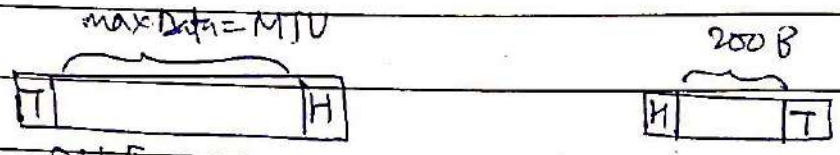
  ↳ "Fragmentation" → Done at NL.

- Segmentation when done wisely, does not require Fragmentation.

- Fragmentation is NOT DONE at source. Because, at source, segmentation is done in such a way that there is NO need of ~~segm~~ fragmentation.



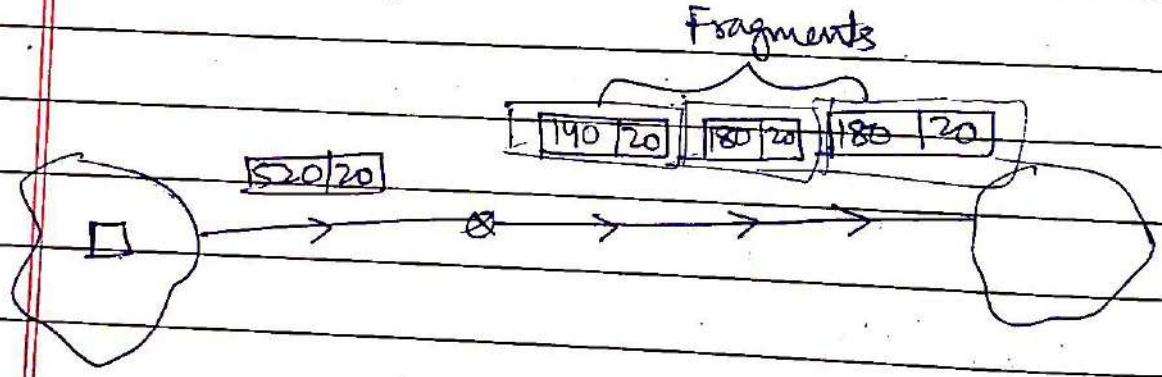
MTU = 520 MTU = 200B



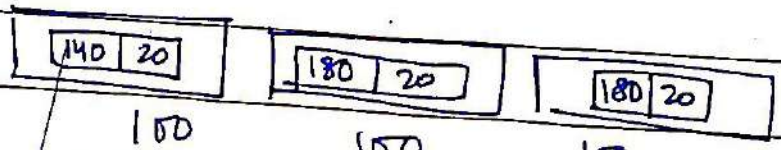
DLL Frame

max # bytes Datagram
= 520 − 20 (Header)
= 500 B

max # Bytes Datagram
= 200 − 20 (Header)
= 180 B.

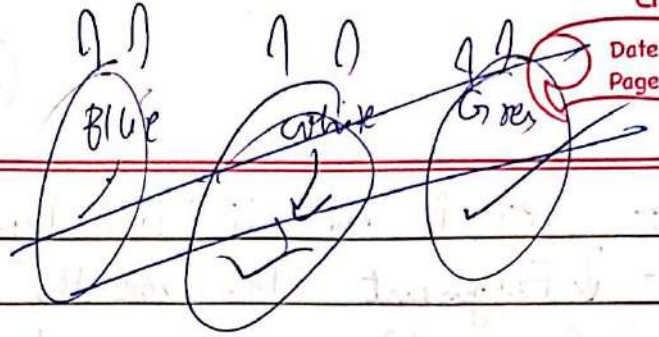


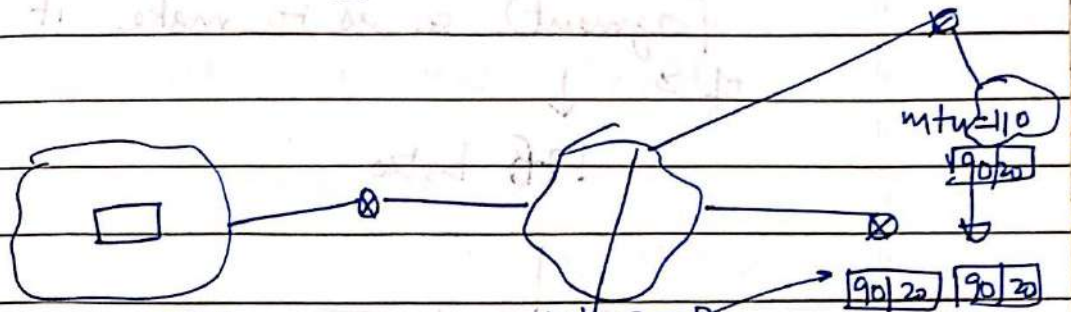
- How to identify that all the fragments are coming from same datagram?
  ↳ IDENTIFICATION NUMBER



100	100	100	ID
2	1	0	~~offset~~
0	1	1	MF

× Not going to work





140 | 20	180 | 20	180 | 20		90 | 20	90 | 20
100	100	100	ID	100	100 ID
360	180	0	offset	270	180 ~~off~~
0	1	1	MF	1	1 MF
5	5	5	HL	5	5 HL
160	200	200	TL	110	110 TL

No. of data bytes ahead in this datagram

4 (0, 180, 270, 360)

fragments offsets

- Worst case fragment offset :-

$\boxed{65515 \mid 20}$

max size IP datagram

$\boxed{1 \mid 20}$ ... D D D D

65514 ... 0 offset

Worst case fragment offset

But fragment offset = 13 bits ∴

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

∴ Divide by 8 and then put offset

Problem:- 180 is not divisible by 8

Sol^n:- ↓ Fragment (Decrease the size of fragment) so as to make it multiple of 8

↓

176 bytes

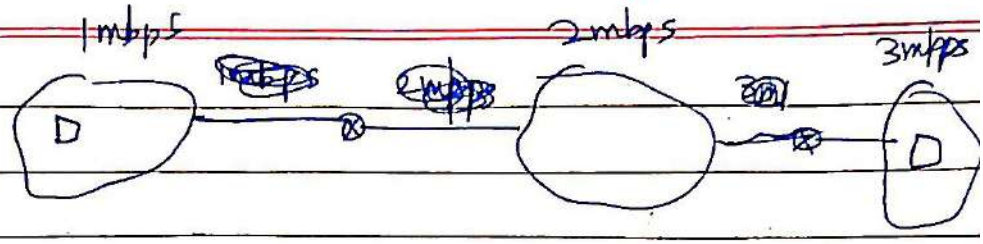
168 | 20	176 | 20	176 | 20	
100	100	100	ID
$\frac{176+176}{8}=44$	$176/8=22$	$0/8=0$	offset
0	1	1	MF
5	5	5	HLF
168	196	196	TL

88 | 20	88 | 20	
100	100	ID
$22+\frac{88}{8}=33$	22	offset
1	1	MF
5	5	HL
108	108	TL

$$\eta = \frac{\text{Useful Bytes}}{\text{Total Bytes}} = \frac{500}{500+4\times 20} = \frac{500}{580}$$

Overhead $= 3\times 20 = 60$ Bytes

BW Util $= \eta \times$ BW



BW util = η x BW. (which B/W?)

take bottleneck B/W (1mbps)

↓

- In throughput formula take min. bandwidth in case of multiple bandwidths given

Fragment:-	I	II	III	IV
	20 | 176	20 | 88	20 | 88	20 | 148
offset	0	22	33	44
MF	1	1	1	0
HLF	5	5	5	5
TL	196	108	108	168
Id.	100	100	100	100

1) Where is fragmentation done?
→ Routers, not at source
↓ why?
Source does segmentation

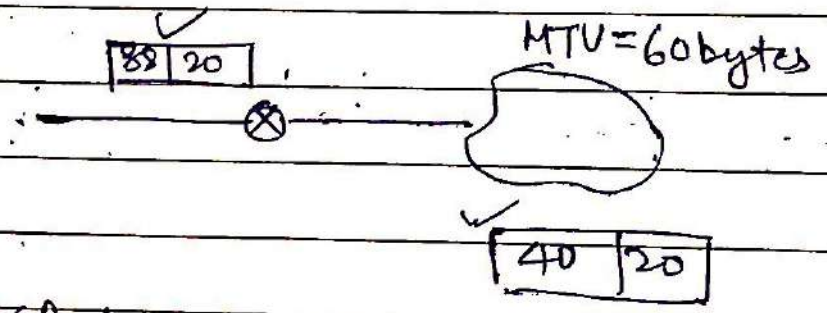
2) Where are segments reassembled?
→ Destination, not at the routers on the way
↓ why?
All the fragments may follow different route

3) Why is reassembling NOT done at routers?

→ all fragments may not meet at a router

→ There might be a need of further fragmentation.

- Now suppose a certain only gets fragment II and it has to transmit it through a network with MTU = 60 bytes



(Refragmentation)

	88 | 20 →	8 | 20	40 | 20	40 | 20
ID	100	100	100	100
offset	22	32	27	22
HLF	5	5	5	5
TL	108	28	60	60
MF	1	1	1	1

- Reassembly Algorithm:-

	20 | 176	20 | 88	20 | 88	20 | 148
Offset	0	22	33	44
MF	1	1	1	0
HLF	5	5	5	5
TL	196	108	108	168
Id	100	100	100	100

* Reassembly at destination

- Using offset and MF, we can identify that datagram is fragmented or NOT

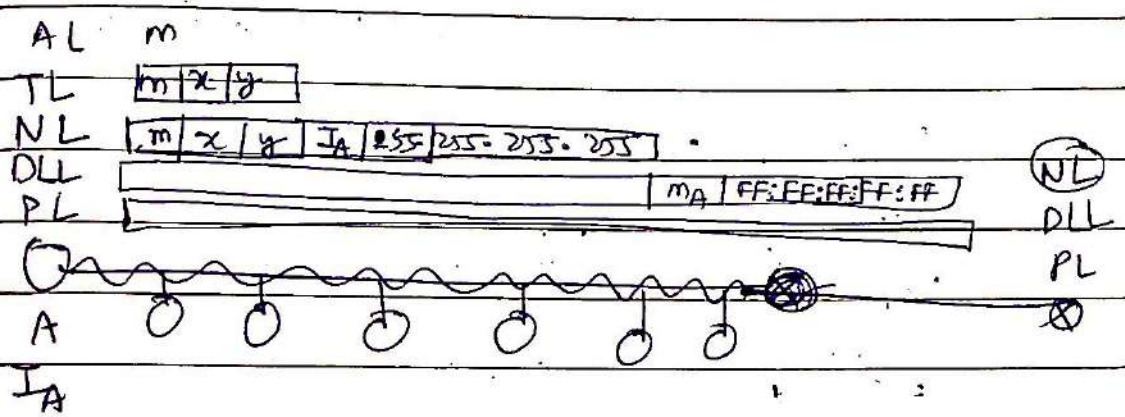
MF	off	
1	0	— 1st fragment
1	≠0	— Between fragment
0	≠0	— Last fragment
0	0	— NOT FRAGMENTED

1) Destination should identify that datagram is fragmented (MF + offset)
2) Destination should identify all fragments belonging to the datagram (Id)
3) Identify I fragment (offset = 0)
4) Identify subsequent fragments
   (offset of Next fragment = off + $\frac{TL - HLF \times 4}{8}$)
5) Repeat 4 until MF! = 0

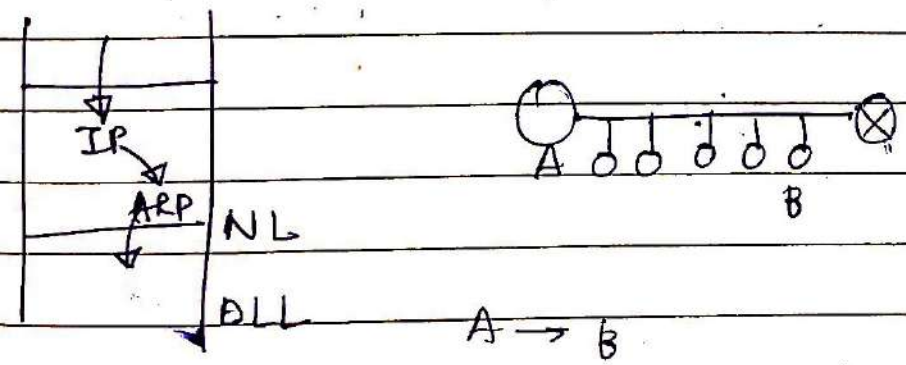
~~13 Hardware and various Devices used in Networking~~

9. Protocols and Concepts at Network Layer

Implementation of Brodcasting



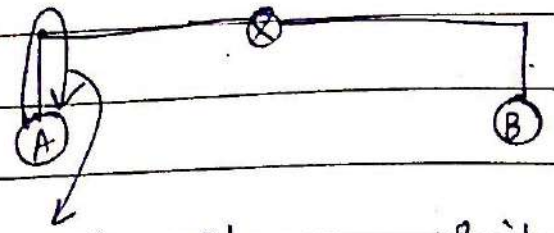
ARP (Address Resolution Protocol)



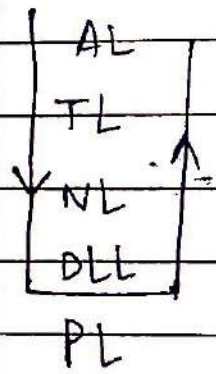
ARP request :- Broadcast

ARP reply :- Multicast

Special Address 127



Testing self connectivity
(Sending packet from A to A)



(Address 127 used)

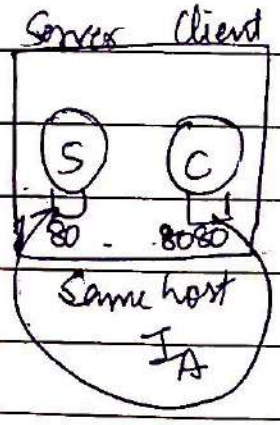
(Loop back address)

127.0.0.1

↓

127.255.255.254

another application:-

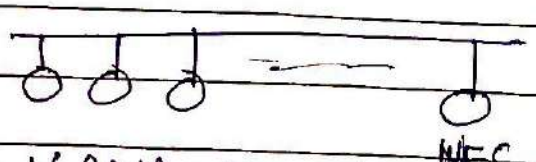


m	8080	80	I_A	127.0.0.1

RARP (Reverse ARP)

MAC → IP

use?

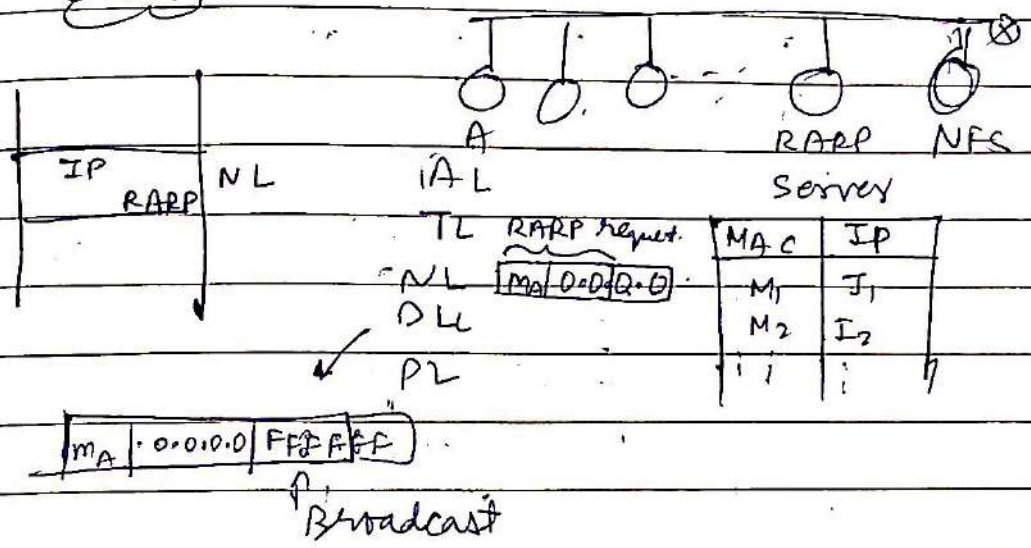


If RAM not present

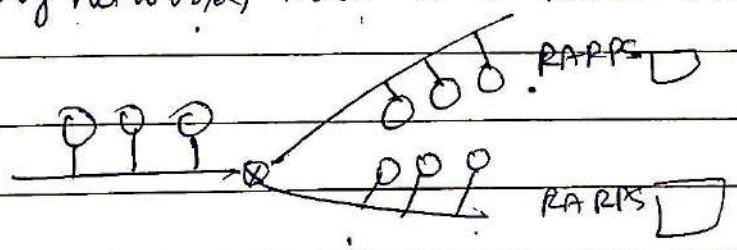
MAC → ROM ✓ IP → RAM ✗

How does a host know its IP address if it does not know its IP address.

→ RARP



In every network, there is a RARP server



Disadvantages :-

- Every network should have a RARPS
- Static mapping tables (IP ⇒ Hosts)

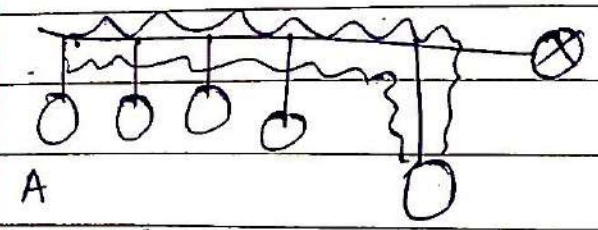
} Obsolete

Date
Page

BOOTP (Bootstrap protocol)

→ Runs on AL

(Evolution of RARP) → Runs on NL



PL

DLL | MA | FF.FF.FF

NL | 0.0.0.0 | 255.255.255.255

TL | R | |

AL R

(Same as RARP)

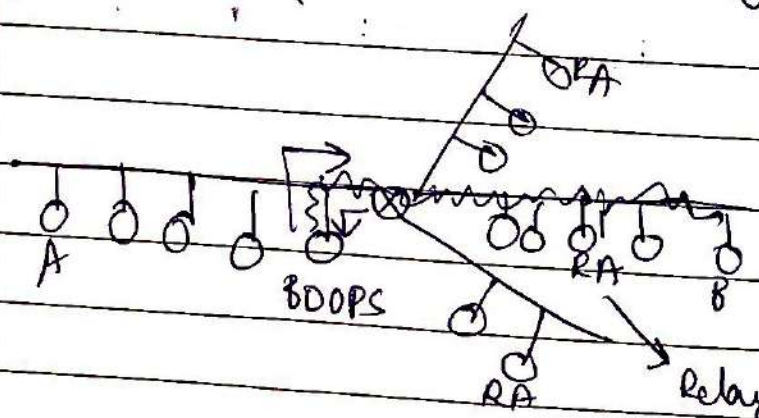
Table (Static) (Same as RARP)

MAC	IP
M_1	11
M_2	12
M3	13

(IP ≥ Hosts)

Same two disadvantages of RARP

But problem 1 is solved using a Special server called relay agent (RA)



Relay agent knows the IP of BOOTPS

Advantage of ~~BOOTP~~ RA :-

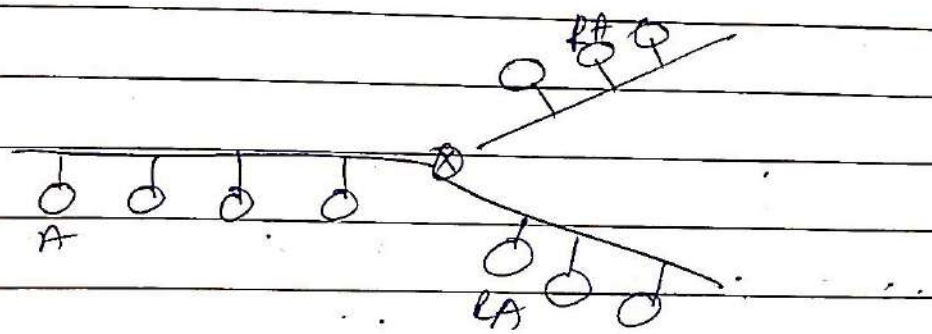
i) Only 1 BOOTPS required

Disadvantage :-

i) Static BOOPS ~~table~~.

DHCP (Dynamic Host Configuration Protocol)

↳ Table need not be static.



DHCP S Table

Static	m_1	m_1	
	m_2	I_2	
	m_3	I_3	
Dynamic	MAC	IP	lease time
	m_A	I_A	10mins

Servers

Pool of IP

Assign ← dynamically

Advantages :-

i) Only 1 DHCP server is enough

ii) Table is dynamic (IP = Online Hosts)

Backward Compatible

RARP → BOOTP → DHCP

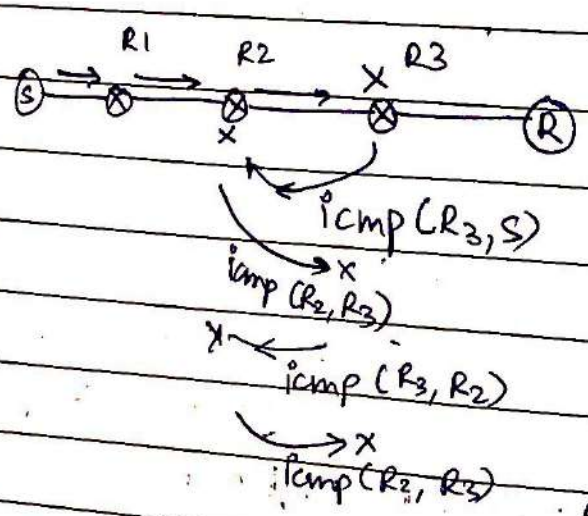
Same port 10.

In practice :- DHCP implemented on Router itself

ICMP (Internet Control Message Protocol)

ICMP

- Error handling or feedback messaging
  - TTL Exceed
  - Parameter problem
  - Source quench
  - Source redired
  - Destination unreachable
- Request and Reply
  - Echo Req and Reply
  - Time stamp req & reply
  - N/W Mask req & reply
  - Router solicitation & advertisement

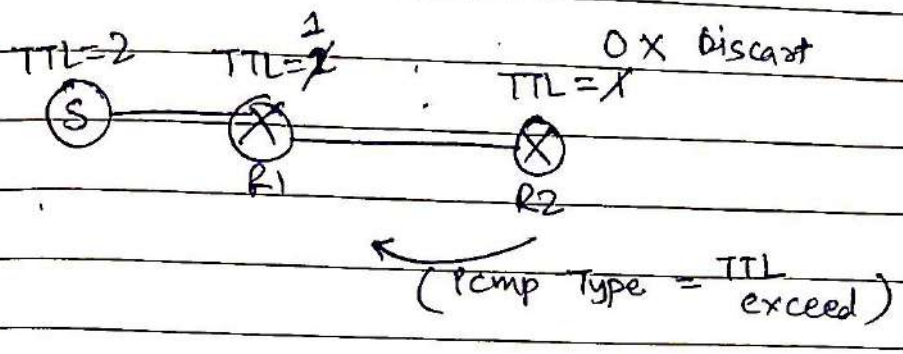


Infinite loop (Halt)

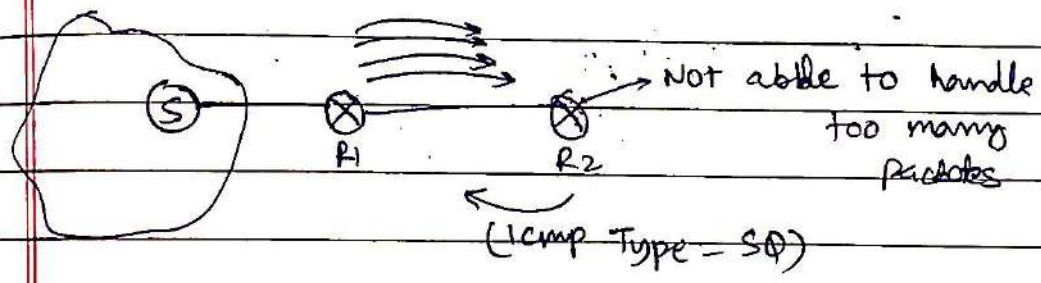
if icmp is sent for a discarded icmp too

Therefore, IP + ICMP Unreliable

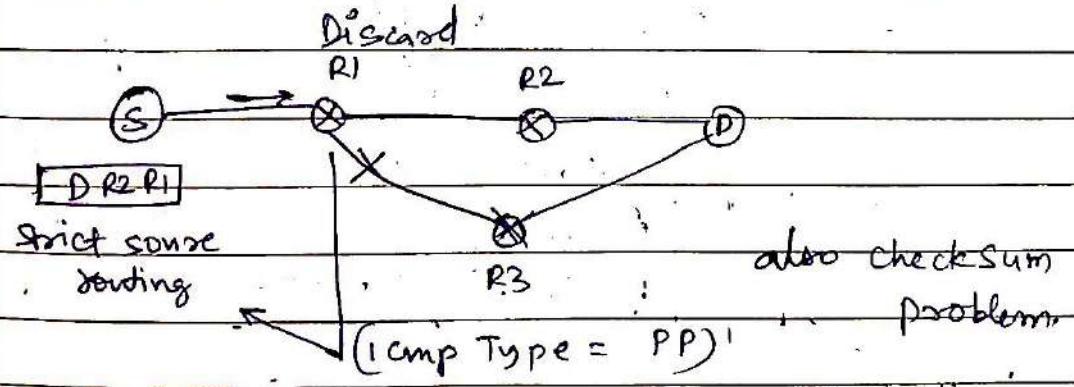
TTL Exceed



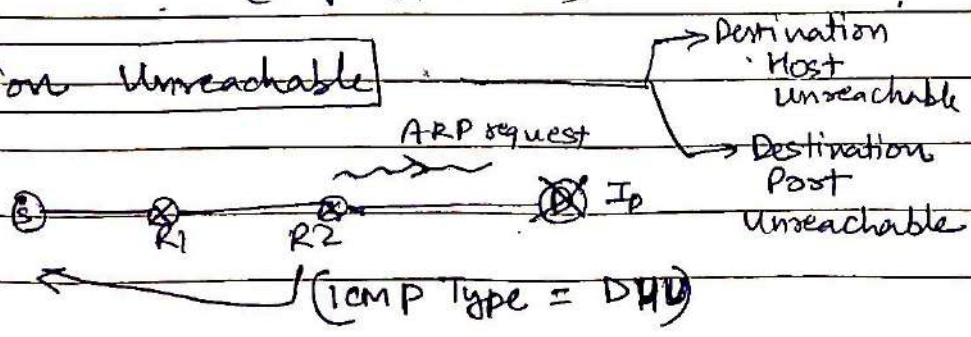
Source Quench

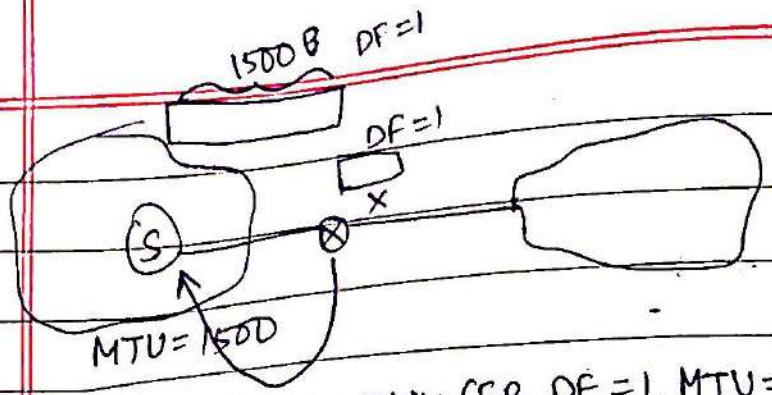


Parameter Problem



Destination Unreachable





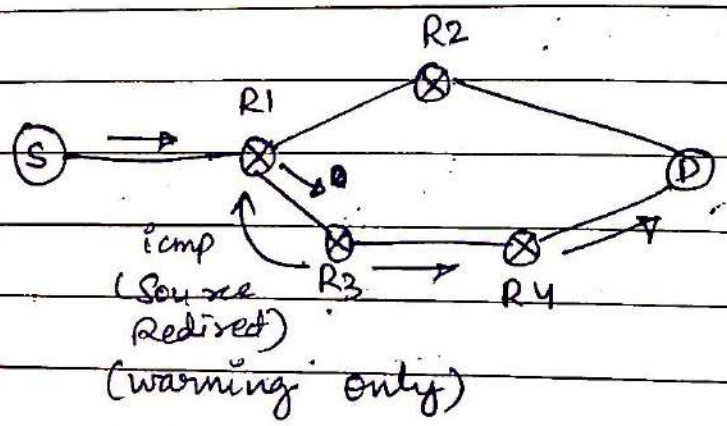
ICMP = DHU (FR, DF=1, MTU=500)

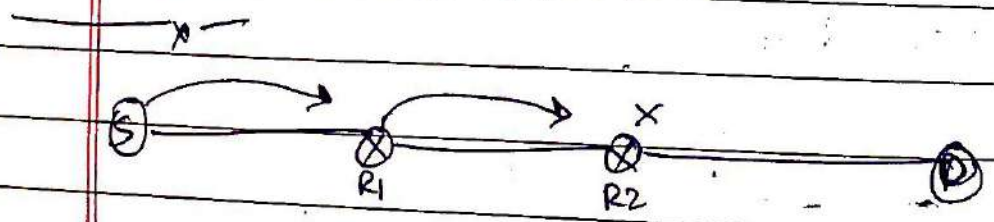
↓

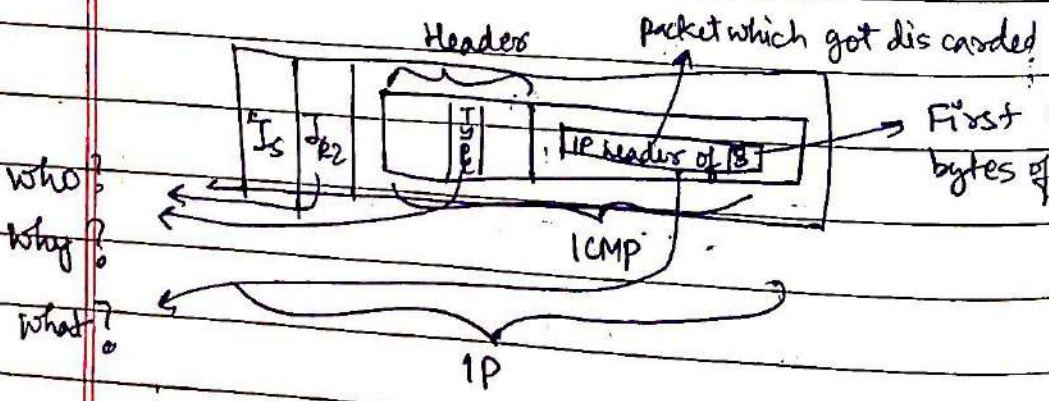
Fragmentation Required

Popular subclass of DHU.

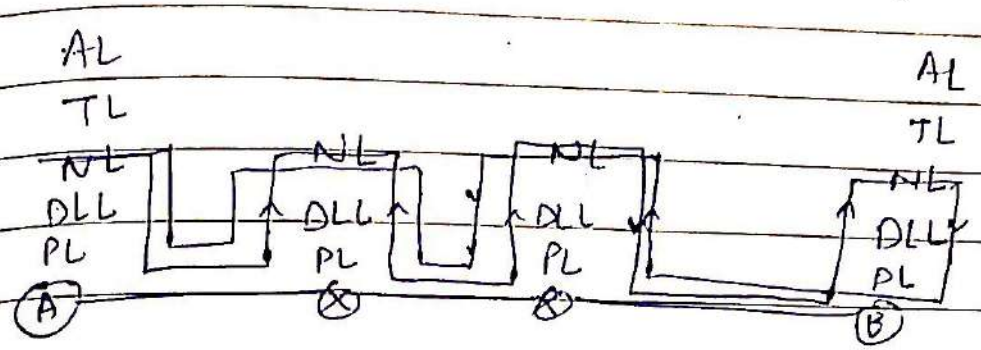
Source Redirect







Echo Request & Reply

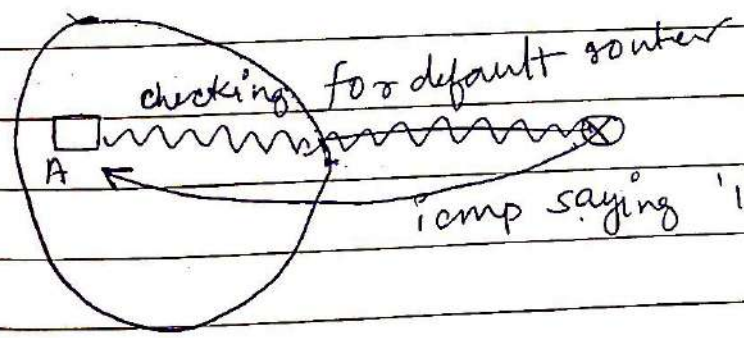


used to test whether the NL of routers and destination are working OR NOT.

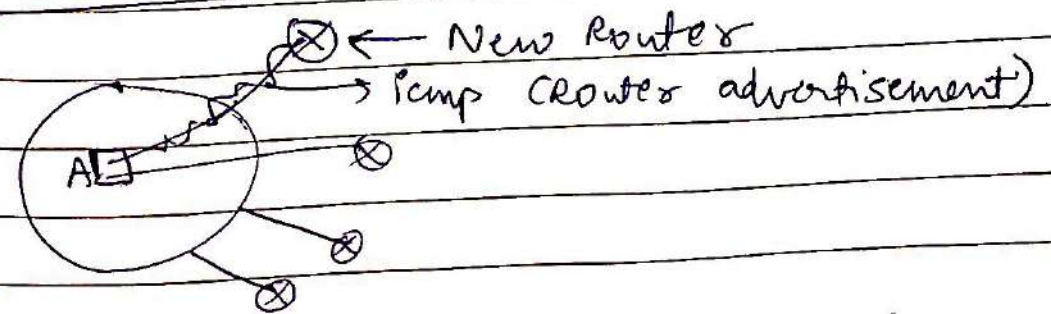
ICMP Echo req / reply :- Ping
Packet internet Grapher

Ping attack :- Send lots of echo requests to server so that is becomes down.

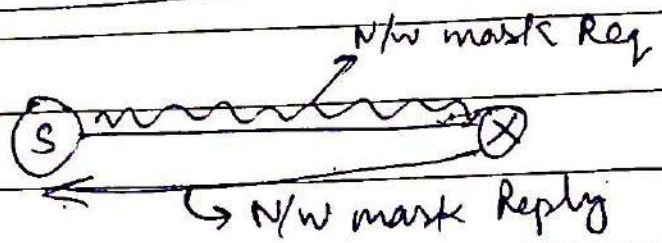
Router Solicitation



Router Advertisement



N/W Mask Req & Reply

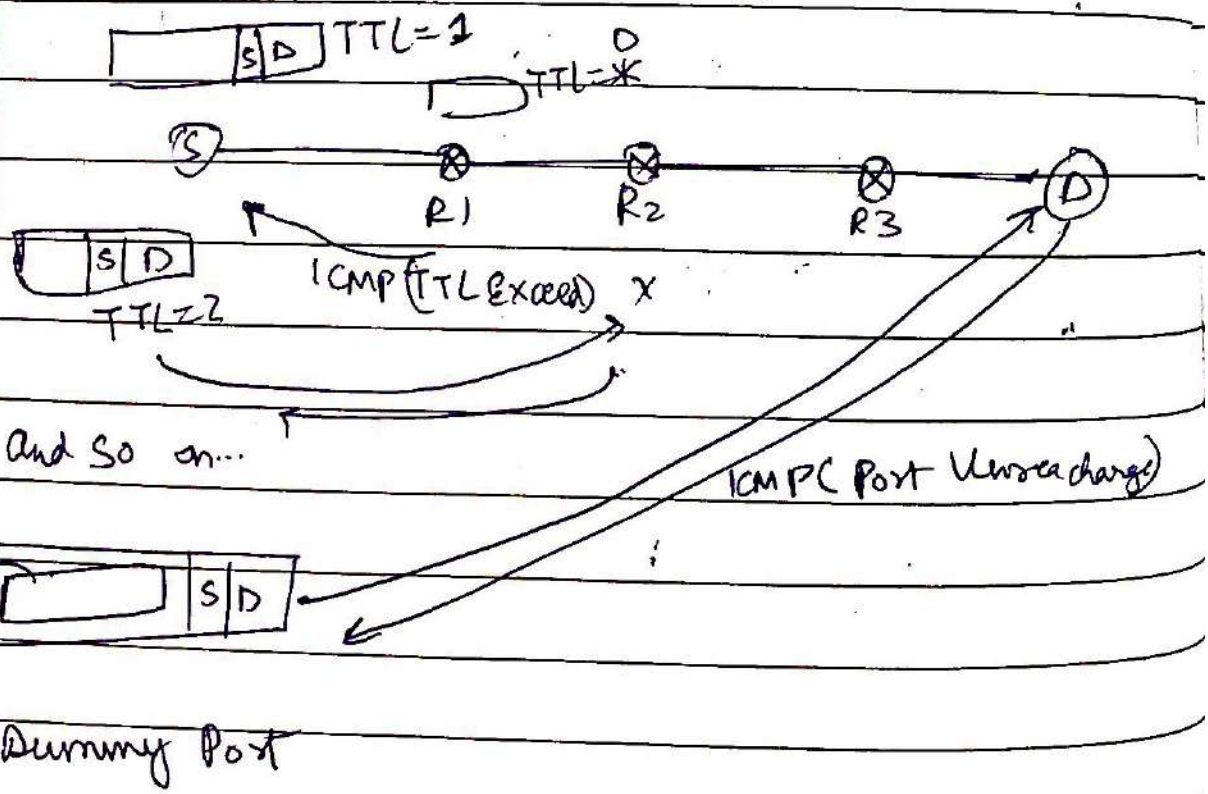


Time stamp req & reply

⚠ Not heavily used

to check the time zone of destination (to synchronize)

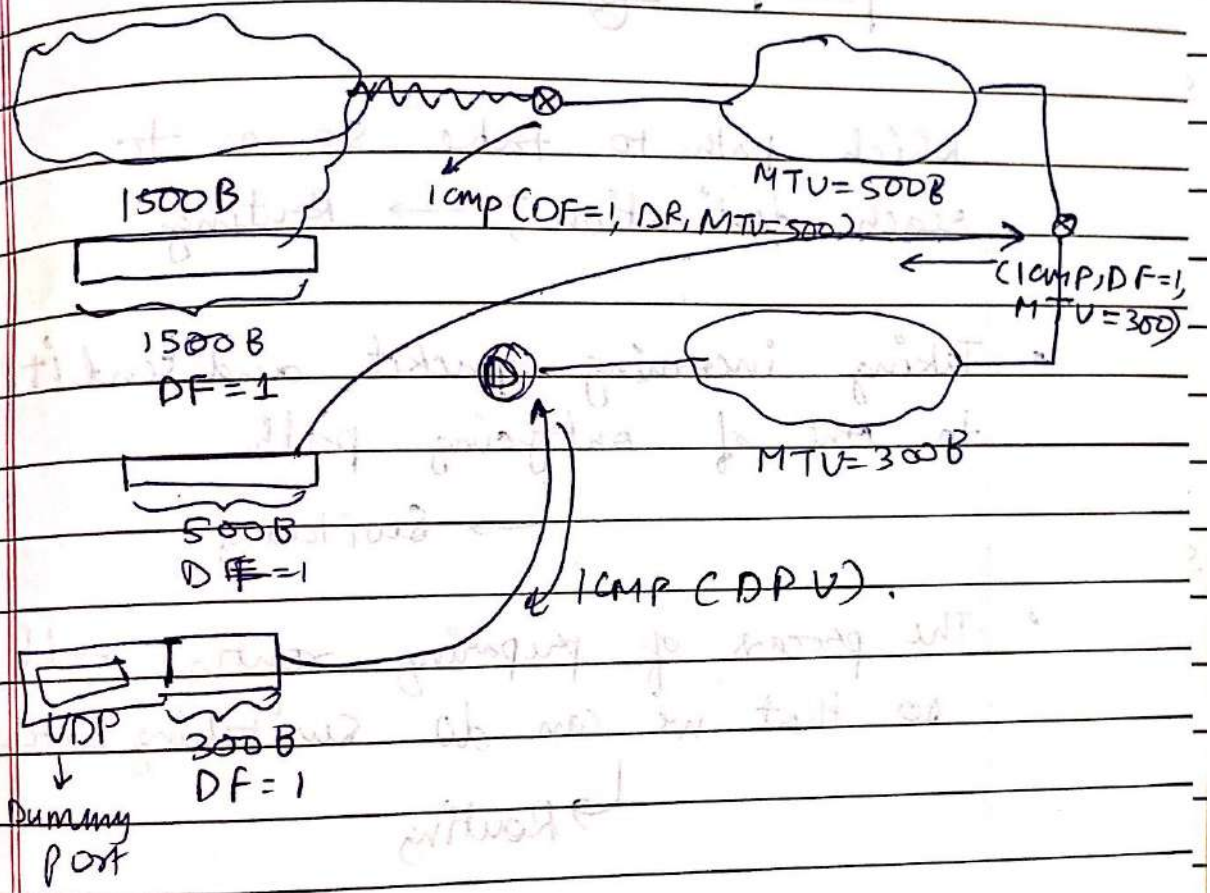
Traceroute - Application of ICMP



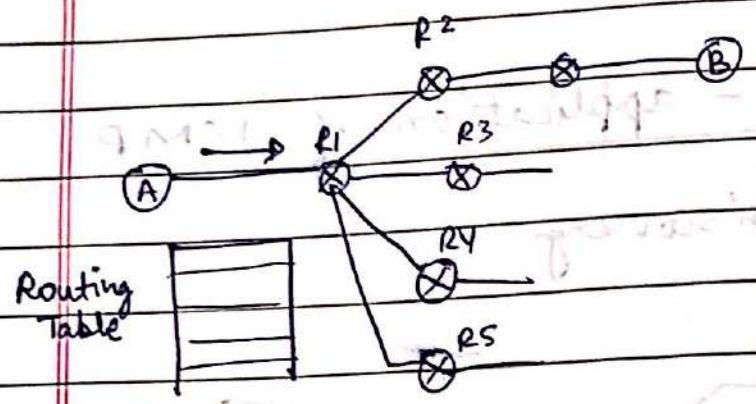
~~PMTD~~

PMTDU - application of ICMP

Path MTU Discovery



10. Routing



Which path to take so as to reach destination? → Routing

- Taking incoming packet and send it to one of outgoing path
  ↳ Switching.

- The process of preparing routing table so that we can do switching better
  ↳ Routing

- When we don't have routing, what should we do?, send packets to all networks possible
  ↳ Flooding

Flooding

Adv:- 1) No routing
2) Shortest path guaranteed
3) Highly reliable

Routing

Disadv:- (i) Routing table required
(ii) No shortest path guaranteed
(iii) NOT much reliable.

Flooding

Disadv:- (i) Destination gets duplicate packets.
(ii) Traffic is very high

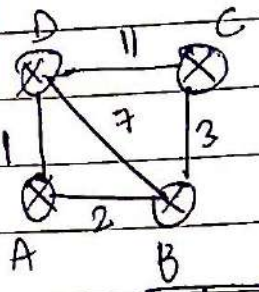
Routing

adv:- (i) No duplicate packets
(ii) Low traffic

Types of Routing Algorithm

- Static (manually)
  Does not change with the changes in traffic and topology
- Dynamic
  (changes with the changes in traffic or topology)
  - DVR (Distance vector Routing)
  - LSR (Link State Routing)

DVR (Distance Vector Routing)



Destination	Distance	Next Hop

Routing Table

Step 1 :-

Des	Dis	N+1
A	0	A
B	2	B
C	∞	–
D	1	D

R.T. of A

Des	Dis	N+1
A	2	A
B	0	B
C	3	C
D	7	D

R.T. of B

Des	Dis	N+1
A	∞	–
B	3	B
C	0	C
D	11	D

R.T. of C

Des	Dis	N+1
A	1	A
B	7	B
C	11	C
D	0	D

R.T. of D

Step 2 :-

At A :- distance vector

DV recieved from B, D

At B :-

DV from A, C, D

At C :-

DV from B, D

At D :-

DV from A, B, C

At A :-

From B

2
0
3
7

AB=2

From D

1
7
11
0

AD=1

New Routing Table

Des	Dis	N+1
A	0	A
B	2	B
C	5	B
D	1	D

At B :-

From A

0
2
∞
1

AB = 2

From C

∞
3
0
11

CB = 3

From D

1
7
11
0

DB = 7

New Routing Table

Des	Dis	N+1
A	2	A
B	0	B
C	3	C
D	3	A

At C :-

From B

2
0
3
7

BC = 3

From D

1
7
11
0

CD = 11

New Routing Table

Des	Dis	N+1
A	5	B
B	3	B
C	0	C
D	10	B

Repeat this one more time to get final result.

Answer :-

Des	Dis	N+1
A	0	A
B	2	B
C	5	B
D	1	D

RT of A

Des	Dis	N+1
A	2	A
B	0	B
C	3	C
D	3	A

RT of B

Des	Dis	N+1
A	5	B
B	3	B
C	0	C
D	6	B

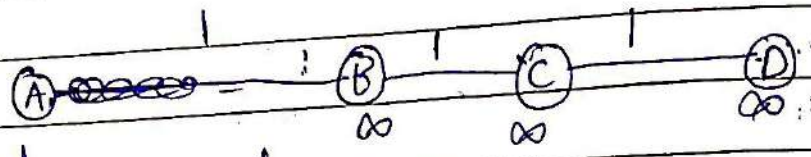
RT of C

Des	Dis	N+1
A	1	A
B	3	A
C	6	A
D	0	D

RT of D

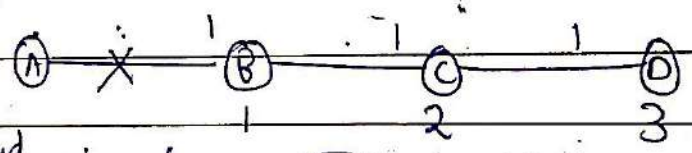
Note :- CD and BD are NEVER used
↓
Remove them

Count to infinity (Bad news spreads slow
Good news spreads fast)



Good news spreading fast

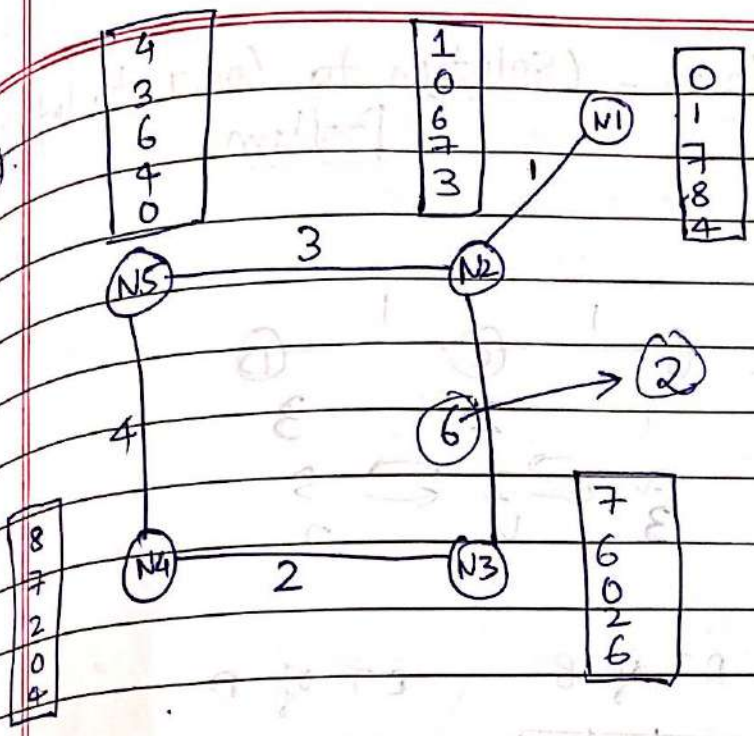
1	∞	∞
1	2	∞
1	2	3



Bad news spreading slow

∞	2	3
3	4	3
5	4	5
5	6	6
7	6	7
7	8	7
9	8	9
9	10	9
⋮	⋮	⋮
∞	∞	∞

eg.



At N3 :-

From N2

1
0
6
7
3

From N4

8
7
2
0
4

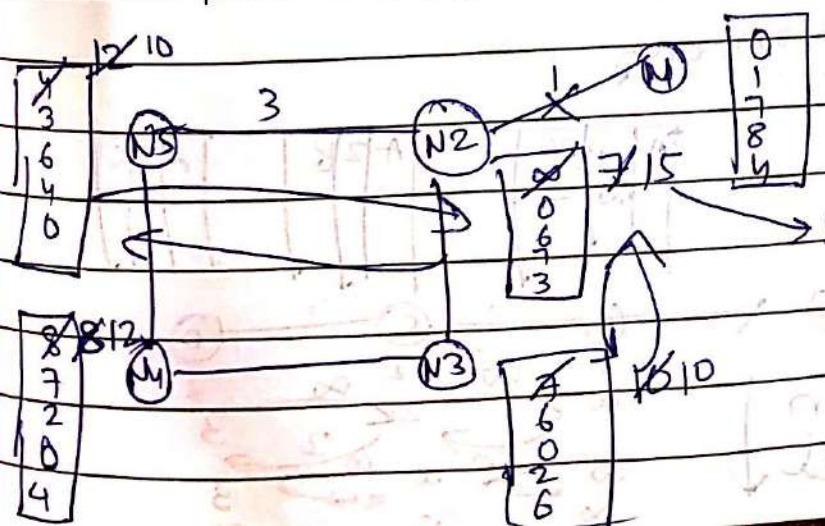
New Routing Table

N_1	3	N_5
N_2	2	N_2
N_3	0	N_3
N_4	2	N_4
N_5	5	N_2

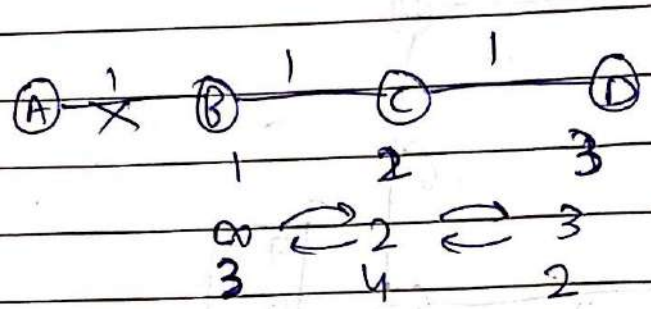
N_3N_2 : 2

N_3N_4 : 2

New Q. :- N_1 is down, How count to ∞ works?



Split Horizon - (Solution to Count to Infinity Problem)



R.T. of C R.T. of B R.T. of D

A	~~2~~ 4	~~B~~ D

A	~~∞~~ 3	~~1~~ C

A	~~3~~ 3	~~2~~ C

If B wants to send a packet to A,

B → C → D

loop

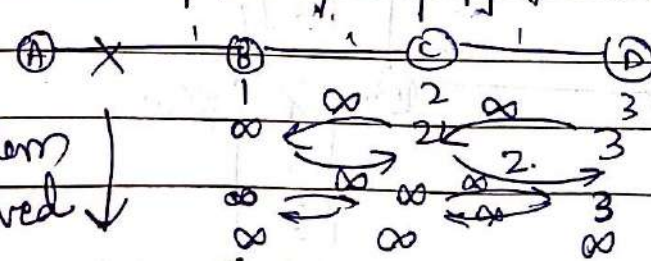
Solution

Split Horizon :

A	~~1~~ ∞	~~A~~ –

A	2	B

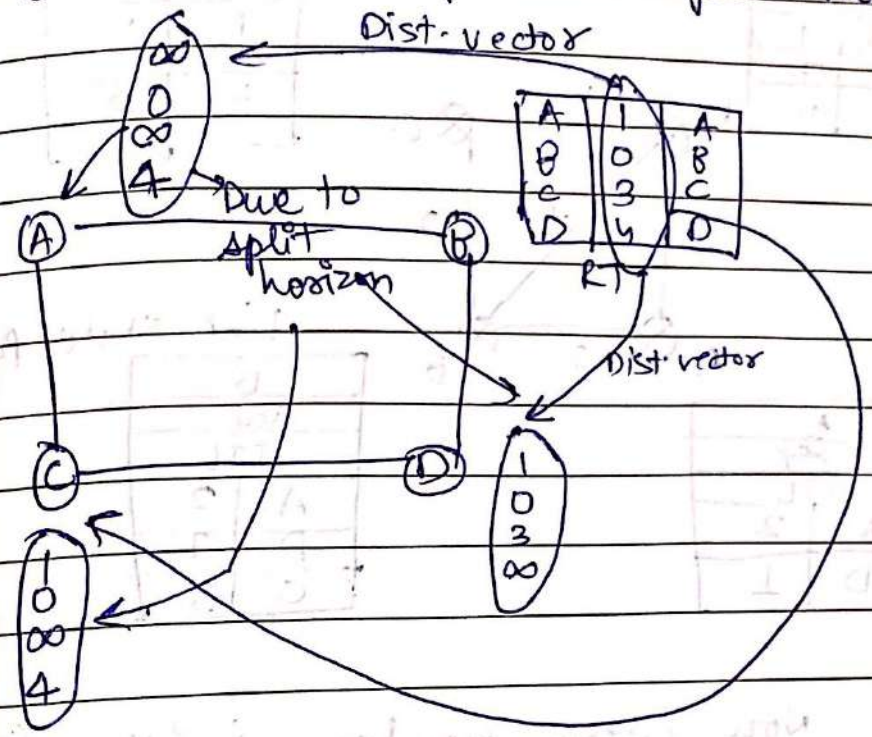
A	3	C



Problem Solved

If a node is dependent on some other node, it should send ∞ to that node (to avoid loops). → split Horizon

eg.



Disadv. of DVR without split Horizon :-

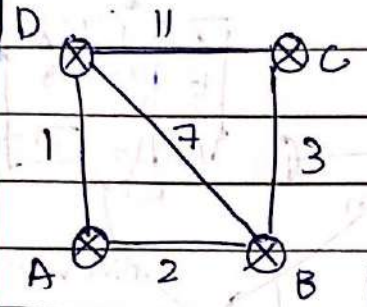
(i) Count to Infinity
(ii) Slow Convergence
(iii) loops

~~Both~~ solved by split horizon

Link State Routing

D	
Seq	
TTL	
A	1
B	7
C	11

C	
Seq	
TTL	
D	11
B	3



A	
Seq	
TTL	
B	2
D	1

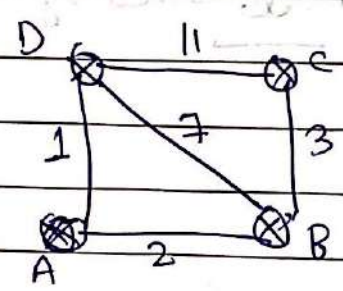
Link State Packet

B	
Seq	
TTL	
A	2
D	7
C	3

Now every router will flood its link state packet to every other router.

DVR – local knowledge
LSR – Global knowledge.

At A :-



A constructs the entire graph in its local memory (A has the global database)

A will apply Dijstra Algorithm (Single source shortest Path) and will construct routing table.

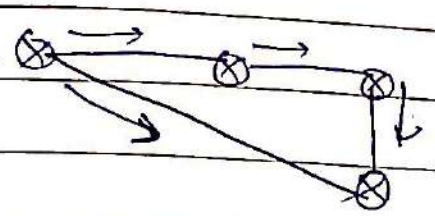
Therefore, LSR converges very fast.

Problem:

~~Too~~ Heavy Traffic (Flooding by every router)

Correction

Sequence Numbers →



To distinguish b/w the latest and old packets (Difference caused due to the delay of arrivals of packets due to flooding)

(A,10) ✓
(A,5) ✗

At A:

Router	latest Seq. No.
B	~~10~~ 15
C	20
D.	30

← (B,8) ✗ Rejected

← (B,15) ✓ accepted

Still, there could be ∞ loop

Solution

Time to live (TTL) flooding of a packet is done by TTL times only.

Date
Page

Another Problem again :-

At A :-

Router	latest
B	~~to~~ 31
C	20
D	30

Now 16
17
18
:
31
32

all rejected

~~0000~~ 1111

(B, 15)

Corrupted

(B, 31)

Solution

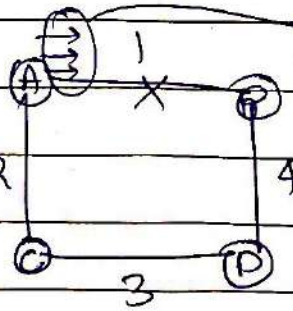
Router	latest	lifetime/validity
B	31	~
C	20	~
D	30	~

A particular packet will be used only till lifetime, after that new updated latest will be used.

Transient Problem :-

Black Hole problem.

looping Problem



A will take time to recognize that B is down, and packets sent by A will be lost.

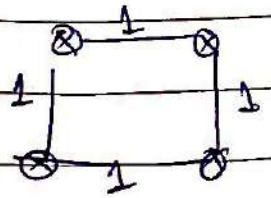
C → A → B
A → C

Difference b/w DVR and LSR

DVR	LSR
1980's	1990's
BWidth is less	Bandwidth is high
local knowledge	Global knowledge.
Bellman Ford type of algorithm	Dijsktra type of algorithm
Traffic is less	Traffic is very high
Periodic updates are done	Periodic updates are down
Converges very slow	Converges faster
Count to ∞	NO count to ∞
Persistent looping	Transient looping
RIP (Routing information Protocol)	OSPF Protocol (open shortest path first)
Computation Simple	Computation Complex

RIP (Routing Information Protocol)
← Implementation of DVR

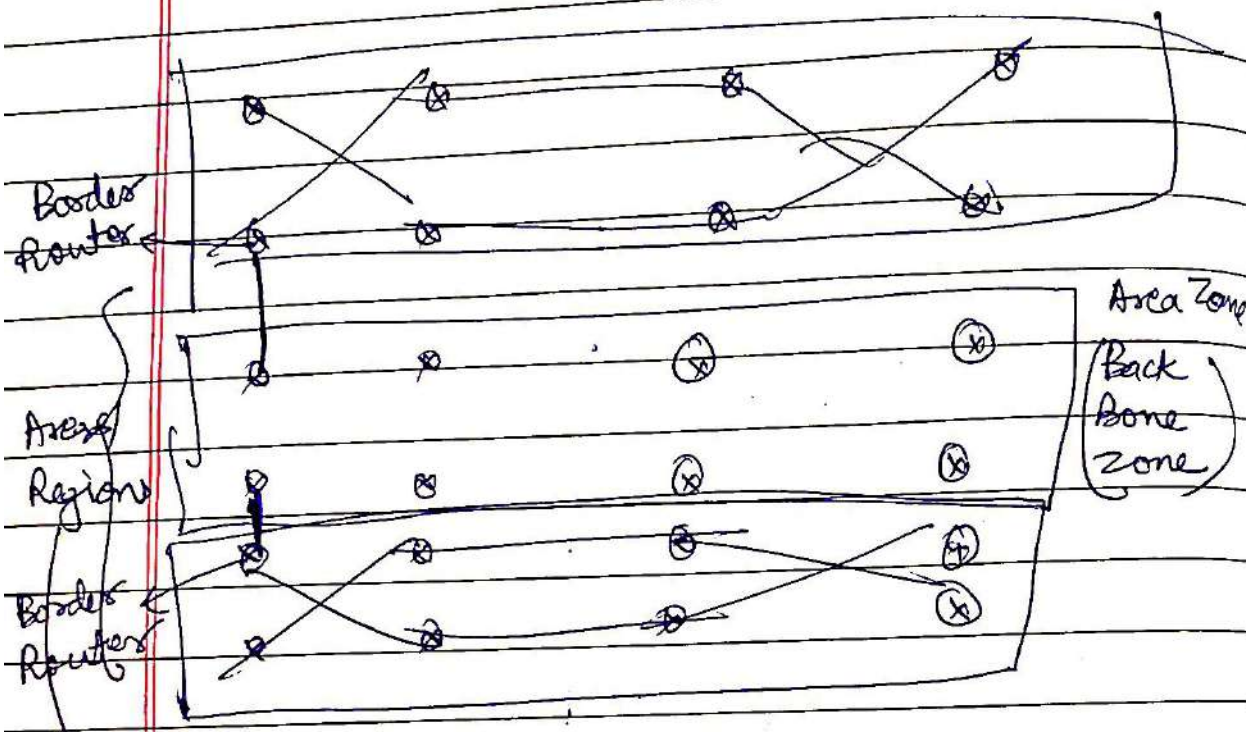
metric :- hop count



∞ → "16"

Date
Page

OSPF – Implementation of LSR
(open Shortest path first)



- Flooding is restricted to only one region at a time.

- Here hop count is NOT used as metric, System admin can set the weights of edges.

EIGRP → Combination of RIP and OSPF
↑
(Enhanced Interior Gateway Routing Protocol)

11. TCP

Transmission Control Protocol

- Transport layer
- End-to End protocol
- Connection Oriented
- Byte-Stream Protocol

TCP Header

4B	Source Port (16)								Destination Port (16)	
4B	Sequence number (32)									
4B	Acknowledgment Number (32)									
4B	Header Length (4)	6 bits Reserved	URG	ACK	PSH	RST	SYN	FIN	Window Size (or) advwindow (16)	
4B	CheckSum (16)								Urgent Pointer (16)	
20B	Options (0-40 bytes)									
40B										
60B	Data									

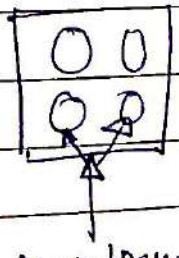
Min. header length = 20B,
Max. header length = 60B.

Source Port and Destination Port

SP → 16 bit
DP → 16 bit

$2^{16} = (0, 65536)$

- End-to End protocol
- Can do Mux / Demux



HTTP :- Port No :- 80
FTP :- 21
Telnet (Remote Login) :- 23

0, 1, 1023 } well known

SMTP :- 25

TCP → connection oriented (Reserved resources)

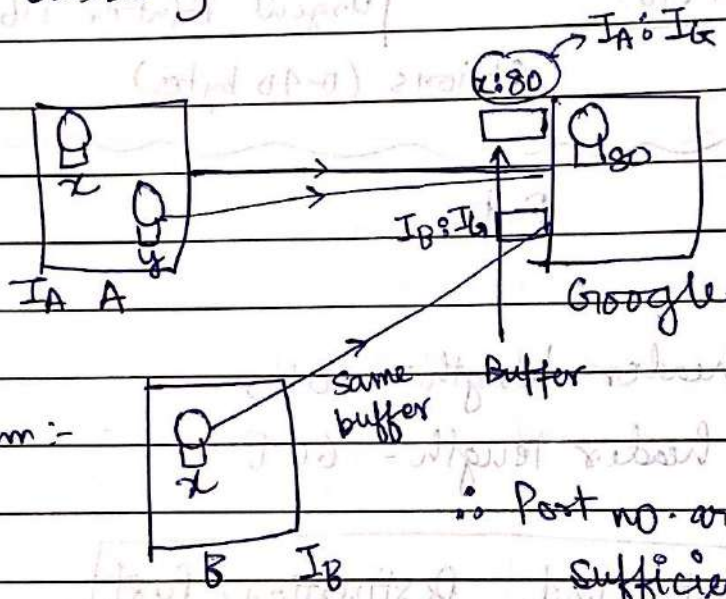
0 – 1023 } well known port numbers

1024 – 49151 } Reserved

49152 – 65535 } General Public

Socket :-

(48 bit)



problem :-

∴ Port no. are not sufficient

Solution :- use IP addresses

↓ Problem

another host from same network access same host on Google

↓ Solution

∴ cannot use only port no. or only IP address

↳ use both as a combination

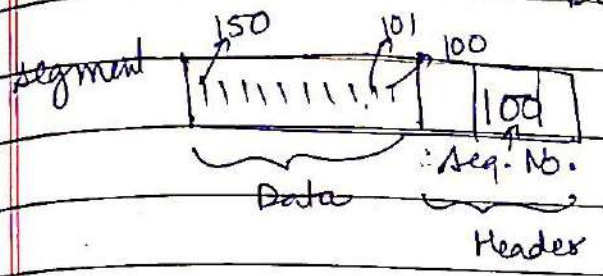
↓ called ($I_A : I_G : x : 80$)

Socket = IP + Port

32 16 bit

Sequence Number

- TCP is a Byte-Stream protocol
- IP is a packet stream
- DLL → HDLC → bit stream protocol



Acknowledgement Number

Sequence No. of segment expected next

For this example,

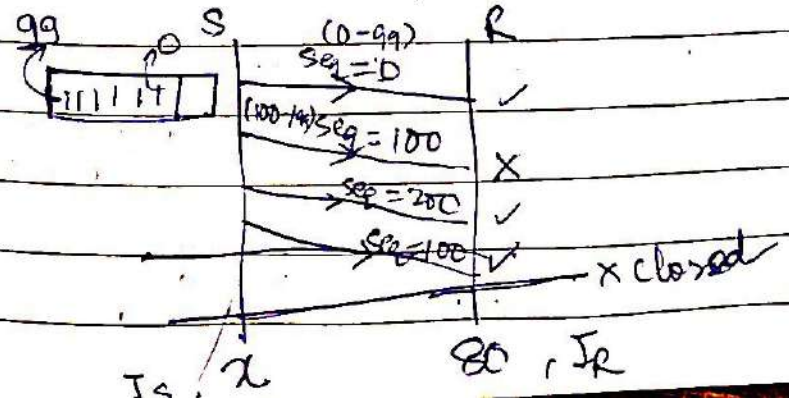
Ack No = 151

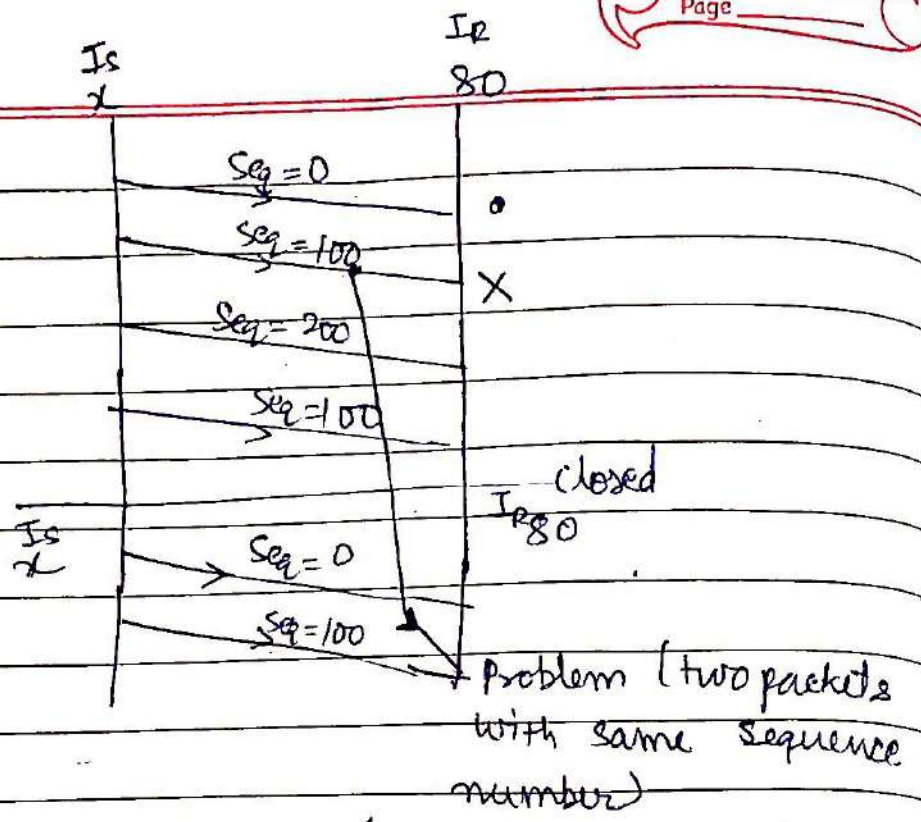
Problem :- How to calculate last byte seq. no.?

↓ Solution

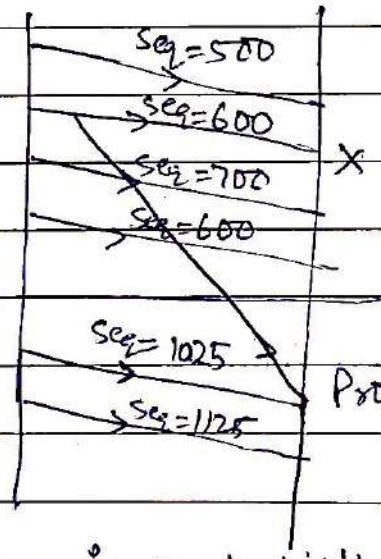
It will calculated from IP header
(Every TCP segment will be contained in datagram)

Sequence Numbers possible :- 0 to $2^{32}-1$





Solution
Don't start with zero
Start with a random sequence no.



Still there is probability that same no. may come up again as starting seq. no.

But, $P = \frac{1}{2^{32}} \simeq 0.00000000 \ldots$ very less.

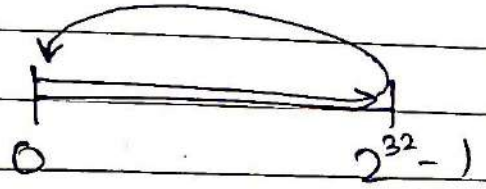
- TCP uses random initial

Wrap around in sequence numbers:-

$2^{32} = (4G)$ seq. no.

$\underbrace{4GB}$ + 1B

after this much data, wrap around and use the same seq. no. (0)



concept will not change for random initial sequence numbers too.

Wrap around time :- Time taken to finish using of all sequence numbers

eg. 2^{32} sequence numbers

BW = 1MBps

1s = 1MB

10^6 B = ~~1 seq no~~ 1s

10^6 seq. no. = 1s

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

2^{32} seq. no = $\frac{2^{32}}{10^6}$ s ≈ 4294.96s

enough large no.
< lifetime
(3 mins)

As long as WAT > LT

wrap around time Life time

eg. If BW = 1 GBps

1s = 1 GB

10^9 B = 1 s

10^9 seq no = 1 s

2^{32} seq. no = 4.294 s

≃ 4 s. << LT

When WAT < LT there is a problem

Possible ↓ solution

Don't increase b/w ~~within~~ beyond a limit

↓ NOT GOOD SOLUTION

Valid sol^n

↓

Increase sequence number bits

eg. For BW = 1 GBps, LT = 180 s.

min (seq. no.) = ?

~~2^{32} seq no~~ 1 s = 1 GB

= 10^9 B

~~1 B =~~ = 10^9 seq. no

$180\,s = 180 \times 10^9$ seq. no.

$= \lceil \log_2 (180 \times 10^9) \rceil = \boxed{42 \text{ bits}}$

32 bits + 10 bits

seq. no. field — Options field (Time stamp).

Header Length

4 bits but length of header = 20 to 60 Byte

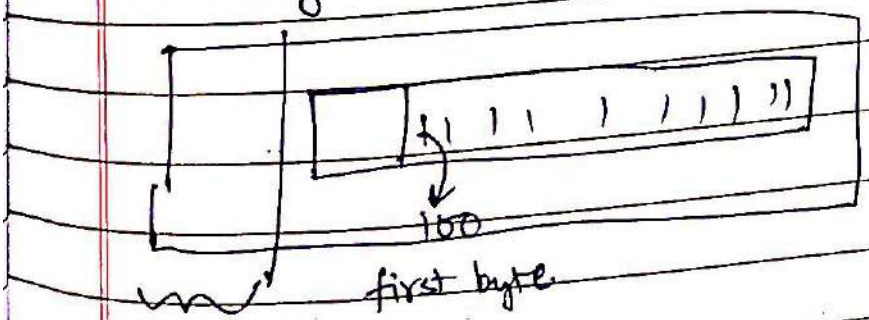
↓

0 to 15 (But have to store 20 → 60)

$\frac{60}{15} = 4 \rightarrow$ Scaling factor

Sequence No :- seq. no. of 1 byte

Acknowledgement No :- Seq. no. of byte expected next.



Total length − header IP = Body size IP = TCP header + Body TCP

→ TCP Body = Total length − IP header − TCP header

seq. (end) = seq. (start) + TCP body.

Date
Page

eg. TL = 1000, HL = 5 } IP datagram
↓ inside which is

HL = 5, seq = 100 (Starting) } TCP segment
Sequencence no's

$$\text{seq (end)} = 100 + (1000 - \underset{\times 4}{5} - \underset{\times 4}{5}) - 1$$
$$= 1060 - 1 = 1059$$

$\therefore$ ~~Ack no = 1060 + 1 = 1061~~

$\therefore$ Ack no $= 1059 + 1 = \boxed{1060}$

~~Flag~~
~~(Synchronization Flag)~~

SYN :-	Synchronization Flag
ACK :-	Acknowledgement Flag

C	S

* TCP is full duplex connection

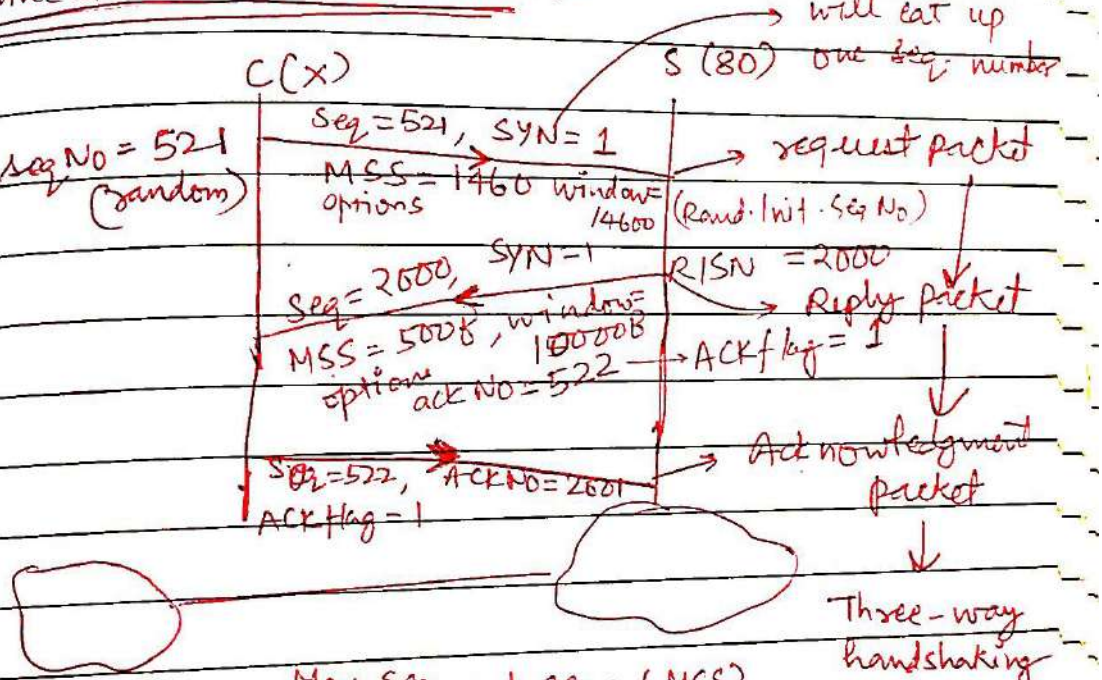
- SYN = 1, 1 Seq No
  ACK = 1, 0 Seq No
  FIN = 1, 1 Seq No
  1 Data Byte, 1 Seq No

* TCP is connection oriented protocol

Phases:-

1. Connection establishment
2. Data transfer
3. Connection termination

Connection establishment :-



Max. Segment Size (MSS)

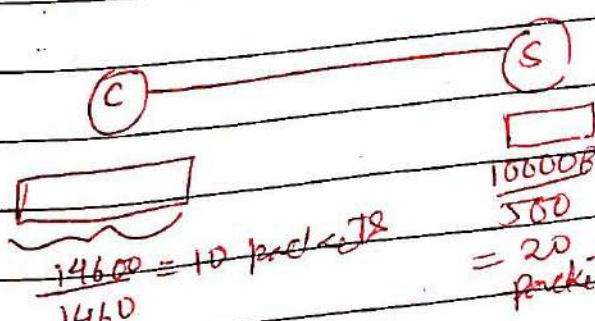
TL [1460 | 20]

NL [1480 | 20]

DLL [T | | H] 1500B

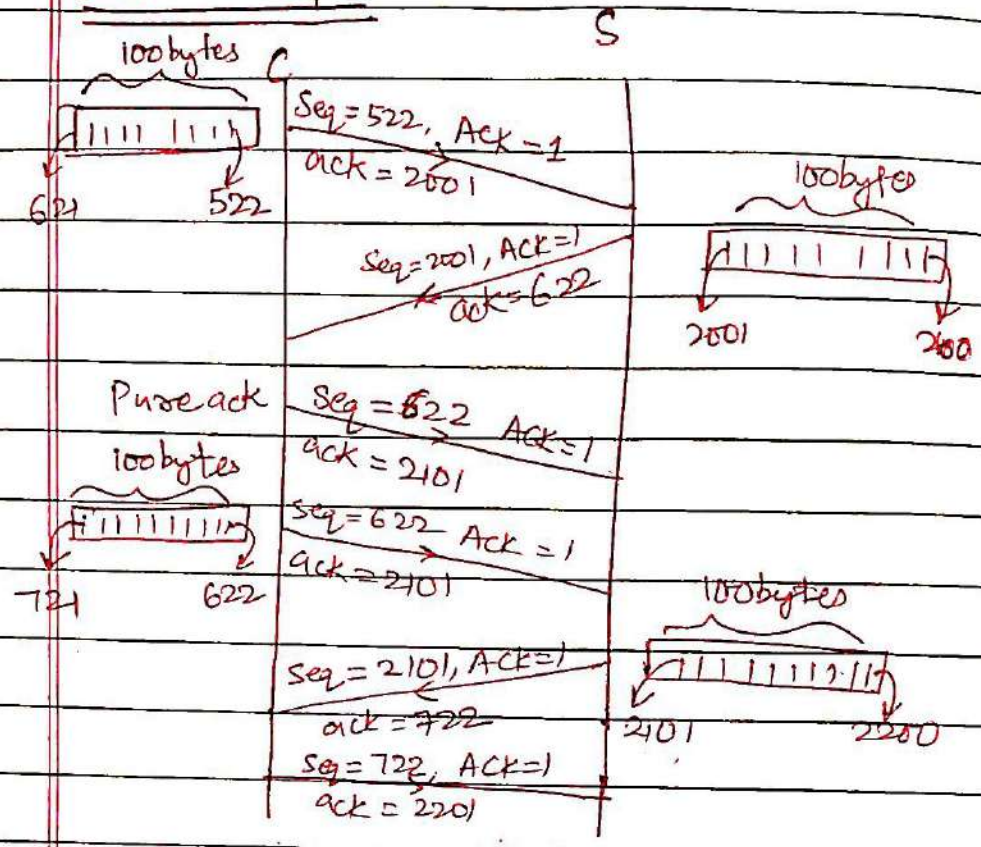
Window Size :

Capacity of the host as a receiver (Flow control)



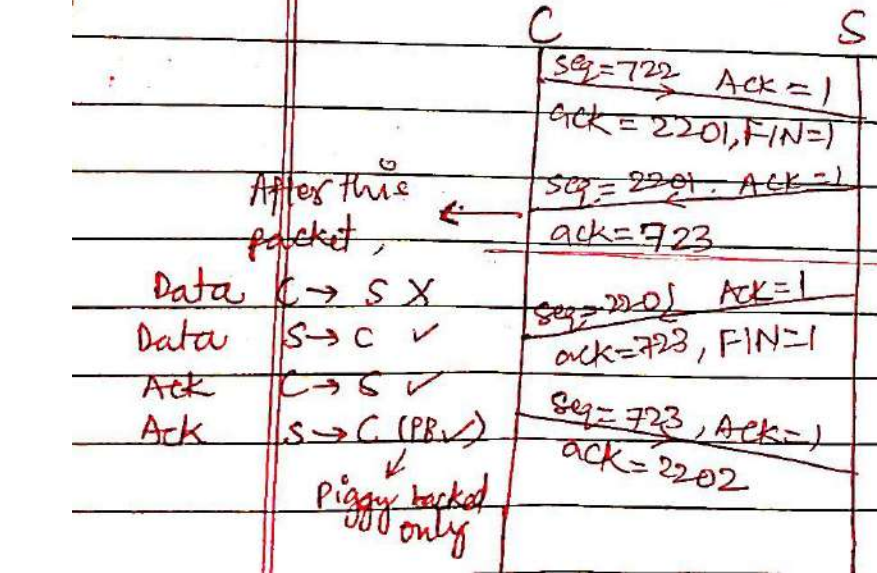
$\frac{14600}{1460}$ = 10 packets

Data Transfer (After establishment)

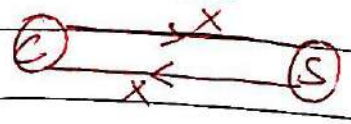


SYN	ACK	
1	0	– I segment/ request packet
1	1	– II segment / Reply packet
0	1	– Acknowledgement present
0	0	– NOT possible

Connection Termination (After data transfer)



FIN - Finish Flag

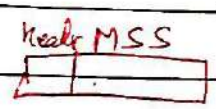


SYN +
ACK +
FIN
Connection Management

used in connection termination

SYN Flag :- Synchronizing Sequence numbers
Ack Flag :- indicate whether Ack num field is valid or NOT.
FIN Flag :- Request for connection termination

PSH Flag - Push Flag.

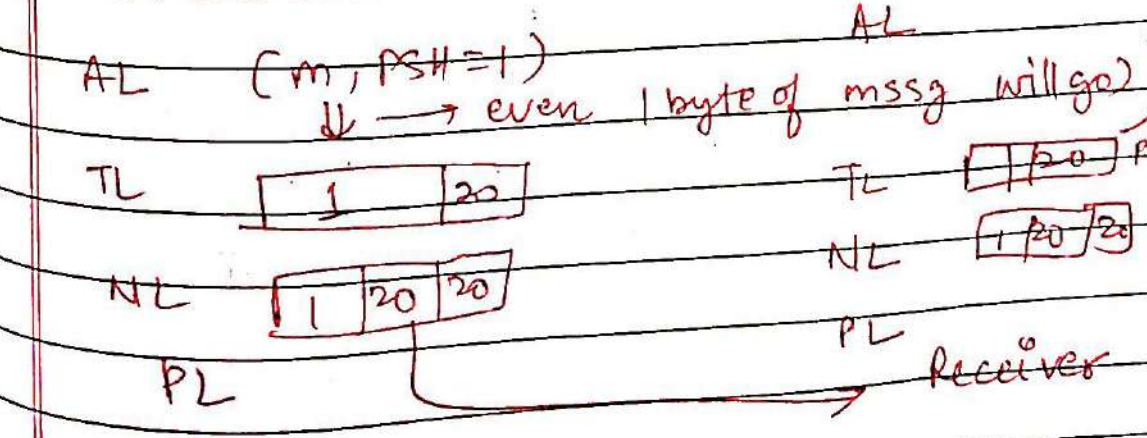


AL
TL
NL
DLL
PL

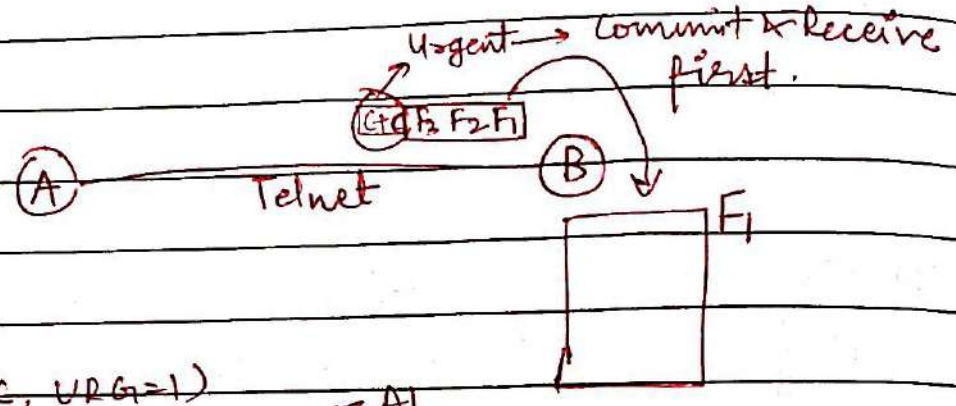
TL does not send the data immediately after getting it from appl^n layer. - "Buffering"

Interactive applications do not want TL to wait for MSS size of data to be sent. (Chat app)

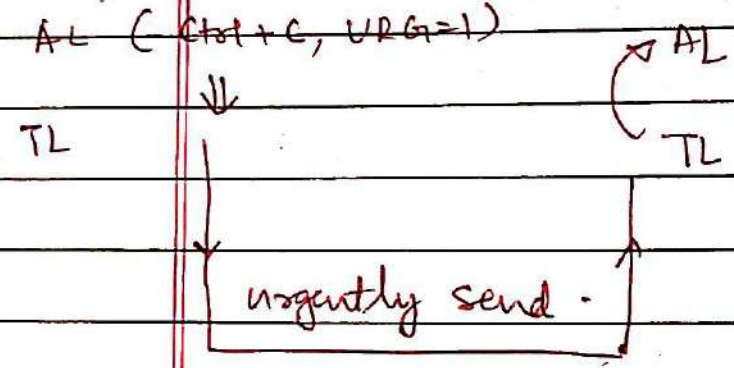
If appl^n layer sends PSH = 1, Buffering will not happen

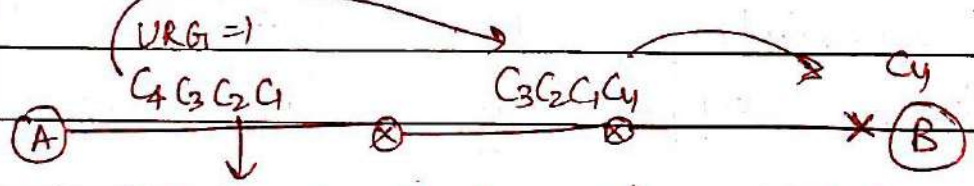


URG Flag & Urgent Pointer



AL (Ctrl + C, URG = 1)



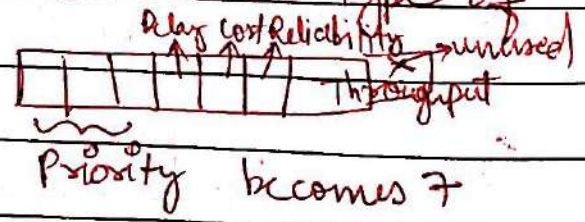


Problem :- Router does not have Transport layer

↓ Solution

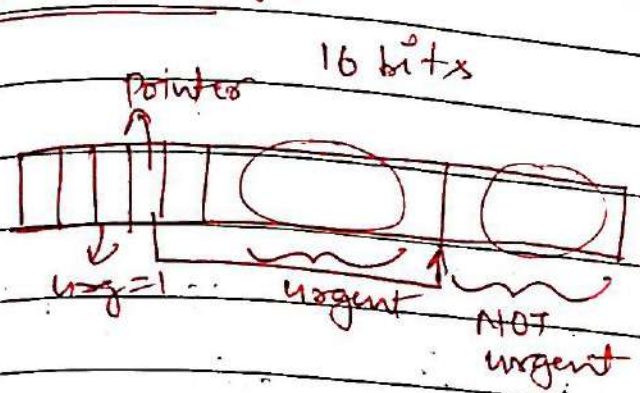
Set priority to 7

So that in NL (in the type of service field)



and at router, packet with max. priority will be sent first

Urgent Pointer :-



If urg = 1
urgent packet
urg pointer
to which part data is urgent

Seq = 1000
URG = 1
urg. pointer = 100

Seq + urg. pointer = 1100
→ seq. no. of large urgent byte

1000 — 1100

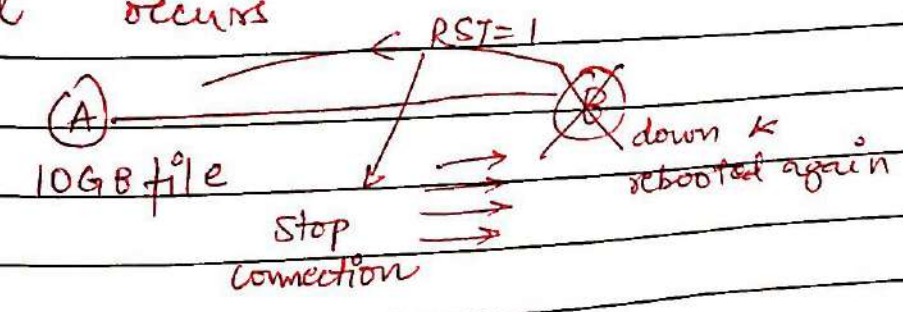
urgent data = 101 bytes

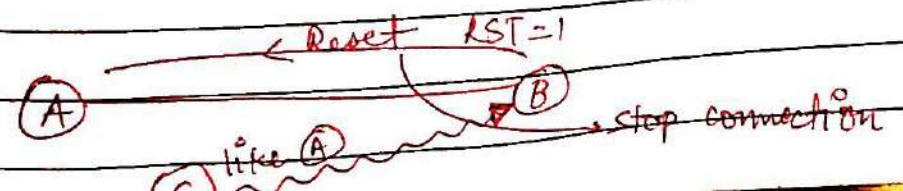
if urg pointer = 100, # bytes urgent = 101

∴ # bytes urgent = (urg pointer) + 1

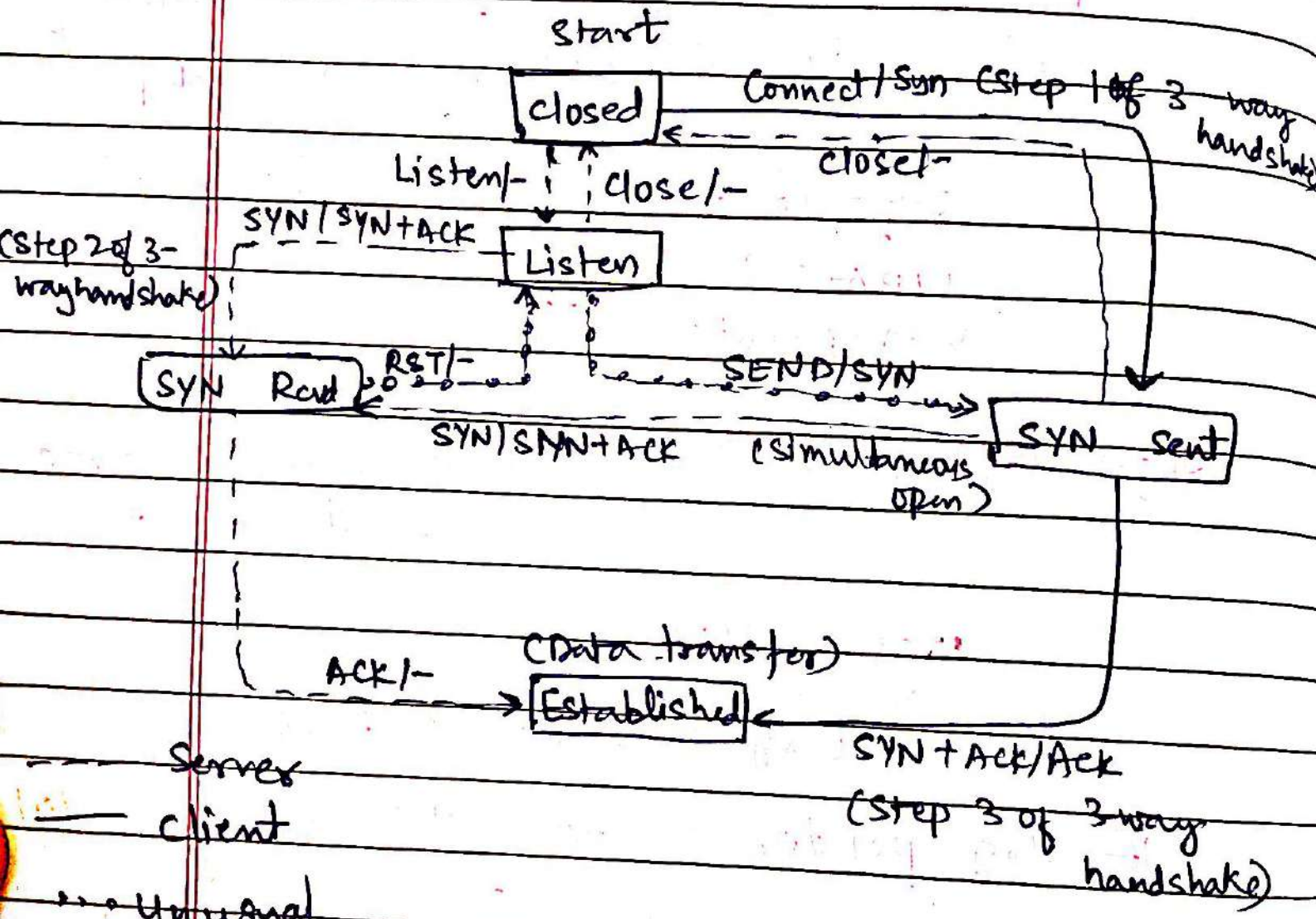
RST Flag

Reset Flag :- when anything WRONG or unexpected occurs

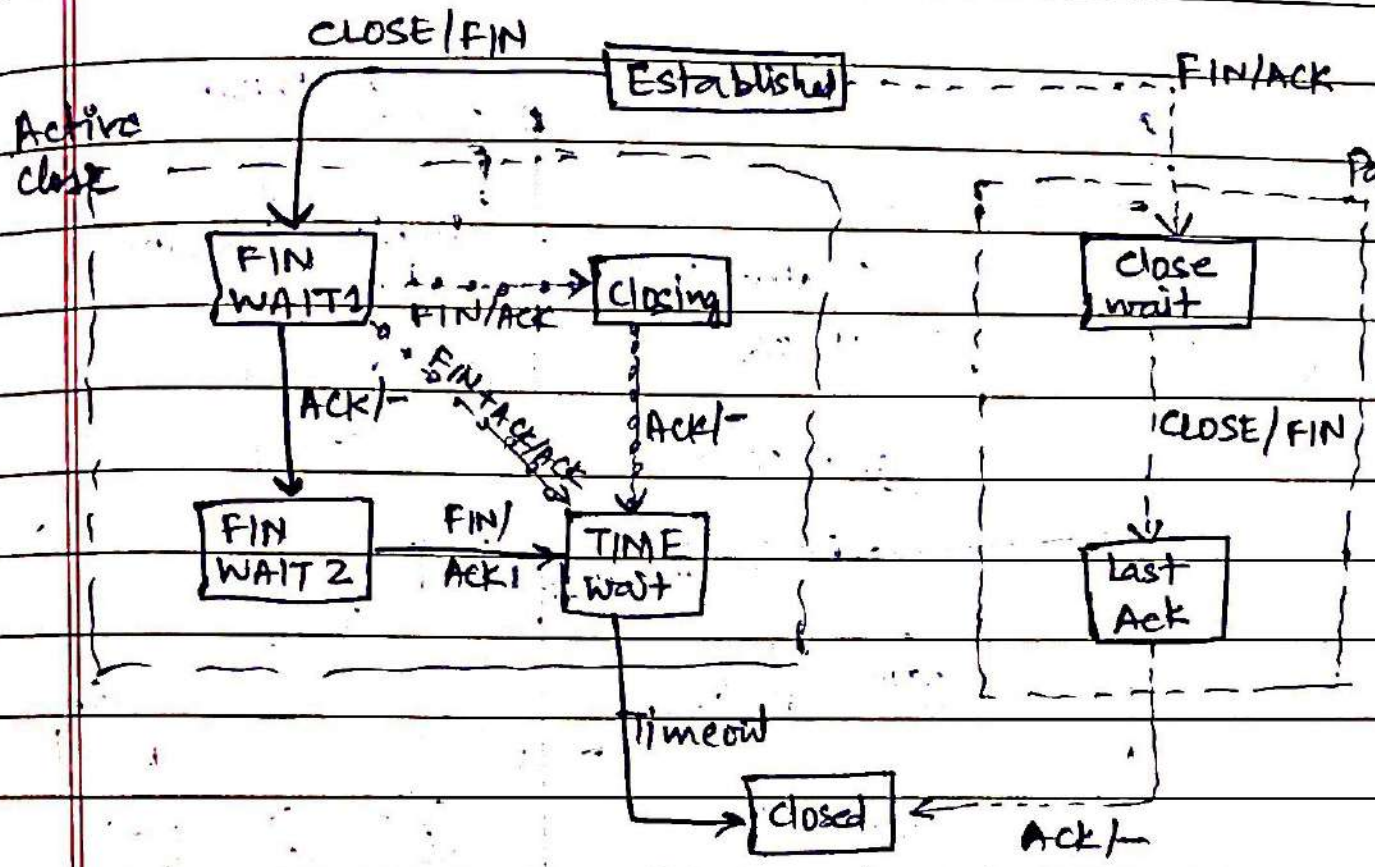




TCP State Transition Diagram
(Connection Establishment)



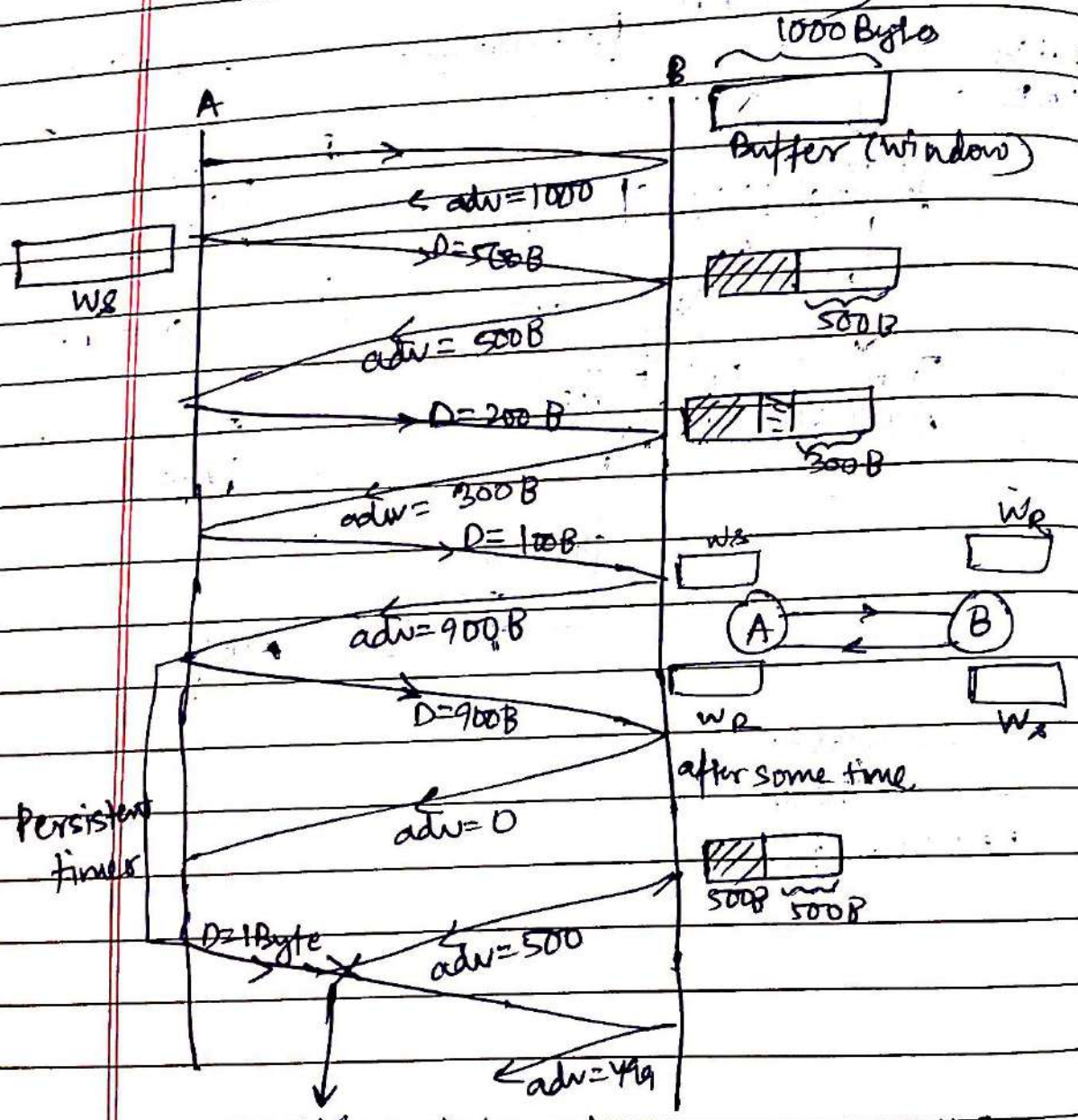
TCP state Transition Diagram
(Connection Release)



---- Server
—— Client
.... Unusual

Window size or Advertisement window

(16-bits)



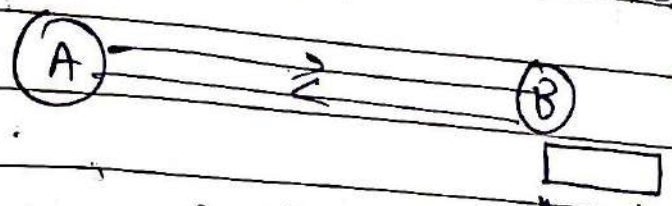
A thinks that B is busy & B thinks that A has no data to send ⚠ Deadlock

↓

Solution → Persistent times

Send 1B data after this time

Disadv :- 16 bits only (65535 max)



max buffer size = 65535 ≃ 64 KB

Solution

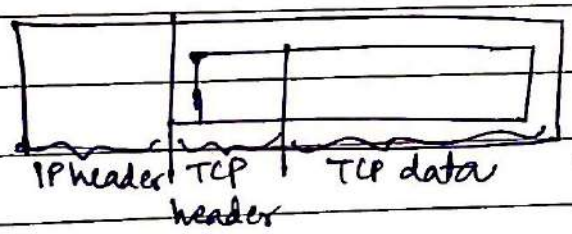
16 bits wind size + 14 bits option

= 30 bits

max = $2^{30} - 1$ = 1GB Buffer size.

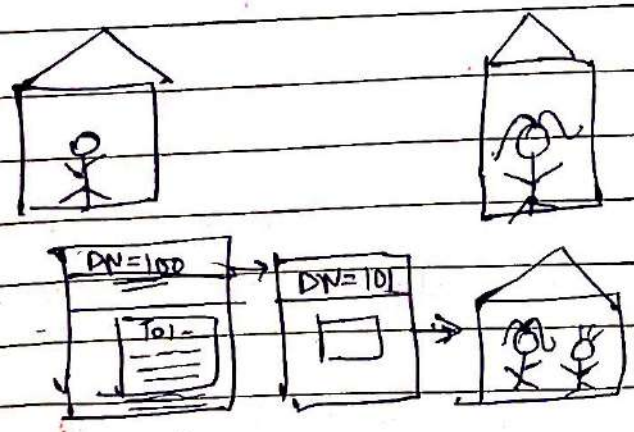
Check Sum

(16-bits)



= on three fields, TCP check sum is Calculated

why?



Solution

write address both on envelope & inside.

TCP Cs = TCPH + TCPD + IPH

↓ might change

solution

don't take complete IPH
use important fields only

SIP		(32)
DIP		(32)
zero (8)	Protocol (8)	TCP Seg length (16)

} Pseudo IP-header

(Not complete IPH)

Options

(1) Time stamp → used when (WAP < L.T.)

(2) Window size extension

(3) Parameter negotiation

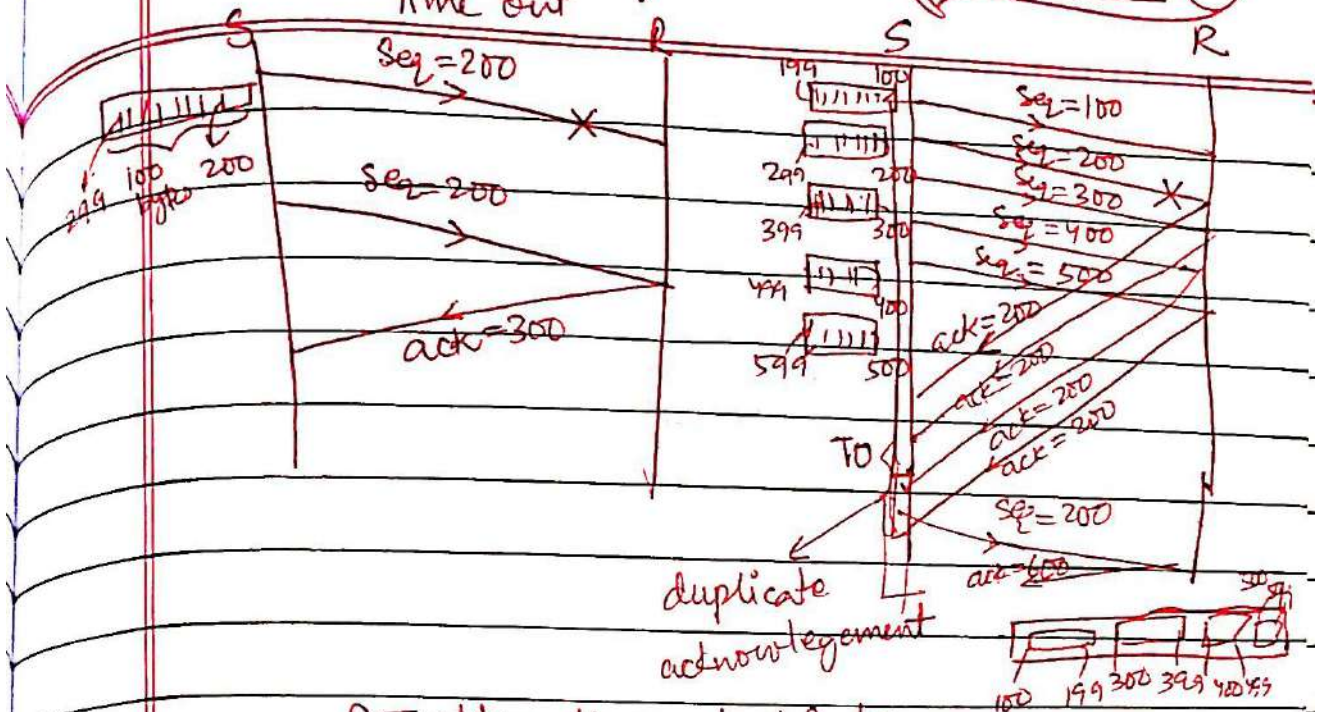
(4) Padding
↓
to make header length 14

Retransmissions in TCP

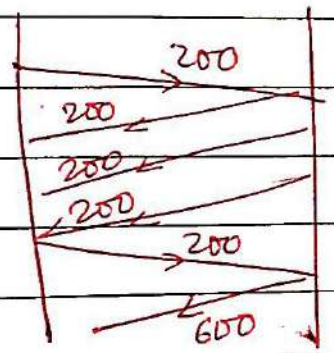
- TCP uses SR + Go Back N
  - → $W_S = W_R$
  - → out of order packets will be accepted
  - → Acknowledgments used are cummulative

75% SR + 25% GBN

Retransmission after Time out

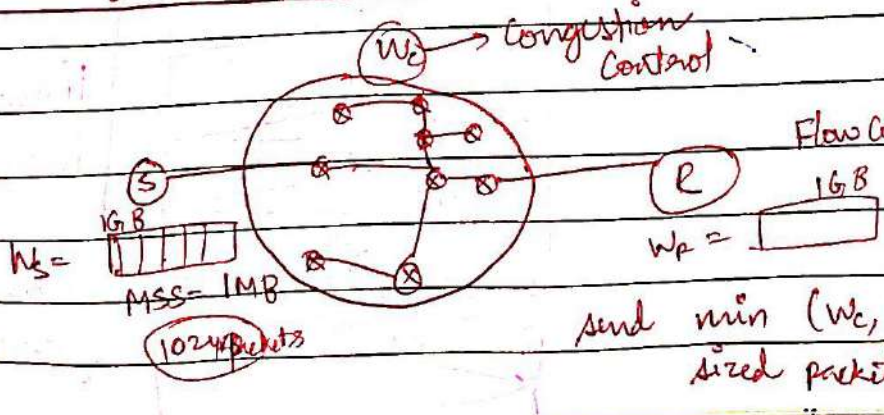


RT after three duplicate acknowledgement
(Early retransmission)

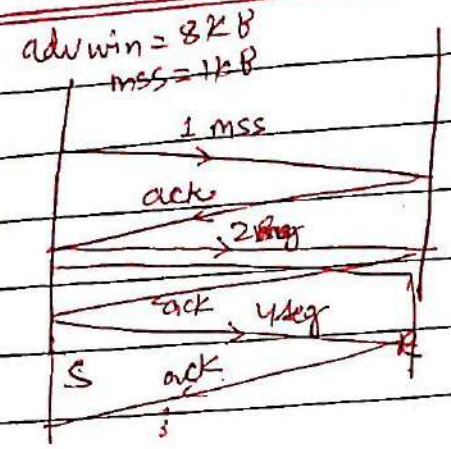


- Time out indicates that congestion is severe but three duplicate ack indicates that congestion is NOT that severe.

TCP congestion Control



eg.



adv win = 8 kB

mss = 1 kB

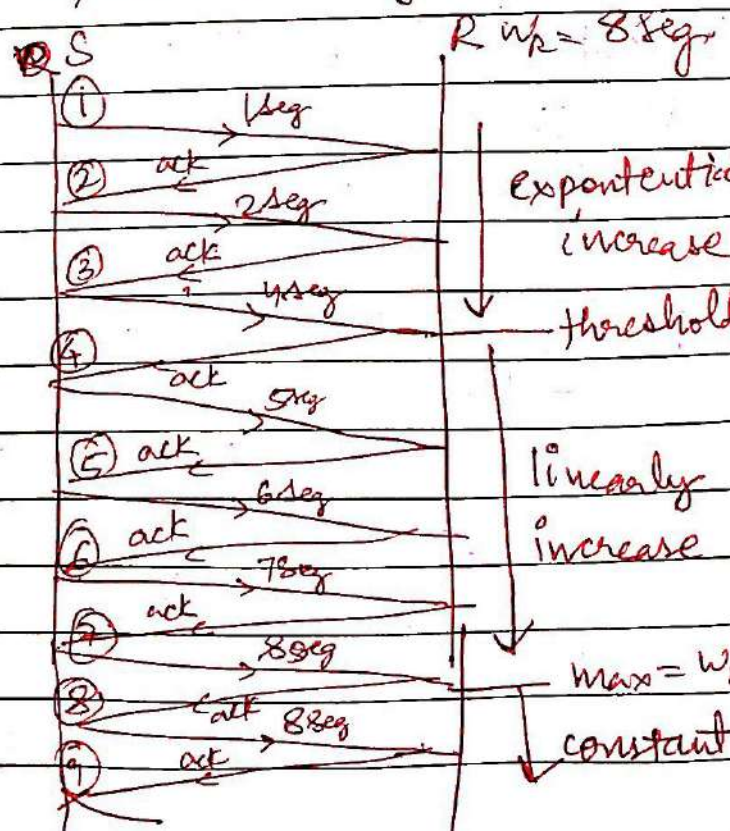
W_R = 8 segment — Receiver window

W_C = 1 segment — Congestion window (take min for first time)

$W_S = \min(W_R, W_C)$
$= 1$

Threshold $= \frac{W_R}{2} = 4$

after 4, ~~send~~ increase linearly



Q. After how many round trip times do we reach max. sender capacity window? →

$\boxed{6\ RTT}$

eg. adv win = 16 KB

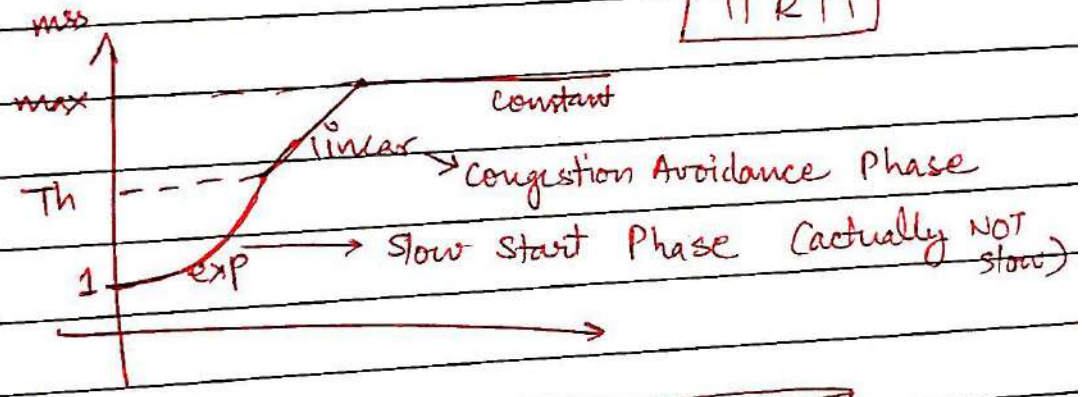
mss = 1 KB

W_R = 16 segments

W_C = 1 segment $Thr = \frac{16}{2} = 8$

1, 2, 4, 8, 9, 10, 11, 12, 13, 14, 15, 16

$\boxed{11\ RTT}$



TCP Congestion Control Algorithm

Phase I :- Slow Start Phase

Phase II :- Congestion Avoidance Phase

Phase III :- Congestion detection :- ~~Phase~~

(a) Time out (Severe congestion)

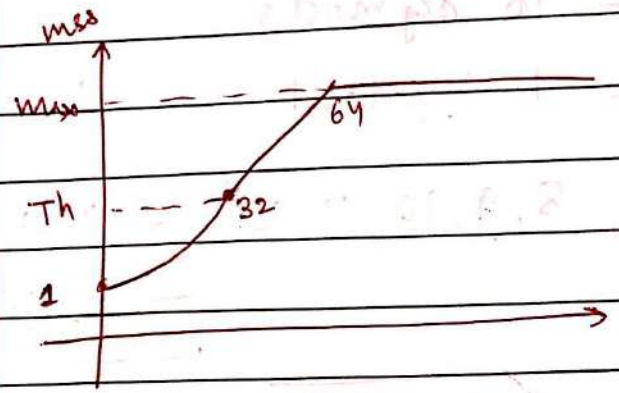
(b) 3 duplicate acknowledgements (Mild congestion)

(c) ICMP source quench

eg. $W_R = 64\,KB$ $\quad$ $W_R = 64$ mss

$MSS = 1\,KB$ $\quad$ $Th = 32$ mss

$\underbrace{1\ 2\ 4\ 8\ 16\ 32}_{\text{Slow Start}}\ \underbrace{33\ 34\ 35 \ldots 63\ 64}_{\text{Congestion Avoidance}}\ 64\ 64\ 64 \ldots$



Congestion Detection :-

eg. 1 2 4 8 16 32 33 34 ↑ TO

new threshold $= \frac{1}{2} W_c = \frac{34}{2} = 17$ mss

Start from
Slow Start phase (when TO occurs)

34 ↑ TO 1 2 4 8 16 17 18 19 20 ↑ 3duplicate

new threshold $= \frac{1}{2} W_c = \frac{20}{2} = 10$ mss

Start from
Condition avoidance (when 3DAck occur)

20 ↑ 3DAck 10 11 12 ↑ TO

new threshold $= \frac{12}{2} = 6$

12 1 2 4 6 7 8

TO

3 dup Ack

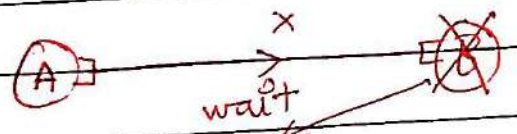
new thr $= \frac{8}{2} = 4$

7 8 4 5 6 7 8 9... 64 64....

3d Ack

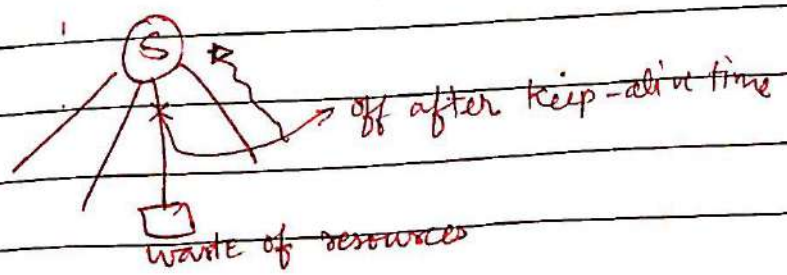
TCP timer management

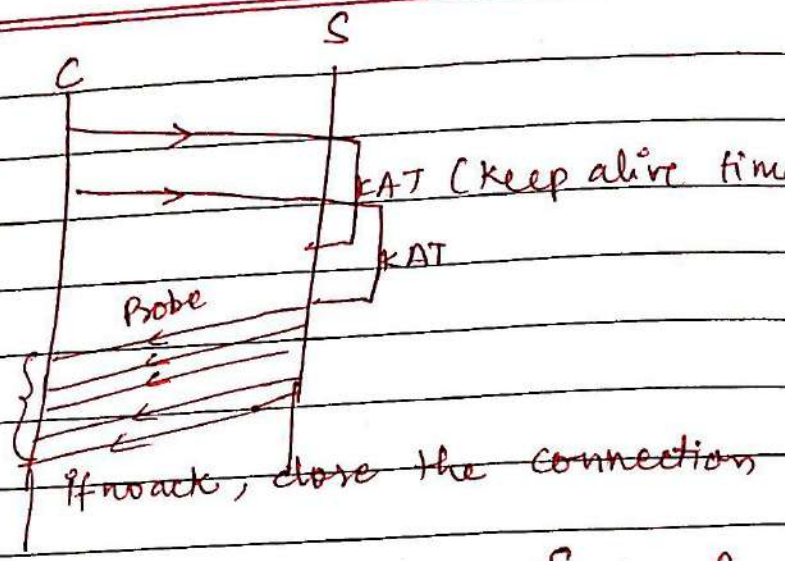
(i) Time - wait timer (Late packets)
(ii) Keep alive timer (close idle connections)
(iii) Persistent timer (window = 0 Probe)
(iv) acknowledgement timer (cumulative ack + piggy bag)
(v) Time- out timer

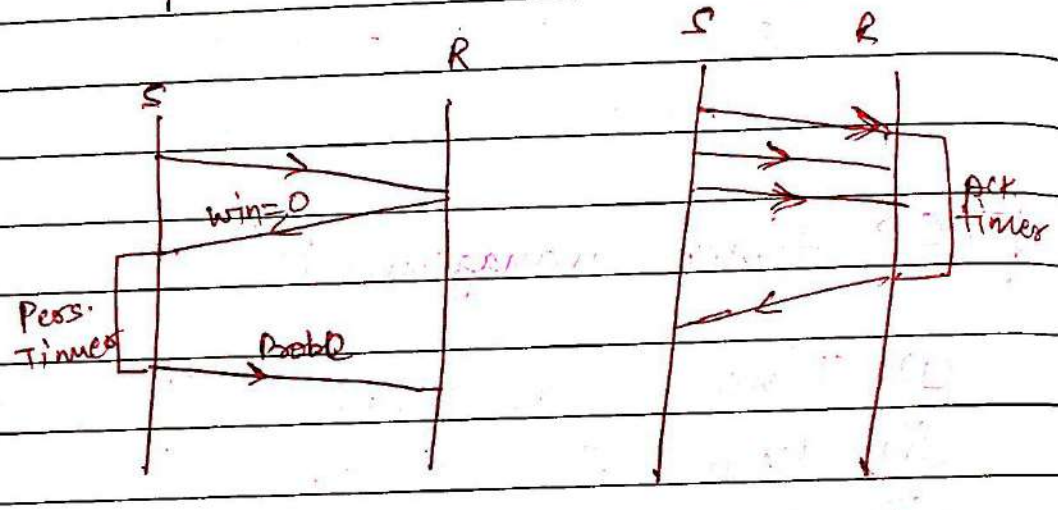


Time-wait → 2* LT time so that delayed packet may arrive safely

Keep- alive → To check of idle timers

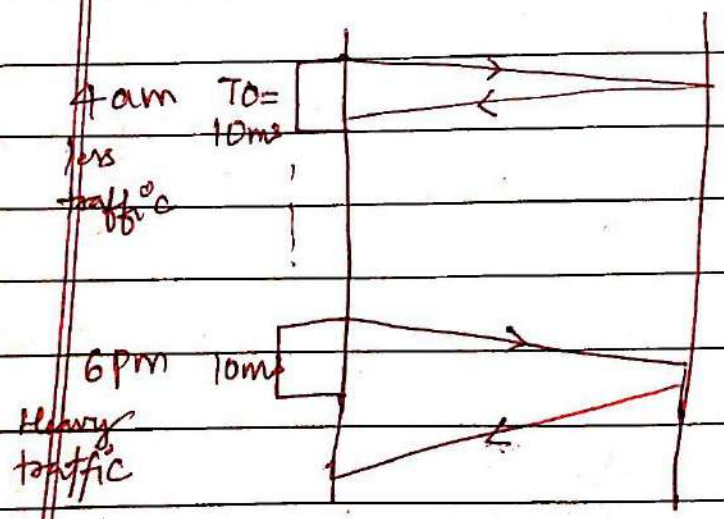






Time out Timer

Time out timer cannot be Static, in TCP.



Algorithms :-

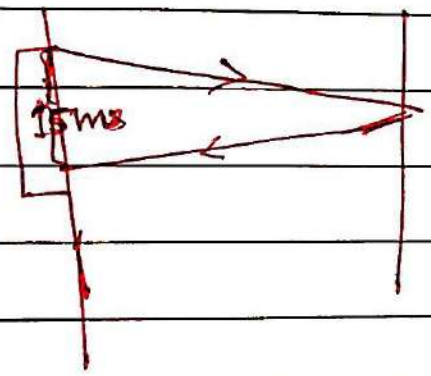
1) Basic

2) Jacobsons

Basic Algorithm for Timeout Timer Computation

Initial R.T.T. = 10 ms (Guess)

T.O = 2 × RTT = 20 ms



Actual R.T.T = 15 ms (Luckily)

T.O. timer too large $\xrightarrow[\text{in}]{\text{results}}$ Unecessary time waste

T.O. timer too small $\xrightarrow[\text{in}]{\text{results}}$ Unecessary retransmissions

Next R.T.T. = α I RTT + $(1-\alpha)$ ARTT

α → smoothening factor

If $(\alpha == 1)$ → static T.O. timer

If $(\alpha == 0)$ → Purely Dynamic T.O. timer

$0 \leq \alpha \leq 1$

~~eg~~ let $\alpha = 0.5$

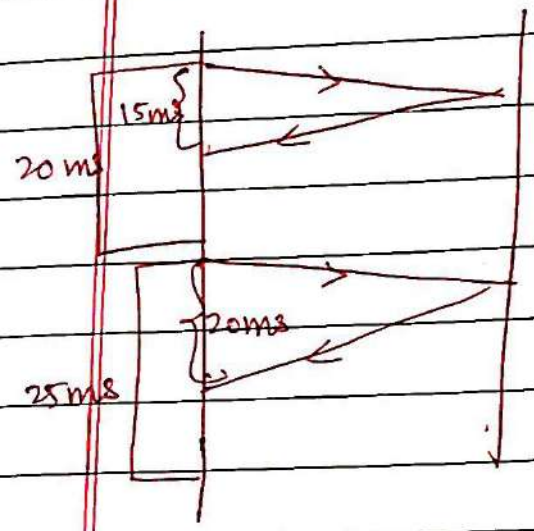
Next R.T.T. = 12.5 ms

~~T.O. = 2 × RTT~~

For next packet :-

IRTT = 12.5 ms

T.O = 2 × RTT = 25 ms



ARTT = 20 ms

NRTT = α IRTT + $(1-\alpha)$ ARTT

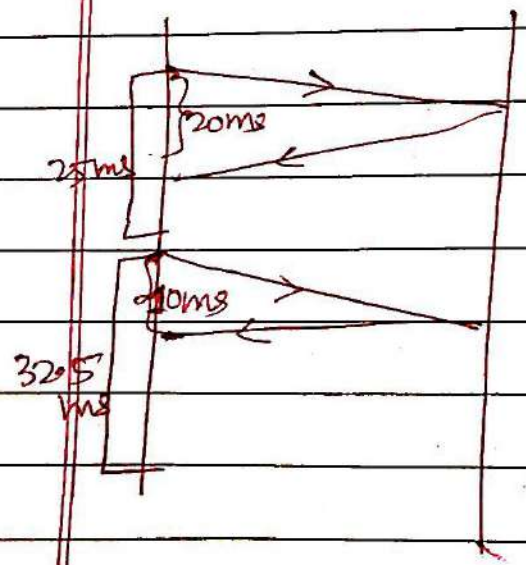
$= \frac{\cancel{12.5}\ 20}{2}$ ms

= 16.25 ms

For next packet :-

IRTT = 16.25 ms

T.O = 2 × RTT = 32.5 ms



ARTT = 10 ms

NRTT = α IRTT + $(1-\alpha)$ ARTT

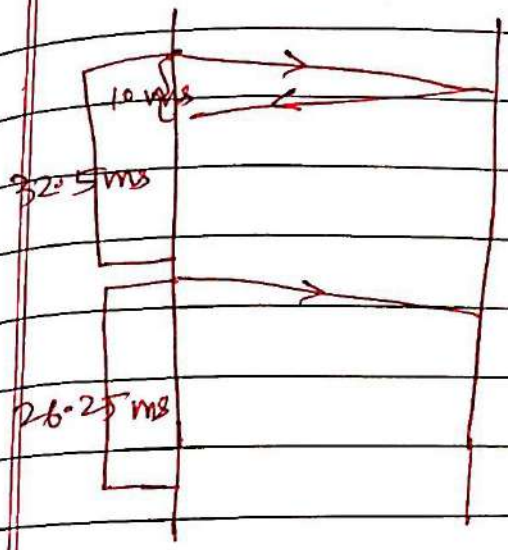
$= \frac{10 + 16.25}{2}$ ms

= 13.125 ms

For next packet :-

IRTT = 13.125 ms.

T.O. = 2x RTT = 2x 13.125 ms = 26.25 ms



and so on
⋮
(dynamic T.O. timer)

Disadv :- Always T.O = 2x RTT
No logic behind it

correction
Jacobson's Algorithm

Jacobson's algorithm for timeout timer Computation

eg. IRTT = 10 ms (Guess)

ID = 5 ms (Deviation Guess)

TO = 4 x D + RTT
= 4 x 5 + 10
= 30 ms

ARTT = 20 ms

~~AD = 10 ms (|IRTT - ARTT|)~~

Date
Page

$NRTT = \alpha(IRTT) + (1-\alpha)(ARTT)$

$= 15\,ms$ (Let $\alpha = 0.5$)

$ND = \alpha(ID) + (1-\alpha)(AD)$

$= 7.5\,ms$

For Next packet :-

$IRTT = 15\,ms$

$ID = 7.5\,ms$

$TD = 4 \times D + RTT$

$= 45\,ms$

$ARTT = 30\,ms$

$AD = 15\,ms$

$NRTT = 0.5 \times 15 + 0.5 \times 30$

$= 22.5\,ms$

$ND = \frac{7.5 + 15}{2} = 11.25\,ms$

For next packet :-

$IRTT = 22.5\,ms$

$ID = 11.25\,ms$

$T.D = 4 \times D + RTT = 67.5\,ms$

$ARTT = 10\,ms$

$AD = 12.5\,ms$

$$NRTT = \frac{22.5 + 10}{2} = 16.25 \text{ ms}$$

$$ND = \frac{11.25 + 12.5}{2} = 11.875 \text{ ms}$$

$$TO = 4 \times D + RTT = 63.75 \text{ ms}$$

∴ and so on

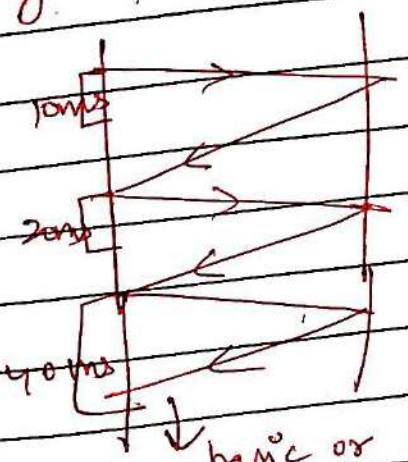
Disadv :- next packets depend upon all preceding packets. (when time out occurs)

↓ correction

Karn's modification

Karn's modification

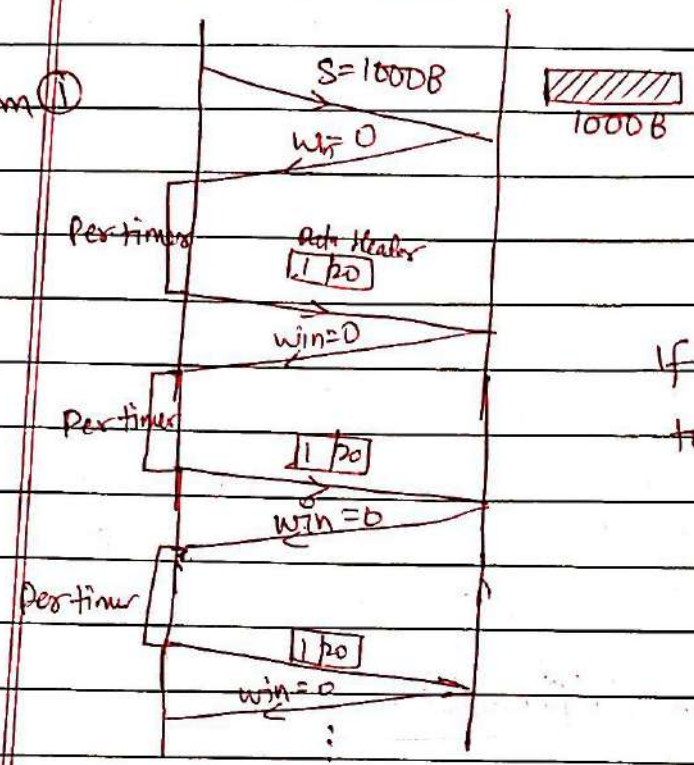
whenever a time-out occurs, the next packet's T.O. timer don't apply basic or Jacbson's, just keep on doubling the T.O. timer value.



↓ basic or Jocobson's continued.

Silly window Syndrome

Problem ①



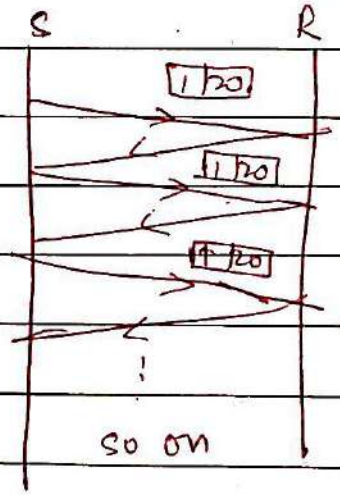
If Receiver is too slow

and so on...

Solution window size = 0 for long time

use Reset flag

Problem ②



If sender is too slow

Nagle's Algorithm

wait for 1RTT and see in this time, how much data is produced by AL and then send that much data in one segment

If 1mss is filled in a time less than 1RTT, send 1mss immediately

eg 1 Byte/ms → Application layer speed

RTT = 100 ms.

mss = 200 Byte,

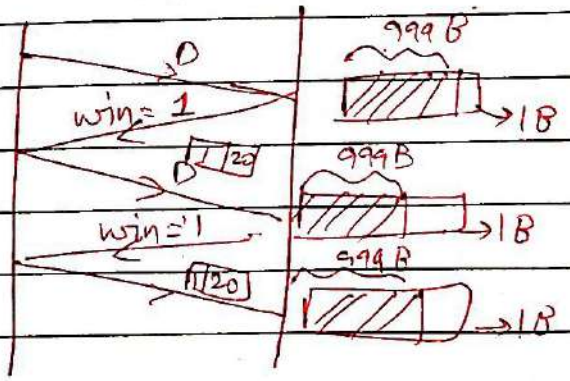
wait for 1RTT ⇒ 100 Bytes < mss
↓ so send
100 Bytes each time

eg if mss = 50 Byte

wait for 50 ms (50 Bytes = mss) Only.

and send 50 Bytes each time.

Problem ③



Receiver consumes 1B only after every tip

Clarck's Solution

Receiver should wait for atleast ½ of the buffer ~~and~~ or 1mss and then advertise its window size

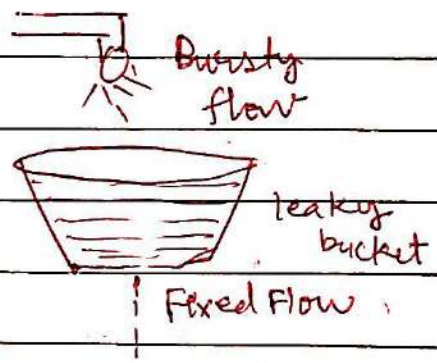
Traffic Shaping

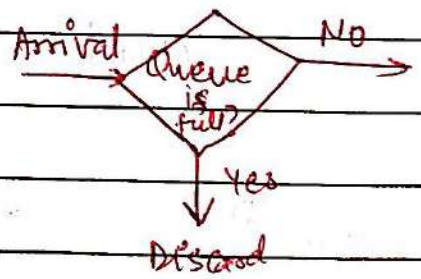
→ Another method of congestion control is to "shape" the traffic before it enters the network.

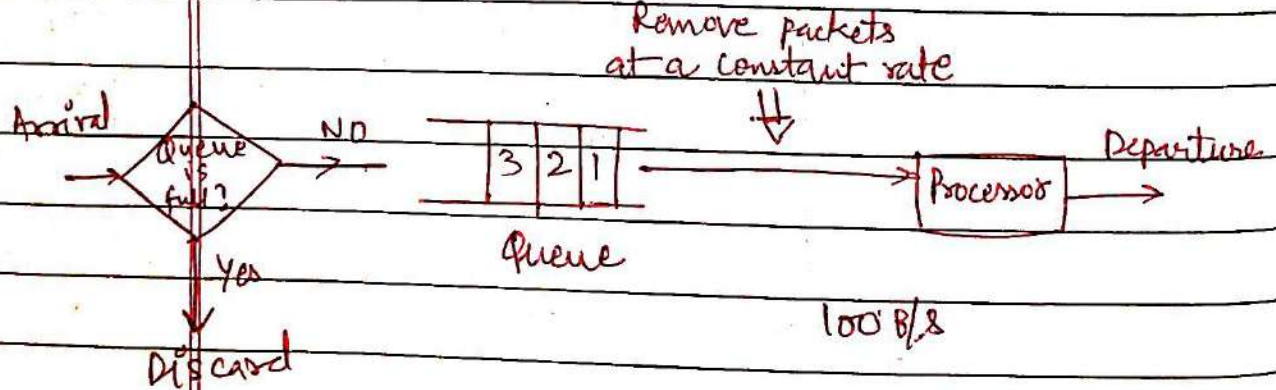
→ Traffic shaping controls the "rate" at which packets are sent.

→ During connection establishment, the sender and the carrier negotiate a traffic pattern (shaping).

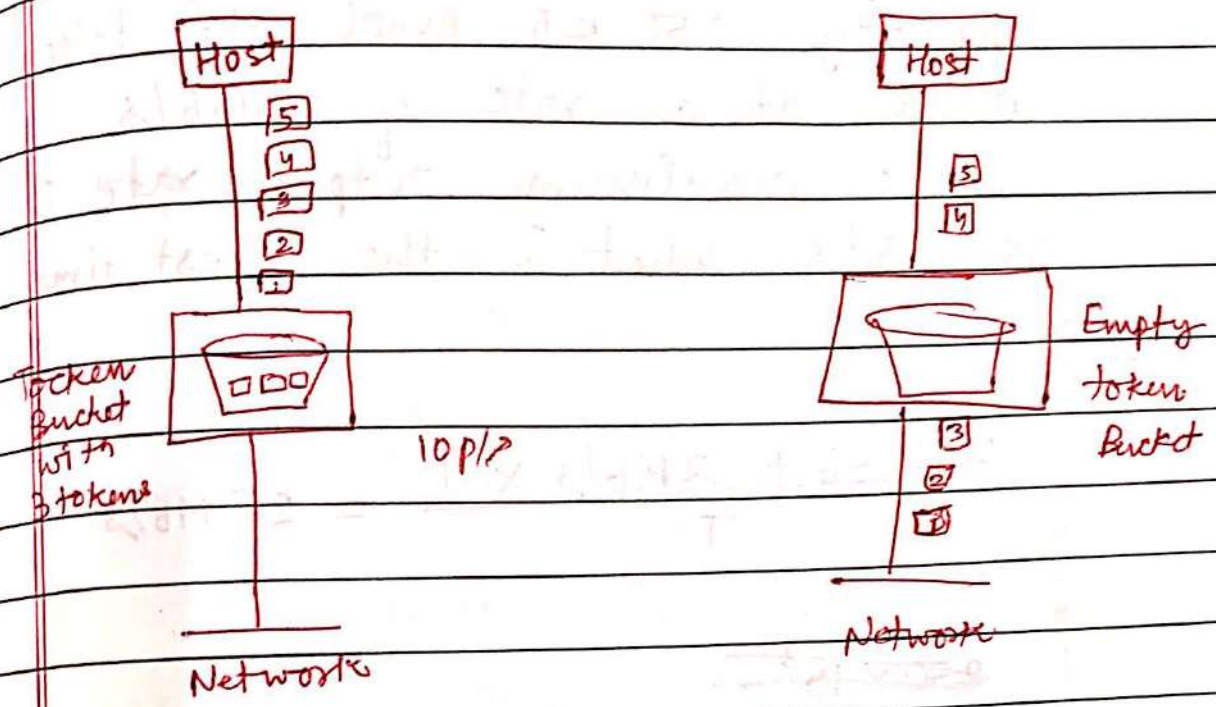
Leaky Bucket







Token Bucket



Let the capacity of token bucket be 'c' tokens and the tokens enter the bucket at the rate of 'r' tokens per sec. The maximum number of packets that can enter the network during a time interval 't' is :-

Max. no. of packets $= c + rt$

$\therefore$ Max. avg. rate $= \frac{(c + rt)}{t}$ packets / sec

eg. ~~Consider~~

Date
Page

eg. Consider a token bucket of capacity 250 KB and the tokens arrive at a rate of 2MB/s. If the maximum output rate is 25 MB/s, what is the burst time?

Sol:-

$$\frac{250\ KB + 2MB/s \times T}{T} = 25\ MB/s$$

$\cancel{250 \times 10^3}$

$$250 + 2 \times 10^3 T = 25 \times 10^3 T$$

$$\Rightarrow 23 \times 10^3 T = 250$$

$$\Rightarrow T = \frac{250}{23 \times 10^3}\ s$$

$$= 10.86\ ms$$

12. UDP

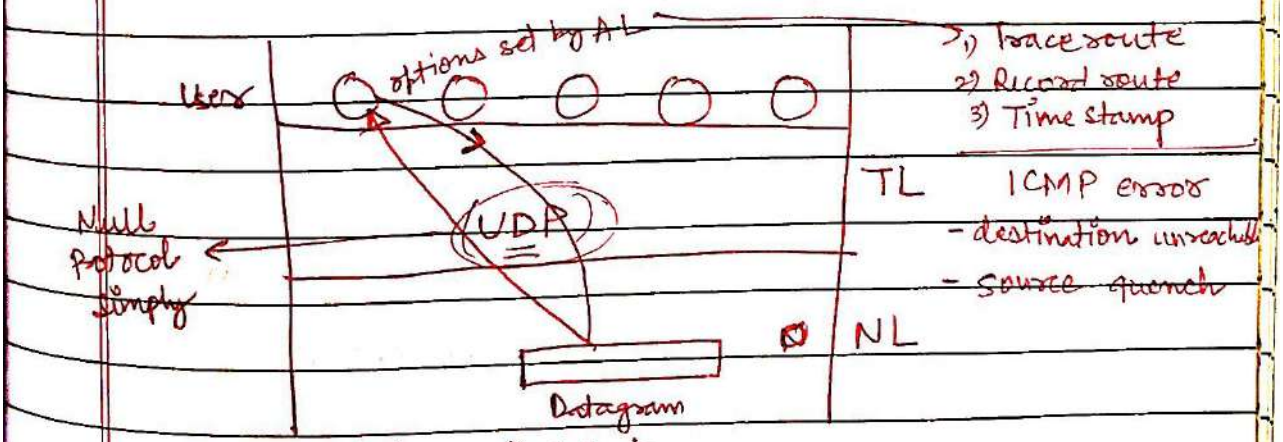
User Datagram Protocol

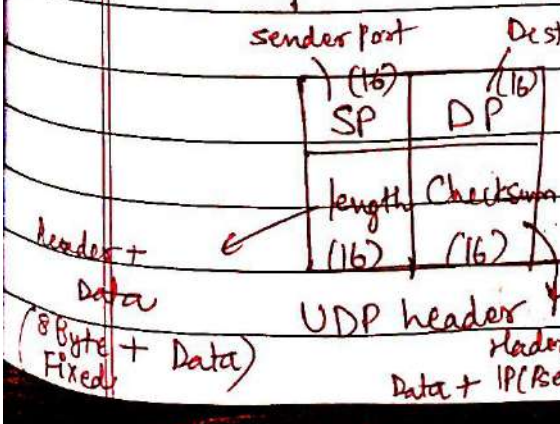
Need for UDP

1) Application needs 1 request / 1 reply
   eg. ① DNS — Domain name server
   ② BOOTP
   DHCP — MAC → IP
   ③ NTP — Network time protocol (Find time)
   ④ NNP — Network news protocol
   ⑤ quote of the day protocol

2) Broadcasting / Multicasting application

3) Fast Application (Rather than Reliability)
   eg ① Online games
   ② Multimedia application





- No connection establishment / termination
- No acknowledgements
- No sequence numbers
- No window size (flow control)
- No flags
- No options

13. Hardware and various Devices used in Networking

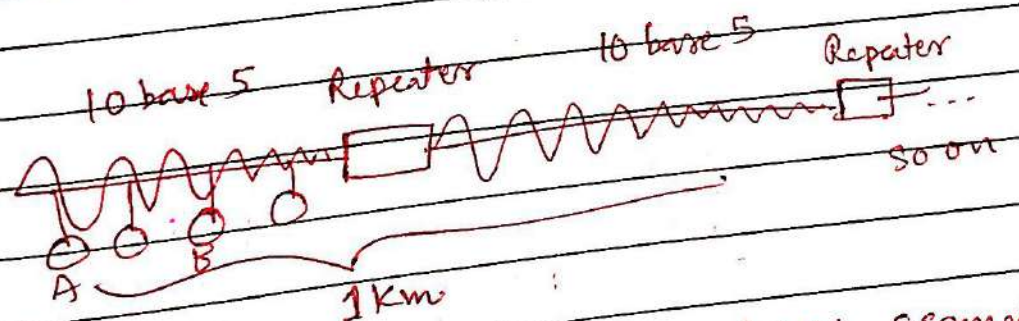
Cables

Ethernet Cables :-

10 Base T (10 MBps, No multiplex, 100 m)
10 Base 2 (" " , 200 m)
10 Base 5 (" " , 500 m)
100 Base T (100 Mbps , " , 100 m)

↳ operate at Physical layer
- ~~Authe~~ Attenuation
- Collisions (Coll. Domain = n)

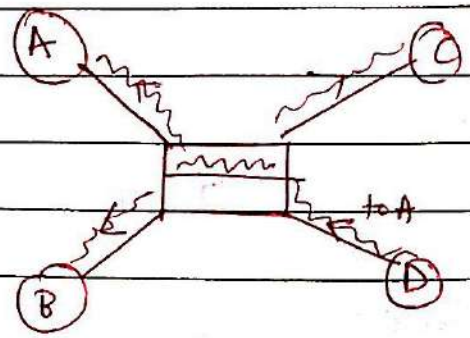
Repeaters



- used to connect two LAN segment
- works at physical layer
- Collisions are possible (Coll. Dom = n)
- Range of LAN is increased
- Unecessarily increase traffic

Hubs

↳ Multi-port Repeaters

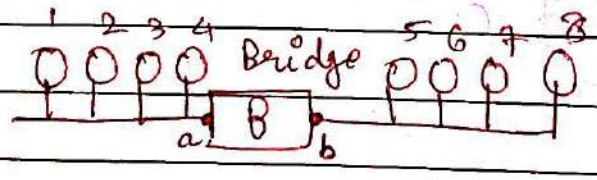


- Traffic is very high
- Physical layer only
- Collisions are possible (coll. Dom = n)
- Flooding

Advantages:-

- very cheap.

Bridges



- Both physical & DLL

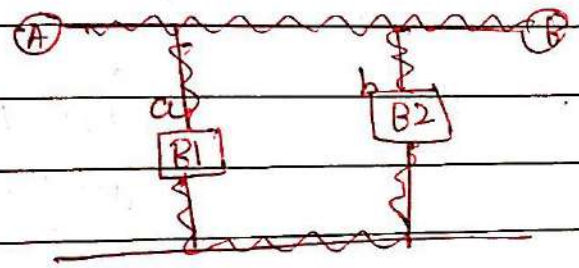
Mapping table

MAC	Port
1	a
2	a
3	a
4	a
5	b
6	b
7	b
8	b

→ Static (Net. Admin types) - Manual

→ Dynamic or learning or transparent

- Thus, message can be filtered
- Forwarding device
- Flooding (when destination NOT in mapping tables)
- Store & Forward
  (No collisions inside)
  (Collision Domain is reduced)



Problem
↓
message in a loop
↓ Solution
Spanning - tree algorithm

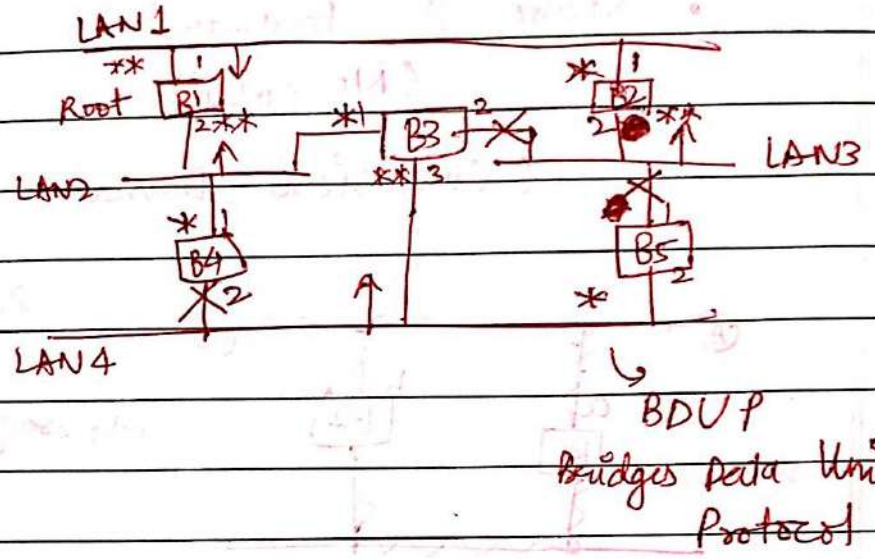
- Theoritically, bridges should be able to connect LANs of different proto types, practically we do not do it. (MTU's may be different)

Spanning Tree Algorithm :-

1) Every bridge has a built in ID. The one with smallest ID is taken as root bridge.
2) Mark one port of each bridge which is closest to root bridge as root port.
3) Every LAN chooses a bridge closest to it as a designated bridge for that

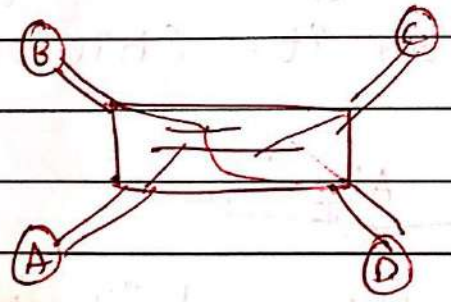
LAN. Make the corresponding port as designated port.

4) Mark the root port and designated port as forwarding port and block remaining



Switch

- Both PL and DLL
- No collision



- Collision Domain = 0
- Traffic is reduced

Disadvan:-

- Costly

LAN components

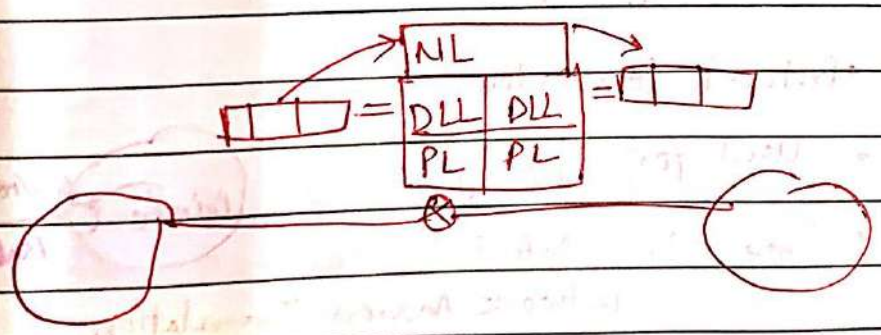
Wire + Hub + Repeater + Bridges + Switches

↓

Any of these devices should not decrease the 'broadcast domain'.

	Broadcast Domain	Collision Domain
Repeater	Same	Same
Hub	Same	Same
Bridge	Same	reduces
switch	Same	reduces (to zero)
Routers	Reduces	Reduces
Gateways	Reduces	Reduces

Routers

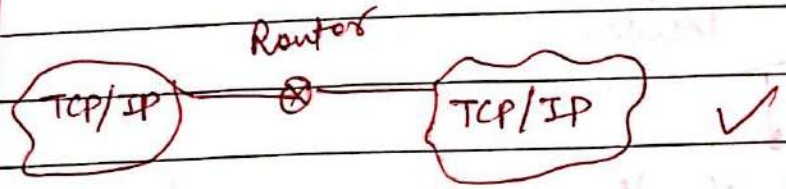


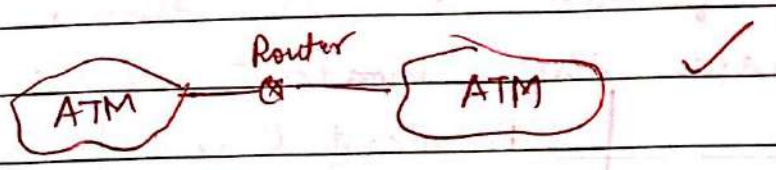
- Forwarding
- Filtering
- Flooding

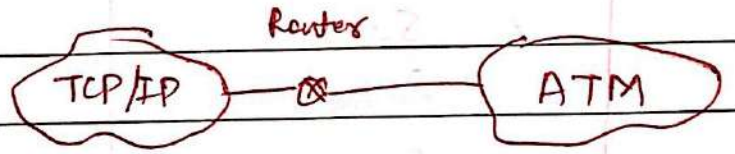
★ Costly

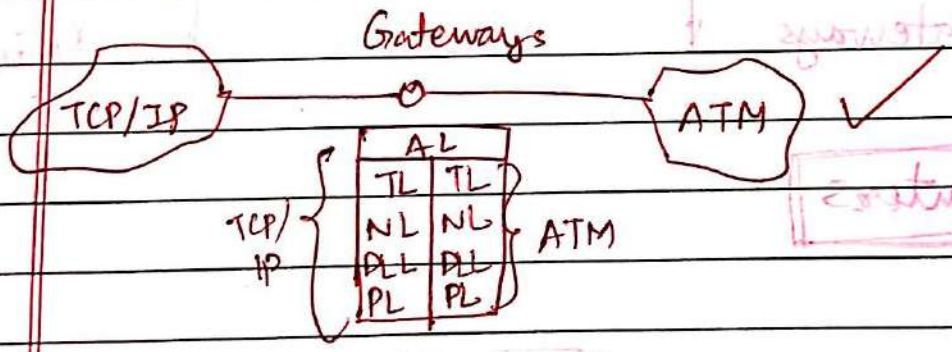
- Routing
- No collisions
- Br/c Domain ↓

Gateways

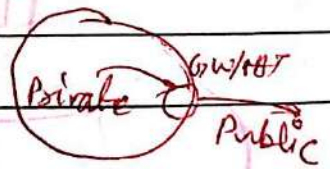






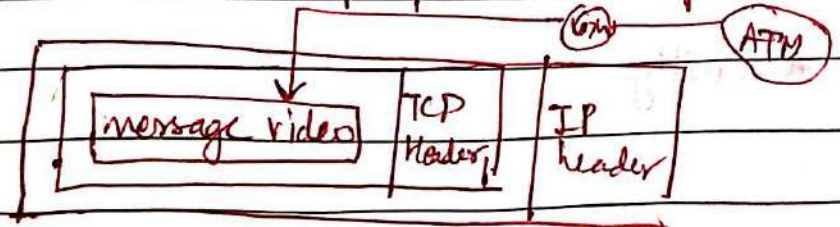


- Protocol Converter
- Used for proxy
- Used as NAT

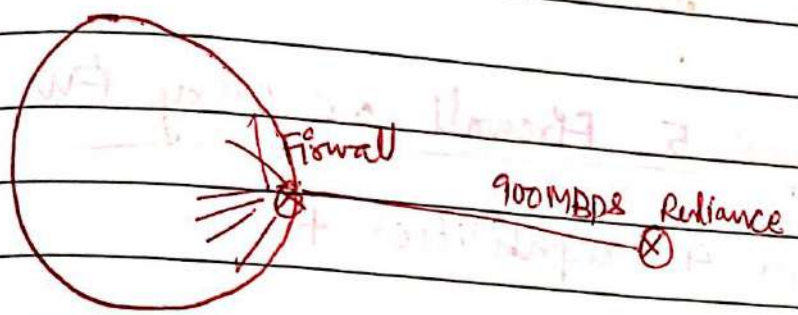


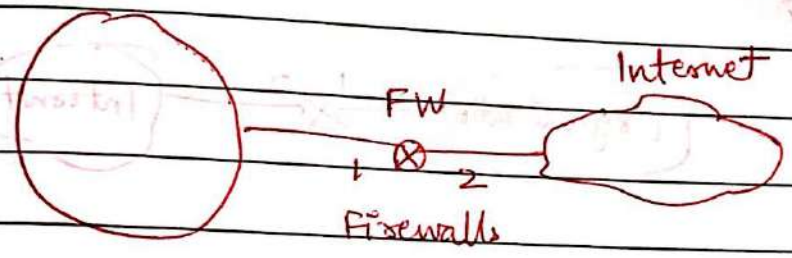
Network Address Translation

- Firewall
- DPI : Deep packet Inspection



Firewalls





Firewall types

- → Layer 3 FW or Packet Filtering FW
- → Layer 4 FW
- → Layer 5 FW or proxy Firewall

Layer 3 Firewall PL, DLL, NL

(i) SIP & DIP → block a host completely

(ii) Protocol (TCP, UDP, ICMP, IGMP)
↳ block a protocol completely

(iii) Block a protocol from a particular host.

Layer 4 Firewall PL, DLL, NL, TL

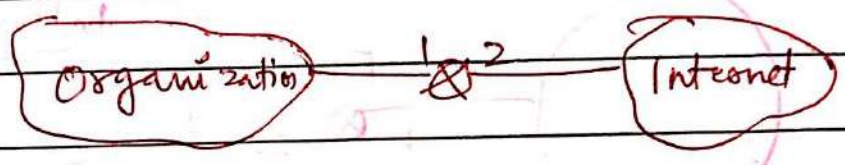
~~Layer 3 capabilities~~ +

(iv) Port number (Block a service completely)

(v) A particular service on a particular host

Layer 5 Firewall or proxy FW

Layer 4 capabilities +



(vi) Application layer data

(Username & Pswd → Authentication)

↳ Block a particular user on a particular host.

Rule Table

SID	DIP	SP	DP	Interface	User
I_{CS}	–	21	–	1	–
–	I_{CS}	–	23	2	–

Code Server

21: FTP

23: Telnet

→ Most powerful firewall.

14. Application Protocols

DNS	HTTP	FTP	SMTP	POP
Domain Name Service	Hypertext transfer Protocol	File transfer Protocol	Simple mail transfer Protocol	Post-office protocol Retrieving e-mail
Name ↓ IP	People see pages through this.	Webpage at home ↓ Server	e-mail direction	

DNS — Domain Name Service

- Easy to remember
- IP's are NOT static

DNS (DN → IP)

(i) Generic domain

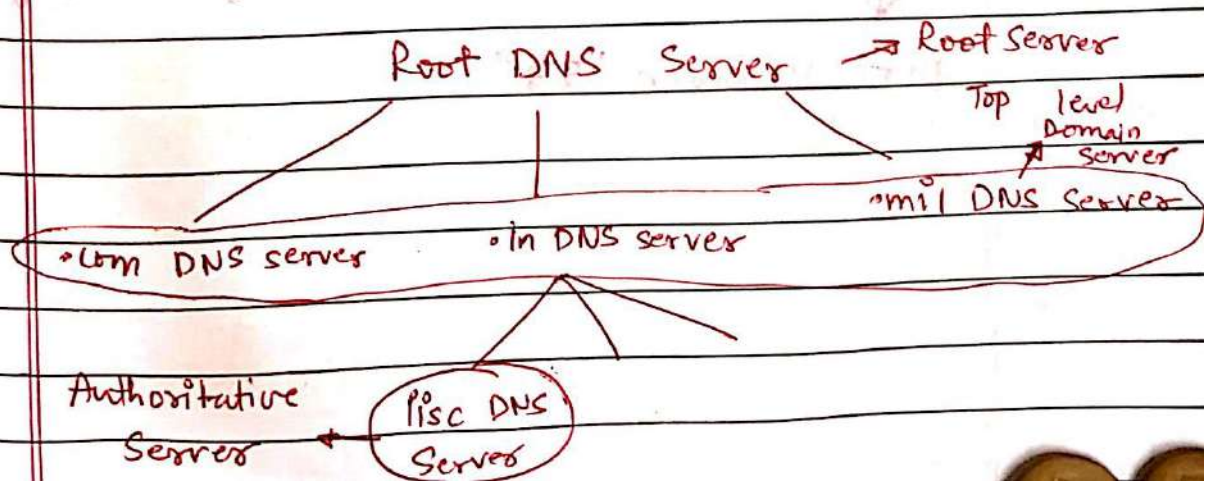
.com, .edu, .mil, .org, .net, ...

(ii) Country domain

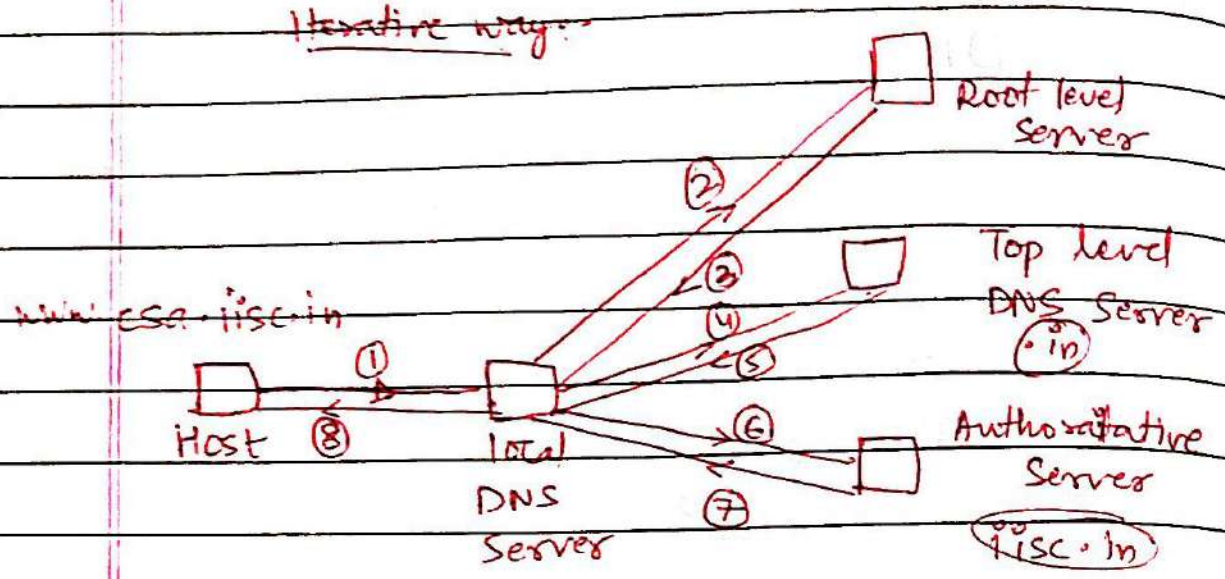
.in, .us, .uk

(iii) Inverse domain

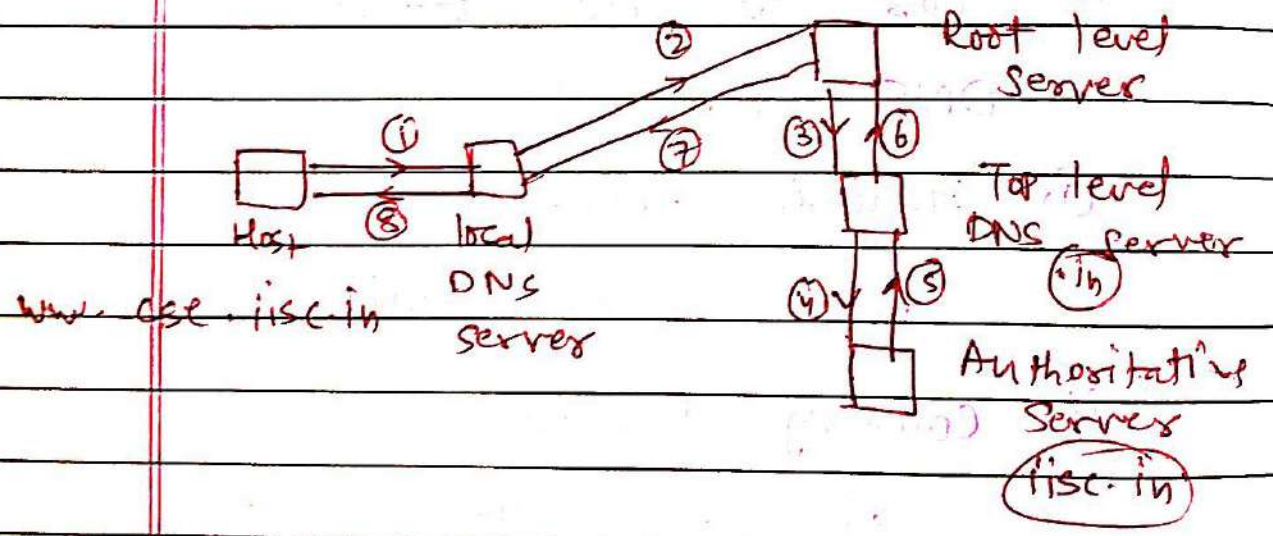
(IP → DN)



Iterative way:-



Recursive way :-



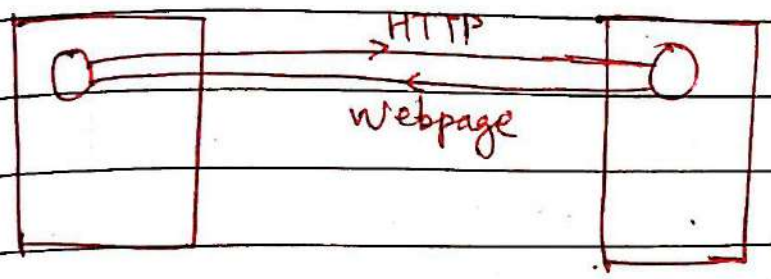
- DNS uses UDP at TL.
  (single request, single reply)

  ↓

  Port number '53'

HTTP (Hypertext Transfer Protocol)

Port number "80"



> Requires reliability.
~~(Not necessisty TCP/IP)~~

> HTTP uses TCP at transport layer.

> HTTP is in band protocol.
(No distinction b/w command & data)

> HTTP is stateless protocol.
(Doesn't remember anything about client)

> HTTP 1.0, non-persistent connection
> HTTP 1.1, persistent connection

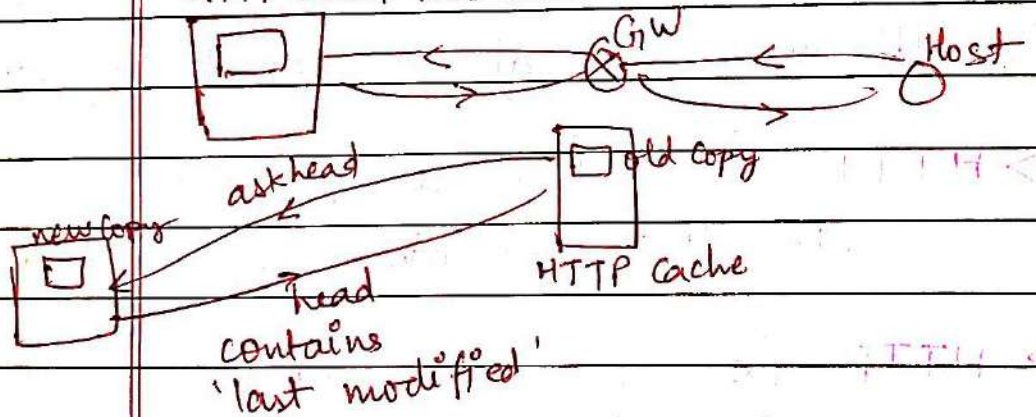
Server need not hold the connection for long time

Server holds the connection for a long time
(BW high)

Methods supported by HTTP :-

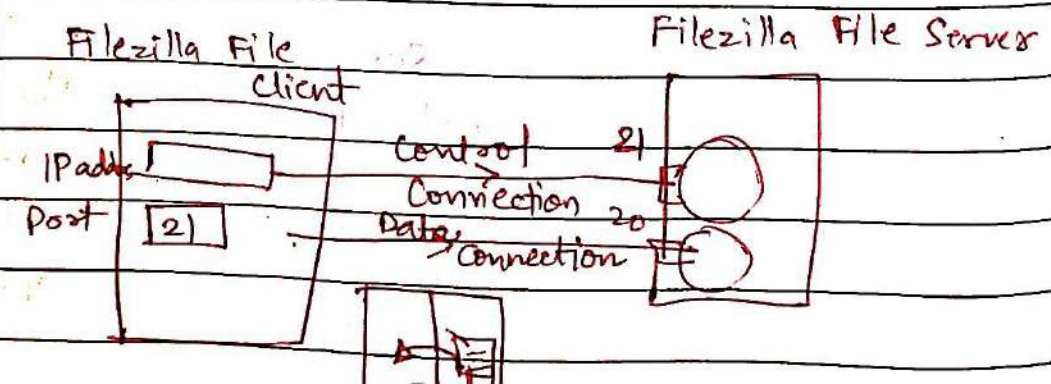
1. Head ✓ → contains metadata of webpage
2. Get → get the webpage
3. Post → fill form
4. Put → upload object
5. Delete → delete object
6. Trace → trace from where data is coming
7. Options → find server is providing 1-6 or not
8. Connect → Https:// → security (authenticated)

HTTP Cache / Web Cache



FTP (File transfer Protocol)

Tectia, Filezilla



- Control connection :- persistent
  (lasts as long as connected)

- Data connection :- non-persistent
  (lasts only till file is transferred)

(Out of band connection)
↳ commands & data in different connections

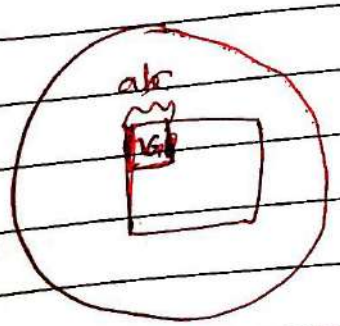
> Requires reliability (uses TCP at TL)
> is statefull protocol (log every activity)

SMTP and POP

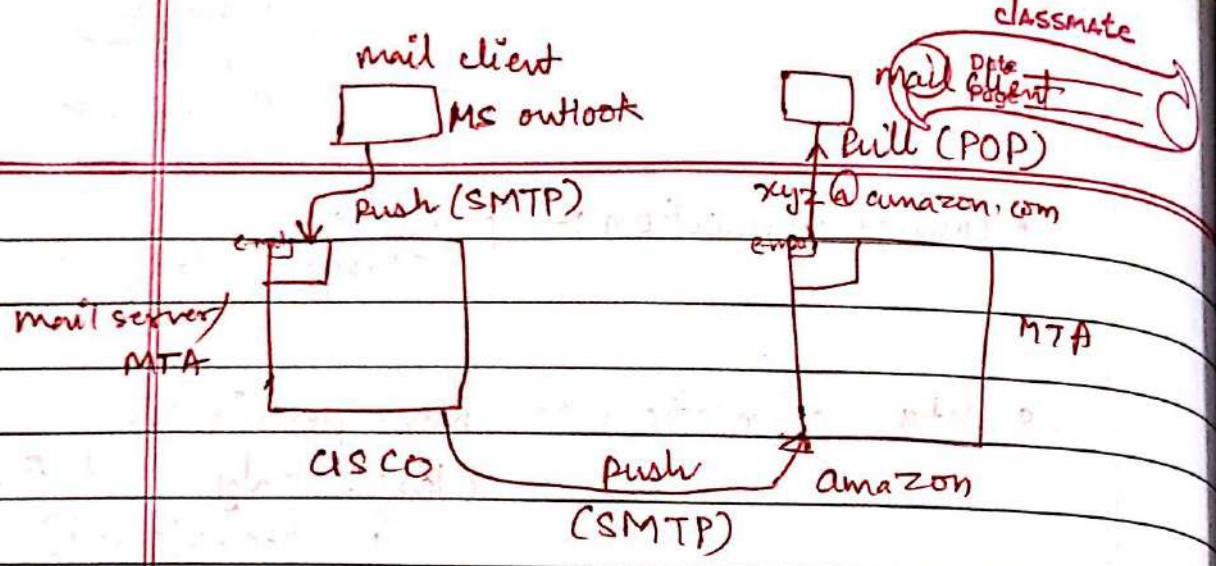
Simple - Mail Transfer Protocol &
Post - office Protocol.

~~FTP~~

gmail :- web based email



abc@ ibm.com

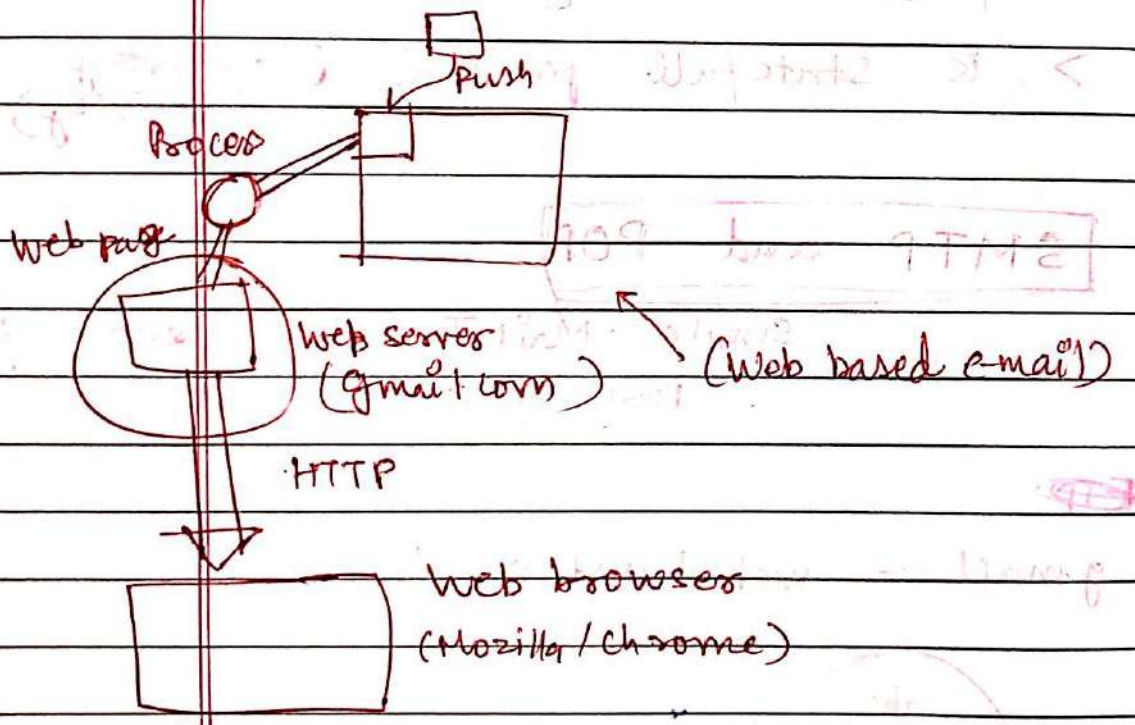


Non-text data → Text data

Text data → Non-text data

→ MIME

Multi-purpose Internet Mail Extension

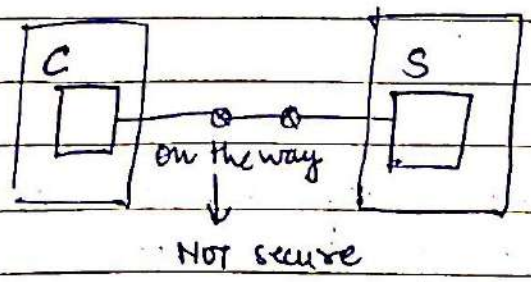


SMTP/POP :- Text based protocol
→ Pushing → Pulling

> Requires reliability
(uses TCP at TL)

> In-band connectivity

19. Network Security

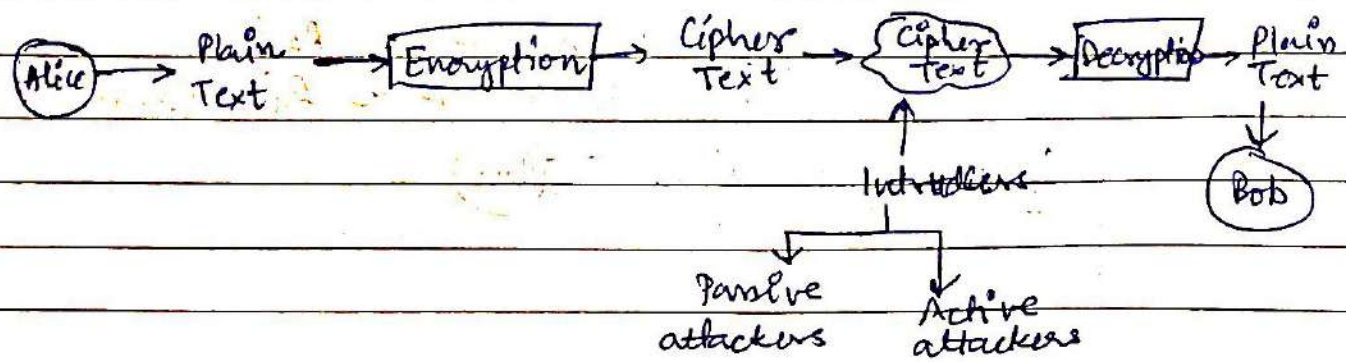


Terminology:-

- Plain Text : Actual Message [Readable Format]
- Cipher Text : Converted Message [Unreadable Format]
- Encryption : process of converting plain text to cipher text
- Decryption : process of converting cipher text to plain text

Cryptography:- Method of storing and transmitting data in a particular form so that only those for whom it is intended can read and process it.

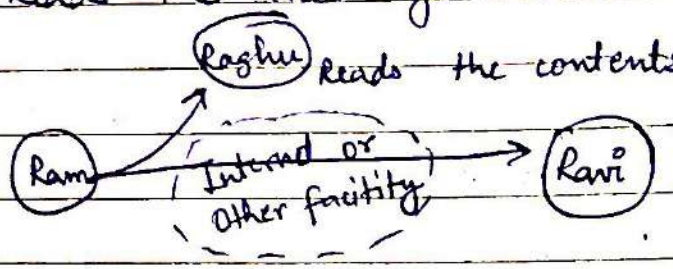
Traditional process:-



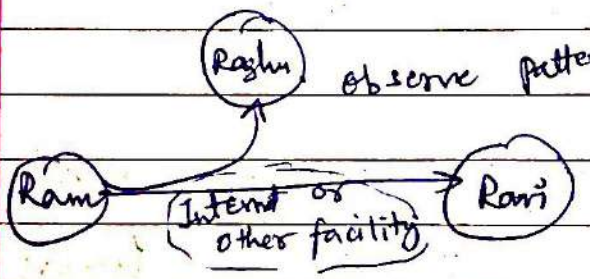
Passive and Active Attackers

Passive Attacker :- They can only see the data, cannot change the contents of the data

*) Reads the message contents

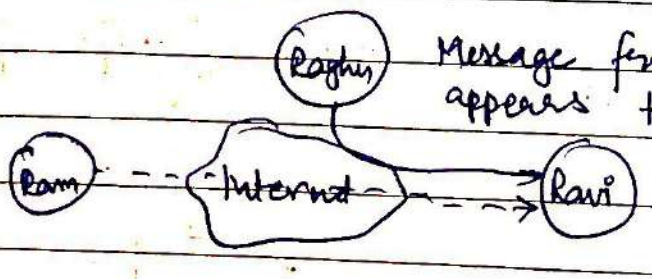


*) Traffic Analysis

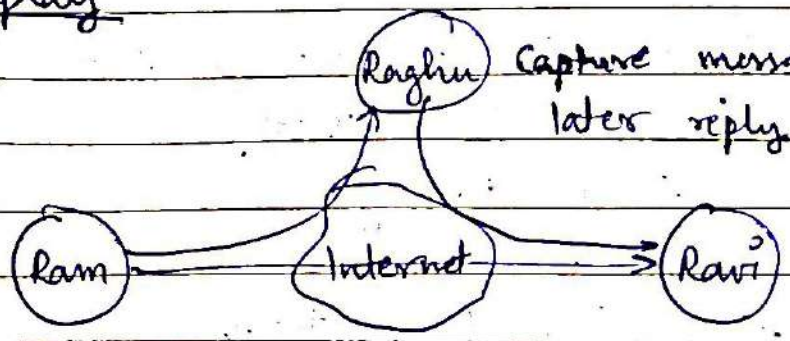


Active Attackers :- Can see the data and can modify the contents of the data

*) Masquerade

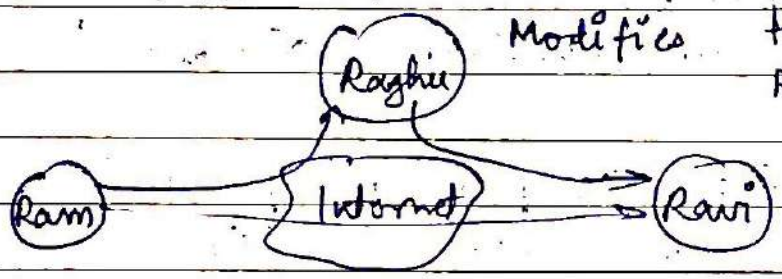


*) Replay



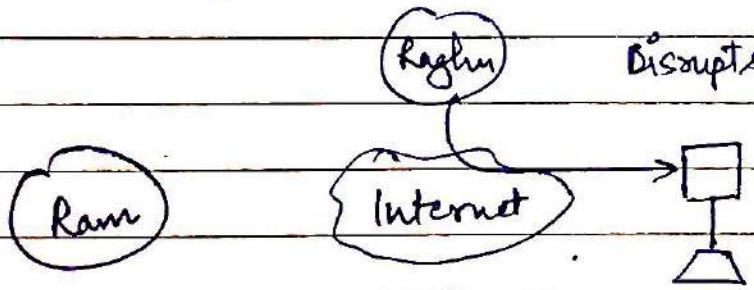
Capture message from Ram to Ravi, later reply message to Ravi

*) Modification of Message



Modifies the message from Ram

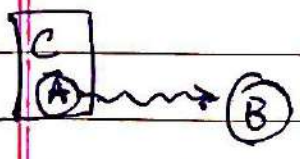
*) Denial of Service



Disrupts service provided by user.

Security Services

⇒ Authentication :- The assurance that the communicating entity is the one that it is claimed to be.



✓ Sure that message is sent by A.

⇒ **Data Integrity** :- The assurance that the data you received is exactly as sent by the authorized entity.

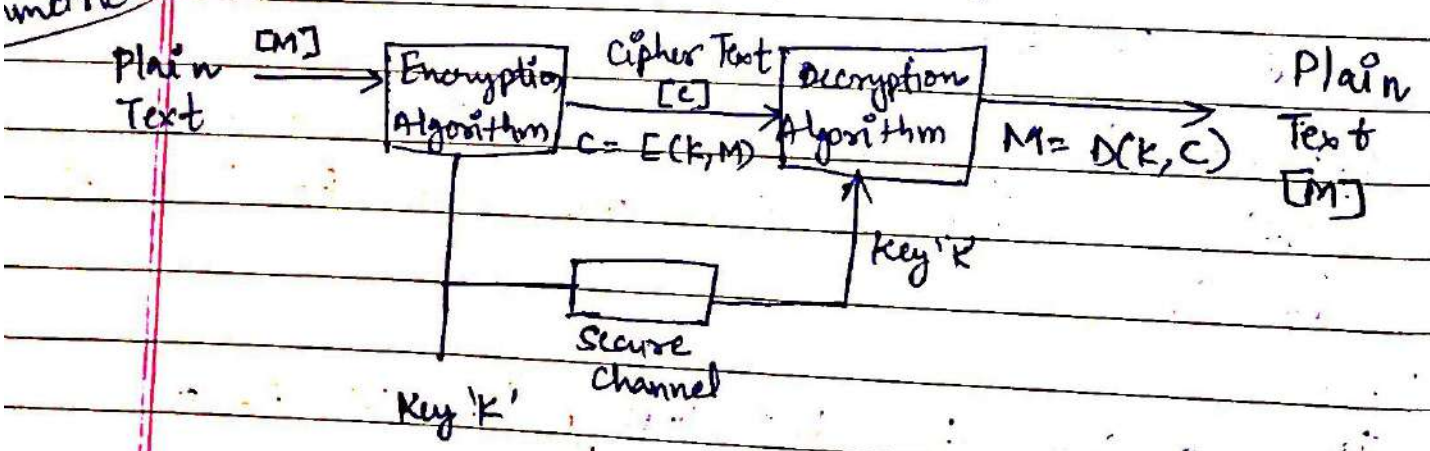
⇒ **Access Control** :- The prevention of unauthorized use of resources.

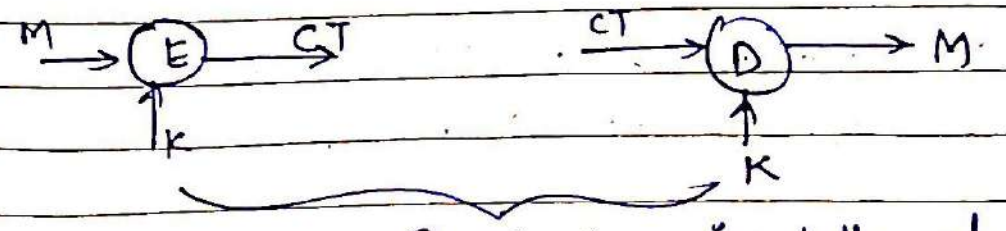
⇒ **Data Confidentiality** :- The protection of data from unauthorized disclosure.

⇒ **Non Repudiation** :- provides protection against denial by one of the entities involved in a communication of having participated in all or part of communication.

Cryptography

- Symmetric — Same key
- Assymetric — Different keys





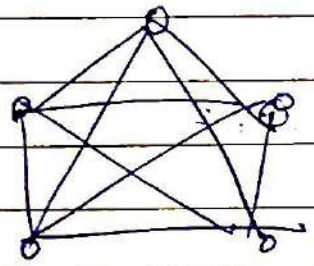
Same key in both phases
Symmetric key cryptography

*) Before starting the communication the key should be exchanged b/w sender and receiver through secure channel.

*) Encryption Algorithms :-

DES [Data Encryption Standard]
AES [Advanced Encryption Standard]

Gate 15



$\boxed{{}^nC_2}$

$= \frac{n(n-1)}{2}$ → Disadvantage

Modullar Arithmetic

*) $a \equiv b \bmod n$:- If 'a' and 'b' leave same remainder when you divide them by 'n'

*) $a \equiv b \bmod n$: If n divides $(a-b)$

eg. $16 \equiv 3 \bmod 13$

*) If $a \equiv b \bmod n$ and $c \equiv d \bmod n$

then $(a+c) \equiv (b+d) \bmod n$

eg. $\left.\begin{array}{l} 17 \equiv 4 \bmod 13 \\ 42 \equiv 3 \bmod 13 \end{array}\right\} \Rightarrow 59 \equiv 7 \bmod 13$

*) If $a \equiv b \bmod n$ and $c \equiv d \bmod n$ then

$(a-c) \equiv (b-d) \bmod n$.

eg. $\left.\begin{array}{l} 42 \equiv 3 \bmod 13 \\ 28 \equiv 2 \bmod 13 \end{array}\right\} 14 \equiv 1 \bmod 13$

*) If $a \equiv b \bmod n$ and $c \equiv d \bmod n$ then,

$(a \times c) = (b \times d) \bmod n$

eg $\left.\begin{array}{l} 8 \equiv 1 \bmod 7 \\ 9 \equiv 2 \bmod 7 \end{array}\right\} \Rightarrow 72 \equiv 2 \bmod 7$

*) If $a \equiv (b \times c) \bmod n$ then

$a \equiv (b \bmod n \times c \bmod n) \bmod n$

eg. $3 \equiv (2 \times 9) \bmod 5$

$\equiv (2 \bmod 5 \times 9 \bmod 5) \bmod 5$

$= (2 \times 4) \bmod 5$

$3 \equiv 8 \bmod 5$ ✓

True

*) If $a \equiv (b+c) \bmod n$ then

$$a \equiv (b \bmod n + c \bmod n) \bmod n$$

eg $11 \equiv (40+23) \bmod 13$

$\equiv (40 \bmod 13 + 23 \bmod 13) \bmod 13$

$\equiv (1+10) \bmod 13$

$\equiv 11 \bmod 13$ True.

<u>Multiplicative Inverse</u>

⇒ For each $a \not\equiv 0 \bmod p$ [p is a prime no.], there is a 'b' such that $a \cdot b \equiv 1 \bmod p$.

'b' is the multiplicative inverse of 'a'

$$a \cdot b = 1 \bmod p$$

$$\Rightarrow b = a^{-1} \bmod p$$

eg. $2 \not\equiv 0 \bmod 7$

$2 \times 4 \equiv 1 \bmod 7$

$4 \equiv 2^{-1} \bmod 7$. ∴ 4 is the multiplicative inverse of $2 \bmod 7$

*) If 'a' and 'n' have no common factors, then 'a' has a mult. inverse mod n.

$\therefore \gcd(a, n) = 1$

eg.1 There is no 'x' such that

$$3 \times x \equiv 1 \bmod 6$$

eg.2 There is a 'x' such that

$$5 \times x \equiv 1 \bmod 9 \qquad \boxed{x = 2}$$

Euler's Theorem.

→ If n is a positive integers and a, n are coprime then $a^{\phi(n)} \equiv 1 \bmod n$

[$\phi(n)$ is the Euler qotient function]

eg $a = 8, \; n = 165$

$$a^{\phi(n)} \equiv 1 \bmod n \quad (\because \gcd(a, n) = 1)$$

$$\phi(165) = \phi(3) \times \phi(5) \times \phi(11)$$

$$= 2 \times 4 \times 10 = 80$$

$$\therefore 8^{80} \equiv 1 \bmod 165$$

(*) $a^{\phi(n) \cdot t} \equiv 1 \bmod n, \; t \in N$

Fermat's Theorem

(Special case of Euler's Theorem)

For any prime no, n, and $a \not\equiv 0 \bmod n$

$$a^{n-1} \equiv 1 \bmod n$$

*) If n is a positive integer and a, n are coprime then

$$a^{\phi(n)+1} \equiv a \bmod n$$

$$\Rightarrow a^{\phi(n)b+1} \equiv a \bmod n$$

eg. $a = 9,\ n = 13 \Rightarrow \phi(13) = 12$

$$\Rightarrow 9^{12} \equiv 1 \bmod 13$$

$$\Rightarrow 9^{13} \equiv 9 \bmod 13$$

$$\Rightarrow 9^{25} \equiv 9 \bmod 13$$

*) If n is a positive integer and a, n are coprime and $b \equiv 1 \bmod n$ then $a^b \equiv a \bmod n$

eg: $a = 5,\ n = 8 = 2^3 \Rightarrow \phi(n) = 8(1-\frac{1}{2}) = 4$

let $b = 17$ $(17 \equiv 1 \bmod 4)$

$$5^{17} \equiv 5 \bmod 8.$$

Euler's Totient function

$$\phi(n) = n-1, \quad n = \text{prime}$$

when (m, n) coprime, $\phi(m*n) = \phi(m) * \phi(n)$

$$\phi(p^n) = p^n - p^{n-1}$$

RSA Algorithm [Rivest, Shamir, Adleman]

→ Using an encryption key (e, n)

i) Represent the message as an integer between 0 and $n-1$

ii) Encrypt the message by raising it to the eth power modulo n, the result is cipher text message c

$$c = p^e \bmod n$$

iii) To decrypt ciphertext message c, raise it to another power d mod n

$$c^d \bmod n = (p^e \bmod n)^d \bmod n$$

$$= p^{ed} \bmod n \quad \text{(has to be)} = p$$

$$\Rightarrow ed \equiv 1 \bmod \phi(n)$$

* The Decryption key $(P_r)(d, n)$
* The Encryption key $(P_r)(e, n)$

$$ed \equiv 1 \bmod \phi(n)$$

Algo for finding n and $\phi(n)$ both too large :-

Algo:-

i) Choose two distinct prime numbers p, q

Let $p = 61$, $q = 53$

ii) Compute $n = p \times q = 3233$.

iii) $\phi(n) = \phi(p \times q) = (p-1) \times (q-1)$

$\phi(61 \times 53) = 60 \times 52 = 3120$

iv) Choose an integer 'e', $1 < e < \phi(n)$, which is coprime to $\phi(n)$

Let $e = 17$

* (e, n) is public key $(17, 3233)$

v) Determine 'd' as $ed = 1 \bmod \phi(n)$

$d = e^{-1} \bmod \phi(n)$

$17 \times d = 1 \bmod 3120$

$d = 2753$

* (d, n) is private key $(2753, 3233)$

PAGE NO.:

Primitive root:

⇒ The number b in $a \equiv b \bmod n$ is called the residue of $a \bmod n$

eg. $7 \equiv 85 \bmod 13$

→ 85 is residue of 7 mod 13

⇒ The residue classes of $f(x) \bmod n$ are all possible values of residue $f(x) \bmod n$

eg. residue classes of $x^2 \bmod 6$ are $\{0, 1, 3, 4\}$

$0^2 \equiv 0 \bmod 6$ $\quad$ $3^2 \equiv 3 \bmod 6$

$1^2 \equiv 1 \bmod 6$ $\quad$ $4^2 \equiv 4 \bmod 6$

$2^2 \equiv 4 \bmod 6$ $\quad$ $5^2 \equiv 1 \bmod 6$

⇒ Let 'p' be a prime number then 'b' is a primitive root for 'p' if powers of b

$1, b, b^2, b^3, b^4, \ldots$

includes all the residue classes of mod p.

*) If p is a prime number, there is a primitive root of p.

⇒ The powers of b form a repeating cycle, and that cycle can't be longer than p-1 [$a^{p-1} \equiv 1 \bmod n$: Fermat's theorem] so b is a primitive root if the cycle as long as p-1

eg. If p = 7, the 3 is the primitive root of p

$3^0 \equiv 1 \bmod 7$ $\quad$ $3^3 \equiv 6 \bmod 7$

$3^1 \equiv 3 \bmod 7$ $\quad$ $3^4 \equiv 4 \bmod 7$

$3^2 \equiv 2 \bmod 7$ $\quad$ $3^5 \equiv 5 \bmod 7$

⋮

repeats

⇒ Given an integer a and a positive integer 'n' with $\gcd(a, n) = 1$, the multiplicative order of a mod n is the smallest positive integer k with

$$a^k \equiv 1 \bmod n$$

eg. (3, 8) $\quad 3^2 \equiv 1 \bmod 8$

$2 \Rightarrow$ order of 3 mod 8

⇒ m is a primitive root Modulo n iff the multiplicative order of m is $\phi(n)$

$$m^{\phi(n)} \equiv 1 \bmod n$$

→ Apart from 1, 2, 4 the only numbers with primitive roots are of the shape p^k, $2p^k$ where 'p' is an odd prime numbers

eg:- 3, 5, 7, 6, 10, 14, 2×7^2 etc.

Discrete Logarithm

The problem of finding x such that $a^x \equiv b \bmod p$ [p is a prime number and a, b are non-zero integers] is called discrete logarithm problem. It is NP-hard problem.

* By assuming $0 \le x < n$, where $n = \text{Ord}_p(a)$ [$a^n \equiv 1 \bmod p \Rightarrow n = \text{ord}_p(a)$] we need to search for the value of x in this range.

eg: $p = 11$ $a = 2$ $b = 9$

$2^x \equiv 9 \bmod 11$

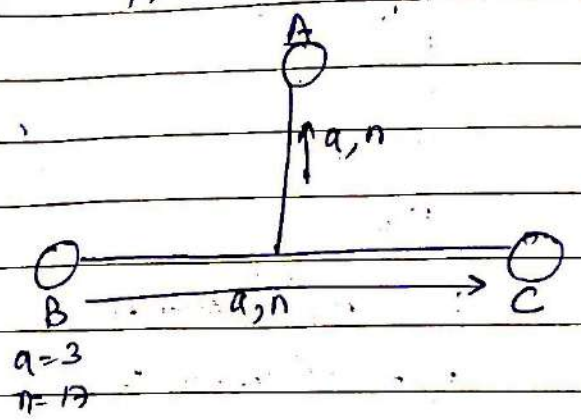
$2^6 \equiv 9 \bmod 11$

$x = 6$

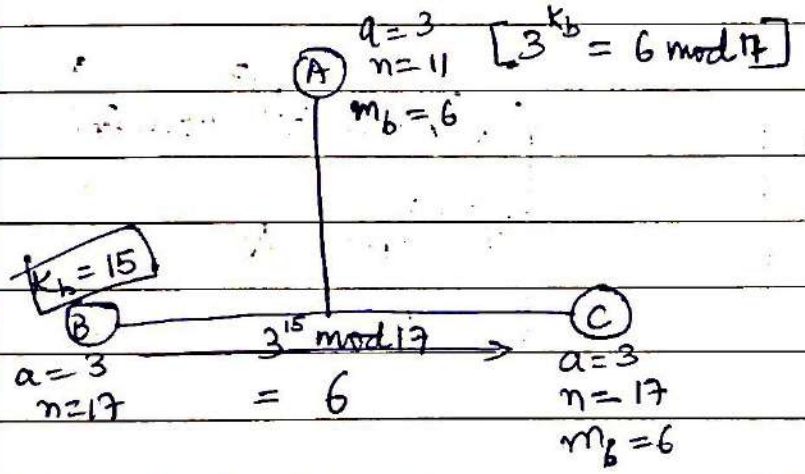
One-way function :- A function $f(x)$ is called one way function if $f(x)$ is easy to compute but, given y, it is computationally infeasible to find x, with $y = f(x)$

Diffie Hellman Algorithm

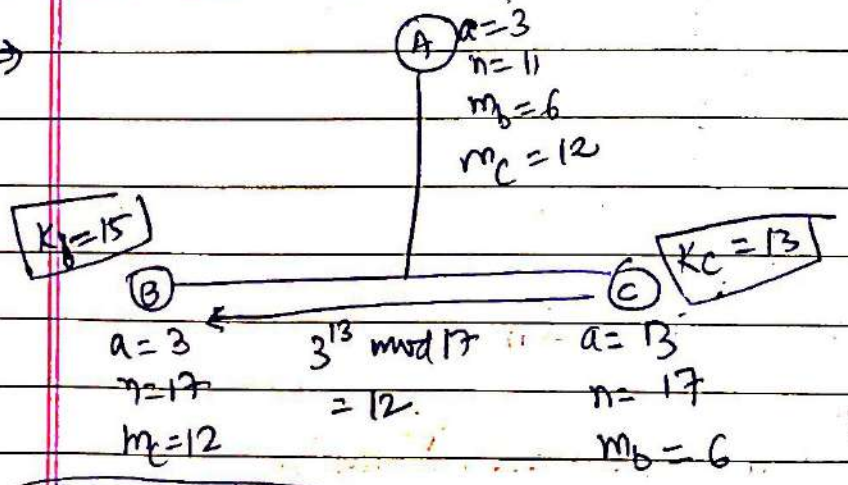
(Diffie Hellman Key Exchange Algorithm)



*) Agree on a generator 'a' and prime modulus 'n'



*) B selects a private random number (K_b) Calculates: $a^{K_b} \bmod n$ and sends it to 'c'



*) C selects a private random number [K_c] Calculates $a^{K_c} \bmod n$ and sends it to 'b'

$3^{K_c} = 12 \bmod 17$

$a = 3$
$n = 17$
(A) $m_c = 12$
$m_b = 6$

→

$\boxed{K_b = 15}$ (B)

$a = 3$
$n = 17$
$m_c = 12$
$K = m_c^{K_b} \bmod n$
$K = 12^{15} \bmod 17$
$= 10$

$\boxed{K_c = 13}$ (C)

$a = 3$
$n = 17$
$m_b = 6$
$K = m_b^{K_c} \bmod n$
$K = 6^{13} \bmod 17$
$= 10$

$$\left[\begin{array}{l} (3^{13} \bmod 17)^{15} \bmod 17 \\ 3^{13 \times 15} \bmod 17 \\ = 10 \end{array}\right] \qquad \left[\begin{array}{l} (3^{15} \bmod 17)^{13} \bmod 17 \\ 3^{15 \times 13} \bmod 17 \\ = 10 \end{array}\right]$$

Summary

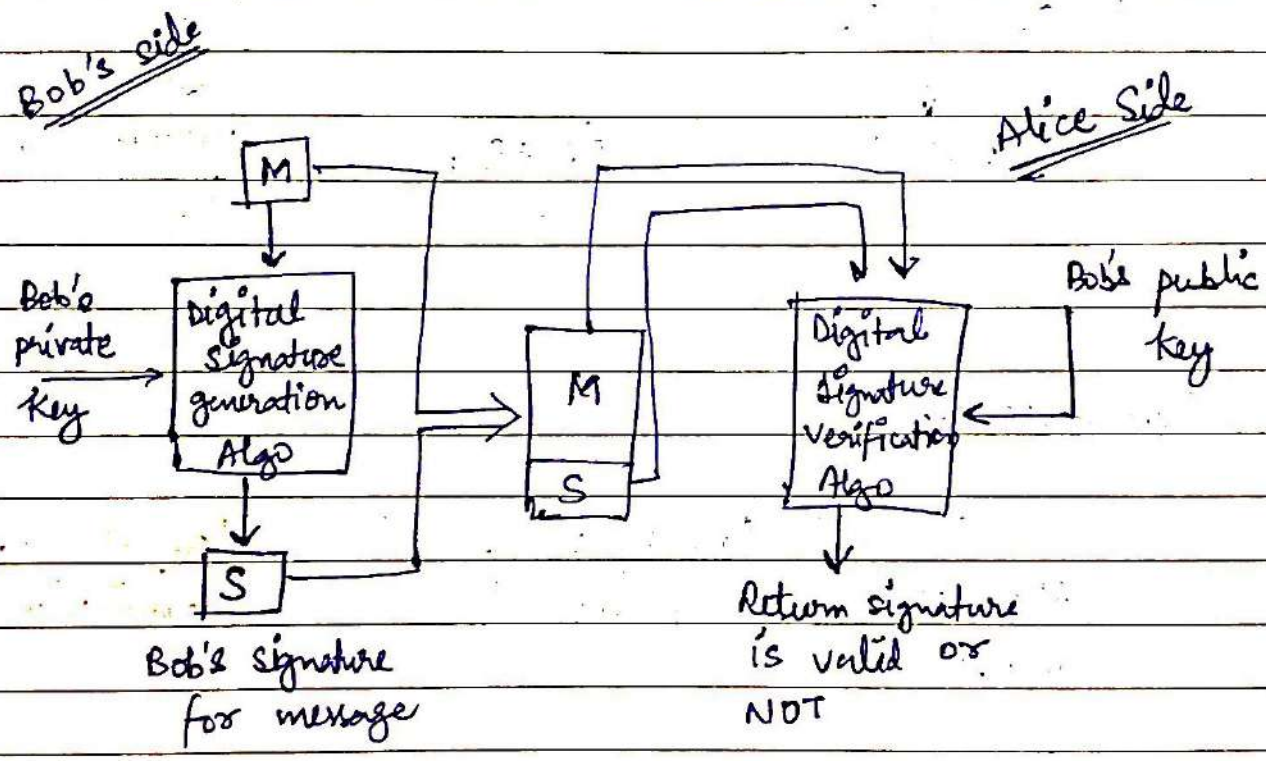
*) Both 'B' and 'C' computes key values as same

	(B)	(C)
Random number	p	q
Computes	$a^p \bmod n$	$a^q \bmod n$
Interchange	$a^q \bmod n$	$a^p \bmod n$
Compute Key	$(a^q \bmod n)^p \bmod n$	$(a^p \bmod n)^q \bmod n$
Key	$a^{qp} \bmod n$	$a^{qp} \bmod n$

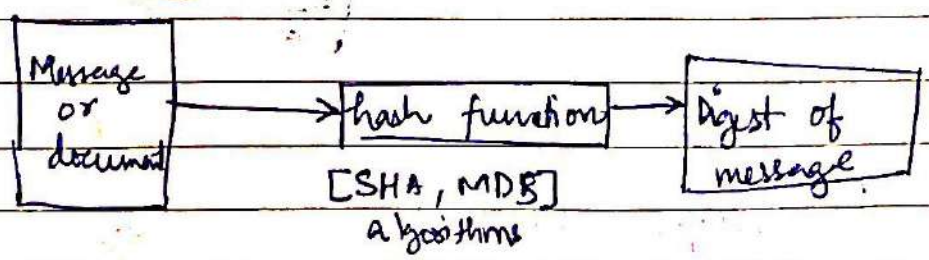
Digital Signatures

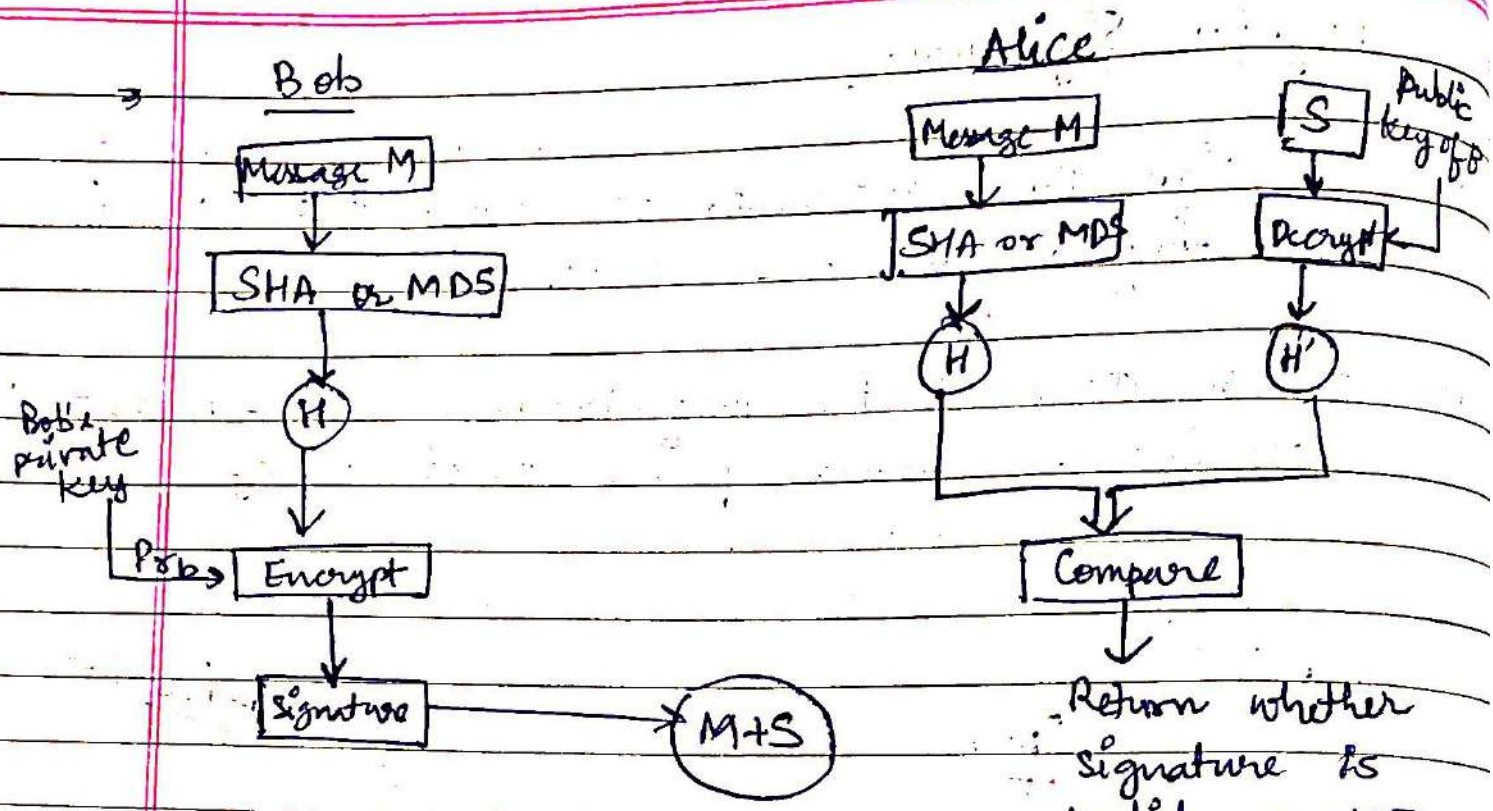
⇒ We sign the documents to show that it is originating from us.

⇒ The signature is the proof to the receiver that the document is coming from correct entity.



⇒ Actually we use the long messages or documents to send to receiver. It is difficult to sign electronically those documents with public key cryptography.

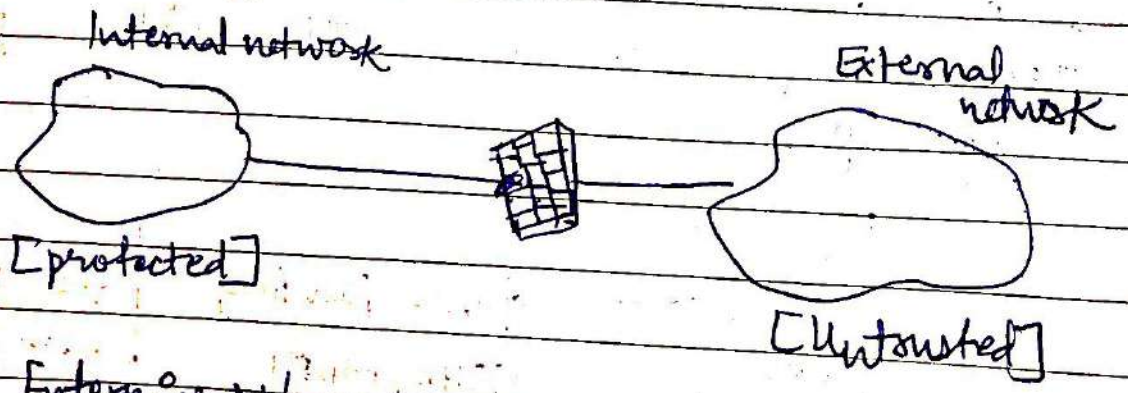




Firewalls :-

⇒ A firewall forms a barrier through which the traffic going in each direction must pass.

⇒ A firewall security policy dictates which traffic is authorized to pass in each direction.



eg. Enterprise N/w

eg. Internet

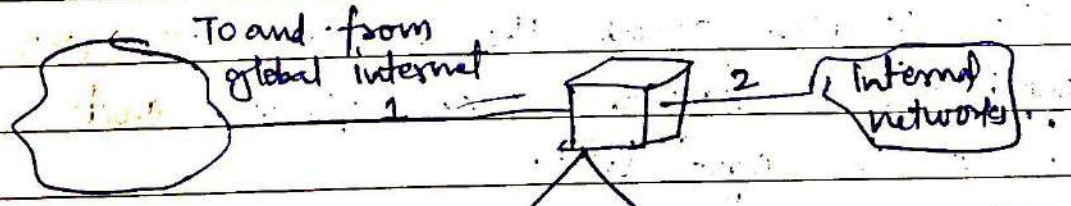
- A Firewall may be designed to operate as filter at the level of IP packets, or may operate at higher protocol layers.

- A Firewall defines a single choke ~~packet~~ point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network.

- **Types of Firewalls** :-

- Packet Filtering Firewall

  *) Packet filtering firewall applies a set of rules to each incoming and outgoing IP packets and then forwards or discards the packet based on information present in TCP or IP headers.

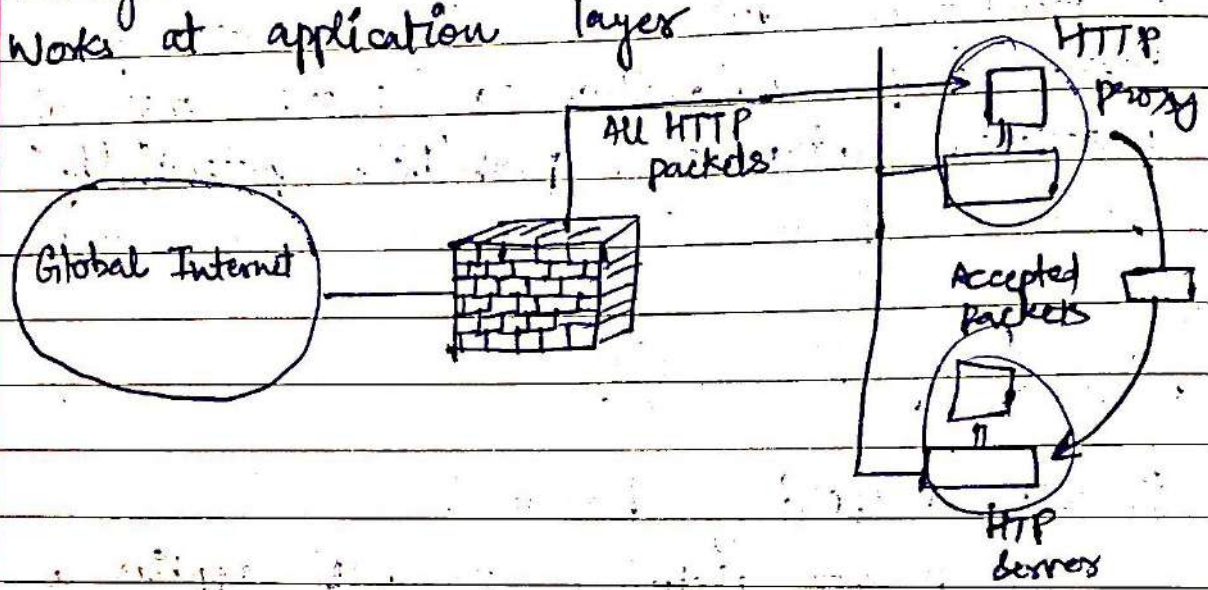


	Interface	Source IP	Source port	Destination IP	Destination port
Block a network	1	131.34.00	*	*	*
Block a service	*	*	*	*	23
Block a host inside	1	*	*	194.78.20.8	*
Block HTTP packets inside	2	*	80	*	*

DATE: / /
PAGE NO.:

Proxy firewall

- Filter the messages based on contents of the message
- Works at application layer



- When user's client process sends a message, the proxy firewall runs a server process to receive the request.
- The server opens the packet at the application layer and finds out if the request is legitimate.

20. IPV6 And Wifi

Need for IPv6 (also called IPng)

- → IPv4 address depletion
- → Real time audio/video transmissions
- → Encryption
- → Authentication
- → Fast processing
- → Additional functionalities

IPV6 header

IPv6 0110
IPv4 0100

VER (4)	Priority/ Traffic class (8)	Flow label (20)	(4 bytes)
Payload length (16)	Next header (8)	Hop limit (8)	(4 bytes)
(128 bits) Source address (IPV6 address)			(16 bytes)
(128 bits). Destination address			(16 bytes)
EH1..			
EH2..			
Data			

40 bytes

Base header length is fixed

IPV4 (Total length)

BH	PL
Header length	Payload length

Traffic class/ Priority

	Version	Priority class	Flow label
RFC	4	8	20
Forouzan	4	4	24

✓ Follow this

Traffic

- low priority — Congestion Controlled — eg:- TCP
- high priority — Uncontrolled — eg. Audio/video

Priority | Research

4

(0, 2 .. 7) (8 9 . . 15)
Congestion Controlled — Uncontrolled

Priority	Meaning
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

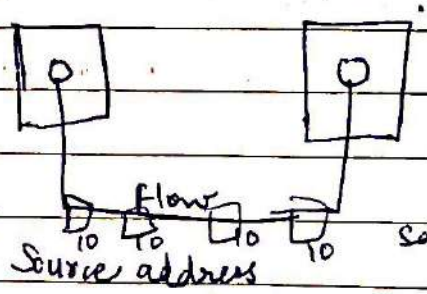
News ← 1
Email ← 2
FTP/HTTP ← 4
Remote login Putty/ssh ← 6
← 7
SNMP, Routing

Simple network Managing Protocol

8, 9, ... 15
A/V → High Definition
A/V → low definition

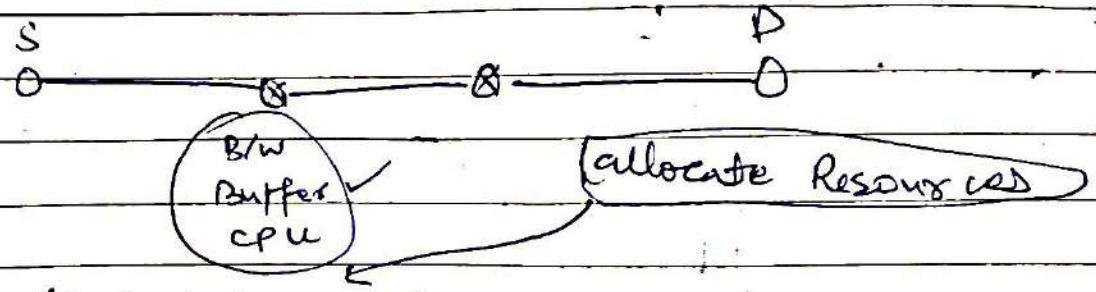
(i) IPV6 → interface
(ii) Source change ✓ router ✓

Flow label



same flow label

i) Source address
ii) Destination address
iii) Priority
iv) Extension header



(i) Control packets

→ Resource Reservation Protocol

→ Real time Transport Protocol

(ii) Extension headers

Flow table

Source	Flow label	Information
A	1	Buffer, B/W
⋮	⋮	⋮

Compare

- There can be more than one flow b/w source & destination (with different flow labels)
- Flow label - all zeros - No flow associated with packet.

Payload length

IPV 4

Payload length = Total length − Header length

↓

NOT fixed

IPV 6

Payload length

(Header length fixed)

↓

EH + Upper layer packet

Next Header Field

Next header codes

Code	Next header
0	hop to hop option
2	Icmp
6	TCP
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted security payload
51	Authentication
60	Destination option

59 No next header (NULL)

eg.

IPv6 header

Next header = 6 (TCP)	TCP header	TCP data

NH = 17 (UDP)	UDP header	UDP data

NH = 43 (SRE)	Source Routing Extension Header NH = 6 (TCP)	TCP header	TCP data

NH = 43 (SR)	SREH NH = 44 (Frag)	FEH NH = 6 (TCP)	TCP header	TCP data

Payload

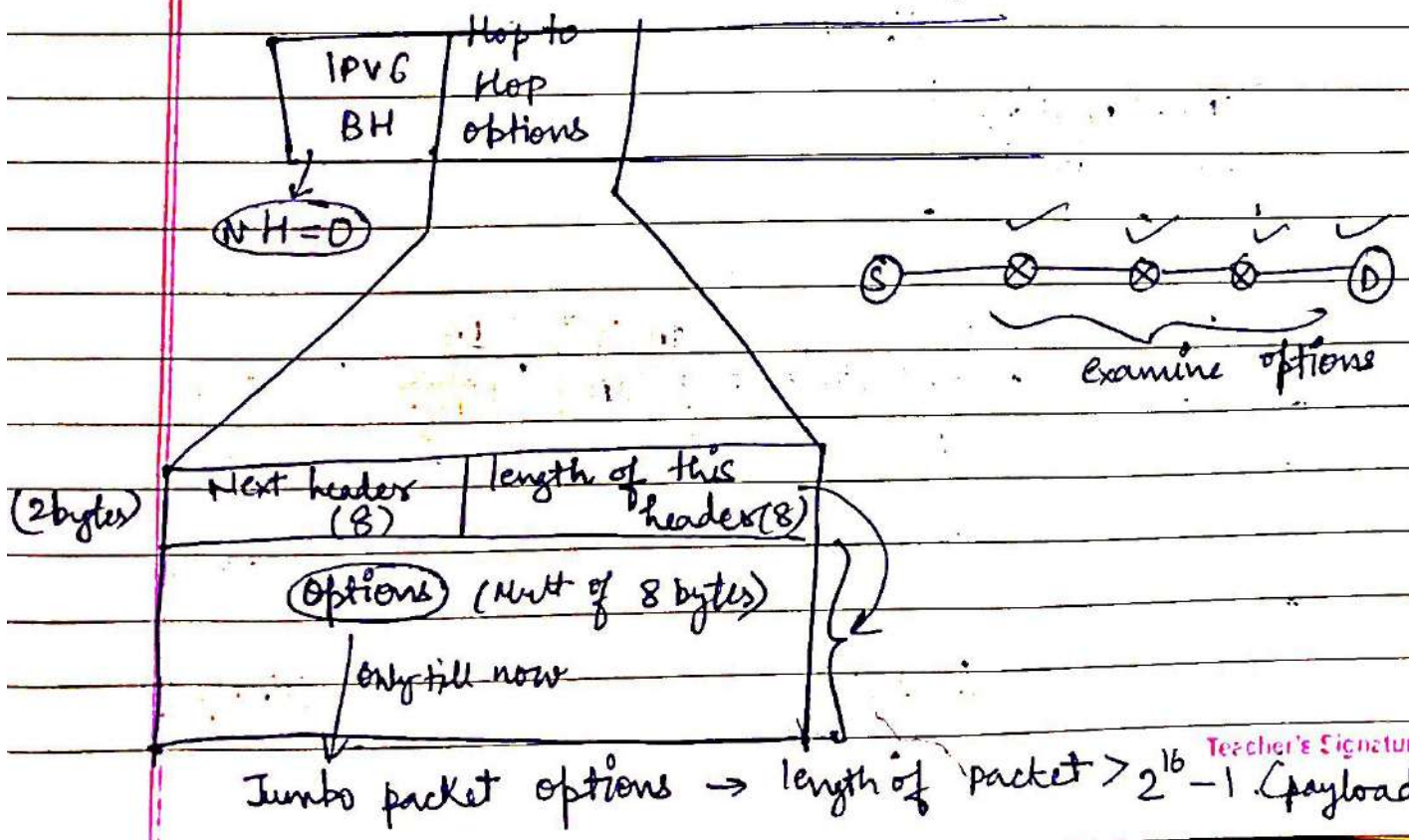
Ordering of Next header Field:-

→ Hop to Hop options header
→ Destination options header
→ Routing header
→ Fragment header
→ Authentication header
→ Encapsulating security payload header
→ Destination header
→ Upper layer header

- These sequence is just convention not a rule
- All the extension headers can be present only once
- Hop by Hop options should always be after base header → rule

Hop to Hop options Header

always following base header (if present)



Routing Extension Header

Source Routing Extension

BH	RH

SA = IPV6(A)
DA = IPV6(R1)

A[1] = IPV6(R2)
A[2] = IPV6(R3)
A[3] = IPV6(B)
Segments Left = 3
(# intermediate Routers)

R1	R2	R3	B
SA = A	SA = A	SA = A	SA = A
DA = R1	DA = R2	DA = R3	DA = B
SL = 3	SL = 2	SL = 1	SL = 0
A[1] = R2	A[1] = R1	A[1] = R1	A[1] = R1
A[2] = R3	A[2] = R3	A[2] = R2	A[2] = R2
A[3] = B	A[3] = B	A[3] = B	A[3] = R3
↓ check SL ≠ 0	↓ check SL ≠ 0	↓ check SL ≠ 0	↓ check SL = 0
↓ dec SL = 2	↓ dec SL = 1	↓ dec SL = 0	↓ end
n − i	n − i = 2	n − i = 3	
⇓ ⇓ 3 − 2 = 1 → index of address visited next	↓ Swap DA = R3, A[2] = R2	↓ Swap DA = B, A[3] = R3	
A[1] → next			
Swap DA & A[1]			
DA = R2			
A[1] = R1			

IPV6 BH	Routing Header

8 bytes

1B	1B	1B	1B	
Next Header	Header Extension length	Routing Type (=0 (Now))	Segments left	(4 bytes)
Reserved = 0				(4 bytes)
Add[1]				(16 bytes)
Add[2]				(16 bytes)
⋮				

Header Extension length

Header length = $\frac{\cancel{8}+16+16}{8}$ = 4.

Exclude

= (No. of add.) × 2

eg If Header Ext. length = 8

Total Header length = 8×8 + 8 bytes
= 72 bytes

Address = $\frac{8}{2}$ = 4

Fragmentation Extension Header

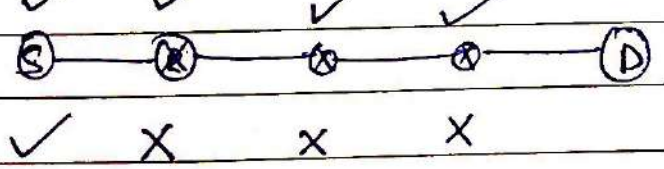
Fragmentation Header:-

Next Header (8 bits)	Reserved (8 bits)	Fragment Offset (13 bits)	Res (2 bits)	m (1 bit)
Identification (32 bits)				

Fragmentation

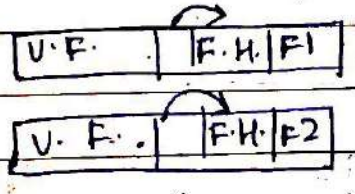
IPV4 ✓ ✓ ✓ ✓

IPV6 ✓ X X X



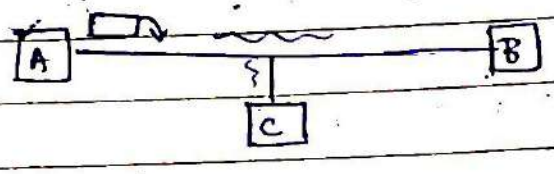
IPV6 BH	H-H	D	R	F	Payload

Unfragmentable | Fragmentable



DATE: / /
PAGE NO.:

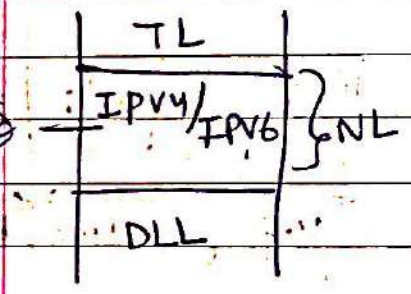
Need for security at Network layer



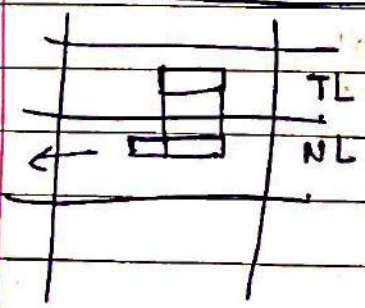
(i) confidentiality
(ii) Message authentication / Message integrity
(iii) Source authentication
(iv) Replay attack protection

IETF : Internet Engineers Task force
↓
IPSec : IP security → Helps both IPV4 & IPV6
↓
Collection of protocols
↓
- Authentication Header
- ESP (Encrypted security payload)

IPSec



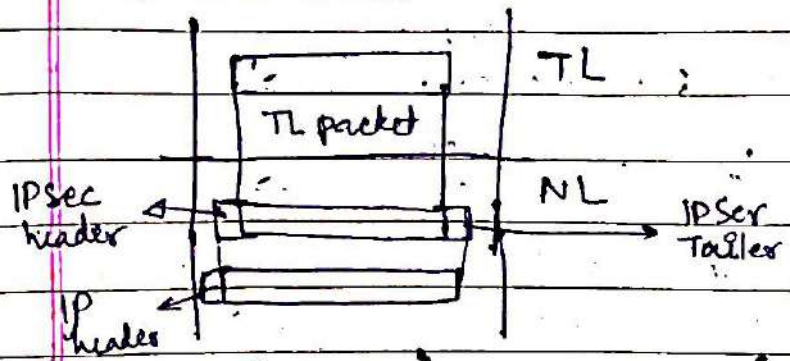
Modes of IP security



→ Transport Mode
→ Tunnel Mode

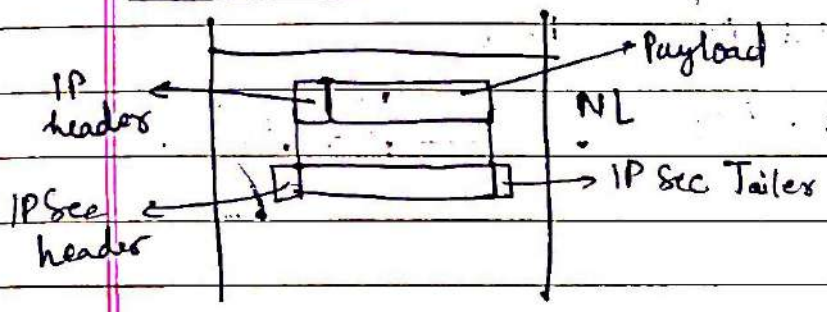
Teacher's Signature

Transport Mode :-



IPSec
- Authentication Header Protocol
- Encrypted Security Payload

Tunnel Mode :-



Authentication Header

⇒ Message Integrity & Source Authentication

IP header	Authentication Header	Transport layer payload	Padding

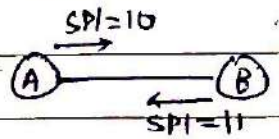
IPV4 header → Protocol field → 51 → Authentication header

(8)	(8)	(16)
Next header	Payload length	Reserved
Security parameter index (32)		
Sequence number (32)		
Authentication data (digest) (variable length)		

(8 bytes)

→ depends upon Algorithm used

$$\text{Payload length} = \left(\frac{HL}{4} - 2\right)$$

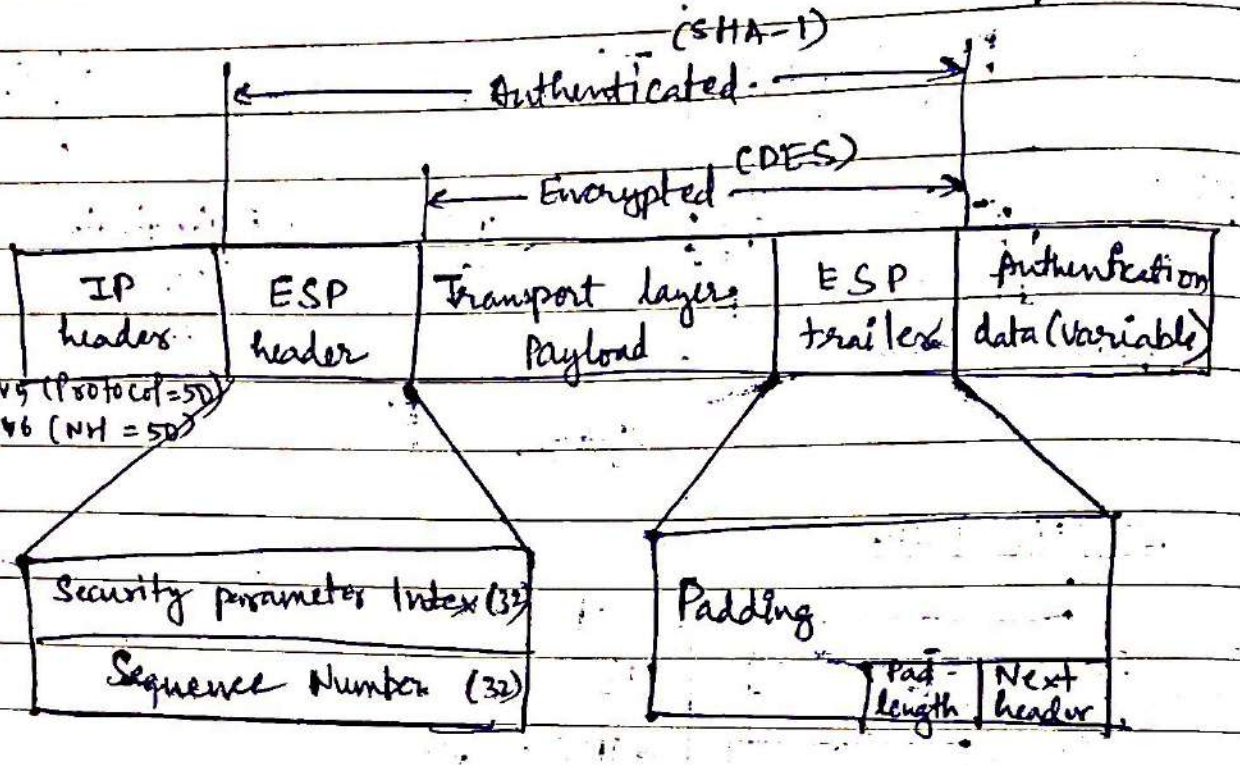


SPI	Hashing Algo	Key
10	SHA-1	K1
11	SHA-2	K2

→ Security Association

Sequence Number :- protects Replay attack

ESP (Encapsulation Security Payload)

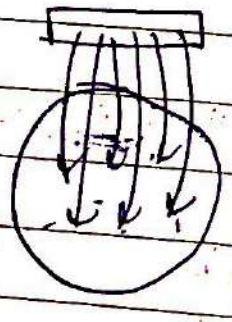


IPV6 Addresses

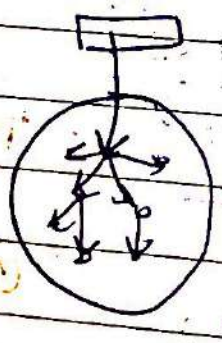
128 bits - 2^{128} addresses

- Unicast → To a interface.
- Multicast → Group of interfaces
- Anycast → Group of interface

Multicast



Any cast



Colon Hexa Representation

eg. ABCD : EF01 : 2345 : 6789 : ABCD : EF01 : 2345 : 6789

16 bits
(2 bytes)

$8 \times 2B = 16B = 128$ bits

eg. ABCD : 0000 : 0000 : 0000 : 0000 : 0000 : 0000 : 1234

≅ ABCD : 0 : 0 : 0 : 0 : 0 : 0 : 1234

≅ ABCD : : 1234

eg. ABCD : 0000 : 0000 : 0000 : 1234 : 0000 : 0000 : 1234

≅ ABCD : : 1234 : 0 : 0 : 1234 ✓

≅ ABCD : 0 : 0 : 0 : 1234 : : 1234 ✓

≇ ABCD : : 1234 : : 1234 ✗

Types of IPV6

Prefix	Type	Prefix	Type
00000000	reserved	011	UA
00000001	~~Unic~~ UA	001	Geographical based Unicast
0000001	UA	101	UA
0000010	UA	110	UA
0000011	UA	1110	UA
00001	UA	11110	UA
0001	Reserved	111110	UA
~~001~~ 100	Reserved	1111110	UA
010	Provider based Unicast	11111110	UA

UA : Unassigned

Teacher's Signature

Prefix	Type
1111 1110 10	link local addresses
1111 111 0 11	Site local addresses
1111 1111	multicast addresses

Provider Based Unicast Address

3	5 bits	n bits	56-n bits	64 bits
010	Registry ID	Provider ID	Subscriber ID	Intra Subscriber

Registry ID	Registry
10000	Multi regional (IANA)
01 000	RIPE NCC
11 000	INTERNIC
00100	APNIC (Asia-Pacific)

IANA
↓
Regional Registry (Registry ID)
↓
ISP (Provider ID)
↓
Organization (Subscriber + Intra Subscriber)

Geography based unicast address

3 bits	45 bits	16 bits	64 bits
001	global routing prefix	Subnet ID	Interface ID

Multicasting

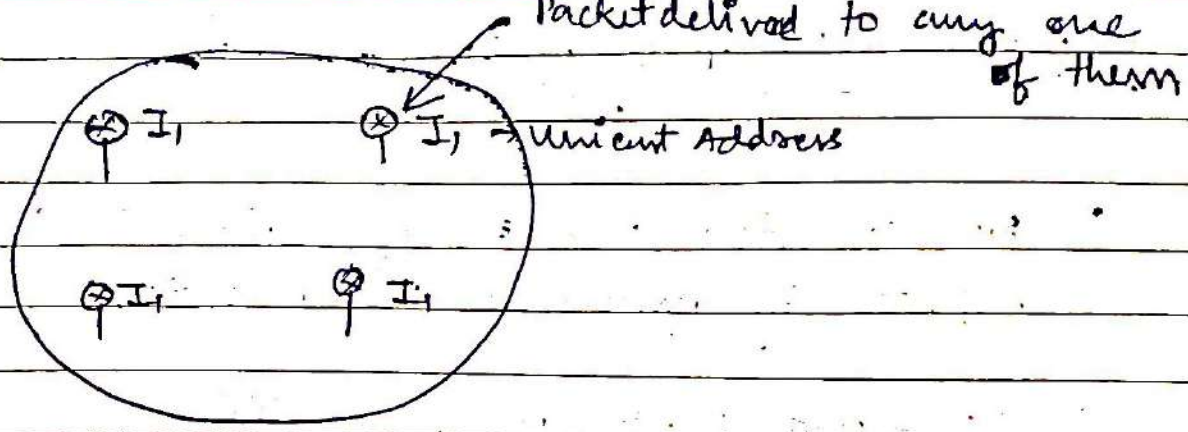
(8 bits)	(4 bits)	(4 bits)	(112 bits)
111 11111	Flag	Scope	Group ID

Flag:
- 0000 - Permanent
- 0001 - Transient

Scope:
- 0000 - Reserved
- 0001 - Node local
- 0010 - Link local
- 0101 - Site local
- 1000 - organizational

Teacher's Signature

Anycast



- Use unicast addresses only.

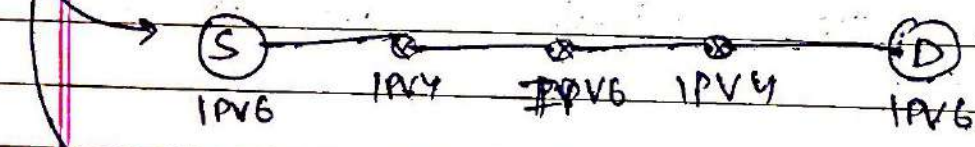
Some Special Addresses

	8 bits		
Unspecified	00000000	120 0's	

	8 bits		
loopback (Self connectivity)	00000000	119 0's	1

	8 bits		32 bits
IPV4 compatible	00000000	88 0's	IPV4 address

	8 bits			32 bits
IPV4 mapped	00000000	72 0's	16 1's	IPV4 address



Local Addresses

	10 bits	70 bits	48 bits
Link Local	1111 1110 10	All 0's	node address

	10	38 bits	32 bits	48 bits
Site local	1111 1110 11	All 0's	Subnet	node address

Comparision b/w IPV4 options and IPV6 extension header

1. No operation and end of options in IPV4 is replaced by PAD1 and Pad N options in IPV6

2. No record route option in IPV6.

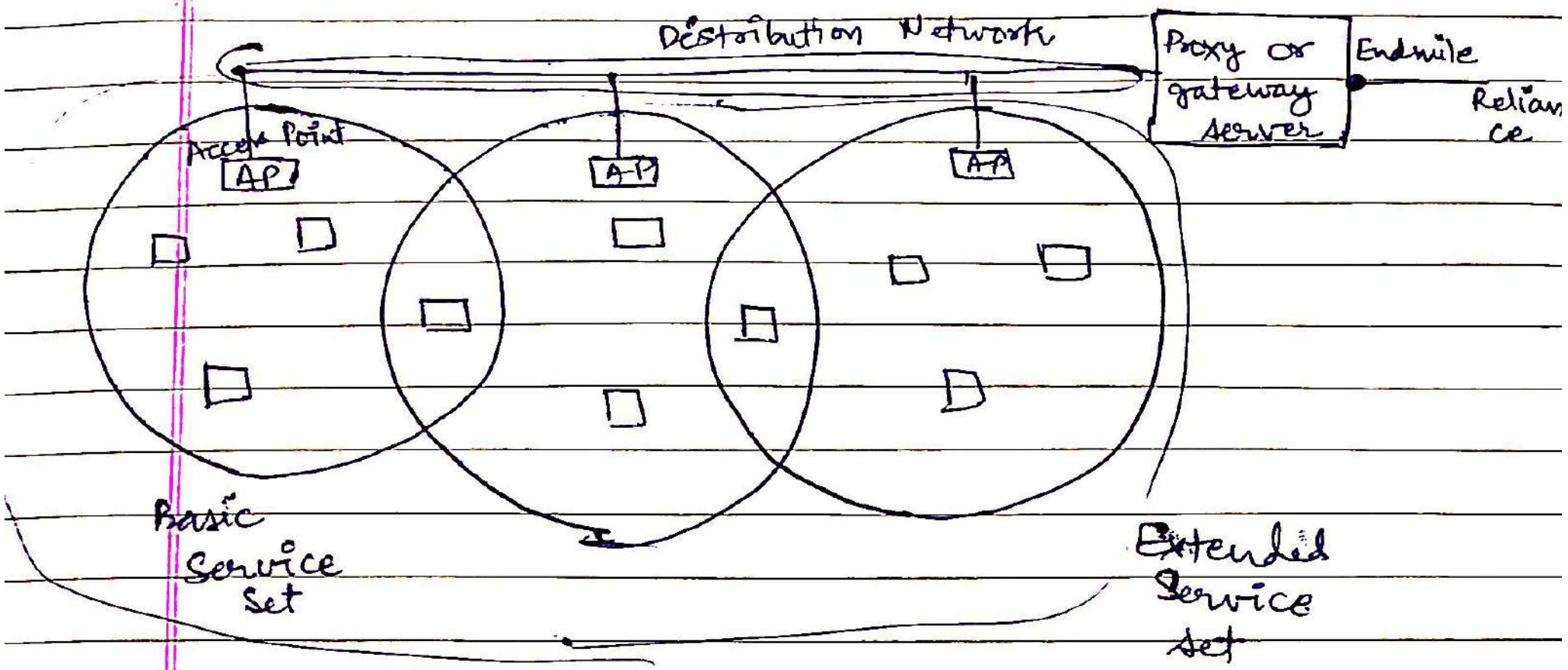
3. No time stamp option in IPV6.

4. The source route option is called source route extension in IPV6

5. Fragmentation ~~holds~~ fields of IPV4 base header are moved to fragmentation extension header in IPV6.

6. The authentication extension header is new in IPV6

7. The encrypted security payload extension header is added 'new' in IPV6.

Introduction to wifi

(Data-link layer protocol)