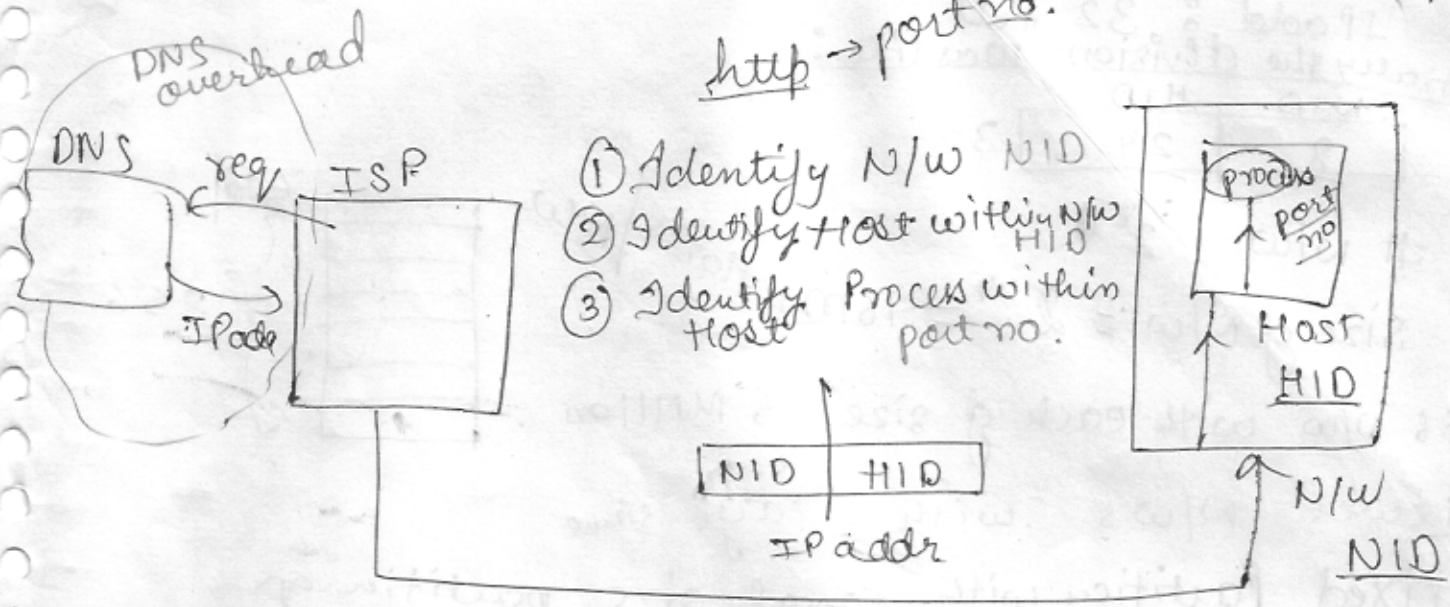
http → port no.



DNS ~~over~~ overhead: DNS converts the req. into IP address which is DNS overhead.

~~If we directly write IPadd, the process is faster.~~

$$\frac{\text{addressspace}}{\#\text{parts}} = \text{size of each part}$$

$$\frac{2^n}{2^K} = \frac{2^{n-K}}{\text{no of parts}}$$

n: Total bits

K: No of bits taken for each part

~~#parts = 2^K~~

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

→ 3 bits

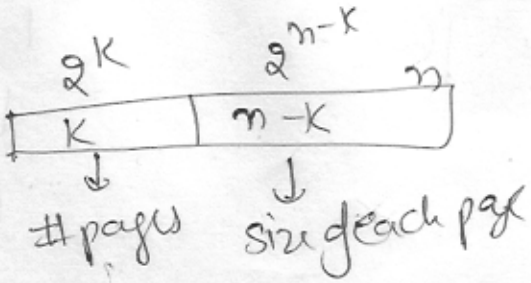
addspace = 2^3

→ 2 bits for a part

~~#parts~~ = 2^2

→ size of each part

$$\frac{2^3}{2^2} = 2^{3-2} = 2$$



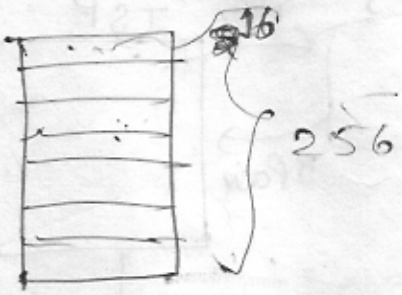
IP add : 32 Bits

Initially the division was like :

NID	HID	
8	24	32

N/w = 2^8

size of N/w = 2^{24} = 16M = Hosts/Nw



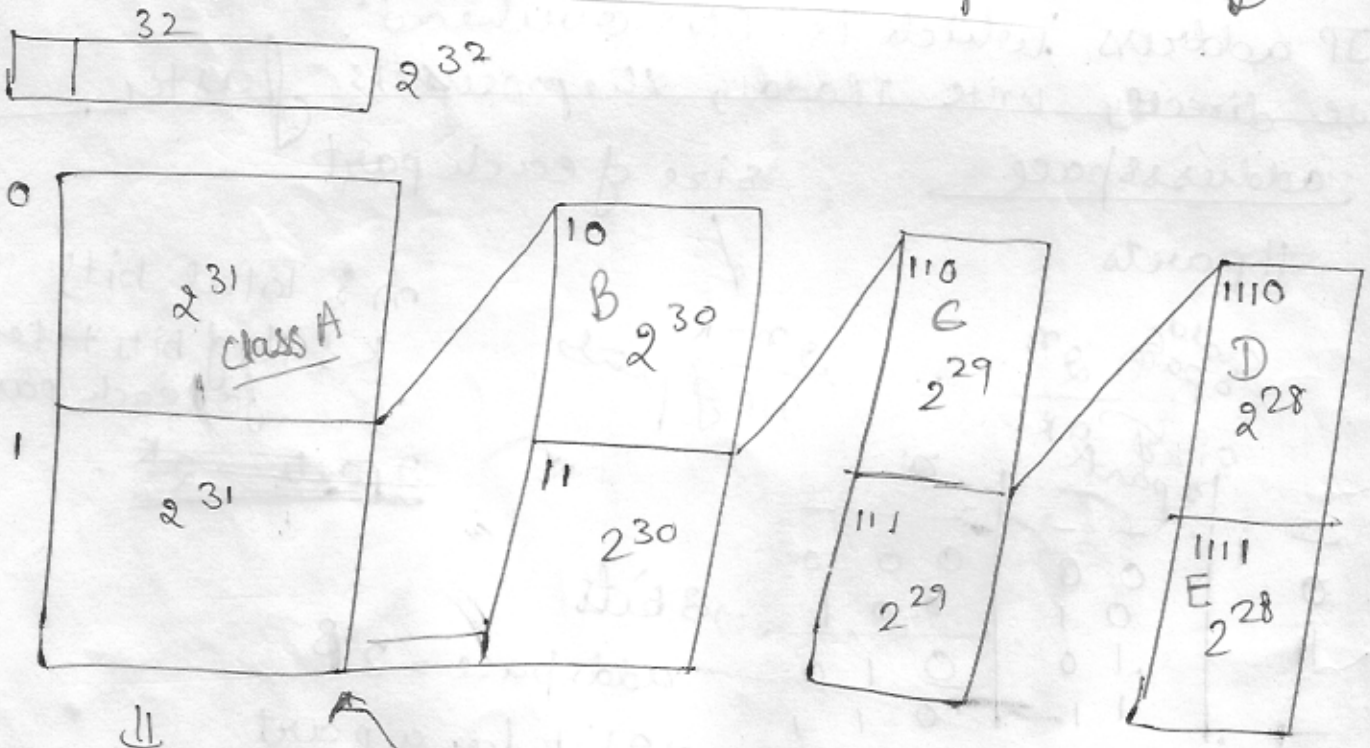
256 N/w with each of size 16 Million.

fewer N/ws with large size

fixed Partition with equal size partition of

NEW DIVISION SCHEME

fixed part with variable size partitions ↓



⇓

Tot add space = 2^{32}

Choosen bit = 1

size of each part = $\frac{2^{32}}{2^1} = 2^{31}$

0 2^{31}

10. 2^{30}

110 2^{29}

1110 2^{28}

1111 2^{28}

gate

Representation of IP addresses

IP address can be represented in many ways.

(A) BINARY REPRESENTATION : 0s & 1s of 32 bits

(B) Dotted Decimal Representation : 32 bits are divided into 4 Bytes and every byte is converted to Decimal number.

ex 10 . 20 . 30 . 40

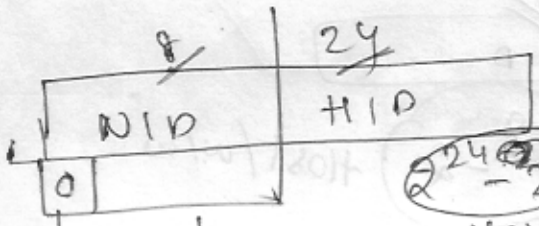
$1 \rightarrow 2^1 - 1$

$11 \rightarrow 2^2 - 1$

$111 \rightarrow 2^3 - 1$

$1111111 = 2^7 - 1 = 127$

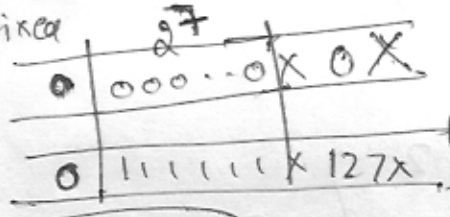
Class A



$2^{24} - 2$ → all 0s & 1s not taken

Hosts possible

fixed



$2^7 - 2$ N/w

1 - 126

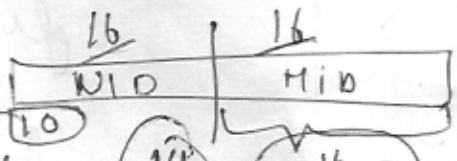
N/w possible in class A each with $2^{24} - 2$ hosts

10 . 24 . 30 . 1

B/w 1-126 ⇒ class A

Class B

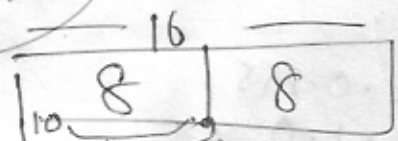
128 - 191



fixed 10

N/w possible : 2^{14}

$2^{16} - 2$ Hosts possible



Only 8 are necessary for knowing the range

10 0 0 0 0 0 0 - 128
- 129
⋮
1 1 1 1 1 1 1 - 191

64

64×2^8 : Total N/w : 2^{14}

Other addresses

128.0 129.0 191.0
128.1
⋮ ⋮ ⋮

128.255 129.255 191.255
256 256 256 Total 2^{14}.

Class C

24	8
110 NID	HID

N/w = 2^{21} $2^8 - 2$ Host/w/w

192 - 223

110 - - - - - ~~192~~
00000 - 192
} 32.
110 11111 - 223

NIDs
32×2^{16}

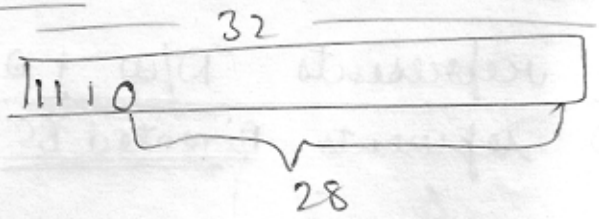
Other ~~IP addr~~ NIDs

192.0.0 223.0.0
⋮ ⋮

Each no from 192 to 223 has 2^{16} NIDs

192.0.255
192.1.0
⋮
192.1.255
⋮
192.255.255 223.255.255
2^{16} 2^{16}

Class D



NO NIDS
NO HIDS

1110 _ _ _ _
0 0 0 0 - 224
225

1 1 1 1 - 239

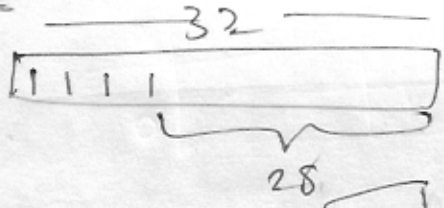
$\boxed{2^{28}}$ Class D addresses are possible

⇓

used for MULTICASTING

224–239

Class E



$\boxed{2^{28}}$ Class E addresses are possible

1111 _ _ _ _
0 0 0 0 → 240

1 1 1 1 → 255

Class E is Reserved for security purposes

240 - 255

Identification of IP addr class

A : 1–126
B : 128–191
C : 192–223
D : 224–239
E : 240–255

Reason : why all 0s & 1s are not taken as IP addr :

Note : All 0s in Host ID represents N/w ID
All 1s in Host ID represents Directed BC addr

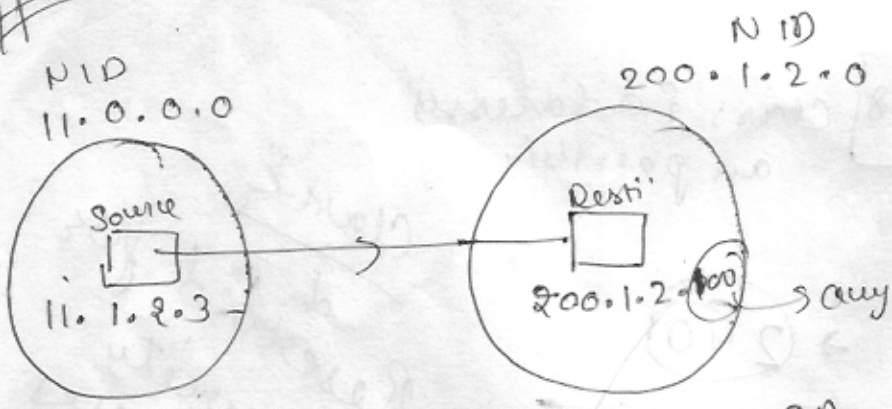
		N/w IDs
20.1.2.3	– class A	20.0.0.0
100.1.5.6	– class A	100.0.0.0
150.2.3.4	– class B	150.2.0.0
200.1.2.3	– class C	200.1.2.0
255.1.2.3	– class E	X No N/w ID
300.0.0.1	– X	

in class B last 2 Bytes are for HID

in class C last 3 Byte are for HID

for N/w ID, only HIDs should be kept 0, others same.

#



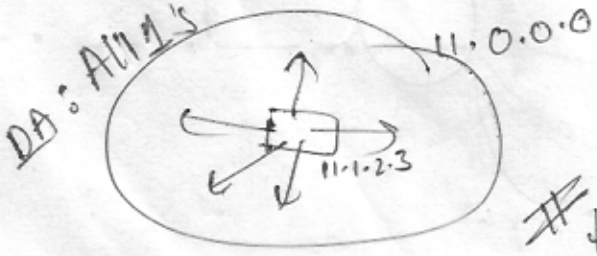
UNICASTING
from 1 host in 1 N/w to another host in diff N/w

	SA	DA
Data	11.1.2.3	200.1.2.100

LIMITED BROADCAST

[packet but is one but is visible to all]

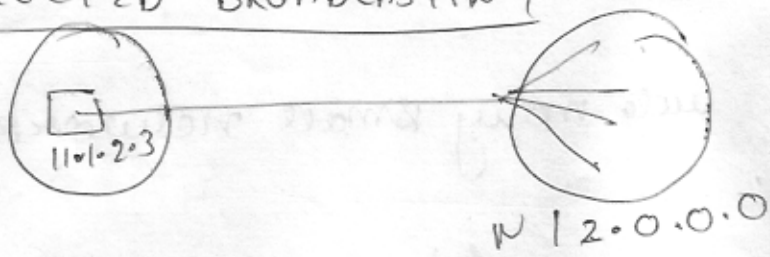
	SA	DA (LBA)
Data	11.1.2.3	255.255.255.255



If Dest addr = 1.1.1.1 all 1s

then packet is send to all hosts within same N/w

DIRECTED BROADCASTING



Data	11.1.2.3	12.255.255.255
	SA	DA

If a host wants to send data packet to all hosts on other N/w the

Dest addr : 1.1.1
NID. 3 ones.

Host ID : 255
→ To send to all Hosts on that N/w

Note : All 1's in HID part is directed Broadcast address (DBA)

Limited Broadcast Addr (LBA) remains same for all
→ All 1s.

	NID	DBA	LBA
11.2.3.5	11.0.0.0	11.255.255.255	255.255.255.255
100.2.3.5	100.0.0.0	100.255.255.255	"
200.1.2.10	200.1.2.0	200.1.2.255	"
175.1.2.10	175.1.0.0	175.1.255.255	"
	⇓ Make only Host ID = 0	⇓ Make only Host ID = 255 ⇓ If somebody from other N/w wants to send packet all host on this N/w	⇓ All 255.

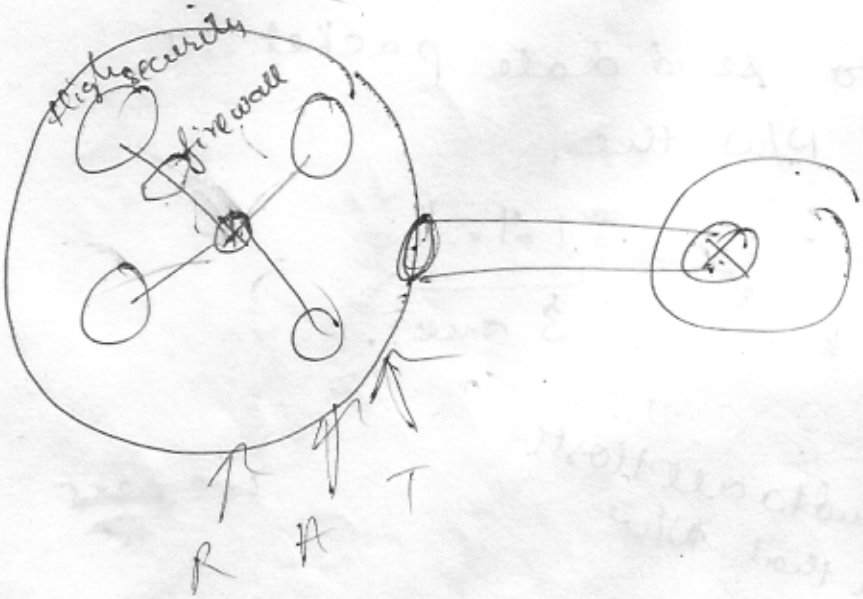
NID	HID	LBA
NID	1	1
NID	0	DBA

Subnetting

Dividing a big network into many small networks is called subnetting.

Reasons for Subnetting:

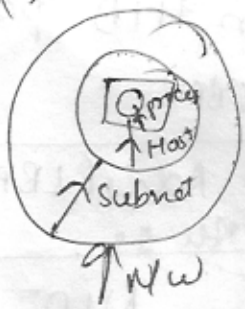
1. Easy Maintainence [with the help of several N/W administrators for each subnet]
2. Security can be improved



for dividing into subnets we need to choose bits. Though we can choose any position, but we should choose from starting

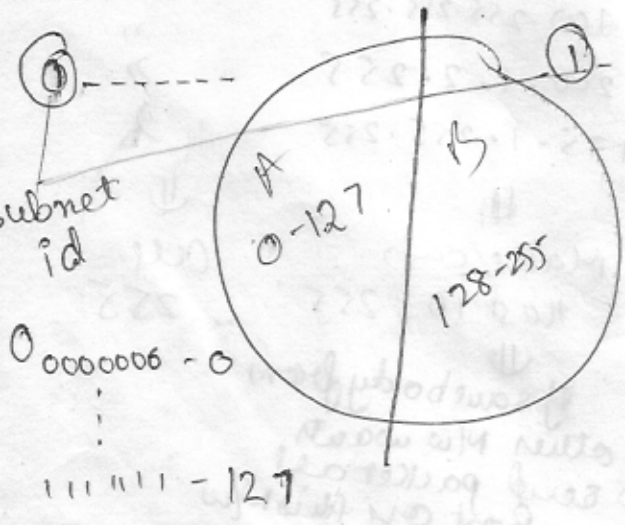
Now, 4 steps:

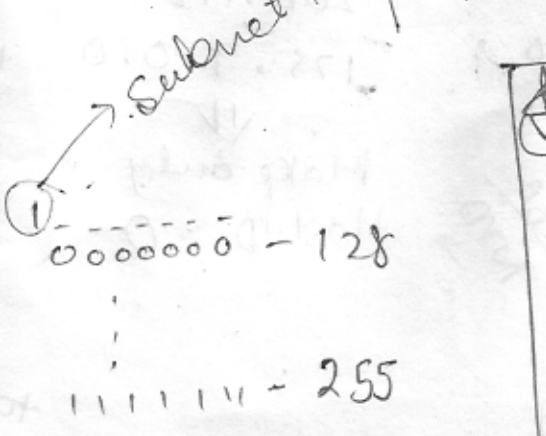
1. Identify N/W
2. " subnet → extra step
3. " host
4. " process



Class C 200.1.2.0 → NID

Subnet id	Block ID BID	NID





from outside:

200.1.2.0 ⇒ NID for entire N/w

200.1.2.255 ⇒ DBA for " "

from Inside:

200.1.2.0 ⇒ subnet ID for part A

200.1.2.255 ⇒ DBA for part B.

200.1.2.128 ⇒ SID for part B

200.1.2.127 ⇒ DBA for part A

Dividing into 4 subnets

↳ we would need 2 bits ($2^2 = 4$)

128 64

[- -] - - - - - -

00 - 0

01 - 64

10 - 128

11 - 192

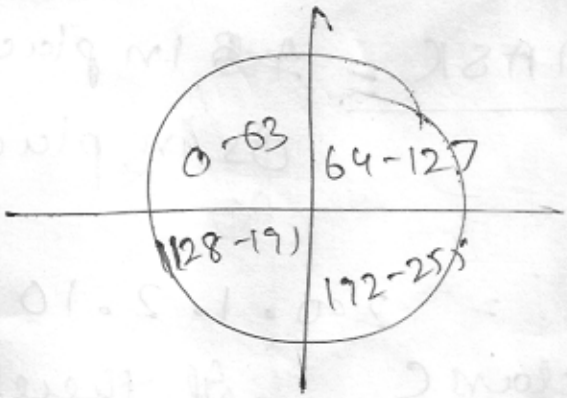
Ranges of 4 subnets

0 - 63

64 - 127

128 - 191

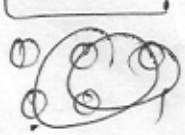
192 - 255



DIVIDING INTO 8 subnets

Choose 3 starting bit

[- - -] - - - - -

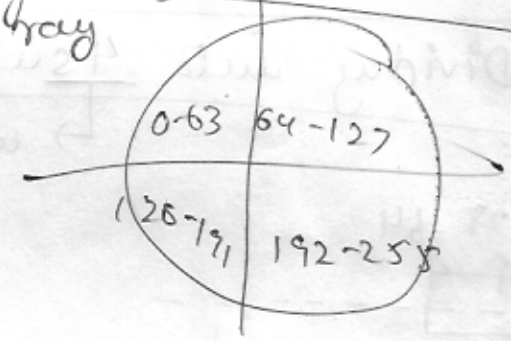


000 — 0 - 31
001 — 32 - 63
010 — 64 - 95
011 — 96 - 127
100 — 128 - 159
101 — 160 - 191
110 — 192 - 223
111 — 224 - 255

ranges for 8 subnets

IP addr — SID → by looking at range

200.1.2.10 - 200.1.2.0
200.1.2.100 - 200.1.2.64
200.1.2.150 - 200.1.2.128
200.1.2.254 - 200.1.2.192



We can calculate SIDs using the the

SUBNET MASK = 1s in place of SID & NID
0s in place of HID

given
IP addr = 200.1.2.10
Since it is class C - If there are 4 subnets then 2 bits
So, subnet mask = 255.255.255.192
11000000
∴ 4 subnets

And any no with 255
same no.

SUBNET MASK

It is a 32 bit no where

#1s ⇒ NID + SID

#0s ⇒ HID

given IP = 200.1.2.10

4 subnets

So, Subnet Mask = 255.255.255.192

SID = BITWISE AND of IP & SM

```
255.255.255.192      BITWISE
200 . 1 . 2 . 10     AND
-----------------
200 . 1 . 2 . 11000000
              00001010
              --------
              00000000
              ⇓
              0
```

Ans 200.1.2.0.

~~COMPLEM~~ Subnet Mask → filter allowing only NID & SID

Complement of subnet Mask = NID & SID = 0
HID = 1

↓

BITWISE AND with this results in HID

0.0.0.00111111

IP addr = 255 . 255 . 255 . 100

SID = 255 . 255 . 255 . 11000000

200 . 1 . 2

200 . 1 . 2 — 69

Same number as AND with 255

If SM is given then

Imp

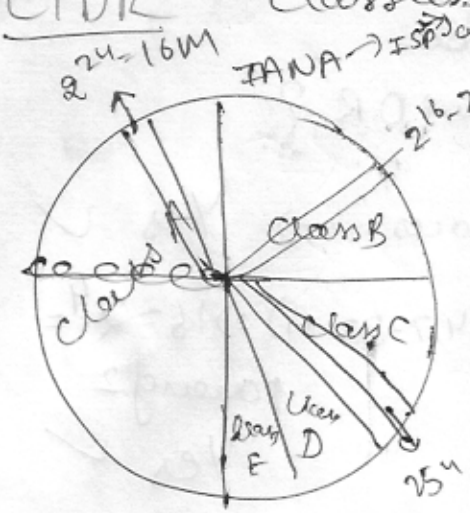
$\#Host = 2^{\#0s} - 2$

$\# SID = 2^{\#extra\,1s}$ [we would need class also]

extra 1s = #1s in SM − #1s in that class

Subnet Mask	#Host	# subnets Class A	# SN class B	# SN Class C
Default SN for Class A 255 . 0 . 0 . 0	$2^{24} - 2$	$2^0 = 1$	×	×
255 . 128 . 0 . 0	$2^{23} - 2$	$2^1 = 2$	×	×
255 . 192 . 0 . 0	$2^{22} - 2$	2^2	×	×
255 . 240 . 0 . 0	$2^{20} - 2$	2^4	×	×
Default SN for Class B 255 . 255 . 0 . 0	$2^{16} - 2$	2^8	1	×
255 . 255 . 254 . 0	$2^9 - 2$	2^{15}	2^7	×
Default SN for Class C 255 . 255 . 255 . 0	$2^8 - 2$	2^{16}	2^8	1
255 . 255 . 255 . 240	$2^4 - 2$	2^{20}	2^{12}	2^4
255 . 255 . 255 . 224	$2^5 - 2$	2^{19}	2^{11}	23
255 . 255 . 248 . 0	$2^{11} - 2$	2^{13}	2^5	↑

CIDR Classless InterDomain Routing



Need: In the world, there are many small networks :
everyone wants class C.
Class A and class B are going to be wasted so, class A & class B are again divided into small networks. This is called CIDR and these small networks are called as BLOCKS.

Rules for CIDR Blocks :

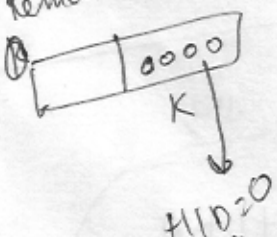
① All the IP addresses in a block should be contiguous.

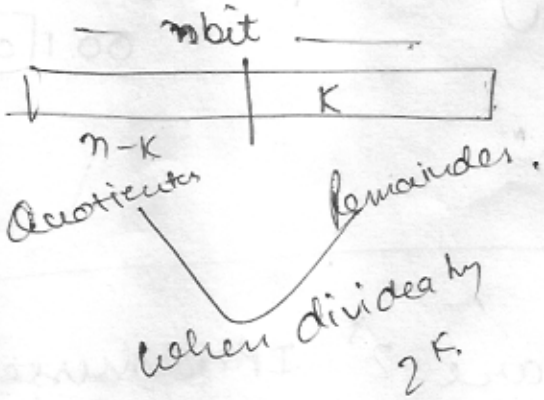
② No of addresses in the block should be power of 2 (2^K)

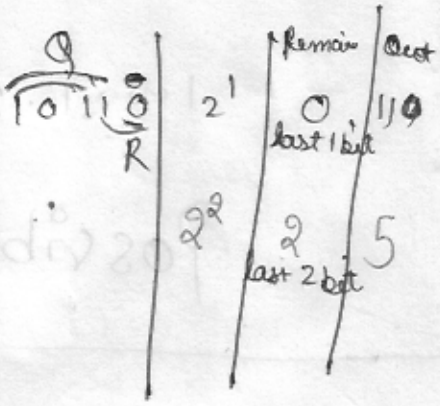
③ First IP address should be divisible by size of block. (2^K) [all zeros in HID part]

⇓

If first IP addr is divisible by 2^K then, remainder = 0







ex 1

100.1.2.32
.33
⋮
100.1.2.47

Is this possible accn to CIDR?

Rule 1 Contiguous ⇒ Yes ✓

Rule 2 size $(47-32+1) \Rightarrow 16 = 2^4 =$ power of 2 Yes ✓

Rule 3 first IP addr 100.1.2.0010 0000

divisible by 2^4

divisible yes ✓

last 4 bits is remainder.

So, possible.

ex 2

100.1.2.32
⋮
100.1.2.63

32 IP addresses

Rule 1 ✓

Rule 2 ✓ 2^5

Rule 3 ✓

last 5 bits

001 00000

divisible

Possible

Since there are 2^5 IP addresses so,

HID = 5

NID = 32 − 5 = 27

NID	HID
27 bits	5 bits

CIDR Rep : 100.1.2.47/27 → This number is enough to represent entire block

100.1.2.47/27

⇓ This '27' implies (32-27=5), 5 HID

47

100 1 . 2 . 0 0 1 | 0 1 1 1 1

HID

NID → 0 0 0 | 0 0 0 0 0 → 32 → NID

⋮

0 0 1 | 1 1 1 1 1 → 64 → DBA

This means 100.1.2.47 belong to block 100.1.2.32 to 100.1.2.64

---

find block g

20.1.2.4/28 → NID

HID = 32 − 28 = 4.

20.1.2. 0 0 0 0 | 0 0 0 0

← 0000 → 0

⋮

← 1111 → 15

Ans: Block 20.1.2.0 to 20.1.2.15.

Q find block g

40.20.15.20/26

32 − 26 = 6 bits for HID

Ans 40.20.15.0 to 40.20.15.63

2	20	
2	10	0
2	5	0
2	2	1
	1	0

---

0 0 | 0 1 0 1 0 0

0 0 0 0 0 0 0 0 → 0

0 0 1 1 1 1 1 1 → 63

Imp find block range

100 . 10 . 20 . 40 / 20

100 . 10 . 000	0100 . 40	
0001	0000 . 00000000	=
	⋮	
0001	1111 . 11111111	=

Ans

100 . 10 . 16 . 0

100 . 10 . 31 . 255

Subnetting blocks

(a)

find the SIDs of 2 parts of

20 . 1 . 2 . 4 / 25

25

0 | 000100

available for subnets.

Ans Block is 20 . 1 . 2 . 0 . ⋮ 20 . 1 . 2 . 127 } 2 parts 0 - 63, 64 - 127

32 - 25 = 7 bits of HID

Ans SIDs

20 . 1 . 2 . 0 / 26 and 20 . 1 . 2 . 64 / 26

Now 1 more bit is used for HID

Another way

7 bits for 2 parts, choose 1 bit

0	_ _ _ _ _ _	
0	0 0 0 0 0 0	= 0
1	0 0 0 0 0 0	⇒ 64.

for subnet

(9) 20.1.2.4/25
Divide in 4 parts & find SIDs

Ans 20.1.2. [ ] [ ] ------ → 4 parts

00	0
01	32
10	64
11	96

Ans

20.1.2.0 /27
20.1.2.32 /27 → ∵ 2 more bits are taken for subnets.
20.1.2.64 /27
20.1.2.96 /27

~~If first bit of HID choosen for subnet.~~
~~20.1.2.[ ]------~~

0 - 127 1st part
128 - 255 2nd part

~~If last bit choosen from HID for SID.~~
~~200.1.2.--------[ ]~~
o/p

even & odd in diff part

Note for getting subnets bits, bits are choosen from HID part and these bits can be choosen from anywhere theoritically but practically only first few bits should be choosen.

In a class C N/W if 4 subnets are reqd then theoritically $^8C_2$ subnet masks are possible
∵ for 4 subnets, 2 bits are reqd

but practically only 1

255.255.255.11000000

= 255.255.255.192

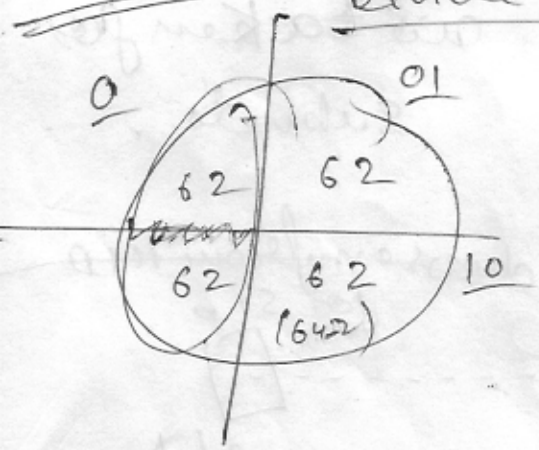
Till Now FLSM → fixed length subnet masking

Variable Length Subnet Masking

Q Divide 200.1.2.0 into 3 subnets of 120, 60, 60

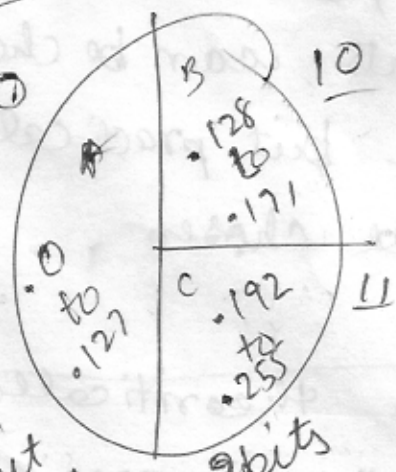
Ist way

Divide in 4 parts & then combine 2 part



SID

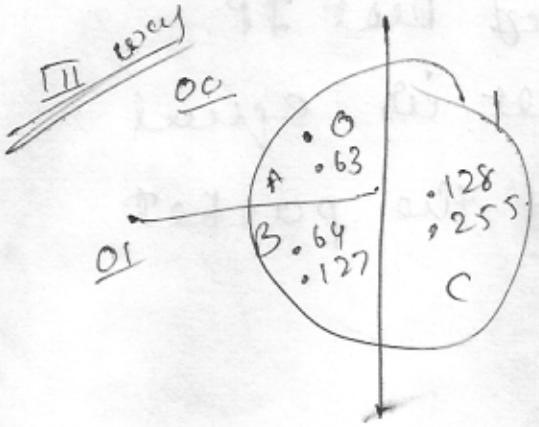
IInd way → Divide in 2 parts. then divide. That part in further 2 part



1 bit needed for subnetting

2 bits

SID	SMS
.0	255.255.255.128 (A)
.128	255.255.255.192 (B&C)
.192	



0 00
01
10
11

SID = .0 SM 255.255.255.192 [A & B]
.64
.128 255.255.255.128. [C]

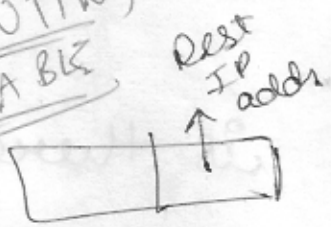
Imp Note : If the subnets are of equal size then their subnet masks are same.

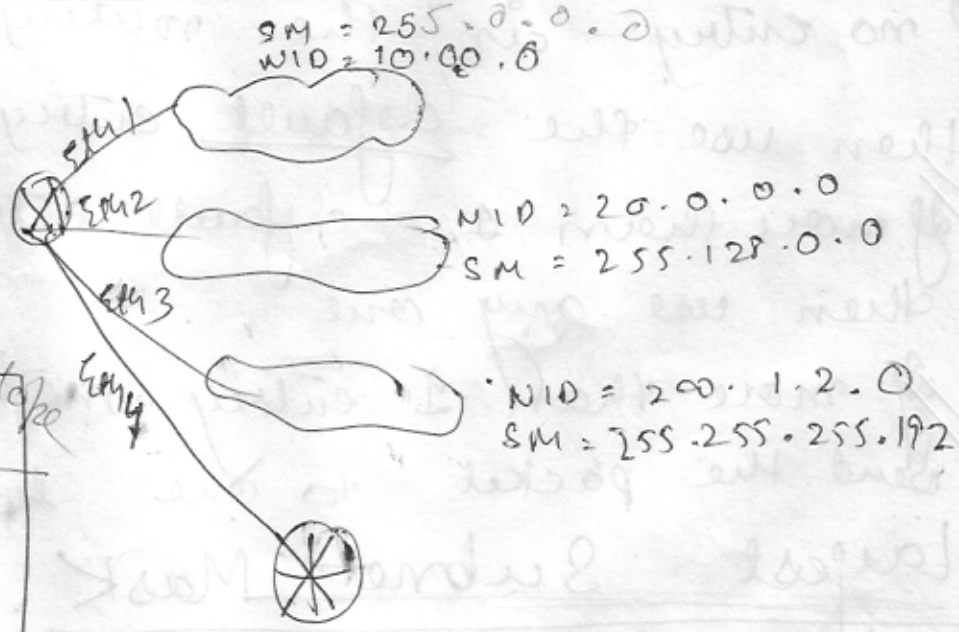
eg A & B in III
B & C in II

But if they are variable size then their subnet masks are also different.

eg A or B & C in III

ROUTING TABLE





Routing Table

Dest NID	SM	Next Hop NH
11.0.00	255.0.0.0	Eth1
20.0.00	255.128.0.0	Eth 2
200.1.20	255.255.255.192	Eth 3
Default		Eth 4

In the Incoming packet, ~~compare~~ Do 'AND' of Dest IP address with SM if the result is equal to NID of that then send the packet to that interface.

Case I

Incoming packet 11.20.30.40

'AND' with 255.0.0.0

11.0.0.0 => matches with NID

So send to Eth 1.

Case II.

20.1.2.3 AND with 255.0.0.0 = 20.0.0.0
255.128.0.0 = 20.0.0.0

Does not go to Eth 1,
go to Eth 2.

If one entry in routing table matches, then send that packet through that entry.

If no entry in the routing table matches then use the default entry.

If more than 1 default entry is there then use any one.

If more than 1 entry matches then send the packet to one having Largest Subnet Mask.

Eg

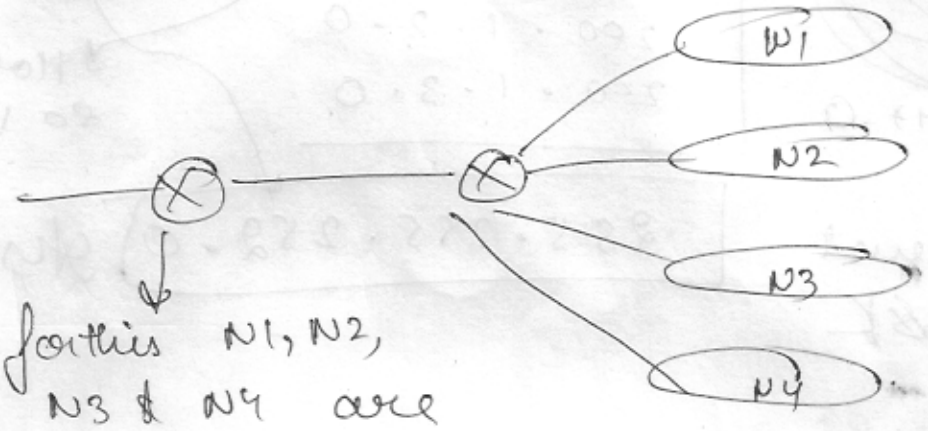
SM
I 255.0.0.0
II 255.255.0.0
III 255.255.192.0

All 3 match then send to largest i.e III

SuperNetting

The process of aggregating many networks into 1 big network is called supernetting.

Adv/Use of supernetting is that it can decrease the no. of entries in the routing table.



for this N1, N2, N3 & N4 are 1 n/w and the routing table will have only 1 entry for these 4, with supernet ID & supernet mask.

Rules for Supernetting

1. All the networks should have contiguous id.
2. Number of n/w should be a power of 2 and all the n/ws should be of same size
3. first ~~IP addr~~ NID should be divisible by size of the supernet.

ex1 200.1.0.0
200.1.1.0
200.1.2.0
200.1.3.0
] eligible for supernetting

ex2 200.1.32.0
200.1.33.0
⋮
200.1.47.0
] Eligible for supernetting

Supernet Mask : It is a 32 bit number in which no. of 1s indicate fixed part and no. of 0s indicate variable part

ex

200 . 1 . 0 . 0
200 . 1 . 1 . 0
200 . 1 . 2 . 0
200 . 1 . 3 . 0
⋮
200 . 1 . 47 . 0

fixed

200 . 1 . 0 . 0
200 . 1 . 1 . 0
200 . 1 . 2 . 0
200 . 1 . 3 . 0

Not completely variable only last 2 bits so 11111100 → Host ID so var.

Supernet Mask: 255 . 255 . 252 . 0 Ans.

ex 200 . 1 . 32 . 0
⋮
200 . 1 . 47 . 0

Ans 255 . 255 . 240 . 0

Supernet mask

	fixed	
32	0010	0000
	0010	0001
	0010	0010
	0010	0011
	⋮	
47	0010	1111

4 bits fixed so, 11110000 = 240

SUPERNET ID

'AND' any NID with supernet mask

(or)

'AND' all the NIDs

(or)

Take the first NID (200.1.0.0) & supernet ID.

Supernet ID : Take any IP address and 'AND' it with supernet mask then you get Supernet ID.

32

NID HID

8 24 Internet Prog.

↓

Classfull (A, B, C, D, E)

↓

Blocks (CIDR

Subnetting - SM

↓

Supernet - SM

After IPV4 Depletion,
private IP addr & IPV6 came in.

Network address Translation [Public and

Internally Reserved private addresses Can be used & only 1 public IP addr (Proxy Server) needs to be purchase

Reserved IP addr for private

10.0.0.0 - 10.255.255.255

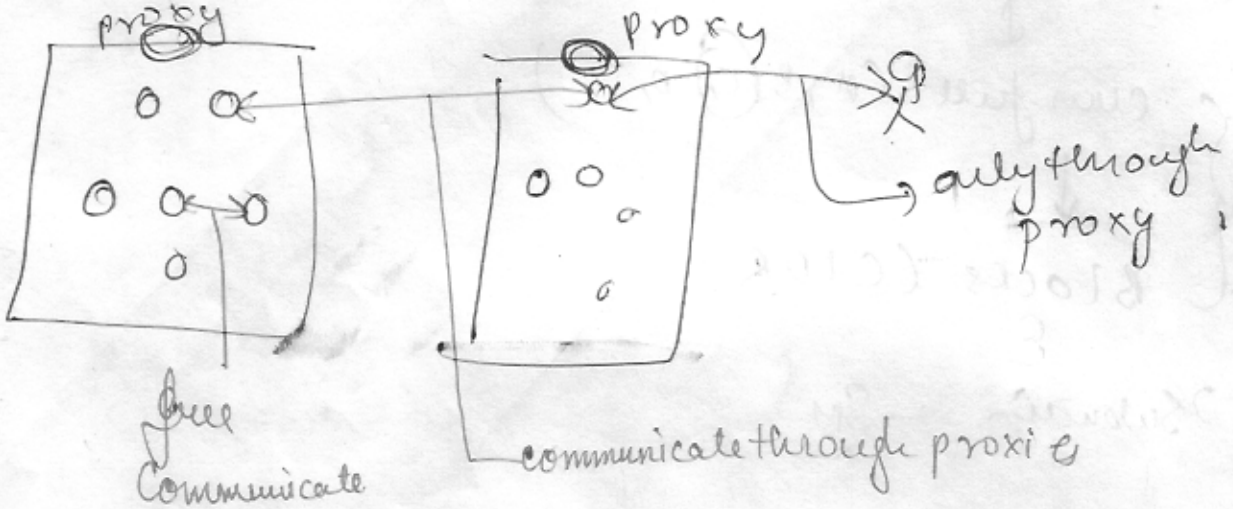
172.16.0.0 - 172.31.255.255

192.168.0.0 - 192.168.255.255

If due to any reason, any packet from private N/w comes to R_1 then R_1 discards the packet if it is from reserved private IP addresses.

IPV4 Exhaustion

Currently all IPV4 addresses have been exhausted so, Network Addr Translation (NAT) is useful.



Solutions of Exhaustion :

1. IPV6
2. NAT

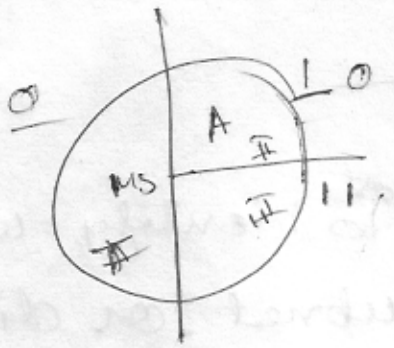
Questions

1) given $\left\{ \begin{array}{l} SM \\ IP \end{array} \right.$ $\overline{SPD?}$

2) given IP_1, IP_2 SM Same n/w or not?

ex) Let an ISP has a block 20.100.16.40/20 and it wants to divide this block into 1 half and give it to Microsoft and in the remaining half 1 quater to Amazon & keep the rest with itself.

20.100.16.40/20



NID = 20

SID

part I

NID = 26

HID = 32 − 20 = 12

1 for SID

20.100.0001|0000.40

20.100.16.0/21

extra bit for SID

part II

20.100.0001|0000

10

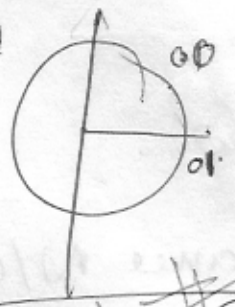
11

20.100.24.0/22

part III

20.100.28.0/22

or



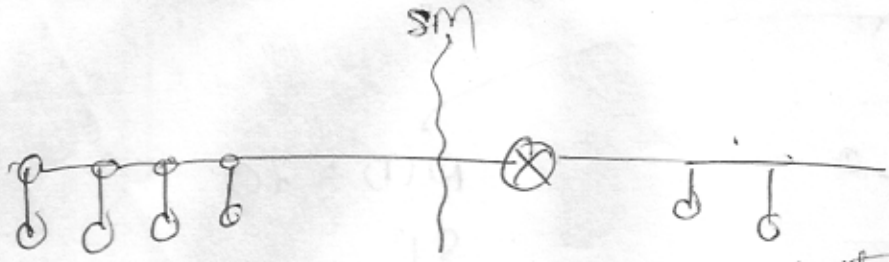
ISP provides
- IPv4
- SM
- DNS
- Default Router

nslookup gives IP address of given website.

A	B		
IP_A	IP_B	IP_A	IP
SM_A	SM_B	SM_B	IP_B
SID_A A	SID_B B	SID_A B	SM_A
			SID_B A

equal then same n/w accn to B

equal then same n/w accn to A



SM ~~acts like~~ will help one host to identify whether other hosts is ~~are~~ in same subnet or different subnet. If in same subnet then directly send packet else send the packet through Router.

Q

$IP_A = 200.1.2.70$ 01001000
$SM_A = 255.255.255.128$ 10...

$IP_B = 200.1.2.$ [illegible]
$SM_B = 255.255.255.192$

$SID_{A_A} = 200.1.2.0$

$SID_{B_B} = 200.1.2.0$

2|10
2|5 0
2|2 1 00001010
0 1

~~Accn to~~

IP_A 200.1.2.70 01001000
$SM_B = 255.255.255.192$ 01000000

$IP_B = 200.1.2.16$
SM_A 255.255.255.128

SID_{A_B} 200.1.2.64

SID_{B_A} 200.1.2.0.

use SM_A with Both IP

Accn to A : A & B are in same N/w

Accn to B : A & B are not in same N/w.

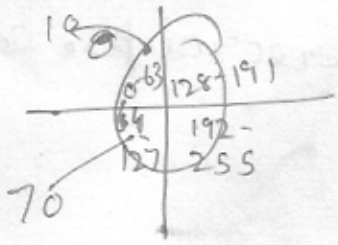
Reason:

accn to A

Same n/w

N/w is divided into 2 subnets (128)
1 bit chos
2 parts

acc to B



10 & 70 in diff subnets

N/w is divided into 4 parts or subnets (192)

2 bits choosey

2^2 parts

Sup

DBA:
20.10.7.255
00000111.11111111

from DBA we can know the point till when definitely not HID. But HID might start from anywhere after that

.111.

from here to left Definitely not HID

11 in HID

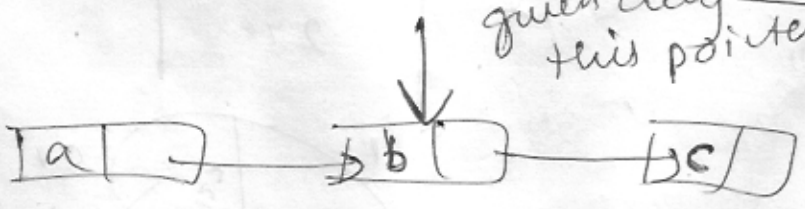
21 NID

NID can have 21 bits or 22 or 23

31 bits

HID

11 bits HID or 10 ... 1 - HID



given only this pointer

however you delete b

Ans swap b & c.

BIT FIELDING

careercups.com

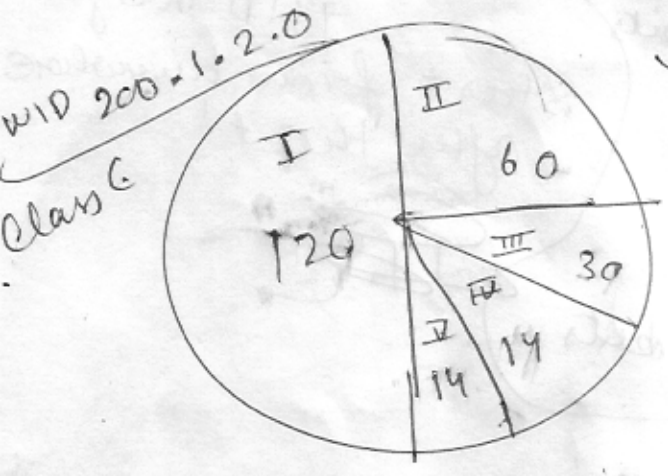
Left shift
Right "
Rotate
How to make all zeros into 1

Common mistakes in Textbooks

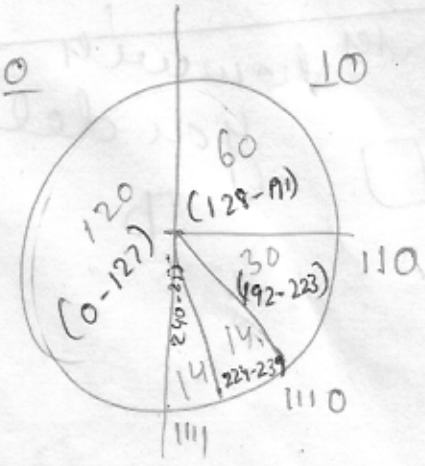
If K bits are chosen for subnet ID part then 2^K subnets are possible not $2^K - 2$

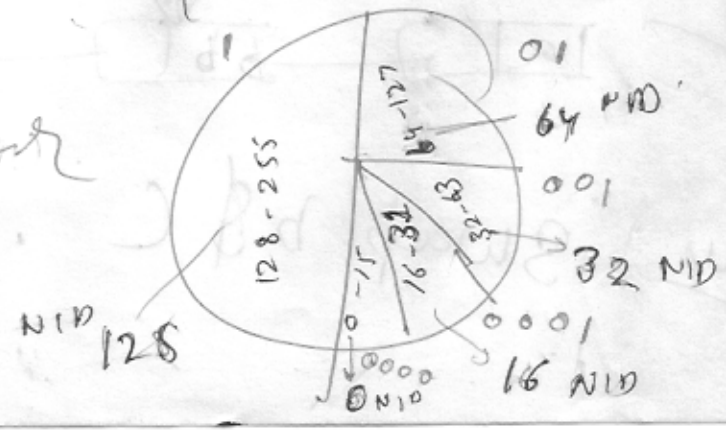
* If variable length subnet Masking (VLSM) is used and if there are 3 subnets then we need not have 3 subnet masks.

Note : All the N/ws of same size will have same SM.



	SM	SID
I	255.255.255.128	200.1.2.0
II	255.255.255.192	200.1.2.128
III	255.255.255.224	200.1.2.192
IV	255.255.255.240	200.1.2.224
V	255.255.255.240	200.1.2.240



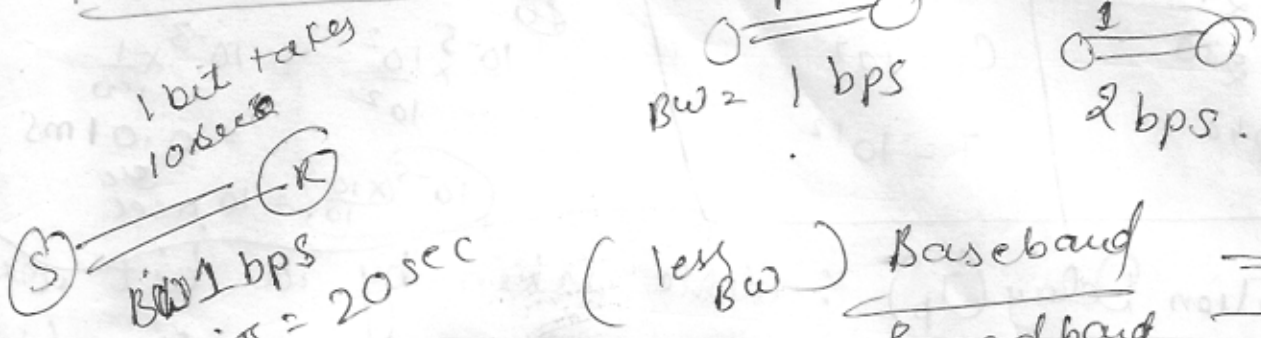


SM = 255.255.255.255 Host alone to send packet

any other, it must be sent through Router.

Q-44-66 → IPaddr & subnetting.

FLOW CONTROL



for 10 bits = 20 sec

$T_t = 10$ sec

$P_D = 10$ sec

(less BW) Baseband

(more BW) Broadband

DELAYS IN CN

Transmission delay : (T_t) Time taken to transfer a packet onto the outgoing link is called Transmission time (T_t)

If 'L' = size of packet in bits and 'B' : B/w in bits per sec

then $T_t = \frac{L}{B}$ sec

BW = B bps

1 sec → B bits

B bits → 1 sec.

1 bit = $\frac{1}{B}$ sec

L bits = $\frac{L}{B}$ sec.

Q If L = 1000 bits and B = 1 Kbps.

then

$T_t = \frac{1000}{1000}$ sec

= 1 sec.

Q If L = 1 Kb and B = 1 Kbps then

$T_t = 1.024$ sec

Data

always powers of 2.

$K = 2^{10}$
$= 1024$

$M = 2^{20}$

$G = 2^{30}$

$T = 2^{40}$

BW

$K = 10^3$
$= 1000$

$M = 10^6$

$G = 10^9$

$T = 10^{12}$

milli $= 10^{-3}$ sec

$\mu = 10^{-6}$ sec

ex 10^{-5} sec.

$10^{-5} \times \frac{10^2}{10^2} = 10^{-3} \times \frac{1}{100}$
$= 0.01$ ms

$10^{-5} \times \frac{10}{10} = 10\ \mu$sec

Propagation Delay (T_p) : Time taken by a bit or signal to travel from 1 end ~~of the wire~~ to other end. ~~of the~~

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

where D = length of the wire in m, and
v = velocity in m/s of wire

Q If length of a wire is 2.1 Kms and velocity of the signal in the wire = 70% of that of speed of light in vacuum.

3×10^8 m/s

Ans $T_p = \frac{2100}{3 \times 10^8}$

$\frac{70}{100} \times 3 \times 10^8$

2.1×10^8

7×10^{-6} sec

$\frac{21000}{21 \times 10^8} = 10^{-5}$ sec

or 0.01 ms or 10 μsec

Q 2) Length of packet (L) = 1000 bits

$B = 1000$ bps

$D = 2 \times 10^8$ m

$v = 2 \times 10^8$ m/s

(A) $T_t = 1$ sec

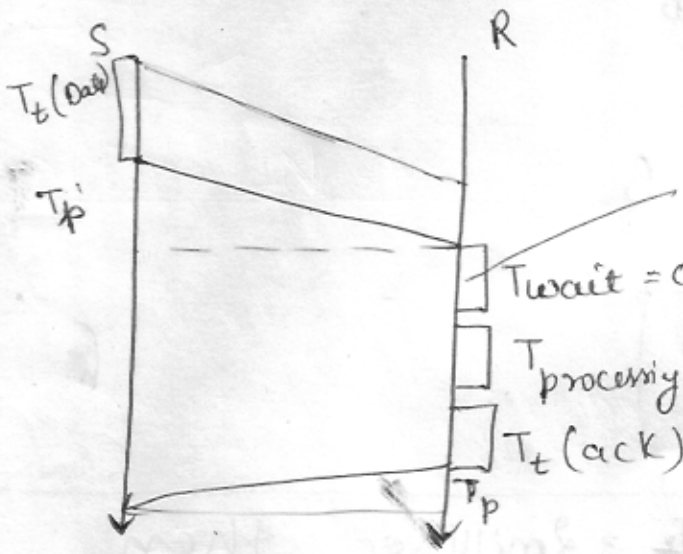
$T_p = 1$ sec

Tot. time taken to send = 2 sec.

- T_t of msg sent
- T_p only once count for whatever is the msg sent

Flow Control Techniques

(1) STOP AND WAIT



Taken as 0

Taken as 0 as ack is very small as compared to data

Stop and wait taken

$$\underbrace{T_t \text{ for data}}_{\text{useful Time}} + \underbrace{T_p + T_t \text{ for ack} + T_p}_{\text{waiting Time}}$$

T_t for ack = 0

$$= \boxed{2T_p + T_t \text{ for data}}$$

$$\eta \text{ (efficiency)} = \frac{\text{useful T}}{\text{Total Time}} = \frac{T_t}{T_t + 2T_p} = \boxed{\frac{1}{1 + 2\frac{T_p}{T_t}}}$$

(divide by T_t on Numerator)

$$\boxed{\eta = \frac{1}{1+2a} \quad \text{where} \quad a = \frac{T_p}{T_t}}$$

B/W utilization
or
Throughput
or
Effective BW

$$= \boxed{\frac{L}{T_t + 2T_p}} \quad \text{no of bits per sec.}$$

$$= \frac{\frac{L}{B} * B}{T_t + 2T_p}$$

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

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

$$\boxed{\text{Throughput} = \eta \cdot B}$$

Q If T_p = 1milli sec. and T_t = 2millisec then what is η ?

A

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

Linked

Q If B = 4Mbps
What is B actually used.

A = 2Mbps ($\because \eta$ is 50%)
Eff - BW

(a) for η to be 50%, what is relation b/w T_p & T_t

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

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

∴ it is sending packet instead of $1+2a$ after sending 1st & before rec. ack.

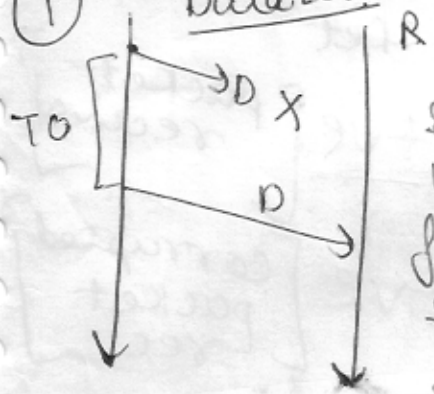
(b) for $\eta = 50\%$ what is smallest length of packet

$2T_pB_2$

$$\boxed{L = 2 * T_p * B}$$

Prob in Stop & wait

(1) Data Loss



S & R wait forever

S (wait for ack)

R [wait for Data]

Deadlock

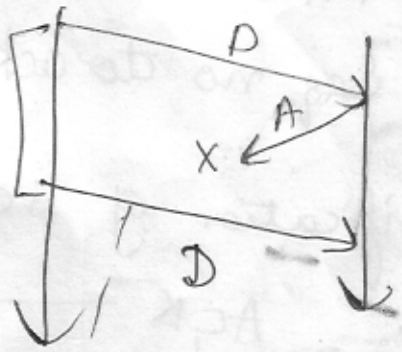
Soln : Time Out (TO)

Stop & wait + TO

= wait ARQ

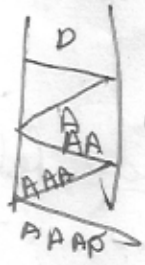
automatic Repeat Request

(2)



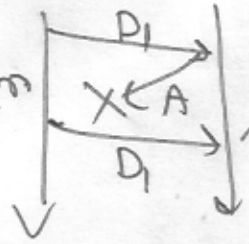
Duplicate Data [R might consider it as next data]

Soln : Sequence no. for data.



we cannot have ack for ack ∵ of BJ problem.

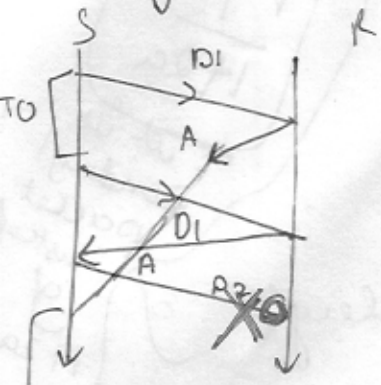
Soln

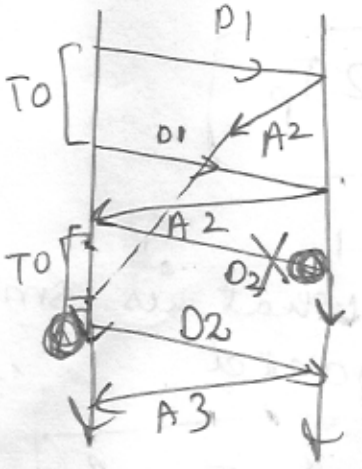


D_1 is already rec so discard this time

③

Delayed ACK





Sender will think ~~Ack~~ D_2 has been received but actually it is lost.

prob MISSING PACKET

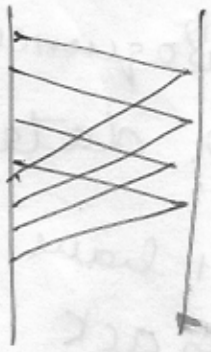
Soln Seq no to ack.

ACK no is always the no. of next expected data packet

Classification of ACK

ACK — +ve [packet received] / -ve [corrupted packet rec.]

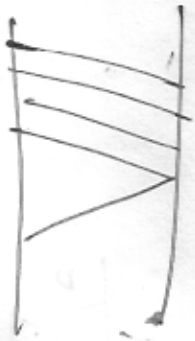
Independent



adv Reliability ↑

disadv Traffic ↑

Cummulative



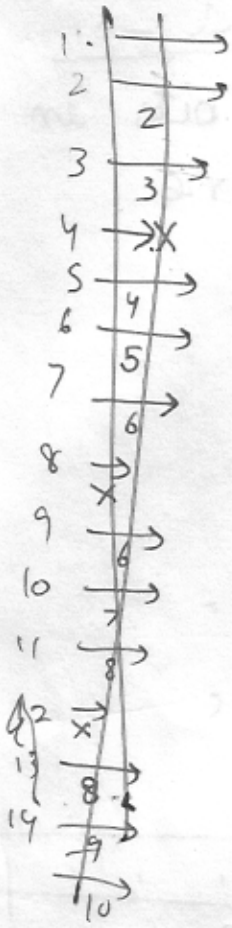
disadv Reliability ↓

adv Traffic ↓

No ack for packet not received

Q) In stop & wait, sender is sending '10' packets in which every 4th packet is lost. How many total transmissions are reqd.

A)



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

Ans 13

3 are re-transmissions

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

Q) If 1000 packets are to be sent on a link using S&W and if error probability on link is 80% then what is the total no of transmissions reqd.

Ans $\frac{80}{100} \times 1000$

800

1 - - . 800800 . . .

$1000\left[\frac{1}{1-\frac{80}{100}}\right]$

$= 1000\left[\frac{100}{20}\right]$

$= \boxed{5000}$ Ans

Imp

"p": error probability

n packets

$n + np + np^2 \ldots$

$n[1+p^2+p^3 \ldots]$

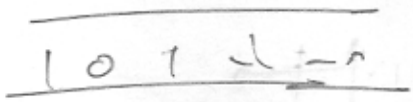
$n\left[\frac{1}{1-p}\right]$

Total transmission reqd for packets

Capacity of a channel / link / wire

$$\boxed{\text{Capacity} = (BW * T_p)} \text{ bits}$$

bps s → delay.

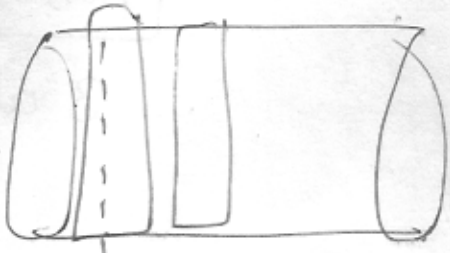


Max No of bits in the wire

Increasing Capacity :

① Inc. BW :

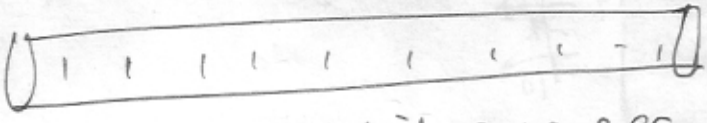
Broadband



1 Mbps
1 Million bits per sec

② Inc. Length :

Inc prop. delay



1bps even though 1 bit per sec but length is more so Capacity is more

Baseband with inc. length

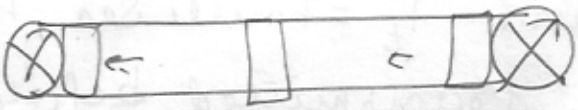
Q If BW = 1Mbps & T_p = 10 sec then what is capacity?

A) Cap = ~~PH * 10~~ ~~10~~ .

$= 10 * 10^6$ bits

If a packet size is 1000 bits then how many packets can be contained in the above channel simultaneously

Ans ~~10 x 10~~ $\frac{10^7}{10^3} = 10^4$ packets.



2.1 Mbps

$T_p = 1$ sec

packet size $= 10^3$ bits

$$Cap = \frac{(2.1 \times 10^6)\text{ bits}}{10^3 \text{ bits}} = 2100 \text{ packets}$$

<u>In full duplex link</u>

$(2100 + 2100)$ packets could be sent

2100 → data packets

2100 → acknowledgements

But in stop & wait we send only 1 packet & wait for ack.

'1' instead of '4200'

$$\eta = \frac{1}{4200}$$

1 → 1 packet sent

4200 → 4200 could have been sent

Stop & wait fails miserably if capacity is high

$$\frac{Cap(\text{bits})}{\text{length}} = \text{no. of packet.}$$

$$\boxed{\begin{array}{l} Cap \uparrow \\ \eta \downarrow \end{array}}$$

$$\eta = \frac{1}{1 + 2 * a} = \frac{1}{1 + 2 * \frac{d}{v} * \frac{B}{L}}$$

2 → full duplex

$\frac{d}{v}$ → prop. delay

B → capacity

L → length of packet

Note: If capacity is high then in stop & wait, η is less.

After 1 packet, $\left(2 * \frac{d}{v} * \frac{B}{L}\right)$ packets could have been sent in the waiting time but only '1' packet, we are sending

Q If $T_p = 1.5$ millisecond and $T_t = 1$ milli sec, then how many packets can be transmitted before waiting for the acknowledgement of first packet.

A

$$= 1 + 2a$$

$$= 1 + 2\left(\frac{1.5}{1}\right)$$

$$= 1 + 3 = \underline{4 \text{ packets}}$$

No of packets that could be sent:

1 packet $\longrightarrow$ T_t sec.

T_t sec $\longrightarrow$ 1 packet

1 sec $\longrightarrow$ $\frac{1}{T_t}$ packets

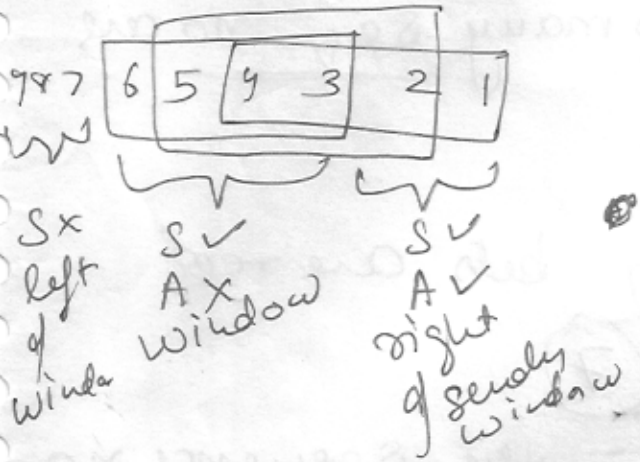
$(T_t + 2 * T_p)$ sec $\longrightarrow$ $\frac{T_t + 2 * T_p}{T_t} = \boxed{1 + 2a}$

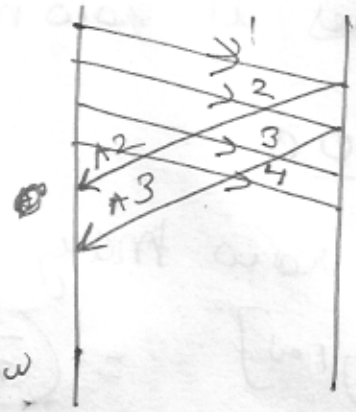
Tot time of Stop & wait

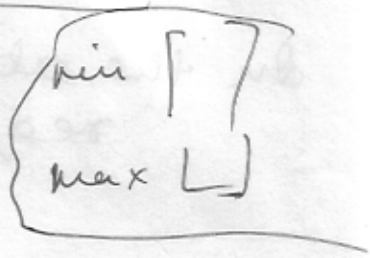
These many packets could have been sent before rec ack of 1st.

Pipeline: In order to ↑ the η, we use a technique called pipelining in which instead of sending 1 packet & waiting for ack, we send a window and wait for the ack.

This window slides after receiving a acknowledgement, so it is called sliding window protocol







Max Sender window size in a sliding windo

$$W_S = \min(1+2a,\ 2^n)$$

$1+2a$ → capacity of LFS; 2^n → number allowed

n = no of bits in Seq no field

$W_S = 1+2a$ → packets that could be send

no of bit reqd $= \lceil \log_2(1+2a) \rceil$

Min no of bits reqd $= \lceil \log_2(1+2a) \rceil$

Max no of bits reqd $= \infty$

Q) If $T_p = 49.5$ millisec and $T_t = 1$ millisec. then what is max sender window size

A) $1 + 2\left(\frac{49.5}{1}\right)$

$= 1 + 99 = 100$

In the above ques if how many seq. no are reqd

Ans = 100

In the above how many bits are reqd

$= \lceil \log 100 \rceil = 7$

Imp

If we have only 5 bits in the sequence no field then what is the sender window size

$= 2^5 = 32$

$\min(100, 32) = 32$

What is the efficiency?

$= \frac{32}{100}$ → packets sending / packets could be sent.

We are not using the capacity completely

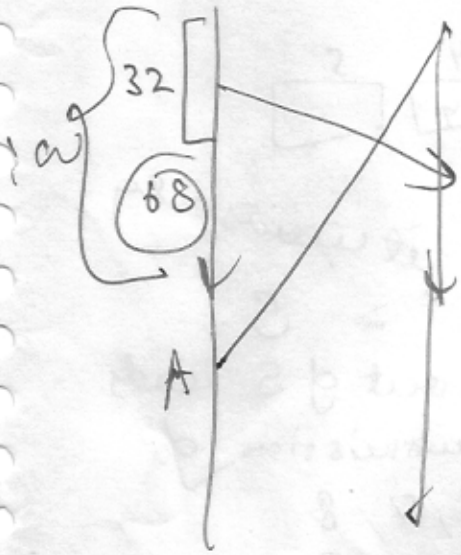
If B = 100 Mbps, then we are using only 32 Mbps.

So, $\eta = \frac{W_S}{1+2a}$ → possibility of sending less packets than $1+2a$ is due to the reason that we can have less no of bits in seq. field

Sliding window protocol

min of $1+2a$, 2^n.

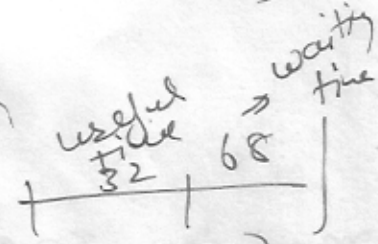
(Effective BW or BW utilised or Throughput) Sliding window $= \frac{W_S}{1+2a} * B = \eta * B$



After sending 32, Sender will wait for 68 more sec for the ack of 1st.

Before ack of 1st, it could have send 100 but sending only 32 so,

$\boxed{\text{Waiting Time} = 68}$



$\eta = \frac{32}{100}$ inefficiency $= \frac{68}{100}$

Sliding window protocol is theoretical concept which talks only about flow control. In practice, we implement it using Go Back N (GBN) and Selective Repeat (SR)

Both have sliding window

Go Back N

① In GBN, N indicates sender window size. If $T_p = 19.5$, $T_t = 1$. what is η of GB10?

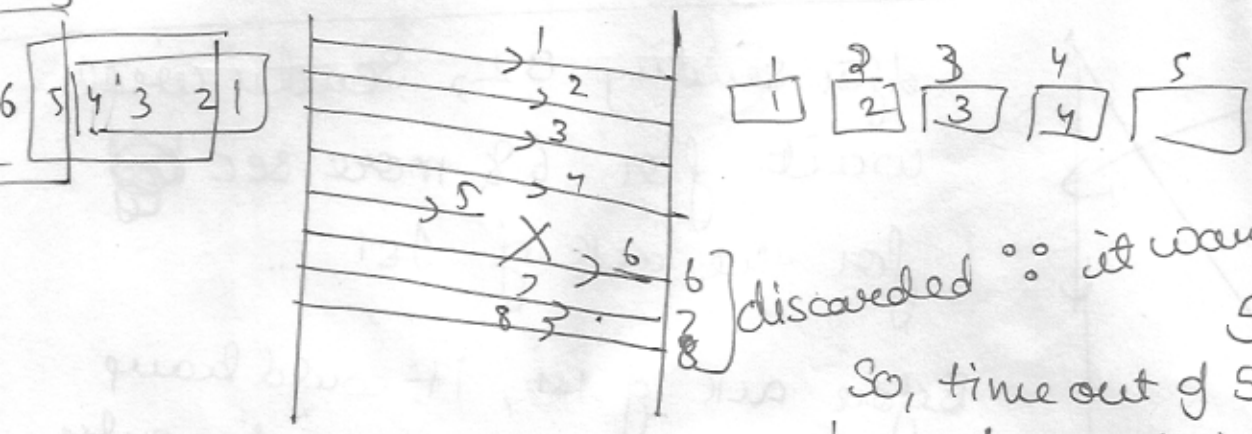
$$\frac{10}{1+390} = \frac{10}{40} = 0.25 \quad 25\%$$

② Receiver window size in GBN is always $\boxed{1}$. So, Receiver will always wait for In order packets and any out of order packets will be

discarded. ∴ a single packet loss will lead to a time out at sender side and this will lead to

Entire window retransmission.

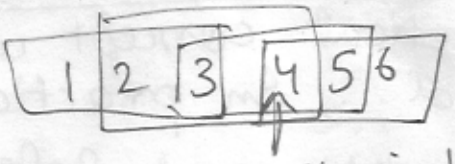
go back (4) - ans



So, time out of 5 leads to retransmission of 5, 6, 7, 8

↓

we don't have to wait for time outs of 6, 7, 8 ∵ After receiving TO of 5, 6, 7, 8, TO is already understood.



If 4th is lost then go back from 6

If sender window size is '3' & every 5th packet is lost then if 10 packets are send in GBN. How many trans are reqd to be?

A

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

↑ ↑ ↑

Ans : 18

Q $w_s = 3$

every 4th packet is lost

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

8 9 10 9 10

(27) transmission

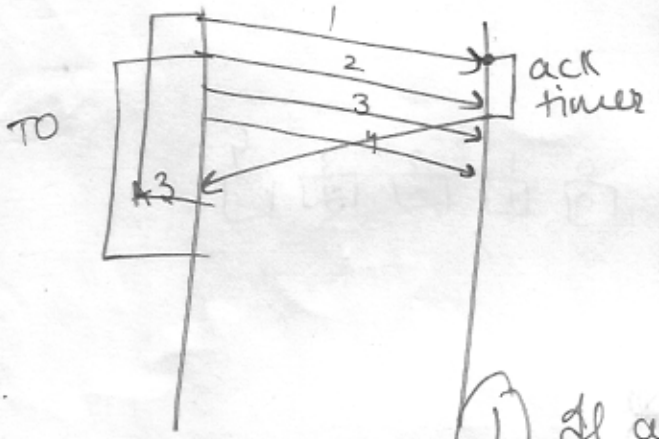
$w_s = 4$

every 6th packet is lost

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

(17) transmissions

(3) GBN uses Cummulative acknowledgement

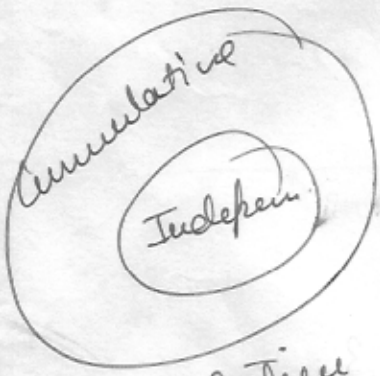


Sender Timeout period > ack timer

(1) If ack timer ↑, more retransmission

(2) If ack time ↓, more no. of acknowledgments

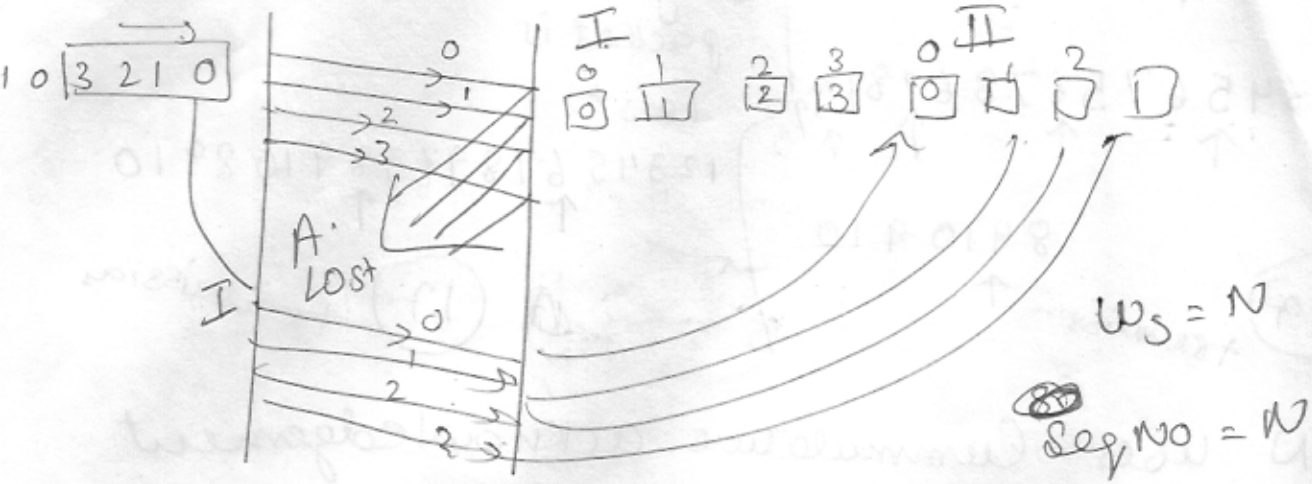
(3) If ACK timer = 0, independent acknowledgm



Cummulative becomes Independent when ack time = 0

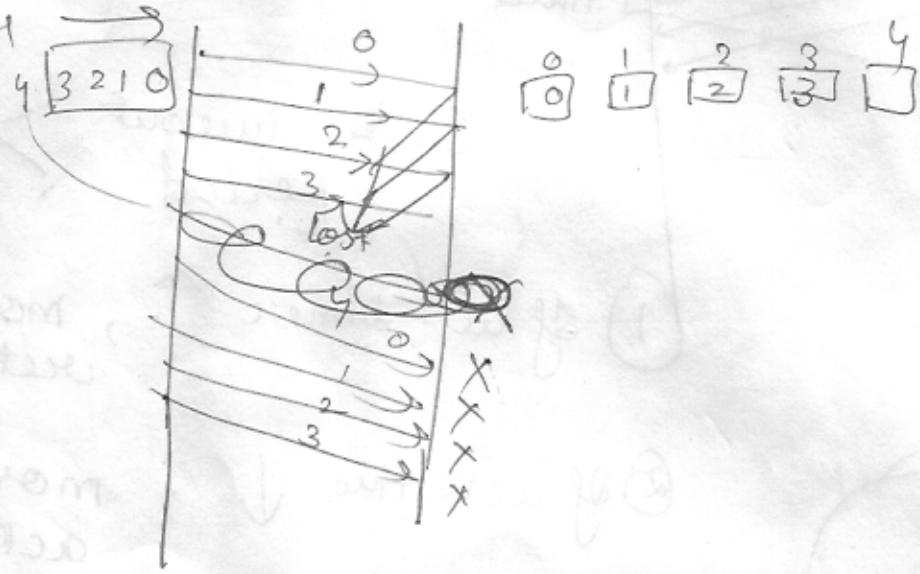
- ack timer starts as soon as a new packet is received
- Every Independent ack is Cummulative ack with ack timer = 0

Relation b/w window sizes & sequence numbers



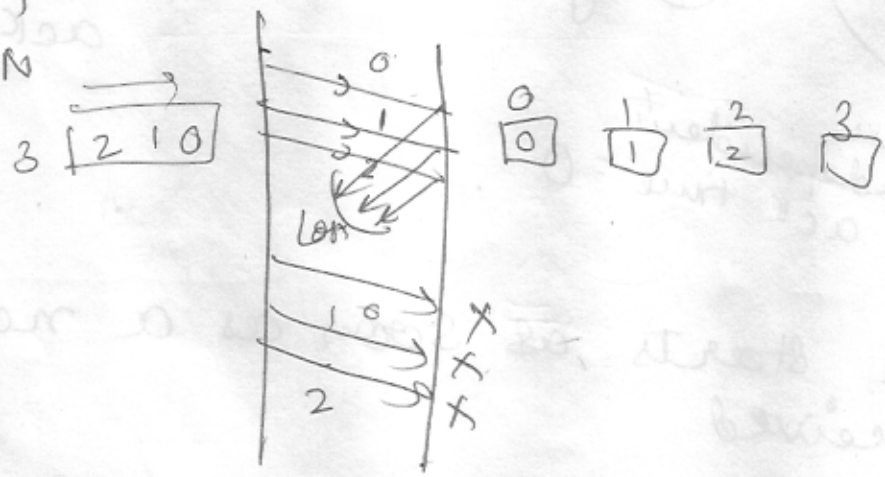
$W_S = N$

Seq No = N+1



$W_S = N-1$

Seq = N



If $w_s = N$ then 'N' seq nos are not sufficient.

so 2 ways to solve the prob

1. Inc. Seq nos by 1
2. Dec. window size by 1

No of seq no needed = Sender size window + Rec. size window

(I) + (J)

<u>Stop & wait</u>

Seq no reqd. = 1+1 = 2

No of bits reqd = 1.

<u>Sliding & wait (GBN)</u>

Seq no reqd = N+1.

I J

GBN :

<u>WS</u>

max = N-1 (seq no)

min = 1

<u>Seq No</u>

<u>max</u> ∞

<u>min</u> N+1 (+ ws)

---

① If sender window size is 'N' in GBN. How many <u>min</u> seq. no. are reqd.

$\boxed{N+1}$ $\boxed{\text{max } \infty}$

min bits = $\lceil \log_2 N+1 \rceil$

② If max available seq no are 'N' then what is the maxm sender window size in GBN

Ans $\boxed{N-1}$ $\boxed{\text{min} = 1}$

③ If maxm no of bits available in seq no field is 'K' then what is the <u>maxm</u> window size.

$\boxed{2^k - 1}$

Condition for Error Control in any sliding window protocol

$$\boxed{W_{Sender} + W_{Receiver} \leq \text{Available seq. Nos.}}$$

If seq no = 4

w_S	w_R	
3	1	GBN
2	1	GBN
1	1	S&W

If seq no = 8

w_S	w_R	
7	1	GBN
6	1	
5	1	
4	1	
3	1	
2	1	
1	1	S&W

In GBN, a packet lost leads to Time out at sender's side & entire window gets re-transmitted.

Also,

In GBN, if a packet received is corrupted, then Receiver silently discards it & Time out happens at sender's side & entire window retransmission is done.

SELECTIVE REPEAT

① In SR, sender window size > 1

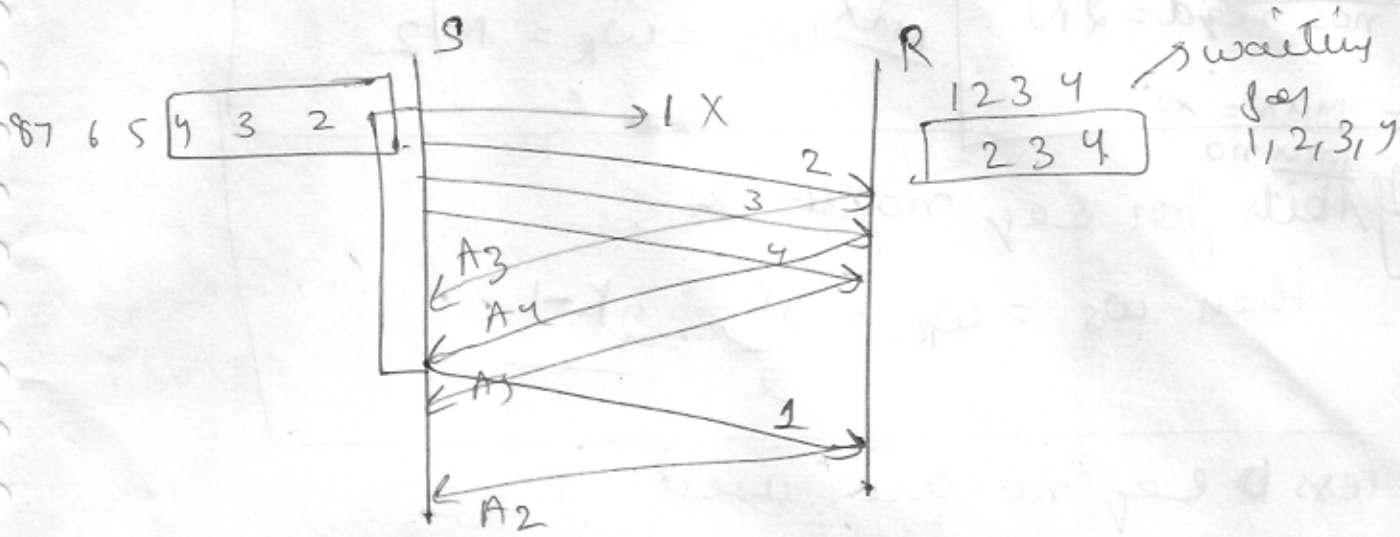
[1 in S&W]

② In SR, Rec window size = Sender Window size which is greater than 1.

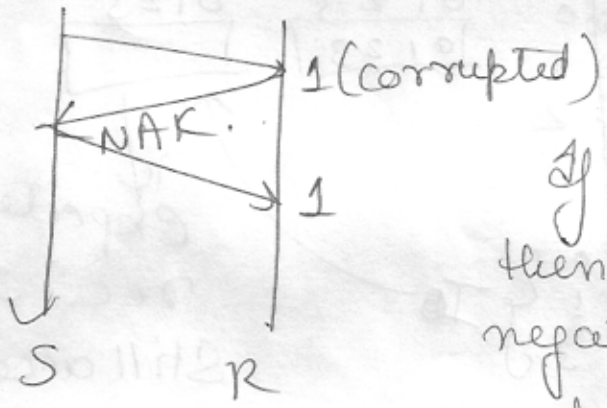
SR → gateways
GBN → routers

Since Rec window size > 1, a rec will rec out of order packets also ie if a packet is lost, then subsequent packets will not be discarded.

∴ a time out will lead to <u>single packet retransmission</u> and not the entire window



③ In SR, only <u>INDEPENDENT ACK</u> are used.

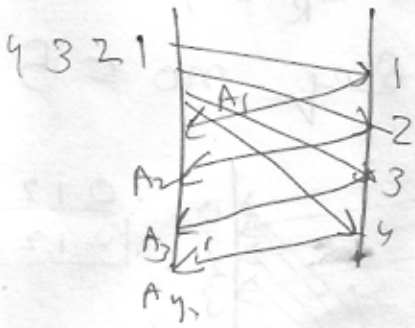


unlike GBN, In SR in case:

If a packet is corrupted then receiver will send negative acknowledgement and sender will re-transmit.

NAK leads to a faster re-transmission than Time out

used when NAK is lost



for 1 – A_1, 1 – A_2 → Theoritically packing

Relation b/w seq numbers & window sizes:

If $w_S = 4$, $w_R = 4$ then seq no reqd = 4 + 4 = 8 → minm. 12 → max

Both must have completely different numbers

$w_S + w_R \le$ Avail seq no.

If $W_S = N$
$W_R = N$
then minimum Seq nos reqd $= 2N$

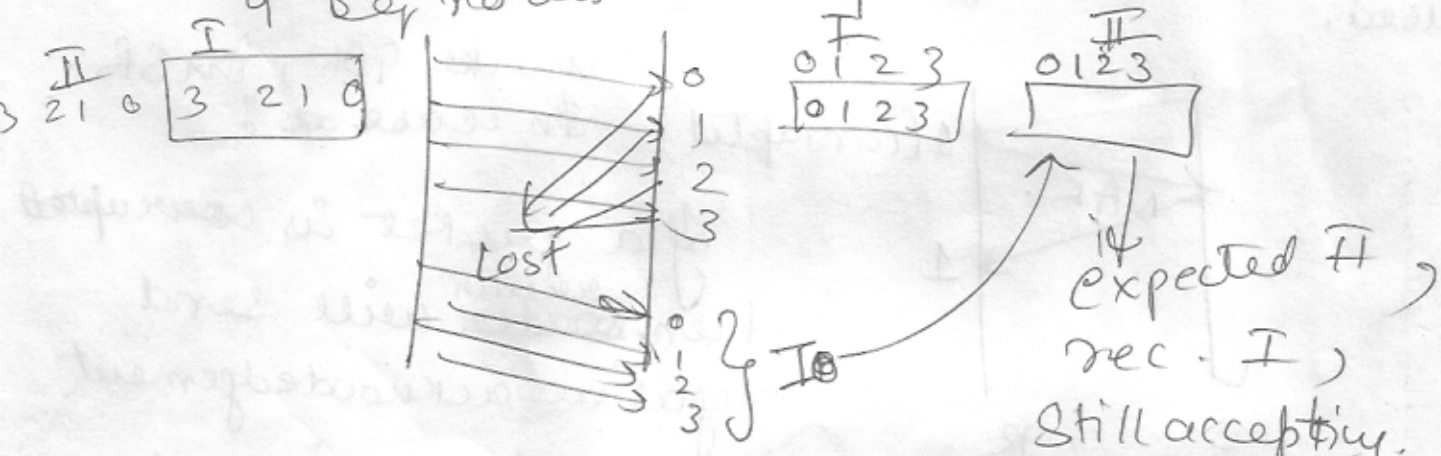
max $= \infty$

If max seq no $= N$
then
max $W_S = W_R = N/2$

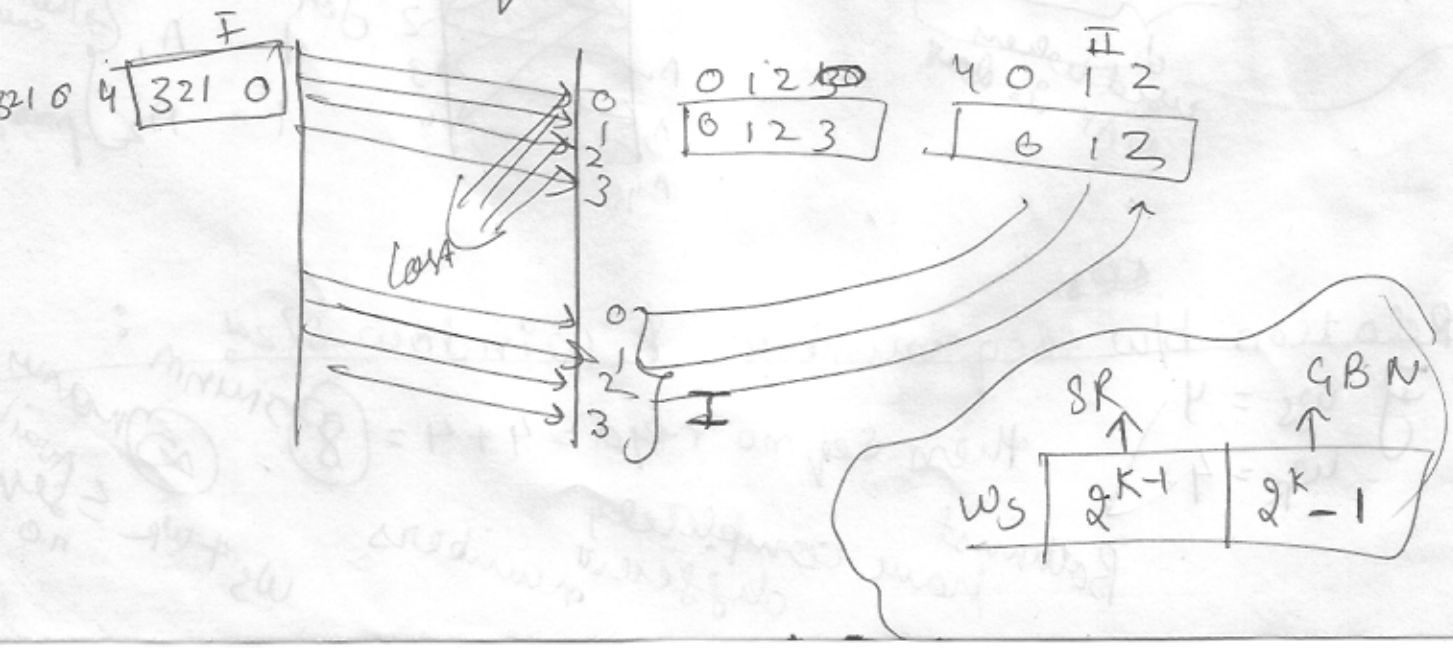
If max no bits for seq no field $= 'k'$
then $W_S = W_R = \frac{2^k}{2} = 2^{k-1}$

If less seq no are used

Assume $W_S = W_R = 4$
& Seq no also 4 then problem :-



Assume $W_S = W_R = 4$
& Seq. no = 5 then problem :



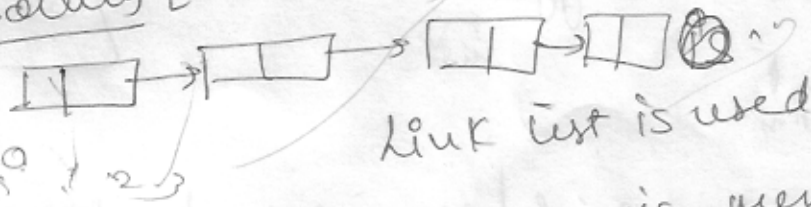
Round Trip time $= 2T_p$

$W_S = N$

	Stop & Wait	GBN	SR
Eff (η)	$1/(1+2a)$	$N/(1+2a)$	$N/(1+2a)$
Buffers	2	N+1	2N
Seq. no.	2	N+1	2N
Retransmission (if 1 is lost)	1	N	1
BW	low	high	moderate
Implementation	easy	moderate	complex

∴ SR ↓ searching ↓ S

R ✓ sorting [∵ Acceptance is done in random order]



Link list is used ∵

Dynamic memory allocation is there.

We do not know initially what is packet size & no of packets.

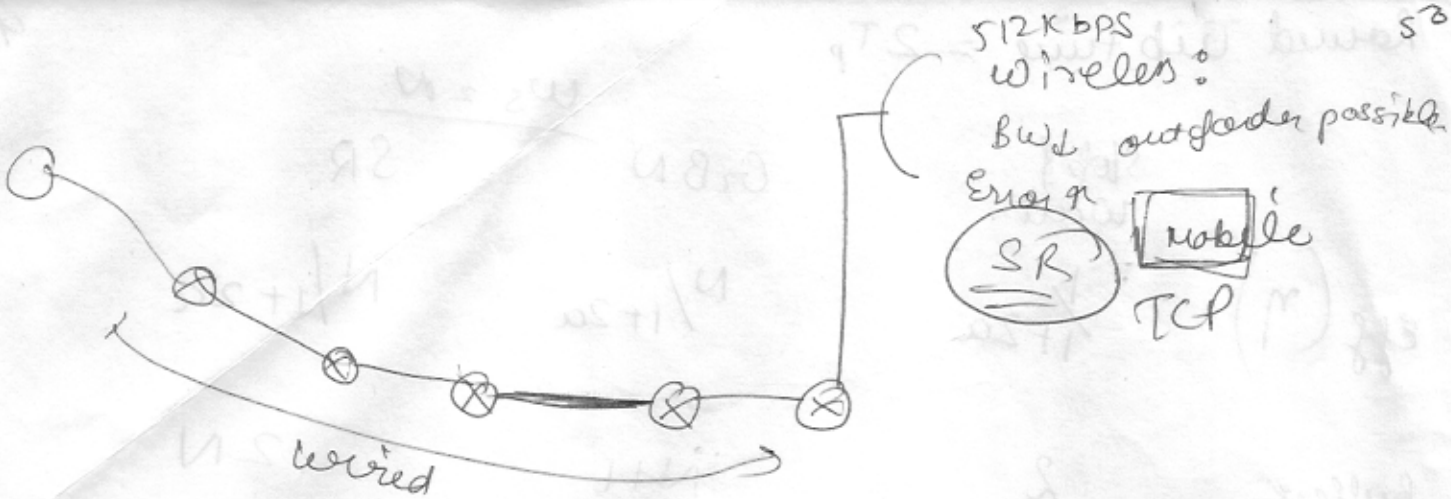
If BW ↓ Error ↑ (wireless) then SR is useful

GBN

NO sorting reqd. Only In order packets are accepted

NO searching reqd. Entire window is retransmitted

→ simpler to implement.



so, GBN.

HDLC : protocol used in Data link layer

TCP (SR + GBN)
75% 25%

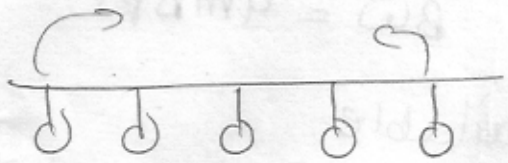
Q If BW is moderate, buffers are sufficient (High) & CPU is sufficient (High) then SR is better
(CPU → for sending & sorting)

Q If BW is high/sufficient, buffers are moderate & processors are moderate then GBN is better.

HDLC implements GBN and
TCP implements (SR + GBN) Comb

Note : • Link to Link protocols used at Data Link layers used as GBN
• End to End protocols used at Transport layer uses SR

Access Control



⇒ using same access line (same frequency) by all leads to COLLISIONS

To prevent collision :

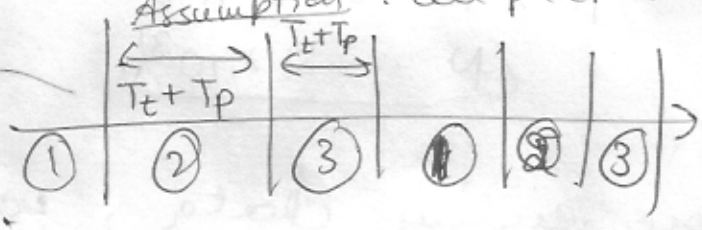
① FDM : frequency Division Multiplex

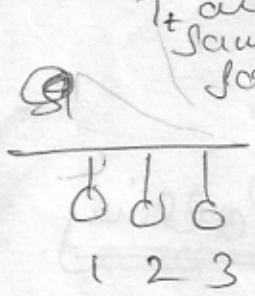
Not useful ∵ 100 diff frequencies mean only 100 stations.

② TDM : Time division Multiplexing

Divide timeline into slots and give one slot to each station in cyclic manner

Assumption : all packets are same size so T_t are same so/ar





Efficiency

$$\eta = \frac{T_t}{T_t + T_p} = \boxed{\frac{1}{1+a}}$$

This efficiency is under high load conditions

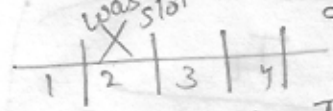
(or)

$$= \frac{\cancel{n} T_t}{\cancel{n}(T_t + T_p)} = \frac{1}{1+a}$$

every station is willing to send

Available BW or Eff BW or Throughput : $\eta * B$

Drawback of TDM : Reservation



Time slot is fixed & if a station is not willing to send then slot gets wasted.

TDMA questions Linked

(Q) If $T_p = 1$, $T_t = 1$, $\eta = ?$

A = 50%

(Q) If BW = 4 mbps

available BW = $\frac{50}{100} \times 4$

= 2 mbps

(Q) If every station needs 1 Kbps BW then How many stations can be connected

(A) $\frac{2 \times 10^3}{} = 2000$

∵ Available BW = 2 mbps

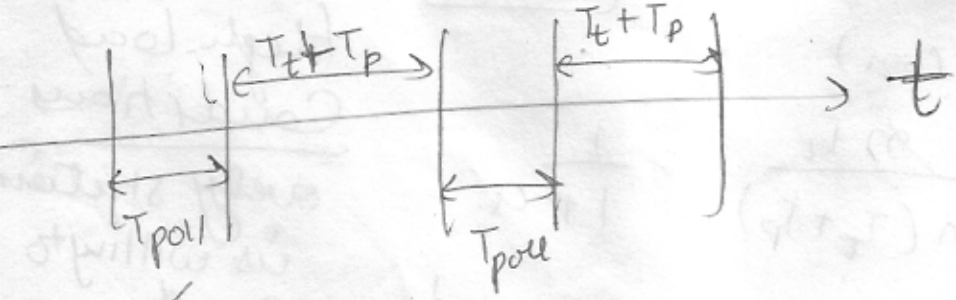
Suppose 'N' stations are connected then

$N \times 1\,Kbps = 2\,mbps$

$N \times 10^3 = 2 \times 10^6$

$N = 2 \times 10^3$

(3) Polling : Before sending data, polling algorithm is run to check who wants to send data & from them 1 is selected and given chance to send data.



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

Take T_p as 0 if not mentioned

Throughput = $\eta * B$

Q If $T_t = 1$ millisec , $T_{poll} = 1$ millisec

$\eta = ?$

(A) $\frac{1}{1+1} = 50\%$

Q If BW is 4mbps then

(A) BW utilization = 2mpbs

Q If each station req. 1Kbps then How many stations ?

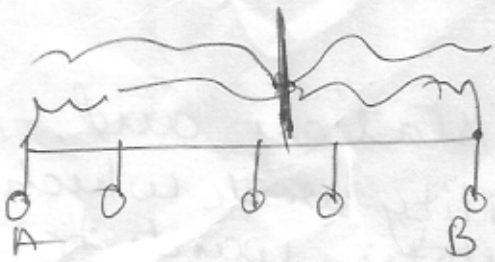
A $N \times 1Kbps = 2 \times 10^6$

$N = 2 \times 10^3 = 2000$

(4) CSMA/CD

senses, multiple access, Collision Detection

- First a station senses the link , if no signal then it transmits
- Since no ~~collisions~~ acknowledgments are there so collision needs to be detected by the sender itself so must keep transmitting till the ~~collided~~ collision msg reaches it back



$T_t \geq T_p$

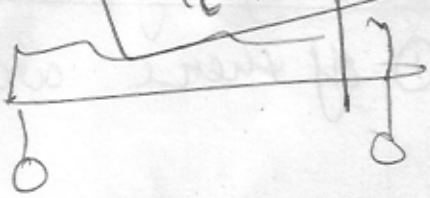
At 10:00 : Both A & B send

At 10:30 : collide

At 11:00 : Both A & B know that collision occ

WORST CAS

$T_t \geq 2T_p$



At 10:00 A sends

At 10:59 B sends

At 12:00 A knows abt core

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

$$\boxed{L_{min} \geq 2 * T_p * B}$$

Q If $B = 1\,Mbps$

$T_p = 1\,ms$

$L_{min} = ?$ in CSMA/CD

Ans

$L_{min} \geq 2 * 10^{-3} * 10^{6}$ bits

$\Rightarrow \underline{2 * 10^{3}}$ bits

a) 2000

b) 2K

Poisson : $\frac{1}{p}$

"p" : probability of success

↳ from previous History (Prior Estimation)

Binomial : ${}^{n}C_{1}\, p(1-p)^{n-1}$

$= np(1-p)^{n-1}$

Efficiency of CSMA/CD :

① If there are 'n' : no of stations and

'p' : probability with which every station wants to transmit the data

then Probability for successful transmit

$P(\text{successful transmission}) = n p (1-p)^{n-1}$ [only 1 transmits]

To find max P:

$$\frac{dP}{dp} = 0$$

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

$$P_{max} = n p (1-p)^{n-1}$$

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

$$P_{max} = n * \frac{1}{n} * \left(1 - \frac{1}{n}\right)^{n-1}$$

$$\underset{n \to \infty}{Lt} P_{max} = \underset{n \to \infty}{Lt} \left(1 - \frac{1}{n}\right)^{n-1}$$

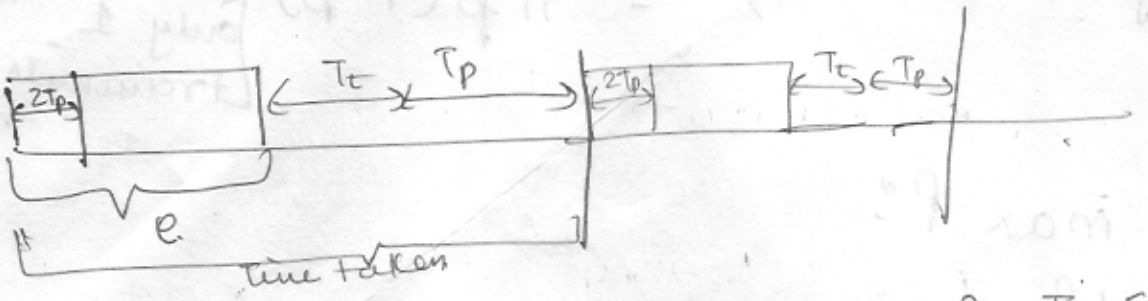
$$= \frac{1}{e}$$

So, $P_{max} = \frac{1}{e}$

~~So, Coll~~ If Probability for success is 'p' then no of tries reqd before 1st success is '$\frac{1}{p}$' which means

here $\frac{1}{P_{max}}$ = no of tries before 1st success = e.

'e' = #times we failed = ~~all cost~~ 'e' collision before 1st success.



Time wasted in each collision $= 2 * T_p =$ contention slot

$\therefore$ total time wasted in collisions $= 2 * e * T_p$.

$$\eta = \frac{T_t}{T_t + 2eT_p + T_p}$$

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

CSMA/CD

Sense - Send - Detect

Condition $\underbrace{T_t \geq T_p}_{\text{best case}}$ $\underbrace{T_t \geq 2T_p}_{\text{worst case}}$

$$\eta = \frac{T_t}{T_t + 2eT_p + T_p} = \frac{1}{1 + 6.44a}$$

P (successful transmission) $= np(1-p)^{n-1}$

$p = \frac{1}{n}$ → that only one wants to

P

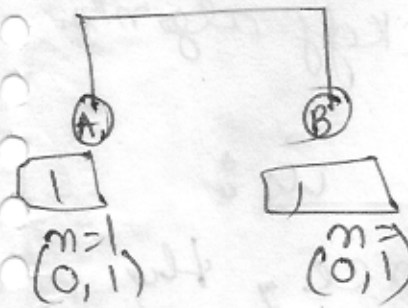
Binary BACK OFF ALGORITHM

This algorithm gives waiting time of nodes involved in collision.

$\boxed{\text{Waiting time (W)} = K * T_{slot}}$ where $\boxed{K \in (0, 2^n - 1)}$

and 'n' = collision number for that packet.

ex



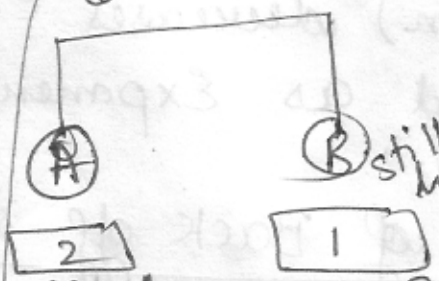
4 comb

0 0 — Collision
0 1 — A sends
1 0 — B send
1 1 — collision

$p(C) = \frac{2}{4} = 50\%$

$p(A) = \frac{1}{4}$ → Assume it sends

$p(B) = \frac{1}{4}$



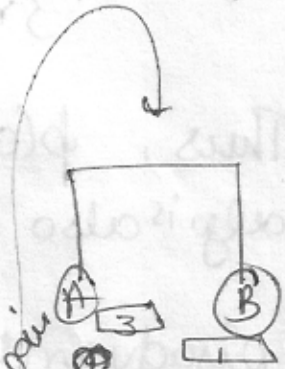
8 comb

0 0 — Coll
0 1 — A
0 2 — A
0 3 — A
1 0 — B
1 1 — Coll
1 2 — A
1 3 — A

$p(A) = \frac{5}{8}$

$p(B) = \frac{1}{8}$

$p(C) = \frac{2}{8} = 25\%$



Assume A sends again

16 comb

$p(C) = 9/16$

$p(A) = 13/16$

$p(B) = 1/16$

Also called as "Exponential" algorithm

Before applying algorithm,

$p(\text{collision}) = 100\%$

After 1st collision, $p(\text{collision}) = 50\%$
After 2nd " $p(\text{"}) = 25\%$
3rd $p(C) = 12.5\%$

Thus, p(collision) decreases exponentially, so algo is also called as Exponential backoff algorithm.

Disadvantage of Back off Algorithm is:

Capture Effect = If A wins a race, the probability of A winning the subsequent races will increase exponentially.

Every node will try for a maximum of 10 times and gives up the race

Disadvantage of CSMA/CD

$T_t = 2T_p$ $T_t = 2 \times T_p \times B$

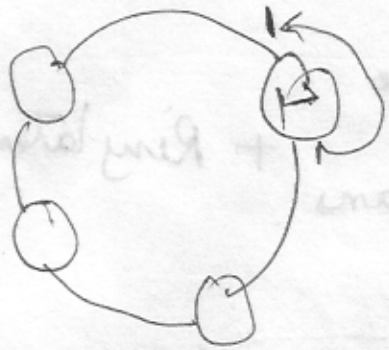
We have a limitation on the minm size of the packet and collision may lead to Capture effect ∴ Token passing is proposed

TOKEN PASSING

secs * BW = bits delay in

sec * velocity = meter

① FDM
② TDM $\frac{1}{Tta}$
③ Polling $\frac{T_t}{T_t+T_p+T_d}$
④ CSMA/CD T_t (min)
⑤ Backoff algo $T_t + 2eT_p + T_p$ capture
⑥ Token passing



n stations
d = distance
v = velocity

K bit delay at every station to store & forward

Total time for 1 bit to take 1 revolution $= \frac{d}{v} + \frac{n * K}{BW}$

<u>Ring Latency</u> : It is the time taken by a bit to start from one point in the ring & return back to the same point.

If d : length of wire
n : total no of stations
k bits : delay introduced at each station

then <u>ring latency</u> is

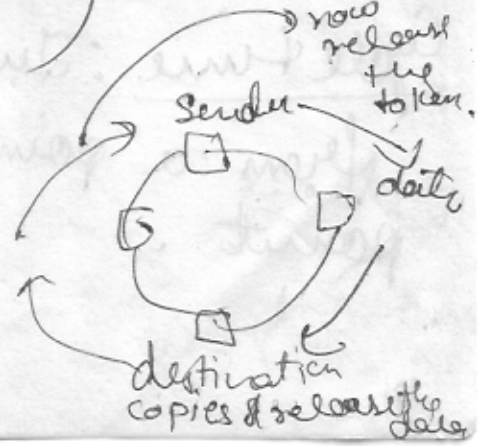
$$\left(\frac{d}{v} + \frac{n * K}{BW}\right) \text{ seconds}$$

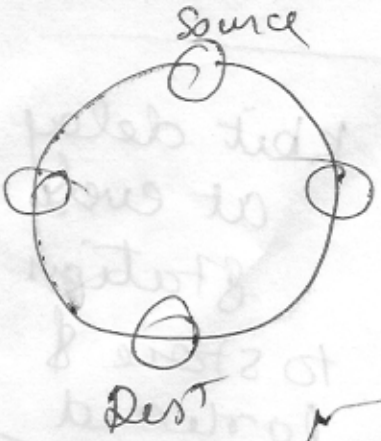
(or)

$$\left(\frac{d}{v} * BW + n * K\right) \text{ bits}$$

$\frac{d}{v} + \frac{nK}{BW}$ seconds

Token Ring : No acknowledgements





Token Holding Time

$$THT = T_{trans} + \text{Ring latency}$$

$$\boxed{\text{Cycle time} = \frac{d}{V} + n(THT)}$$

RL

$THT = RL + T_t$

$CT = \frac{d}{V} + n\,THT$

for transferring 'n' packets.

Useful time for transferring 'n' packets $= nT_{trans}$

$$\text{Efficiency} = \frac{nT_t}{T_p + n(T_t + T_p)} \quad \left(\frac{nT_{trans}}{T_p + n\left(T_t + T_p + \frac{nK}{B}\right)} \right) \text{Take } K = 0.$$

DELAYED TOKEN REINSERTION

In this strategy a station will hold the token and then transmits the data and waits for the data to come back.

$$THT = T_{trans} + RL$$

Cycle time: Time taken by the token to start from a point and returns back to the same point.

$$CT = \frac{d}{V} + n(THT)$$

$$= T_{prop} + n\left(T_{trans} + T_{prop} + \frac{nK}{B}\right)$$

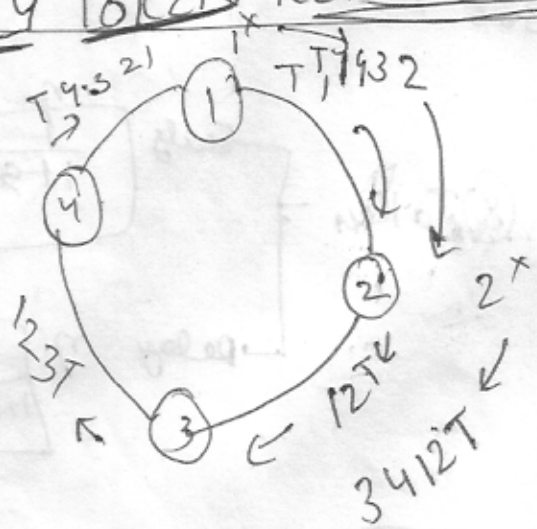
In this total cycle time, the token is seen by n stations & every station will transmit one packet ∴ n packets are totally transmitted in 1 cycle time.

$$\eta = \frac{\text{useful time}}{\text{Total time}}$$

$$= \frac{n * T_T}{T_p + n(T_T + T_p)} = \frac{1}{1 + \left(\frac{n+1}{n}\right)a}$$

where $a = \frac{T_p}{T_p}$

EARLY TOKEN REINSERTION



THT : T_{trans}

cycle time : $\frac{d}{v} + n(THT)$

$= T_{prop} + n T_{trans}$

$$\eta = \frac{n T_{trans}}{T_{prop} + n T_{trans}} = \frac{1}{1 + a/n}$$

$a + \frac{a}{n}$

η increased here.

In this strategy, a station will hold the token, release the data & release the token immediately so it is

called early token reinsertion.

Note: Early token reinsertion is used under heavy load conditions ∵ it has high efficiency.

η

① S&W → $\frac{1}{1+2a}$

② pipelining ⇒ $\frac{N}{1+2a}$ (GBN, SR)

③ Aloha → $G * e^{-2G}$ max $\eta = 0.184$

④ Slotted Aloha ⇒ $G * e^{-G}$ max $\eta = 0.368$

⑤ FDM → $\frac{1}{N(1+a)}$

⑥ Polling = $\frac{T_{trans}}{T_{trans} + T_{prop} + T_{poll}}$

⑦ Ethernet = $\frac{1}{1+644a}$

⑧ Tokens → Early: η $\frac{1}{1+a/n}$

→ Delay: η $\frac{1}{1+a+a/n}$

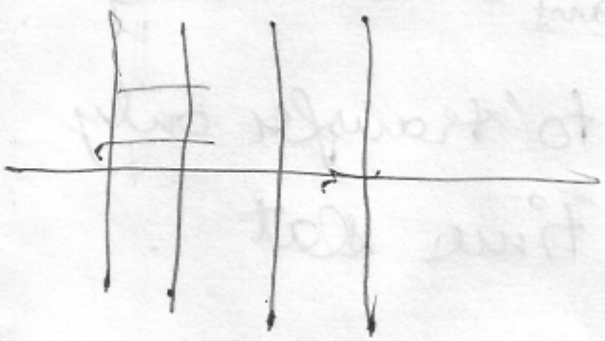
Aloha → Anyone can send data w/o sensing the carrier & ack are there so, no need to detect collisions.

Aloha & Token ring ∵ obsolete methods

Vulnerable time: $2T_{trans}$ & $\eta = G * e^{-2G}$.

$\eta = 0.184$

Slotted Aloha :



Vulnerable time = T_{trans} *1

$\eta = 0.368$

Pure Aloha : In this method, any station could transmit data at any time w/o sensing the carrier.

Collisions are possible.

Acknowledgements are used so no collison detection is needed.

After collision, stations will wait for a random amount of time called Back off Time.

Vulnerable Time in pure aloha is = $2T_{trans}$

Efficiency $(\eta) = G * e^{-2G}$ when G : No of stations that will transmit in a time slot 'T trans'.

for maxm value of η

$$\frac{d\eta}{dG} = 0$$

$G \times e^{-2G}$

$$\Rightarrow G = \frac{1}{2}$$

$\frac{1}{2}$ station → 1 T_{trans}

1 station → 2 T_{trans} ⇒ max η no collision

When $G = \frac{1}{2}$, we get maxm efficiency of 0.184.

Slotted Aloha : In this protocol, time is divided into slots of T_{trans} each and every station is forced to transfer only at the beginning of a time slot.

Vulnerable time : T_{trans}

∴ efficiency $(\eta) = G * e^{-G}$

for max efficiency $= \frac{d\eta}{dG} = 0$

$\Rightarrow G = 1$.

Max $\eta = 0.368$

Error Handling

- Error Detection
  - → Sending copy
  - → checksum
  - → CRC
  - → Parity.
- Error Correction
  - → Hamming

CRC

If CRC generator is 4 bits then CRC is 3 bits (4-1=3)

M: 10110

CRC G = 1101

```
      1
1101 | 10110 0000
       1101
       ----
        00

        111
1101 | 10110 00
       1101
       ----
       01100
        1101
        ----
         1100
         1101
         ----
          010
```

→ CRC.

Sender will send =

1011010100

↳ Append CRC

At receiver's end :

```
1101 | 1011010100 |
       1101
       ----
       1100
       1101
       ----
        1101
        1101
        ----
        000000
```

→ all zeros. | error free

corrupted bit

If receiver, received 1010010100

```
1101 | 1010010100 |
       1101
       ----
       1110
       1101
       ----
        1110
        1101
        ----
         11100
         1101
         -----
         001100
```

→ not all zeros (error exist).

CRC generator is also called CRC generator polynomial.

ex $x^4 + x^2 + x + 1$ → CRC polynomial

$\Downarrow$

$1 \cdot x^4 + 0 \cdot x^3 + 1 \cdot x^2 + 1 \cdot x + 1 \cdot x^0$

CRC code ⟸ 1 0 1 1 1

ENCODING

Manchester Encoding : In this :

⌉ = '1' ⌋ = "0"

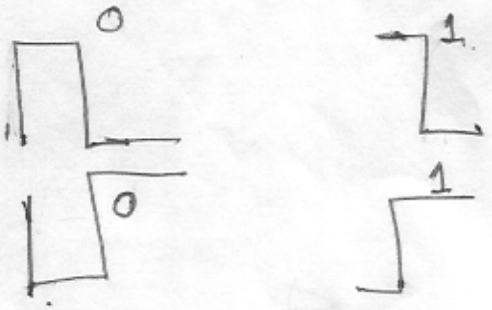
Baud Rate : No of voltage changes per second.

for 1 bit = 2.

for 10 bits = 20

Baud Rate (manchester Encoding) = 2 * Bit rate

Differential Manchester Encoding



In this also

Baud rate = 2 * Bit rate

1 (all 1s) = J

0 (all 0s) = K

Invalid

Shanon's formula

$$BW = 2 * f * \log_2 V$$

V = # of voltage levels used.

S/w in CN

- Mandatory
  1. Error Control
  2. Flow Control
  3. Multiplexing & Demultiplexing
- Optional
  1. Routing
  2. Encryption
  3. Checkpointing

RA → design (ISO-OSI model) → Implement → Test

Different Models

1. ISO-OSI
2. TCP-IP
3. ATM
4. IEEE
   - 802.1
   - 802.3
   - .5
   - ⋮
   - 11

(3) ATM, (4) 802.1, 802.3 — Not in syllabus

S/W in Computer Networks:

Responsibilities:

Mandatory

1. EC
2. FC
3. Mult. & Demul
   M to 1 — 1 to M
4. Addressing

⋮

(70)

Optional

Routing
Encryption
Checkpoints (in Torrents)

⋮

Nearly 70 functions are req in CN.

There are several Reference Models which talk about these functions.

ex ISO-OSI (7 layers), TCP/IP, ATM, IEEE etc

ISO OSI

- App L → user Interface
- Presentation layer → optional
- Session L → Thin
- T L → Thick
- NL → complicated
- DLL → H/w+S/w
- PL → Pure H/w

User/Application Oriented Interface

N/w oriented

ISO-OSI is ideal model.

TCP/IP contains only 5 layers: (Most practical model: many of the functions which are not reqd are removed.)

AL → (Message)
TL → TCP → (Segment)
NL → IP → (Datagram)
DLL → (frame)
PL → PDU

packet

I

Physical Layer: Responsible for electrical, mechanical, functional & procedural characteristics of the physical Links.

- It deals with types of transmission modes:

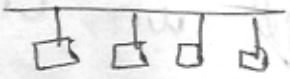
TV a) SIMPLEX : Data can travel only in 1 direction.

walkie talkie b) HALF DUPLEX : Data can travel in both directions, but one at a time.

modems c) FULL DUPLEX : Data travels in both directions at the same time.

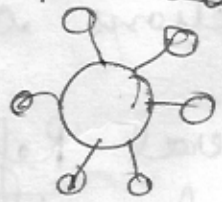
Physical Layer deals with Topology of the LAN.

a) Bus

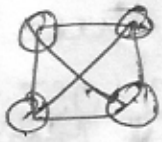


b) Star

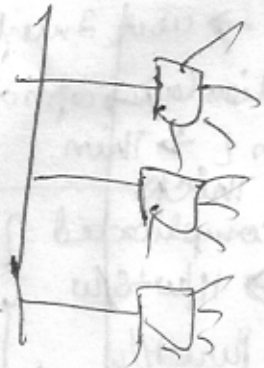
c) Ring



d) Mesh



e) Tree/Hybrid/Backbone



• Type of Physical Medium is also the responsibility of Physical layer.

a) Cu wire → Electrical signal

b) Optical fibre → Light

c) wireless → Electromagnetic waves.

• Encoding is also taken care by physical layer.

ex a) Manchester

b) Differential Manchester

c) Integral Manchester etc.

Note: Ethernet uses Manchester encoding

Token Ring uses Differential Manchester Encoding

II Data Link Layer

Responsibilities:

(i) Flow Control → (SW, GBN, SR) most popular.

(ii) framing

(iii) Error Control → (checksum, CRC)

(iv) Physical Addressing

(v) Access Control → (CSMA/CD, Aloha, Token passing, TDM, polling)

Framing: framing is of 2 types

FL, VL → length, ED

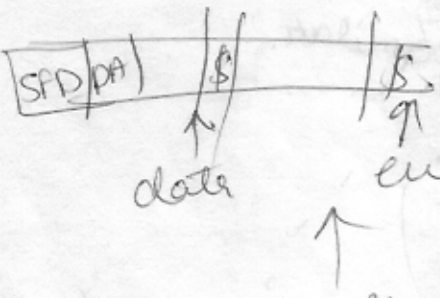
i) fixed length (ii) Variable length

→ In fixed length framing, all the frames should be of same size ∴ Padding is reqd if we have to send data of smaller size.

→ Variable length:

We can have length field to represent length. If this field gets corrupted then we may end up reading more/less bytes than required.

* we can have End Delimiter. In this the disadvantage is that whenever data matches with end delimiter, receiver may think that it is the end of the data. ∴ we need stuffing.



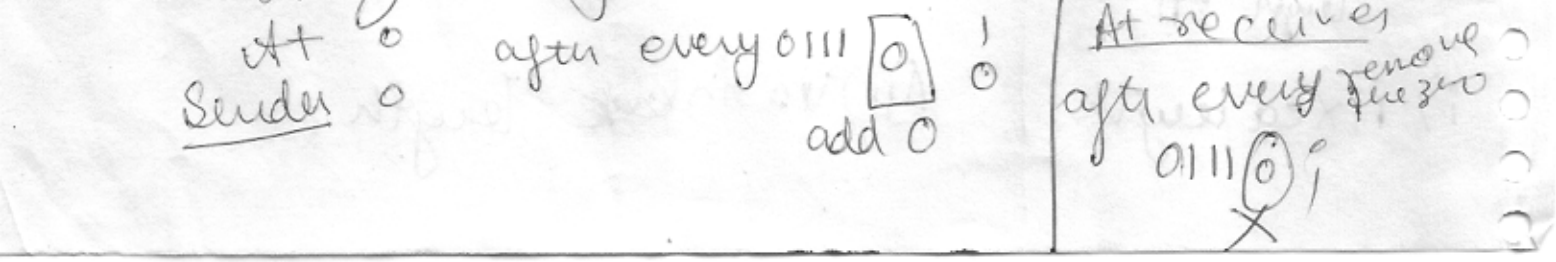
Stuffing is of 2 types: Char, Bit

Character Stuffing: In this, a spcl character is used to represent end of the frame. whenever data matches with this spcl character, we add an escape character to distinguish the data. It is obsolete

Bit Stuffing: If the end delimiter is 01111

At sender: after every 0111 add 0

At receiver: after every 0111 remove zero



End delimiter = 01111 1

At Sender: 01111 [0] 1 0 (Insert) 01111 [0] 1 0 (X)

End delimiter 10000

At Sender: 1000 [1] 0 1 (Insert) At Receiver: 1000 [1] 0 1 (X remove)

<u>Physical Addressing</u> : Any number which is used to identify a <u>host</u> uniquely in a <u>LAN</u> is called <u>physical address</u>.

<u>Logical Address</u> : Any number which is used to identify a <u>host</u> uniquely within the entire <u>world</u> is called <u>logical address</u>.

<u>MAC address</u> : It is a <u>48 bit</u> number which is present on <u>ROM</u> of <u>NIC</u> (N/w Interface card) (H/w number)

It is provided by <u>Vendor</u>

1st	2nd	3rd	4th	5th	6th
Vendor ID		Date		Serial NO.	

unique throughout the world

6 Bytes (48 bits)

<u>MAC</u> address is <u>permanent address</u>

<u>IP address</u> : It is a <u>S/w number</u> (RAM) which is globally unique

• Can MAC address be used as Physical address?

→ Yes

• Can MAC address be used as logical address?

→ Yes, but IP at Network layer does not use MAC as logical address but uses IP addr as logical address ∵

IP addr is → Hierarchical
→ easy routing [gives the LAN identification for]

• Can IP be used as physical address?

→ Yes, ~~but not used~~

MAC can be used as logical addr but not used ∵ routing tables will become very large & MAC cannot distinguish geographically

LAN TECHNOLOGIES

(IEEE 802.3)

1) Ethernet : • It uses BUS Topology.

• The access control used is CSMA/CD

• There are no acknowledgements

• Data Rates in Ethernet are 10 Mbps, 100 Mbps & 1 Gbps (fast ethernet)

• Manchester Encoding is used

• Collisions are possible in ethernet

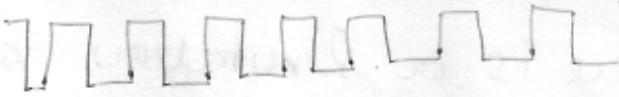
Frame format :

7 Bytes	1 Byte	6B	6B	2B		4 Bytes
Preamble	SFD (Start frame Delimiter)	DA (Destn addr)	SA (Source addr)	Length	Data (1500B Max, Min: 46 B)	CRC

Ethernet uses MAC address

← added by Physical layer → PL Header

Preamble is 7 Bytes of '10101010 .... so on'



Valid data never contains 7 bytes of 10

SFD is 1 Byte → 10101011 → Last chance for clock synchronization

Both are used to indicate the start of frame & also used for clock synchronization.

$\underbrace{101010}_{7B} + \underbrace{10101011}_{1B}$

Both are Constant [so, added by physical layer]

DA & SA : These are physical addresses & MAC addresses are used as physical addr in Ethernet.

Length : It indicates size of the frame.

CRC : If CRC generator is 'N' bits then CRC will be 'N-1' bits.

Here CRC is 32 bits so, CRC generator is 33 bits

	max	min
Data	1500 B	46 B
frame excluding 12 header	1518 B	64 B

This minimum size of frame limitation is there so that collision Detection can be done

Calculated by $L \geq 2 * \frac{d}{v} * B$

In order to avoid Monopolisation {one client very large} there is a limitation on Maxm size of data that can be send in a Ethernet frame.

CRC is added at the end of the frame ∵ CRC is calculated as the data is being transmitted onto the link ∴ CRC is available only after entire data is transmitted.

Disadvantages of Ethernet (802.3) :

① It is not applicable for Real Time applications ∵ of collisons

② It is not applicable for Interactive applications (ATM, chatting) ∵ of Minm pack size limitation.

③ It is not applicable for Client/Server application ∵ they are no priorities

IEEE 802.4 - Token Bus

TOKEN RING (IEEE 802.5)

① Encoding used is Differential Manchester

② Data Rates in Token Ring are 4 Mbps & 16 Mbps

③ Access control method used is TOKEN passing ∴ Collisions are not possible.

∴ Token Ring can be used for Real time app & Interactive app [No collison → No detection ↓ No limitation on minm packet size]

- There are Priorities in Token Ring ∴ it can be used for Client / Server Application.
- There are no acknowledgments in Token Ring.

copies & releases data itself ∴ data could be a Broadcast data & others might also need it.

Ethernet & Token ring do not require encryption ∴ all are Trusted parties → replacing with

Problems in Token Ring:

(1) Source Problem : (a) ORPHAN PACKET : when source is down after transmitting a packet, the packet will revolve in the ring forever

(b) Corrupted Packet : whenever packet gets Corrupted, the source will not recognize it

(2) Token Problem

(a) Captured : A token may never be released by a source.

(b) Lost or Corrupted ξ

③ **Destination Problem**

(a) Unreachable / Down

(b) Busy

④ **LINK / RING PROBLEM** : Cut in the ring.

~~SOLUT~~

SOLUTIONS

Source: a & b : On every packet, we have a special bit called "Monitor Bit (M)" which is used for stamping.

Initially M=0, it will be changed to "1" when a monitor sees it.

If a packet arrives with M=1 at a monitor, it indicates, it is a orphan packet.

② Token : a & b : Monitor will maintain minimum Token return Time (TRT) & maxm TRT after releasing a token. If token does not return within this interval then Monitor will regenerate the token.

Minm TRT : Ring Latency (No computer has data to send)

Maxm TRT : Cycle Time (Every comp holds token)

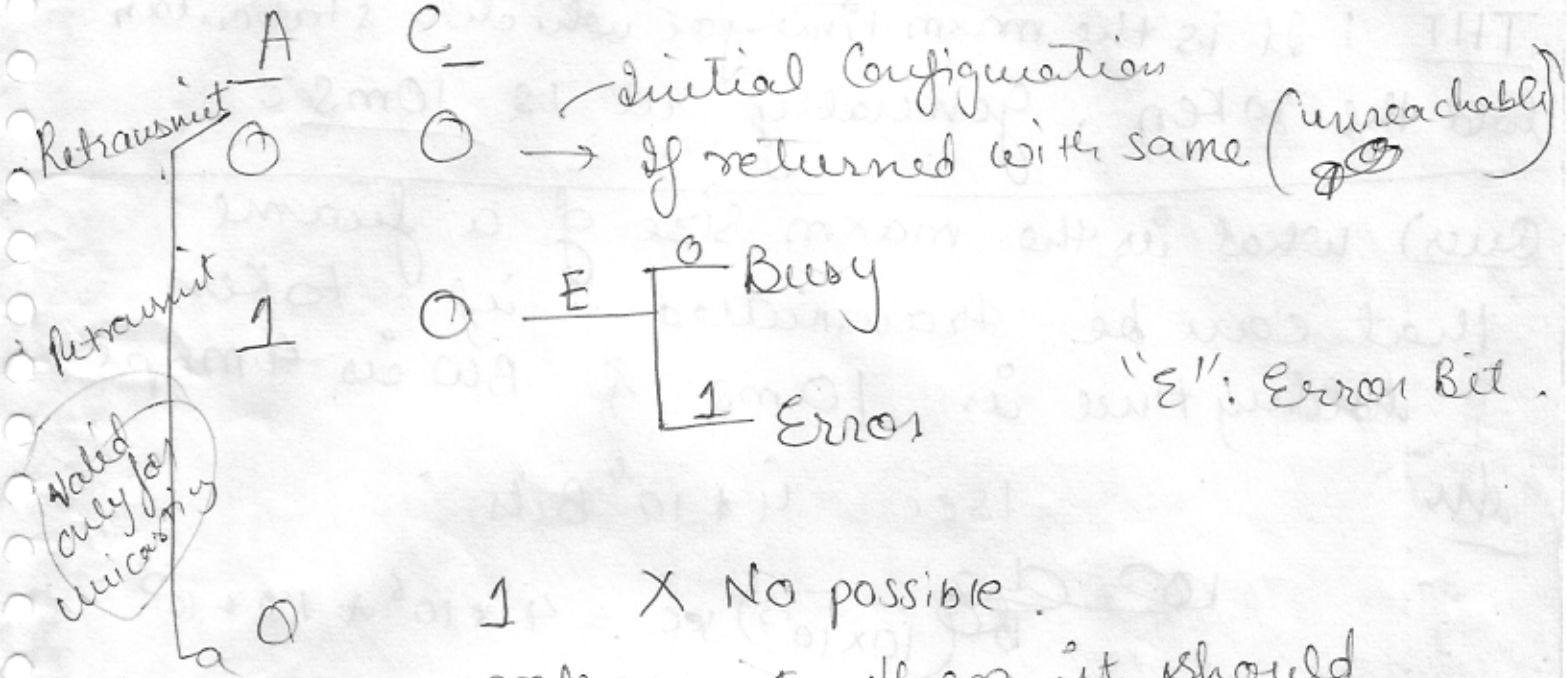
$\frac{d}{v} + m * T_t$ | $\frac{d}{v} + m(T_t + RL)$

Early token insertion | delayed token re insertion

③ Destination : a & b :

Packet will contain 2 bits 'A' & 'C'.

'A' indicates that destination is available &
'C' indicates that destination has copied



When source retransmits then it should reset the monitor bit so that monitor does not consider it as "orphan packet".

These Bit Combinations are not Valid for multicasting & Broadcasting.

Most of the problems are solved by "Monitor Station". But Monitor could also have some problems.

Monitor Problem

(i) DOWN (AMP)

(i) (ii) Malfunctioning

In order to find out whether a monitor is available or not, Monitor should periodically send Heart Beat Message (AMP) [Active Monitor Presence]

(ii) If the code of monitor is corrupted in such a way that the monitors sends AMP packets but does not provide any other service then manual intervention is reqd. [NO SOLN]

THT : It is the maxm time for which a station can hold the token. generally it is 10msec.

Ques) What is the maxm size of a frame that can be transmitted if token holding time is 10 ms & BW is 4 mbps.

Ans

$$1 sec = 4 * 10^6 \text{ Bits}$$

$$(10 \times 10^{-3}) sec = 4 * 10^6 * 10 * 10^{-3}$$

$$= 40000 \text{ bits}.$$

A station can transmit data only if it holds the token & a station can hold token for 10 ms.

TOKEN RING FRAME FORMAT:

① SD | AC | ED — Token frame

② Data frame

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

FC — Type of control

00 - Data
11 - Control

SD : (8 bits)

J	K	0	0	J	K	0	0

AC :

P	P	P	T	m	R	R	R

ED : | J | K | I | I | J | K | I | E |

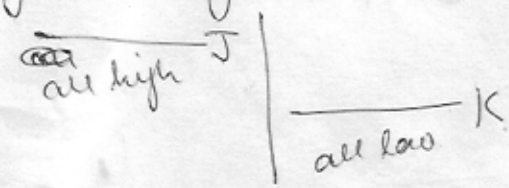
Info bit — 1: More packets are coming

E — error

FS : | A | C | 0 | 0 | A | C | 0 | 0 |

SD (Start Delimiter) : It is used to indicate start of the frame.

J & K are Invalid Differential manchester codes.



SD = Not standardised.
Any comb containing the Invalid J, K codes

AC

| P | P | P | T | M | R | R | R |

used for Reservation

used for priorities for client / Server

1 : Token frame
0 : Data frame.
→ used to distinguish token from other frames

Monitor Bit

FC frame Control : This byte indicates if the frame is Data frame (or) Control frame & what type of control.
↳ used for maintenance
Ex AMP frame, Back on frame & when cut in the wire

DA & SA : Indicates Destn & Source mac addresses.

ED : End delimiter : used to indicate end of the frame, "E" : Error

TR → ⊗ → 4999 Bytes, then fragmentation

"I" : Info bit : Indicates that more ~~Informa~~ frames are following ie fragmentation has been done : necessary when large

FS

Q) why 2 copies of A & C are used?
∵ CRC is not calculated on FS.

Q) Why CRC is not calculated on FS.
∵ CRC is calculated by sender & AC is changed by Receiver.

Network layer : It is mainly responsible for host to host connectivity.
(system to system) (H-H, S, R, LA, Frag, CC)

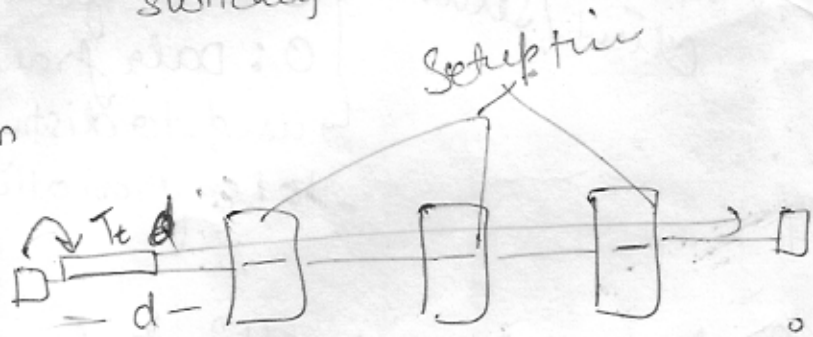
Transport layer End to end : process to other

Link to link : wire to wire.

SWITCHING :

- circuit switching (H/W) soln
- Packet switching
  - Virtual circuits → ATM
  - Datagram → IP
- Message switching

Circuit Switching : → Not store & forward



In this method, path is setup before data is transmitted. for this, time taken is ST [Setup time]

- Data is not broken into packets.
- No Headers are needed.
- Same path is followed by the entire data.
- Data arrives INORDER. No reordering is reqd.

Let us assume "M" : total size of message in bits to be transmitted

"B" : BW (bits per sec)

"X" : # of Hops (links)

"d" : ~~distance~~ length of each hop

"v" : velocity of signal

TD : Tear Down time

then

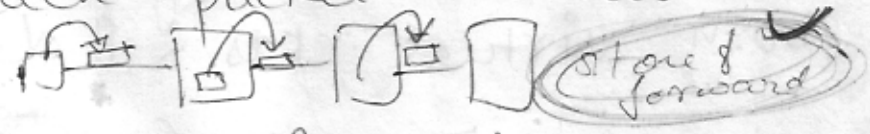
$$\text{total Time} = \boxed{ST + \underbrace{\frac{m}{B}}_{T_t} + \overbrace{\frac{Xd}{v}}^{T_p} + TD}$$

[Since it is not store & forward so, only 1 time transmit. It is JUST One time transmit & forwardin]

It is Implemented at Physical layer so, it is H/W soln & not the S/W soln.

PACKET SWITCHING → connection less

DATAGRAM : In this message is divided into small packets & each packet carries a header.



- Path is not fixed. It is dynamically found out whenever a packet arrives.
- All packets may take different routes.

∴ out of order packets may be possible, & REORDERING is required.

$$\text{Total Time} : \boxed{X\frac{m}{B} + \frac{Xd}{v}}$$

→ store & forward so, X times transmit

VIRTUAL CIRCUIT :

- In this message is divided into packets. Only 1st packet will contain a header & all the packets follow the same path as the 1st packet.
- All the packets follow the same path
  All the packets arrive in order
  Reordering is NOT required.
- In virtual circuit, Resource Reservation is done.
  CPU, BW & Buffer
  so it is Connection-oriented
- VC are reliable.
  Datagram is unreliable (connection-less).
  Some packets may be dropped down of no resource since resources are not allocated
- Total Time taken to transfer a message is same in both virtual ckts & datagram

If $ST + TD > (X-1)\frac{M}{B}$ (small data) $\Rightarrow$ packet switching is better

$ST + TD < (X-1)\frac{M}{B}$ (large data) $\Rightarrow$ circuit switching is better

for Bursty data, Ckt Switching is better.

ex of Datagram : IP

ex of virtual ckt : ATM

VOIP

These applications use Datagram service at Network layer and so transmission will be unreliable.

ex Skype, viber

IP Header

(C A T E)

(bits)

Version (4)	HL (4)	Type of Service (8)	Total length (16)	
Identification (16)			DF / MF	Fragment offset (13)
Time to Live (8)	Protocol (8)		Header Checksum (16)	
Source IP (32)				
Destination IP (32)				
options				

20 bytes static

variable 0-40 Header size = min 20 | max 60

Version (4 bits) : It indicates the current version of IP.

0100 : IPV4
0110 : IPV6

HL (Header length) : 4 bit field : indicates length of the header.

Scaling : is done

$\boxed{HL * 4 = \text{Header size}}$

Max size = 60
field length = 4 max no = 15 (111)
So, $\frac{60}{15} = 4$

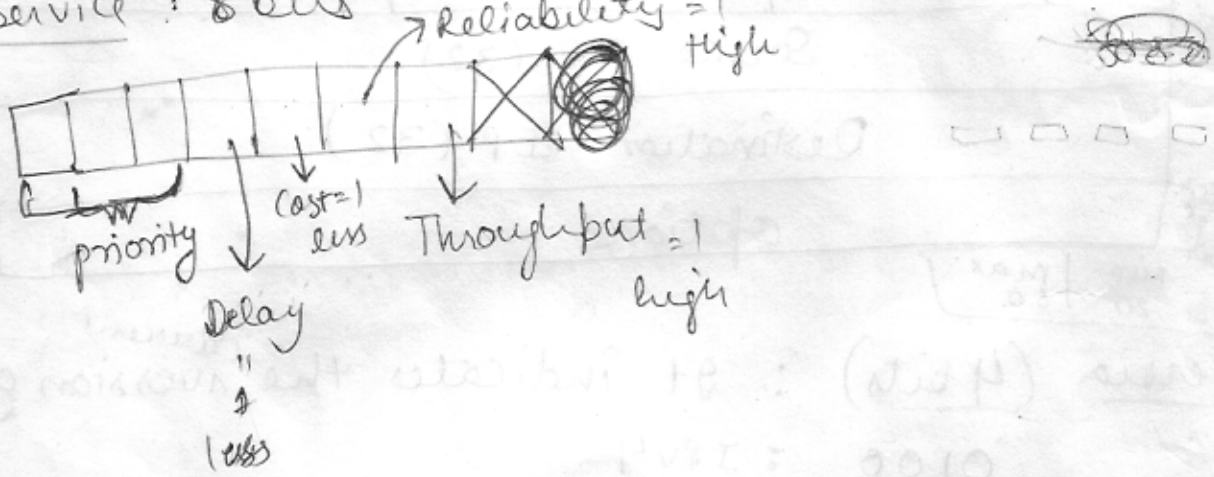
Q If HL field is 5 & total length field is 1000 then what is the length of data.
20

Ans Header size = 5 * 4 = 20

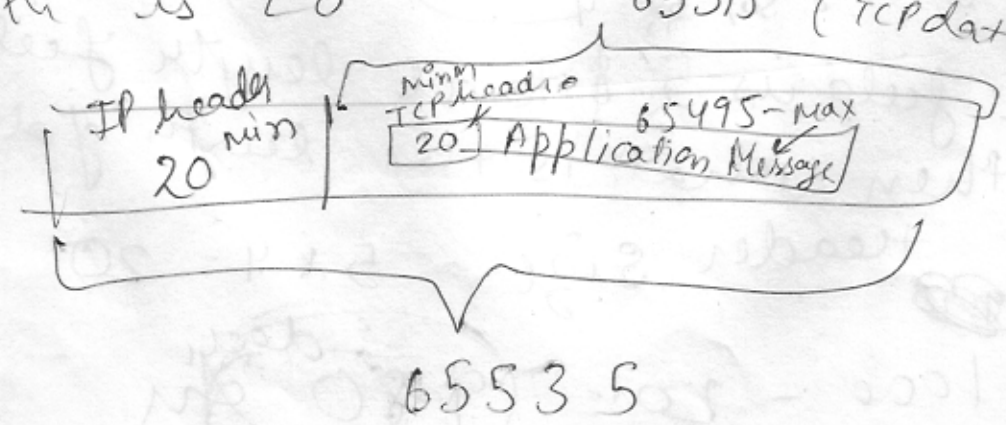
1000 − 20 = 980 Ans

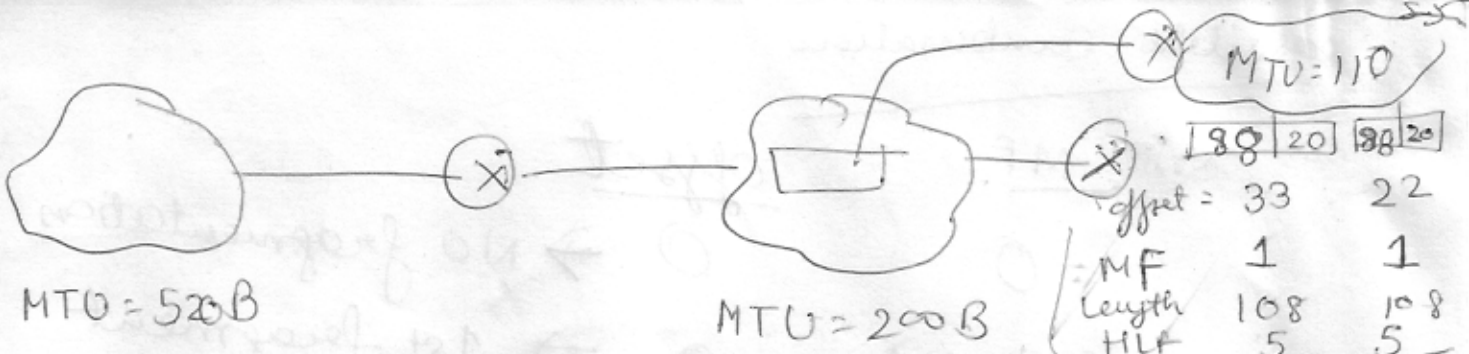
HL	HL field (Divide HL by 4)
20	5
24	6
22 ↳ 22 + 2 pad	6
50 ↳ 50 + 2 pad	13

Type of Service : 8 bits



Total length (16 bit) : It can represent a maxm of 65535 ($2^{16}-1$). ∴ maxm size of a IP packet is 65535 ie maxm size of TCP data is 65515 ∵ IP header minm length is 20





	88 | 20	88 | 20
offset =	33	22
MF	1	1
Length	108	108
HLF	5	5

500	20

Off = 0
MF = 0
Length = 520
HLF = 5

	148 | 20	176 | 20	176 | 20
offset	352/8 = 44	176/8 = 22	0
MF	0	1	1
HLF	5	5	5
Length	168	196	196

Initially = 500 + 20 (1 packet)

Sent = 500 + 80

overhead = 60 Bytes

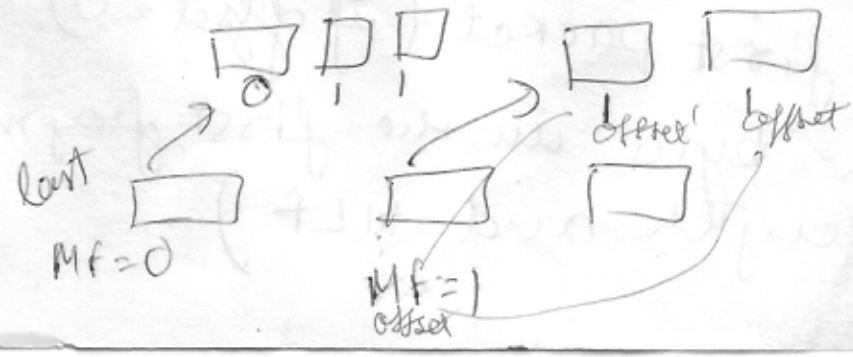
fragments = 4

$\eta = \frac{\text{useful Bytes}}{\text{Total Bytes sent}} = \frac{500}{580}$

offset : No of data bits ahead
65536 8
$2^{13}-1$

useful BW = η * BW

Note :
MF of ~~first~~ last ~~packets~~ fragment = MF of parent
offset of first child = offset of parent



possible combinations

	MF	offset	
	0	0	→ NO fragmentation
fragmented	1	0	→ 1st fragment
	0	×0	→ Last "
	1	×0	→ Intermediate "

A packet gets fragmented only at the Intermediate routers.

Q Where should the re-assembly of fragments be done?

Ans At the Destination

Not at routers :: ① all fragments may take different routes so, they may not meet at 1 router.
② Re-fragmentation may be needed.

RE-ASSEMBLY ALGORITHM at the destination

① If MF = offset = 0, the datagram is not fragmented else it is fragmented.

② Collect all the fragments of 1 datagram using IDENTIFICATION NUMBER

③ Identify the first packet (if offset = 0)

④ Count the data bytes in the first fragment (using Total length and HLF)

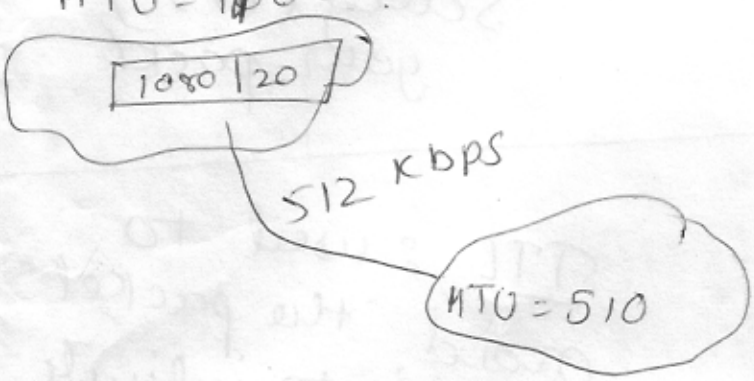
⑤ Divide the data size by 8, say 'x'.

⑥ Identify the next packet which has 'x' as offset.

Keep repeating the above steps till MF = 0.

ex

MTU = 1000



490
490
980

3 packets

overhead = 40

$$\eta = \frac{1080}{1080+60}$$

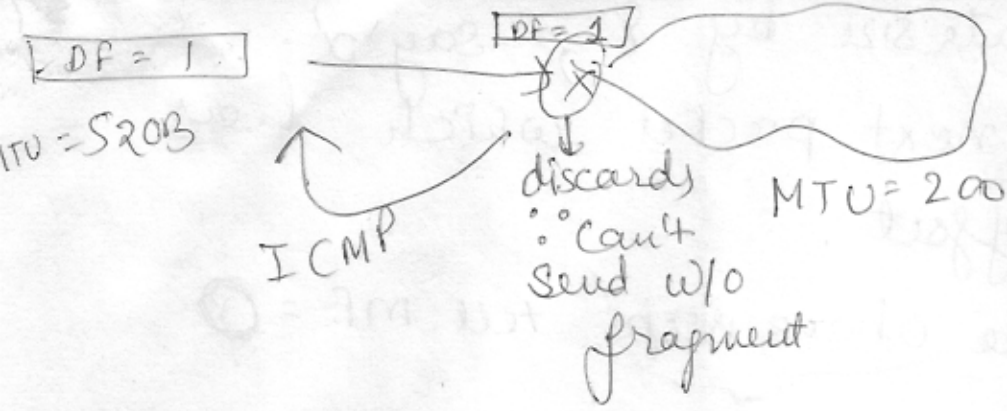
$$\text{useful BW} = \frac{1080}{1080+60} \times 512$$

In IP, if a router discards any packet then it sends ICMP packet to source.

IP = Best effort service

∴ Not reliable
∴ ICMP may also be discarded.

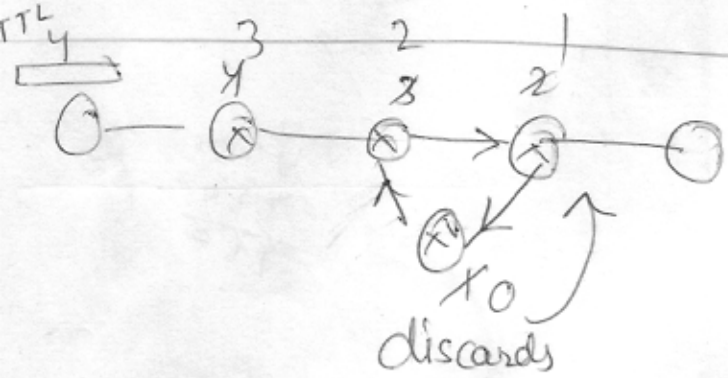
DF : Dont fragment ⇒ This field (1) indicates that this datagram should not be fragmented



- Parameter Problem

↓

router could not forward the packet ∵ of the parameter setting (DF) of your packet.



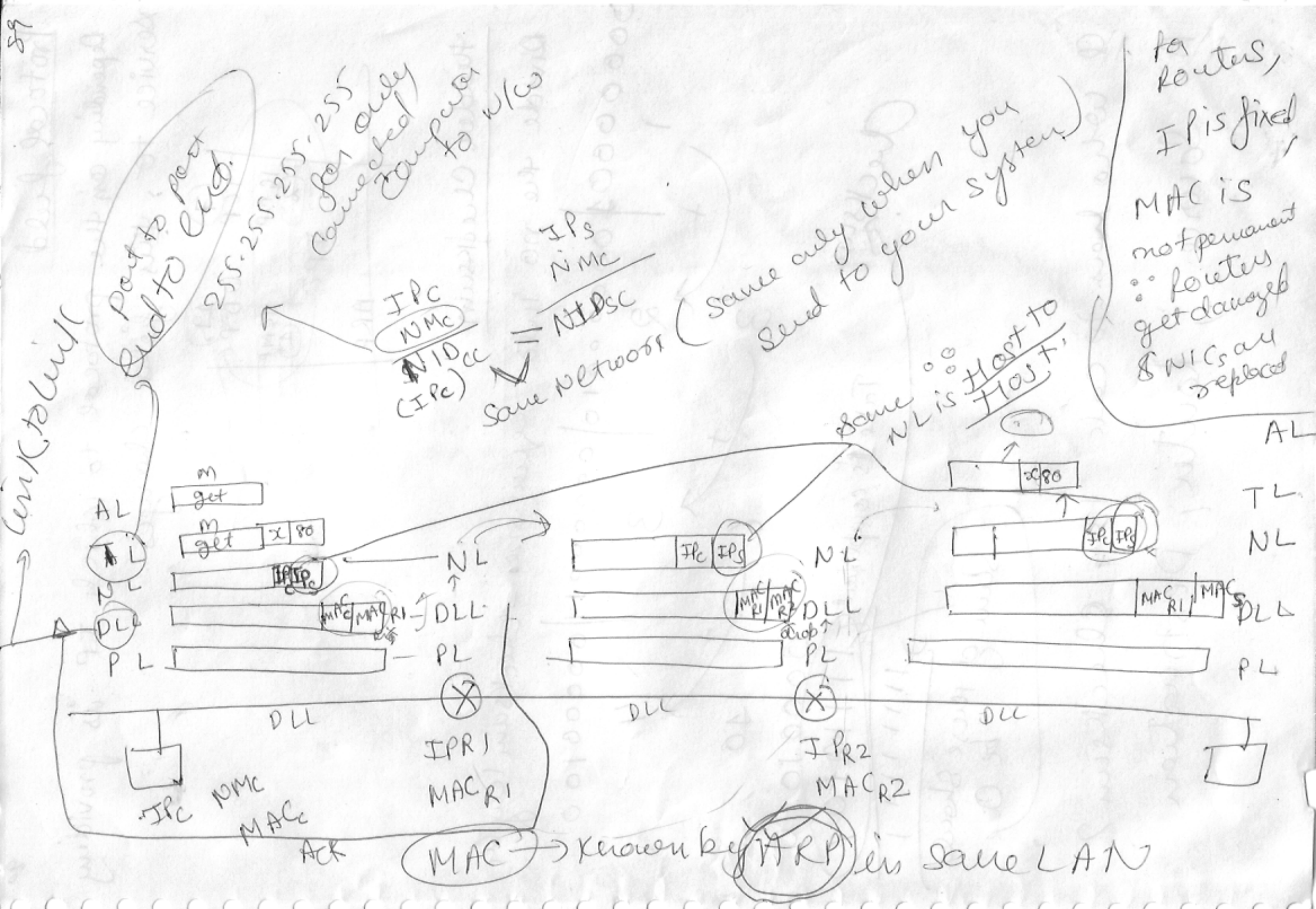
TTL : used to avoid the packet to go into infinite loops.

Sender will know through the type of ICMP that packet was discarded ∵ of TTL was less.

Note

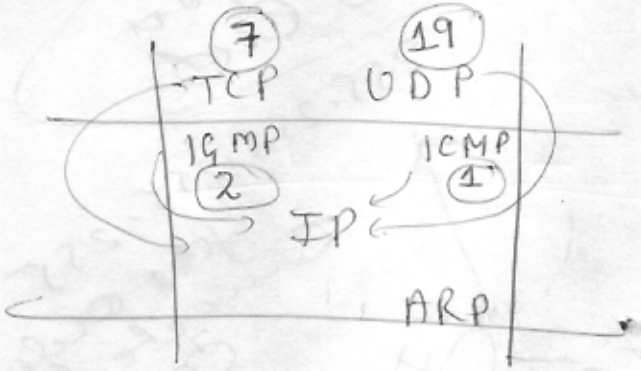
The main difference b/w Transport Layer & DLL is DLL is LINK to LINK (mac addresses using ARP)

whereas TLL is END to END (port nos)


Link to link
port to port
end to end
255.255.255.255
for only connected computer to w/co
IPc
NMc
NID(IPc)cc
IPs
NMc
NIDsc
same network
(same only when you send to your system)
for routers, IP is fixed, MAC is not permanent ∴ routers get damaged & NICs are replaced
same ∴ NL is Host to Host
AL
TL
NL
DLL
PL
m
get
x
80
IPc
IPs
MAC R1
MAC R2
MAC c
MAC s
IPR1
MAC R1
IPR2
MAC R2
NMC
MAC c
ACK
MAC → known by ARP in same LAN

Protocol field

Depending on the protocol to which IP is providing service to, this field changes.



Header Checksum

Divide the no into equal sizes = checksum length

00000001 | 00000010 | 00000011 | 00000100

1 2 3 4

1 + 2 = 3

3 + 3 = 6

6 + 4 = 10

00001010

Take 1's compliment 11110101

Checksum

11111111

Sum of these should be 0

Q who has to calc. the checksum?

A Source, Router, Destination

Q why is checksum calculated only on header.

∵ otherwise routers will get overloaded

Q why should router calculate the checksum.

∵ sum of the fields in the header changes in the packet

Q what are the fields that change?

A TTL, offset, MF, length, options

TTL → will definitely change.

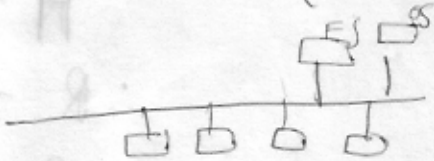
may change.

① used to test self connectivity (NIC is working or not)

② Both server & client can be tested on the same system.

NID	HID	IP (valid)
valid ✓	✓	
✓	0's	NID
✓	1's	DBA
1's	1's	LBA
1's	0's	NM
0's	✓	Host within a n/w
0's	0's	I dont have IP
127	other than 0's & 255.255.255	Loopback

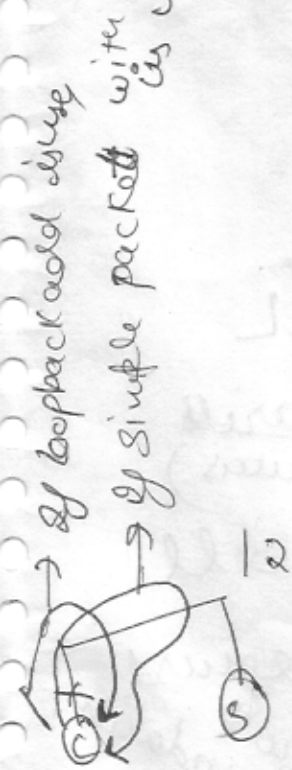
Known Ask
IP → MAC (ARP)
MAC → IP (RARP)



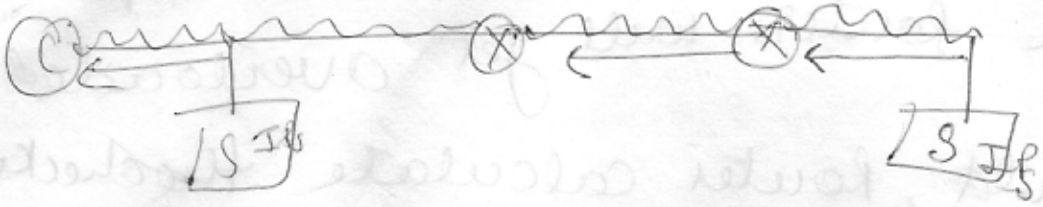
• The packet will return to source w/o going to n/w.

used to check: Self connectivity

In this RA = 127. Such a packet will never appear on n/w



ARP Address Resolution Protocol



FFFFFF	IPs —
48 bits	

→ broadcast

ARP Request

H→S
H→R
R→R
R→S

Mc	Ms	ARP Reply IPs MAC

← Unicast

ARP Request is Broadcast at DLL and ARP Reply is Unicast.

- ARP can be used in following 4 cases

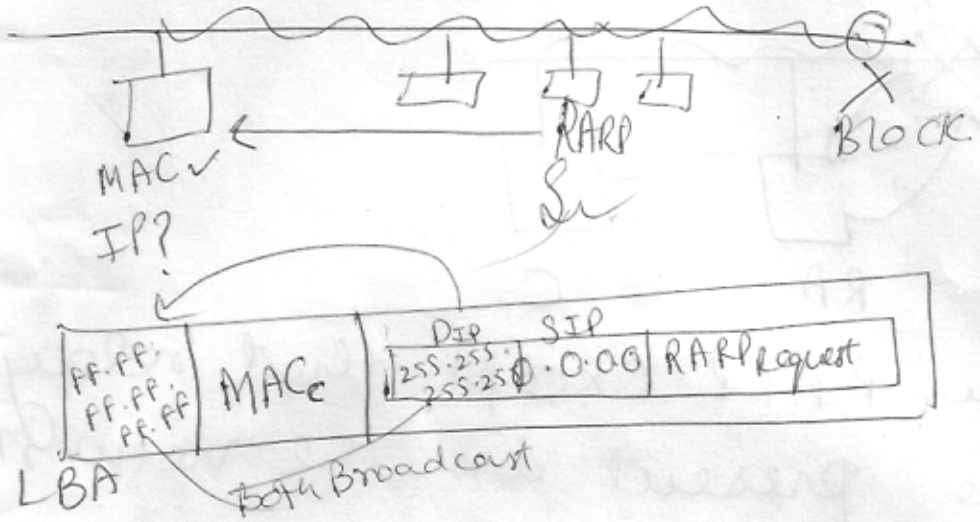
H → H
H → R
R → R
R → H

Note: Any Broadcasting done at DLL should never cross Network Boundaries (Routers)

∴ a router will block or filter all broadcast packets. ∴ ARP request packet can NEVER cross N/w Boundaries.

RARP (Reverse ARP)

A Host can have
≥ 1 IP addr
&
≥ 1 MAC addr



FF.FF.FF.FF.FF.FF	MAC_c	DIP 255.255.255.255	SIP 0.0.0.0	RARP Request

LBA

Both Broadcast

RARP Request Broadcast

MAC_c	MAC_{RARP}	IP_c	IP RARP	RARP Reply

RARP Reply

Unicast

RARP Table

Static

MAC	IP
MAC_c	IP_c
M_1	IP_1
⋮	
M_n	IP_n

RARP Request is Broadcasting at NL/DLL ✓. So, it cannot cross N/w Boundaries ∴ every N/w should have 1 RARP Server within the N/W

Disadvantage

If a N/w has 4 subnets then every subnet should have RARP Server

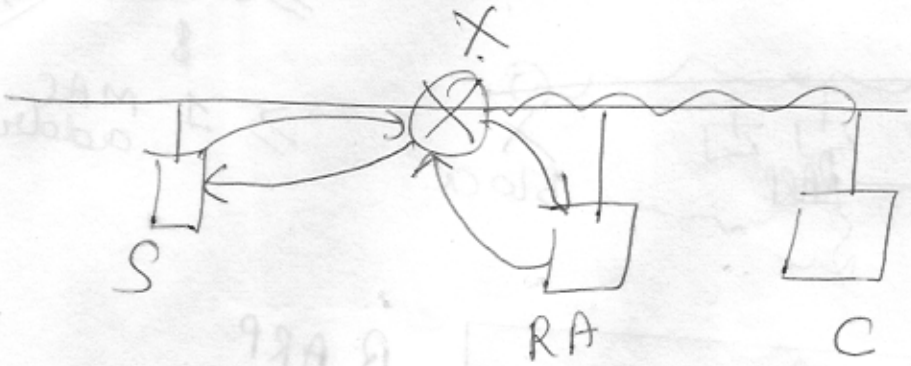
Disadvantage

In RARP, mapping table is Static ie

IP address $\geq$ # m/cs of mac

independent of no of users online at any time

BOOTP Server



It is same as RARP except that relay agents are present in the network where BootP server is not present

1's	M_C	1's	0's	Request

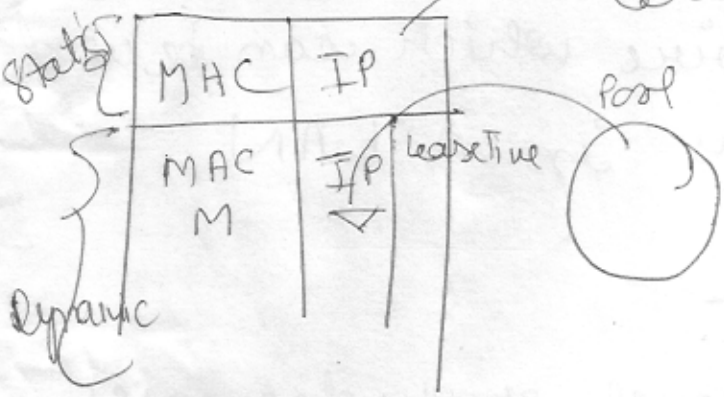
M_R	M_{RA}	IPs	IP_{RA}	Req

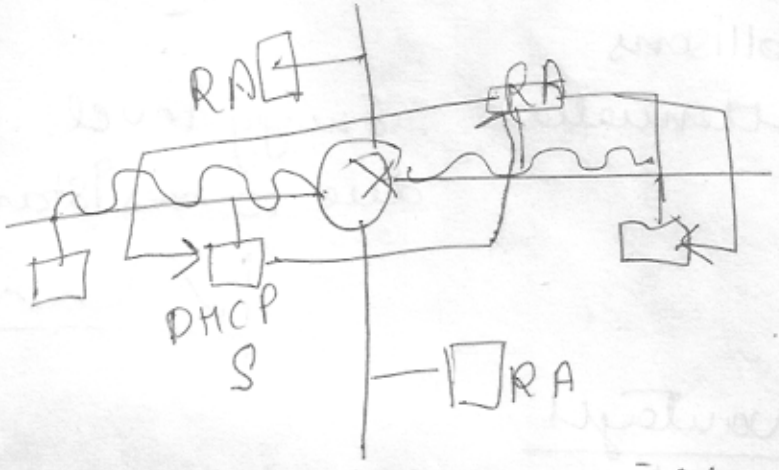
M_S	M_R

M_R	M_S	IP_{RA}	IPs	Reply

M_C	M_{RA}	I_C	I_{RA}	

DHCP Dynamic Host Config protocol → for those which are always active & running





RARP = N/w Layer

Boot P → App Layer

DHCP is backward compatible with Bootp ie a client can send a request in Bootp format and still gets a reply from DHCP server.

ADVANTAGES OF DHCP server:

① Only 1 DHCP Server is enough for any number of N/ws (subnets).

② An IP address can be allocated & deallocated for a m/c automatically.

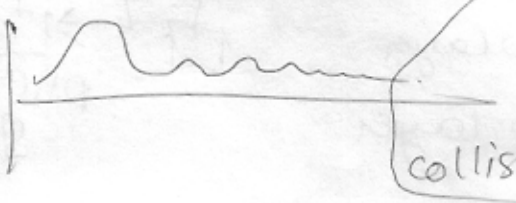
③ If maxm no of users online is "m" then we need "m" no of IP addresses.

H/W used in N/Ws :

I) **Coaxial Cable** : It is a wire which can be used to connect all the systems in a LAN.

Disadvantages

1) Collisions
2) Attenuation : Energy level slowly decreases due to resistance in wire



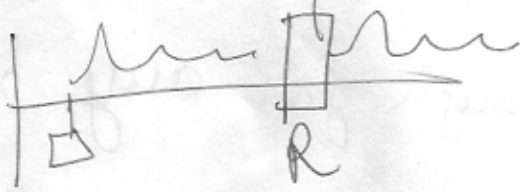
'n' systems
collision domain = 'n'

Advantages

1) Cheap 2) Easy to maintain

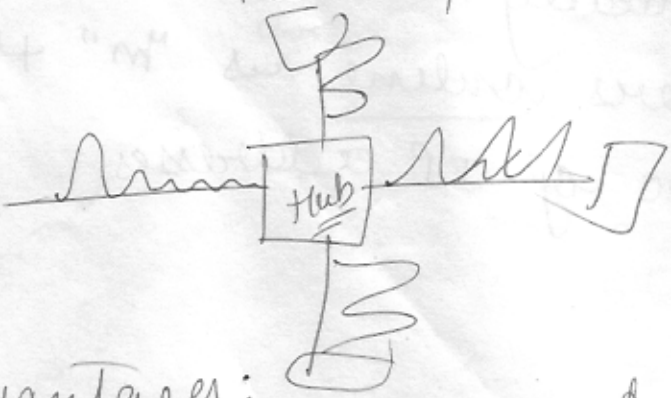
Only physical layer is needed.

II) **Repeater**

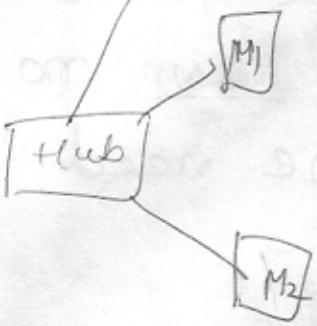


→ Regenerates the energy in the existing signal.

III) **HUB** : • Connecting Device
• Multi port Repeater.



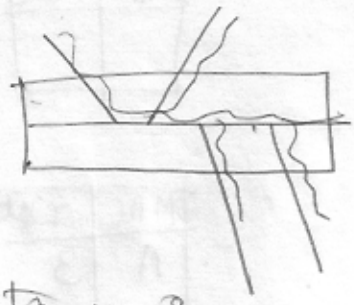
should be directly connected to M/Cs



Advantages:

1) Easy to connect & disconnect ∵ of power button.
2) Maintainence is easy

3) It is cheap



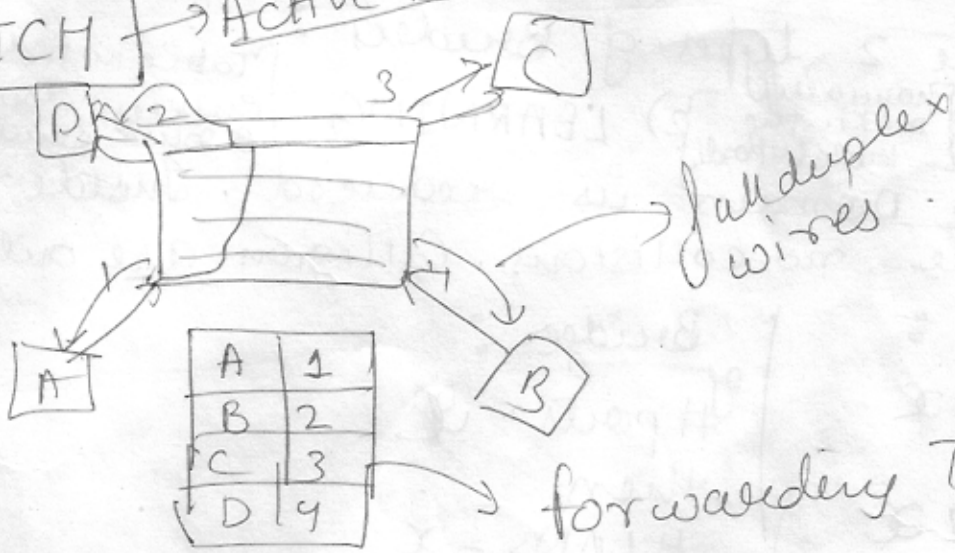
Hub also called as Layer 1 switch

Disadvantages :

1) Collisions are possible in Hub
Collision Domain still - 'N'. [if #systems = N]
not reduced
2) It is passive device (No software)

Only Physical Layer is needed.

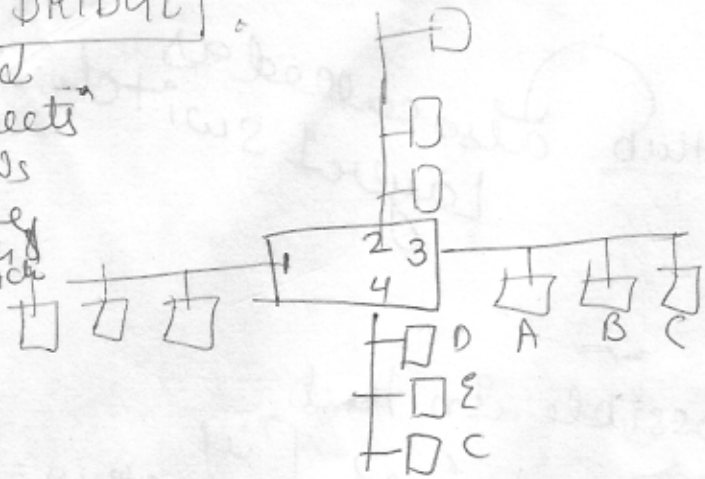
IV SWITCH → Active Device (Has S/w.



- Switch will have physical layer & Data link Layer.
- Switch is used to connect Hosts directly but not LAN.
- Collision Domain is reduced to 0.
It is a store & forward device & uses forwarding table to forward a packet appropriately
- Cost of a switch is 4 to 5 times more than a Hub

IV BRIDGE :

Connects LANs unlike switch & hub which connect only m/cs.



MAC	Interface

Table → Initially empty

MAC	Interface	Expiry time
A	3	

entry done when A sends a message to any one

- Bridge is a connecting device which contains physical layer and DLL.

Bridge is responsible for forwarding and filtering (discarding). → Systems should not be moved to different LAN

There are 2 types of Bridges :

1) STATIC [manually making the table updated] 2) LEARNING [Table initially empty, entries done only when a source sends.]

- Collision Domain is reduced. Inside the Bridge, there are no collisions. Collision are only within the LAN.

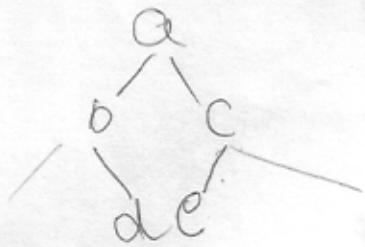
Switch :	Bridge :
# ports = x then # m/cs = x	# ports = x then # LANs = x, # m/cs = any number

All the devices (coaxial cable, Hub, Switch, Bridge) are used to connect systems in same LAN. ∴ these devices should not block any packet broadcasted at DLL. ∴

Broadcast Domain is not changed in these devices

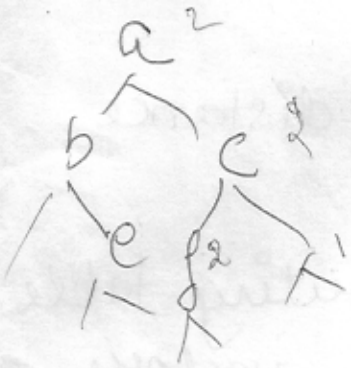
Bridge & Switch reduces Collision Domain but not Broadcast Domain.

Reduces	Collision Domain	Broadcast Domain
Hub	✗	✗ (NOT reduced)
Switch	✓	✗
Bridge	✓	✗



1st time Pre
2nd time In
3rd time Post

Pre abdce
In bdaec



Pre - abecf
In - beafc

ROUTER : • Routers are connecting devices which are used to connect 2 or more N/ws.

- Routers contain physical layer, DLL & NL
- Collision Domain & Broadcast Domain are reduced.

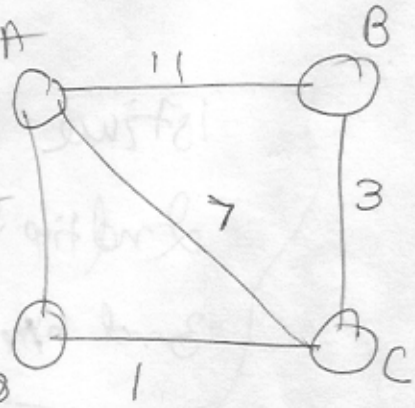
Routers
- use → Flood
  - adv: Reliability ↑, Shortest path guaranteed
  - Disadv: Traffic ↑, Duplicate packets ↑
- use → Routing
  - Static (Not change depending on traffic & topology)
  - Dynamic (They will change depending on T & T)
    - DVR
    - LSR

DVR (Distance Vector Routing)

Step 1: Every router will compute routing tables with local knowledge (only neighbours)

A	0	A
B	11	B
C	7	C
D	2	D

A



A	11	A
B	0	B
C	3	C
D	∞	–

Dest	cost	Next Hop
A	7	A
B	3	B
C	0	C
D	1	D

D:

A	2	A
B	∞	–
C	1	C
D	0	D

Step 2: Every Router exchanges its distance vectors with neighbours

Step 3: At every Router new Routing table will be computed based on distance vectors from the neighbours

At B:

A	11	A
B	0	B
C	3	C
D	∞	–

from A

0
11
7
2

from C

7
3
0
1

⇒

A	10	C
B	0	B
C	3	C
D	4	C

$BC = \min \begin{cases} BA + AC \\ 11 \quad 7 \\ BC + CC \\ 3 \quad 0 \end{cases}$ (3)

$BA = \min \begin{cases} BA + AA \\ 11 \quad 0 \\ BC + CA \\ 3 \quad 7 \end{cases}$ (10)

$BD = \min \begin{cases} BA + AD \\ 11 \quad 2 \\ BC + CD \\ 3 \quad 1 \end{cases}$ (4)

Shortcut

At B

A	11	A
B	0	B
C	3	C
D	∞	–

from A

0
11
7
2

BA 11

from C

7
3
0
1

BC 3

At B

A	10	C
B	0	B
C	3	C
D	4	C

exception

At C

A	7	A
B	3	B
C	0	C
D	1	D

from A

0
11
7
2

CA 7

from B

11
0
3
∞

CB 3

from D

2
∞
1
0

CD 1

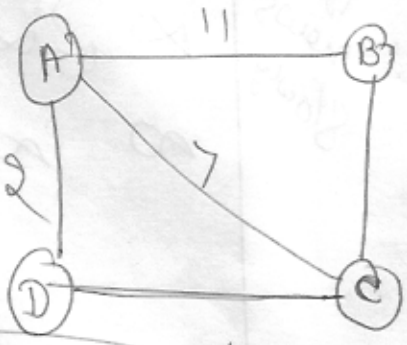
At C

A	3	D
B	3	B
C	0	C
D	1	D

exception

Answer

A	0	A
B	6	B
C	3	D
D	2	D



A	6	C
B	0	B
C	3	C
D	4	C

A	2	A
B	4	C
C	1	C
D	0	D

A	3	D
B	3	B
C	0	C
D	1	D

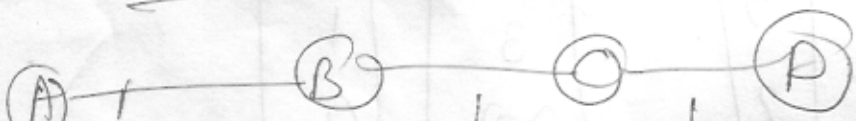
Possible questions

Q1) given either a graph or Routing table and distance vectors, find the new Routing table.

Q2) given a graph, find the final routing table

Q3) What is the number of edges which are unused.

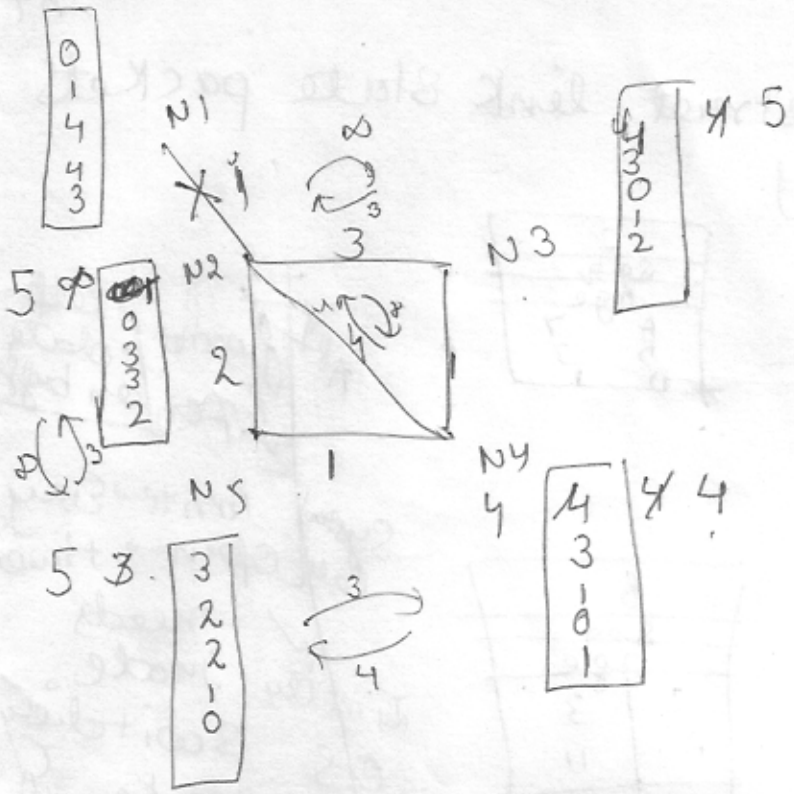
~~RSR (Link State Routing)~~



∞	∞	∞	
1	∞	∞	good news travels faster.
1	2	∞	
1	2	3	
∞	2	3	
3	4	3	
5	4	5	Bad news slowly
5	6	5	
7	6	7	
7	8	7	
9	8	7	
⋮	⋮	⋮	
	∞	∞	

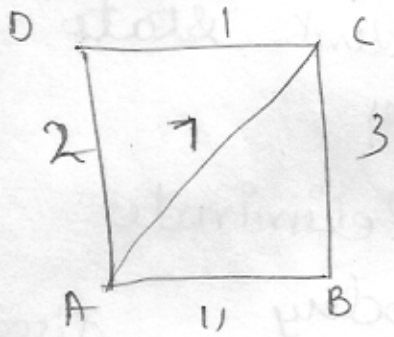
Count to ∞ problem

3	2	3
∞	2	3
∞	∞	∞
∞	∞	3
∞	∞	∞.



After link b/w N1 & N2 breaks, find distance vectors after 2 rounds

LSR (LINK STATE ROUTING)



Seq no →
Age →
Flooding is used
Diff → DVR: Send only distance vector
LSR: Send entire packet

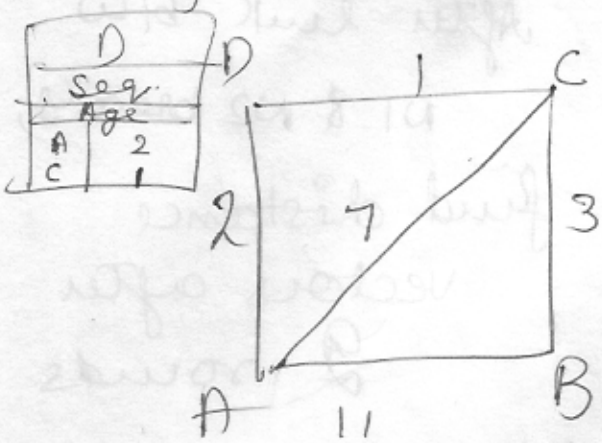
① find Link State packets
② flood to packets
③ Each node, will have information of entire n/w which will be represented using adjacency list or adjacency matrix

Sparse → List
Dens → Matrix

④ Run single source shortest path on each node.

Disadv
① Flooding
② Mem reqd for running single source shortest path is large

① Every Node will construct link State packets using local knowledy



C	
Seq	
Age	
A	7
B	3
D	1

B	
Seq	
age	
C	3
A	11

A	
Seq	
age.	
B	11
C	7
D	2

Age → Implemented using Systemcalls.

API	fwrite fopen · keep · date on buffer
System Call	write · Every open · time needs mode.
Directly OS	switching to Kernel mode

② Every node will flood its link state packets to all other nodes.

③ Sequence Numbers are used to ① eliminate duplicate packets ② Damn the floodery.

B	Seq no.	Age
	10	11

if ← [9] X dont forward discard

if ← [10] X "

if [11] ✓ Accept update the table & ~~forward~~ flood it

④ Age used to limit validity of a packet.

⑤ Every Router will prepare the entire graph using information from all other routers

At D

A	2	A
B	4	C
C	1	C
D	0	D

finally made

from C	from B	from A
D 1	A 11	B 11
A 7	C 3	C 7
B 3		D 2

⑥ Every router will run single source shortest path algorithm on the graph with itself as the source.

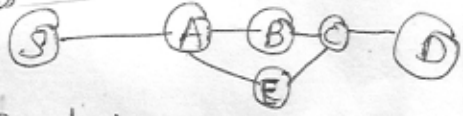
Note: Either link state packets or distance vectors will be produced periodically forever ∴ these algorithm are dynamic.

There is NO COUNT to ∞ problem in LSR ∵ if one link breaks, it is known to all nodes unlike DVR in which only neighbours know.

Disadv of LSR

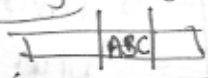
① More packets are exchanged as compared to DVR.

OPTIONS : used by N/w administrator for troubleshooting



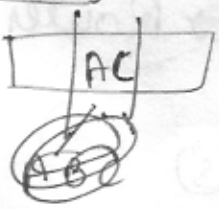
SOURCE ROUTING : It is 2 types:

1) STRICT Source Routing : Source will specify the entire route



has to take ABC.

2) LOOSE Source Routing : Source will specify partial route



can take ABC or AEC

Options field : 40 Bytes

1 Router : 4 Bytes

Tot Router possible $= \frac{40}{4} = \cancel{10}\ 9$

But we have max Routers allowed to be specified $=$

left out 4 Bytes used to specify other info like whether it is strict/loose etc.

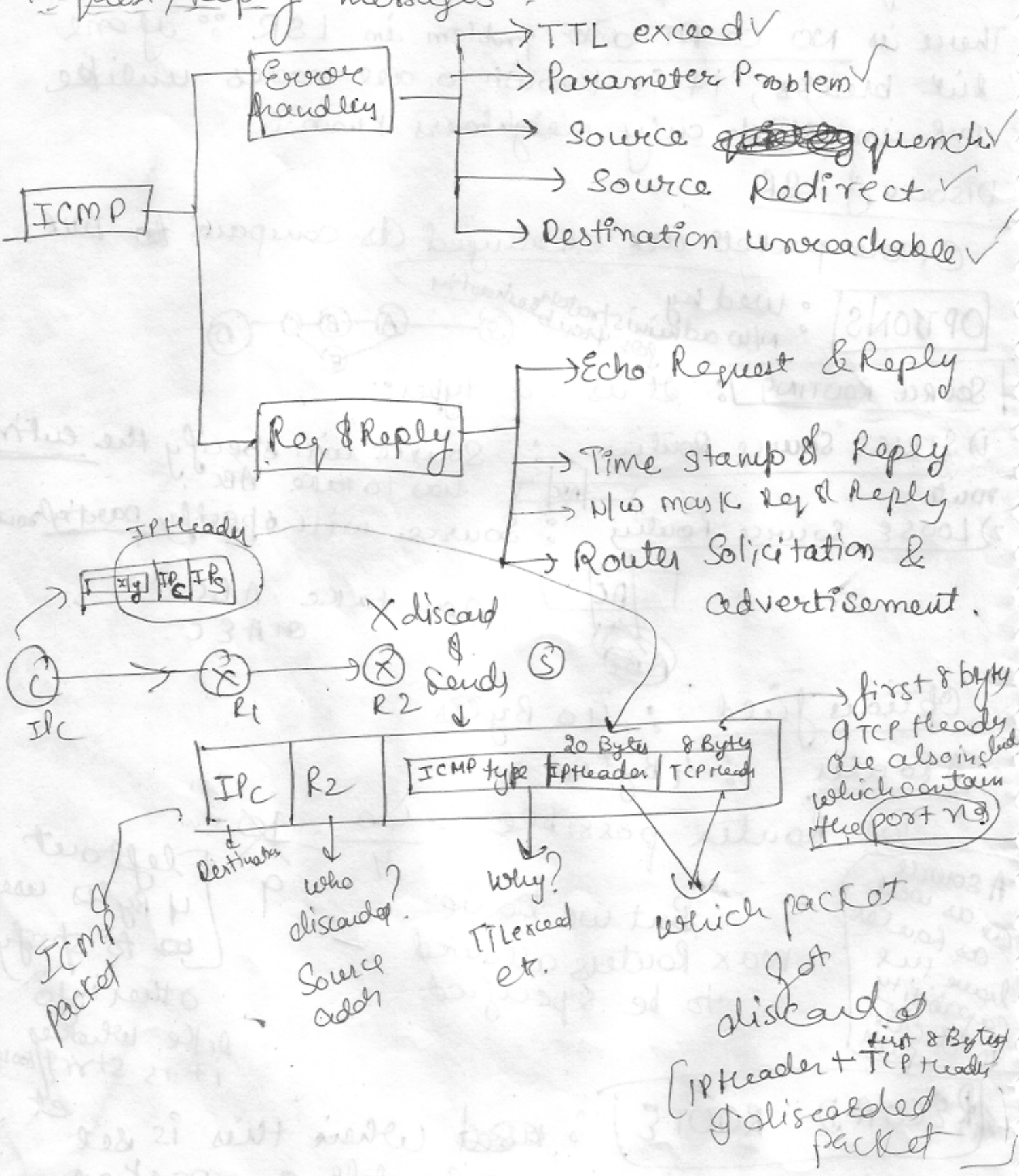
A source as well as Router have the capability to decide path

RECORD ROUTE : When this is set the Router writes in IP addr on packet

At max '9' Routers can be recorded. used by source to record the path which

packet follows. Using this only Destination knows the route followed.

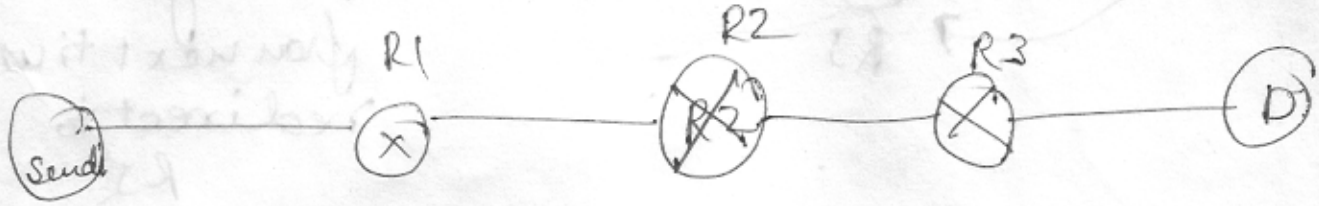
ICMP: It is generally used for feedback messaging or error reporting. It may also be used for Request/Reply messages.



Main purpose of TTL :

① To avoid Infinite loops

TTL also used to TRACE ROUTE : not the primary purpose of TTL.



- This program is used to find what are all the routers on the way from source to Destination.
- It is implemented using TTL and ICMP.

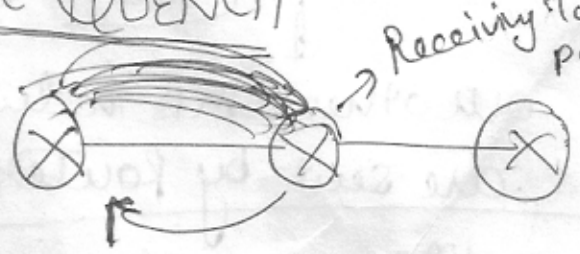
① Create a UDP packet with unused port no. and at N/w layer, set TTL to 1 with Destn IP as valid IP.

② Increment the TTL if ICMP received is TTL exceeds and resend the packet.

③ Keep repeating step 2 until we receive an ICMP packet with Destn Port Unreachable

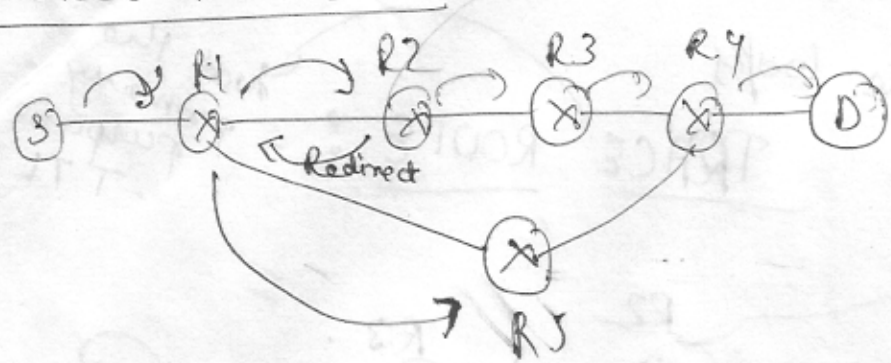
↳ Different from Destn unreachable

SOURCE QUENCH



A router can stop its neighbouring router or a host from sending lot of packets.

SOURCE REDIRECT



warning for this time, R2 is forwarding from next time redirect to R5

Parameter Problem

a) when DF = 1, & packet is larger than MTU of a Router.

b) In Strict Source routing

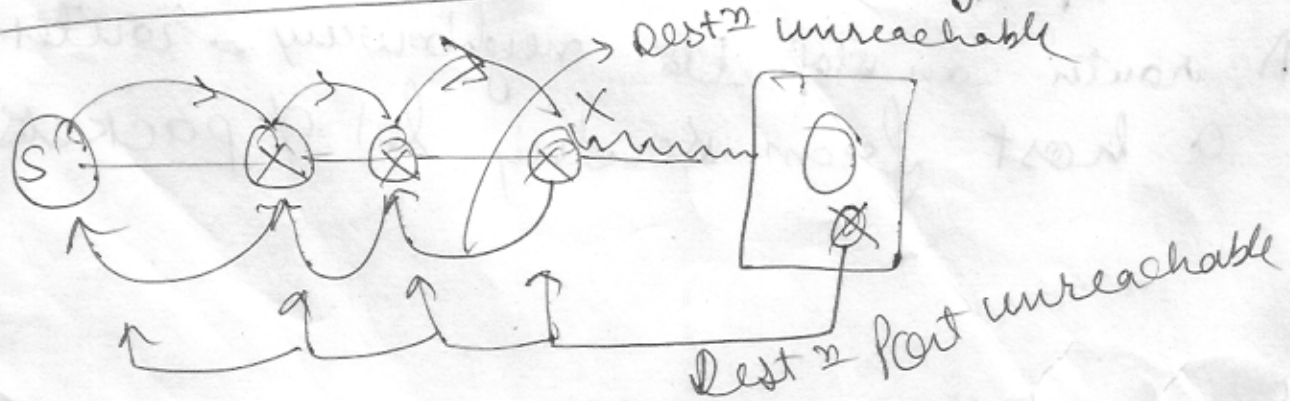
c) HLF < 5 ⇒ Not allowed

If any parameters in the header are not ~~allowed~~ letting the router to forward the packet then this ICMP will be ~~rejected~~ generated.

Destination Port Unreachable : Sent by Dest^n Host

↓

all other ICMPs including Dest^n unreachable are sent by Routers.



Request / Reply messages

Eco Request & Reply : Used to test if the dest^n IP is working or not and Routers on the way are working or not.

Eco Request starts from N/w Layer of the source and Eco Reply starts from N/w Layer of Dest^n

while
{
ping (—)
}

Wireshark : Used to see all the packets received & leaving the system

Ping is a application which uses Eco-Request and Reply.

Timestamp Request / Reply : Used to find out time at Dest^n for synchronisation

Router solicitation : Used to find what are all the routers connected to a network.

Advertisement : A Router which is newly connected to a n/w can use advertisement to specify its presence.

N/w mask Req & Reply : Any system can ask the router for N/w mask & get it using ICMP req & reply.

TRANSPORT LAYER

Main Responsibility: • END TO END CONNECTIVITY

• In case the application needs Reliability and N/w layer is providing unreliable service then transport layer can optionally take care of lost packets, missing packets and packet re-ordering. (EC, FC, Segmentation, Congestion Control, Pkt numbers)

Difference b/w TCP and IP:

TCP → connection oriented like a phonecall.

IP → Best effort service (Postal).

TCP

Source Port (16)								Destination Port (16)
Sequence number (32)								
Acknowledgement number (32)								
TCP Header length (4)	6 bits unused	URG (1)	ACK (1)	PSH (1)	SYN (1)	RST (1)	FIN (1)	Window size (adv window) (16)
checksum (16)								urgent pointer (16)
Options (32)								
Data (32)								

min size = 20B

max size = 60B

20 Bytes mandatory

0-40 Bytes

also contains 10 bits timestamp field

also contains 14 bits for Adv. window size

TCP: Reliable, End to End

Byte stream protocol

every Byte will be numbered.

Ports possible

0 ⋮ 1023 } well known server ports

1024 ⋮ 49151 } → Reserved for future use by IANA

$2^{16}-1$ } freely available

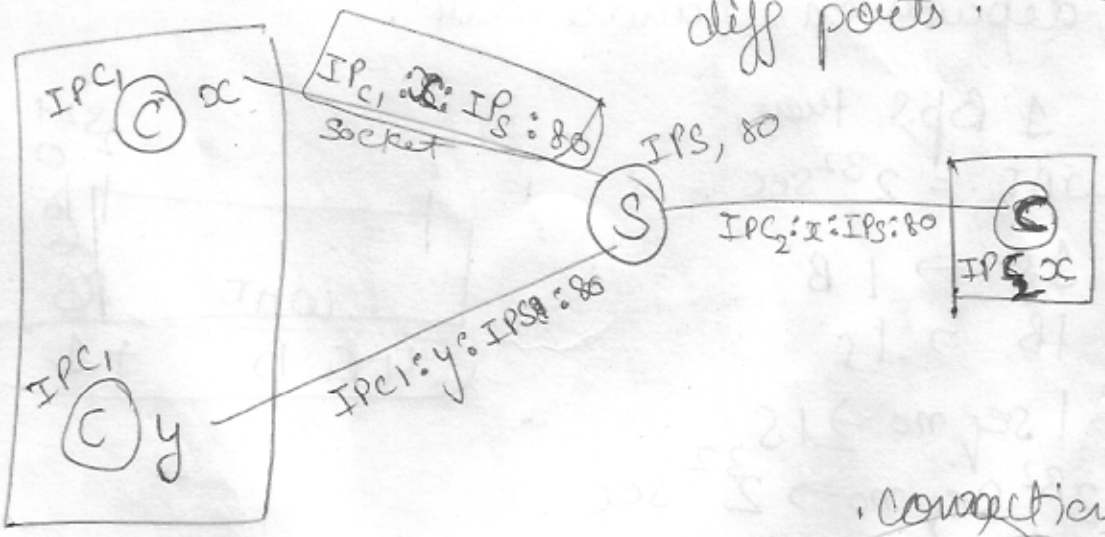
****

We use a combination of $\boxed{\text{IPaddress \& Port no}}$ for a connection

↳ resources.

① Diff systems Client & Server → may choose same port nos. but have diff IPs

② *On same system, diff ports → diff port nos but same IP.



IPadd → 32 bits
PortNo → 16 bits } 48 bits

connection / Socket

Total Connections Possible : 2^{48}

Socket is a 48 bit no. which is combination of IPaddress & port no & it is used to uniquely identify connection

Seq no : 32 bits

Total Seq no possible : $2^{32} = 4G$.

Every byte sent out by TCP will carry sequence number ie every Byte is numbered (Byte stream protocol)

At TCP, Bytes are counted.

Seq no : 32 bit field, 2^{32} sequence nos are possible

Thus 4GB can be sent out by giving unique nos

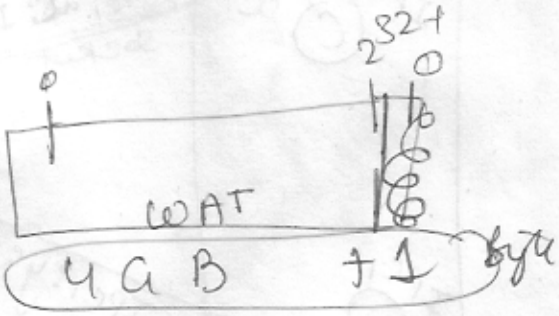
AL can give data to TL of any size.

WRAP AROUND : The process of using up all seq nos & reusing the same sequence number is called wrap-around.

WAT : Time taken to wrap around.
It depends on Bandwidth.

If B/w is 1 Bps then,

WAT $= 2^{32}$ sec.



1S → 1B

1B → 1S

1 seq no → 1S

2^{32} seq no → 2^{32} sec.

$= 4294967296$ sec.

If B/w is 1MBps then WAT = 4294.96 sec.

1MB → 1sec

1MB seq no → 1sec

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

2^{32} " " $= 2^{32}/10^6$ sec $= 4294.96$ sec.

$\frac{2^{32}}{BW}$ = WAT

Lifetime (LT): It is the amount of time for which a Byte can be alive in the Internet before being received by the receiver. In the current Internet, it is found out to be 3mins = 180sec

If B/W = 1GBps then TAT = 4.294sec

$$\frac{2^{32}}{10^9} \text{ sec}$$

Here WAT < LT so problem of Duplicate packets.

Always LT should be less than WAT.

always LT < WAT

for 1GBPS, if we design a protocol then what would be choosen seq no field size.

thus 180 * 1GB

(180 * 10^9) seq nos.

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

for seq. no.

Seq no = 180

10^9

seq no = 180 * 10^9

seq no size = $\log_2 180 * 10^9$

seq no field

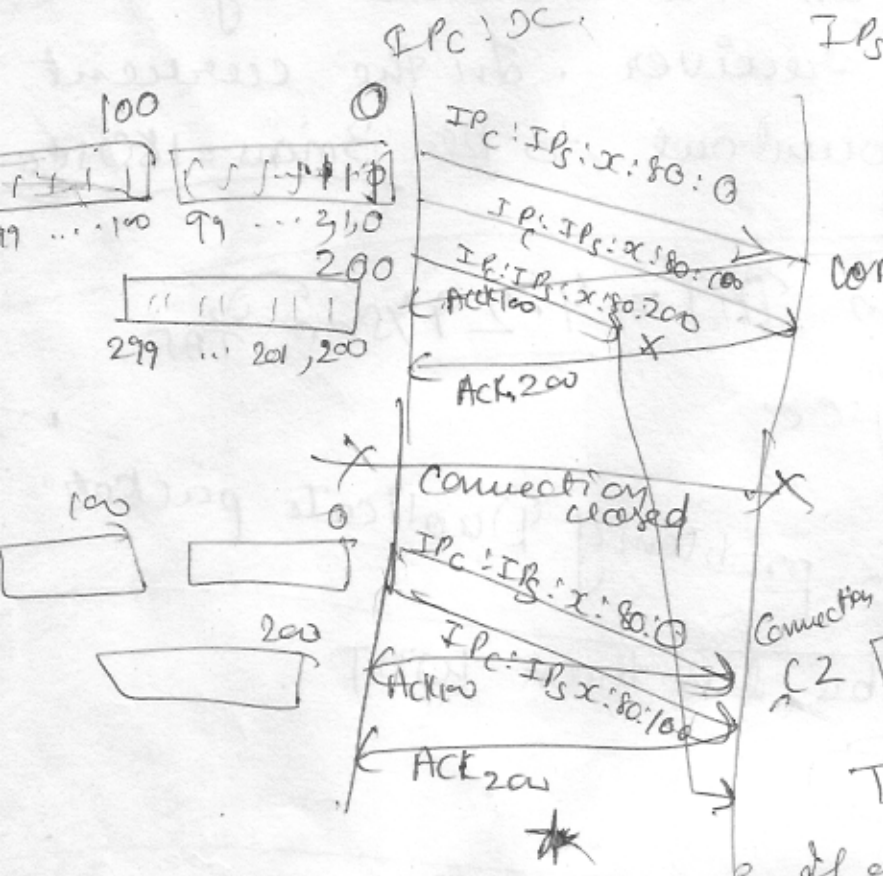
Thus 42 bits are reqd in sequence no field to avoid wrap around in BW of 1GBps

Thus 10 bits extra from seq no bits

These 10 bits are included in Time stamp field

(Obtain)

Q1 : TCP, UDP



1 ack for 100 Bytes → uses cummulative acknowledge.

TCP — 25% GBN
— 75% SR.

If a packet lost, only that transmitted,

Out of order packet are also accepted.

IP, port, seq no

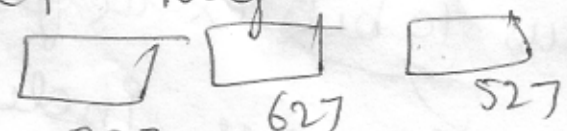
↓

all these will be same for packet from same system to other system in different connection.

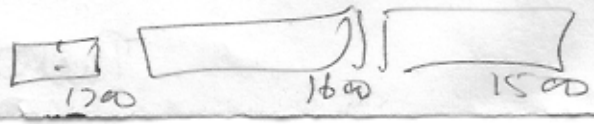
Though all Bytes are numbered but we give sequence number of 1st Byte to the entire packet

RULE : Instead of starting seq no from 0, use any RANDOM initial seq no.

for eg C1 might start from 527:



C2 from 1500 :



TCP Header Length : 4 bit field but maxm header size is 60 Bytes ∴ scaling factor of '4' is used $\left[\frac{60}{15} = 4\right]$

15 → maxm no possible with 4 bits

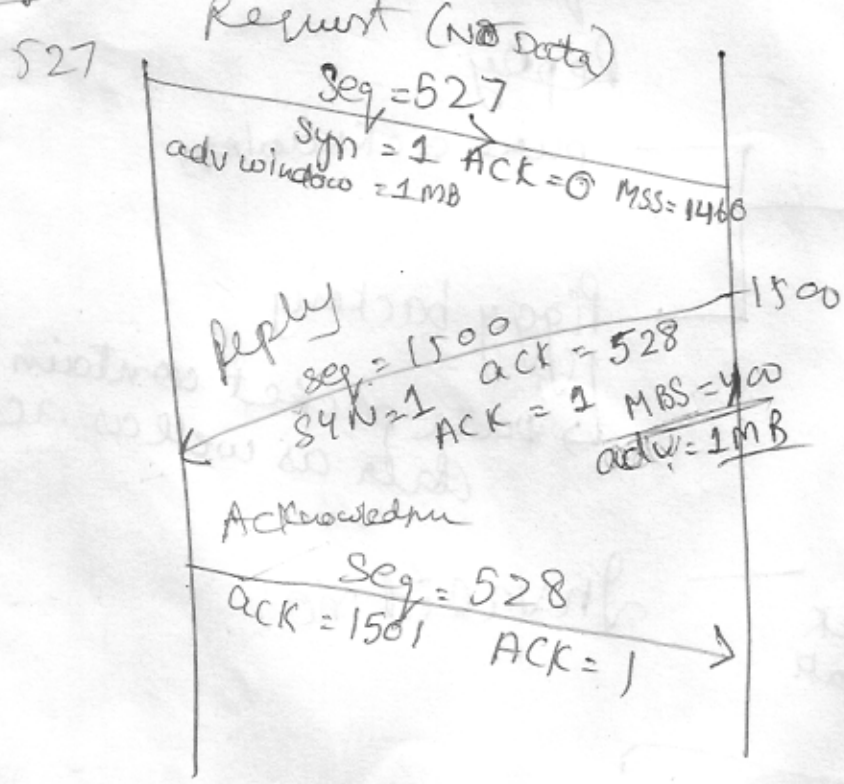
SYN flag : 1 : This is first packet & this packet has first seq. no.

ACK flag : 0 : This packet does not contain a valid ack or data

1 : Acknowledgment field contains a valid number

for data packet as well as ack.

Imp



Max seg which a client can receive should be 1460 & a total of 1MB data should be send beyond.

3 way Handshake

Req - Reply - ACK.

Establishing Connection

AL

any amount of data

TL 65495

max 1460 data

max 65515 data

+20 TCP Header

+20 IP Header.

Min of 2

NL & DL

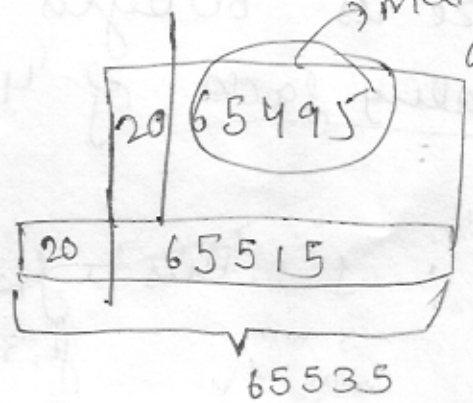
1500 DL

AL

TL — TCP

NL — IP

DLL



max TCP data in 1 segment if cop use TCP/IP

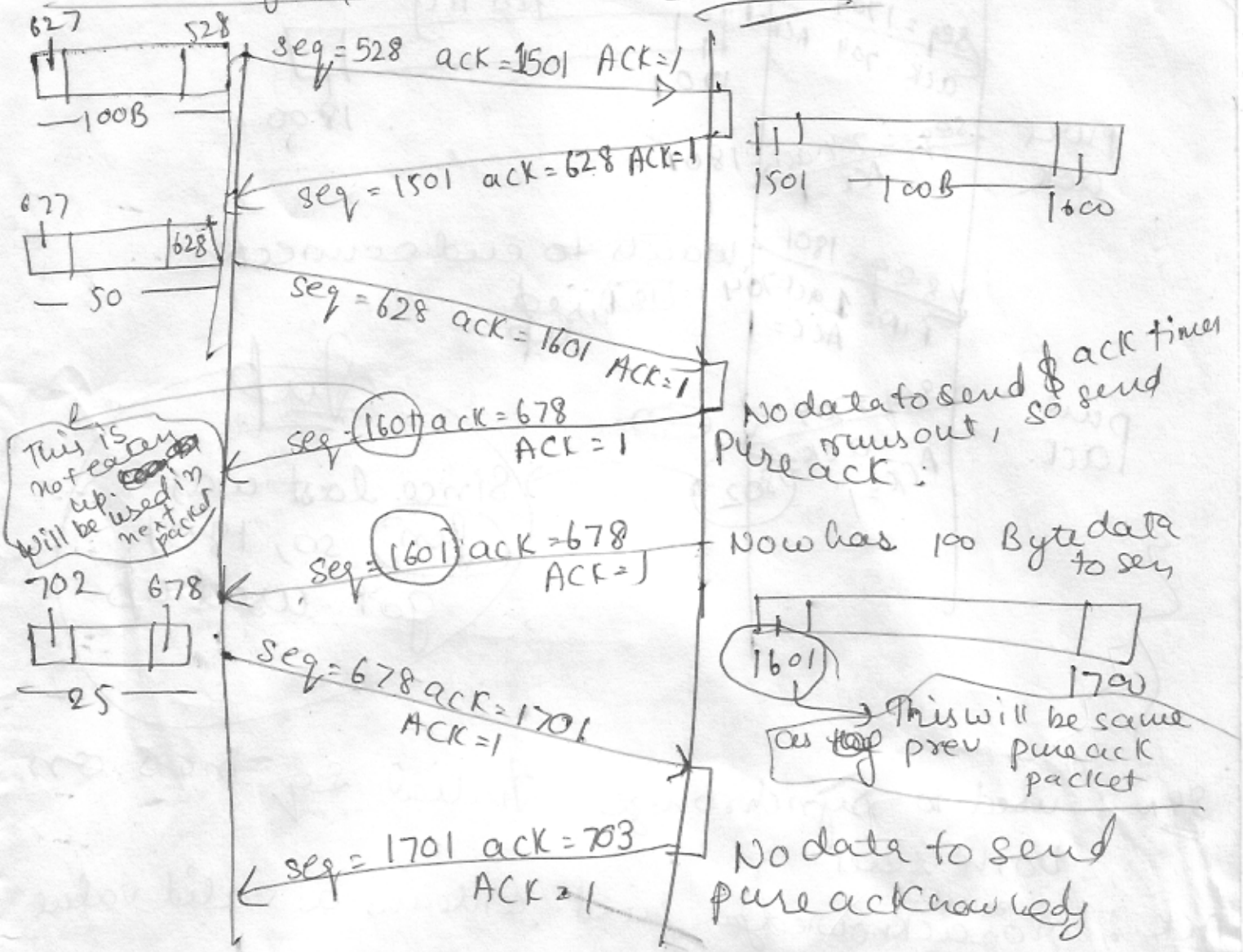
Ack — 0 → only for 1st packet
1 → for all other packet data Ack

SYN	ACK	
1	0	— Request
1	1	— Reply
0	1	— pure acknowledgy / Piggy backing → Data packet contain data as well as ack.
0 (Not the first packet)	0 (No ack No data)	— Invalid no.

pure ack ⇒ 0 doest not eat seq no.
SYN eats 1 seq no
FIN = 1 seq no
Data = 1 seq no

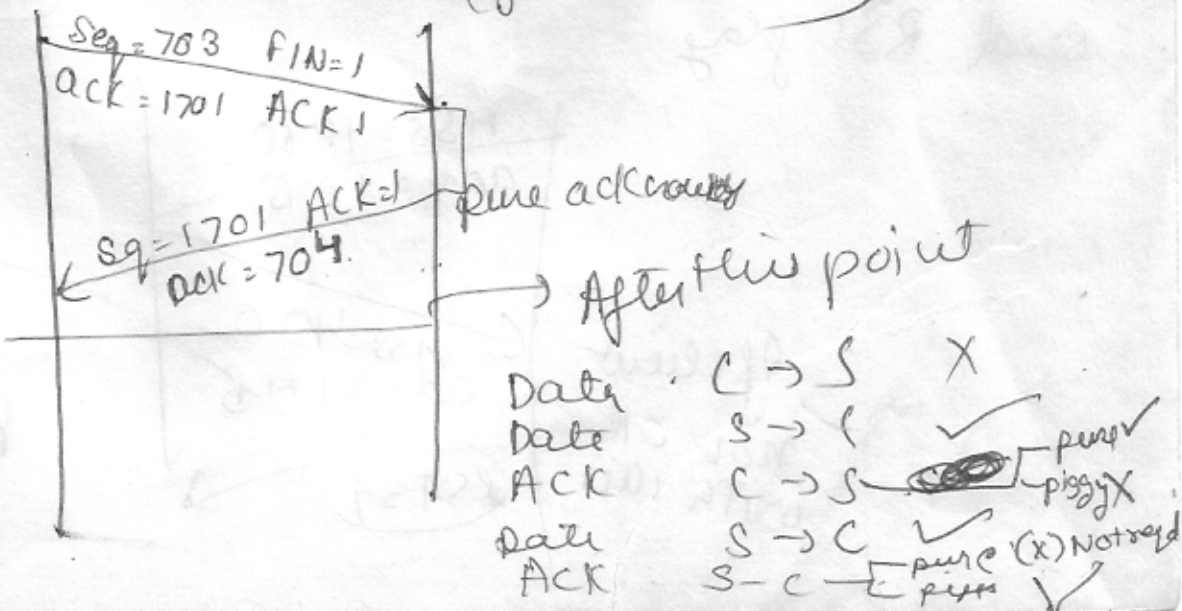
seq no used for this will be used again for next data packet.

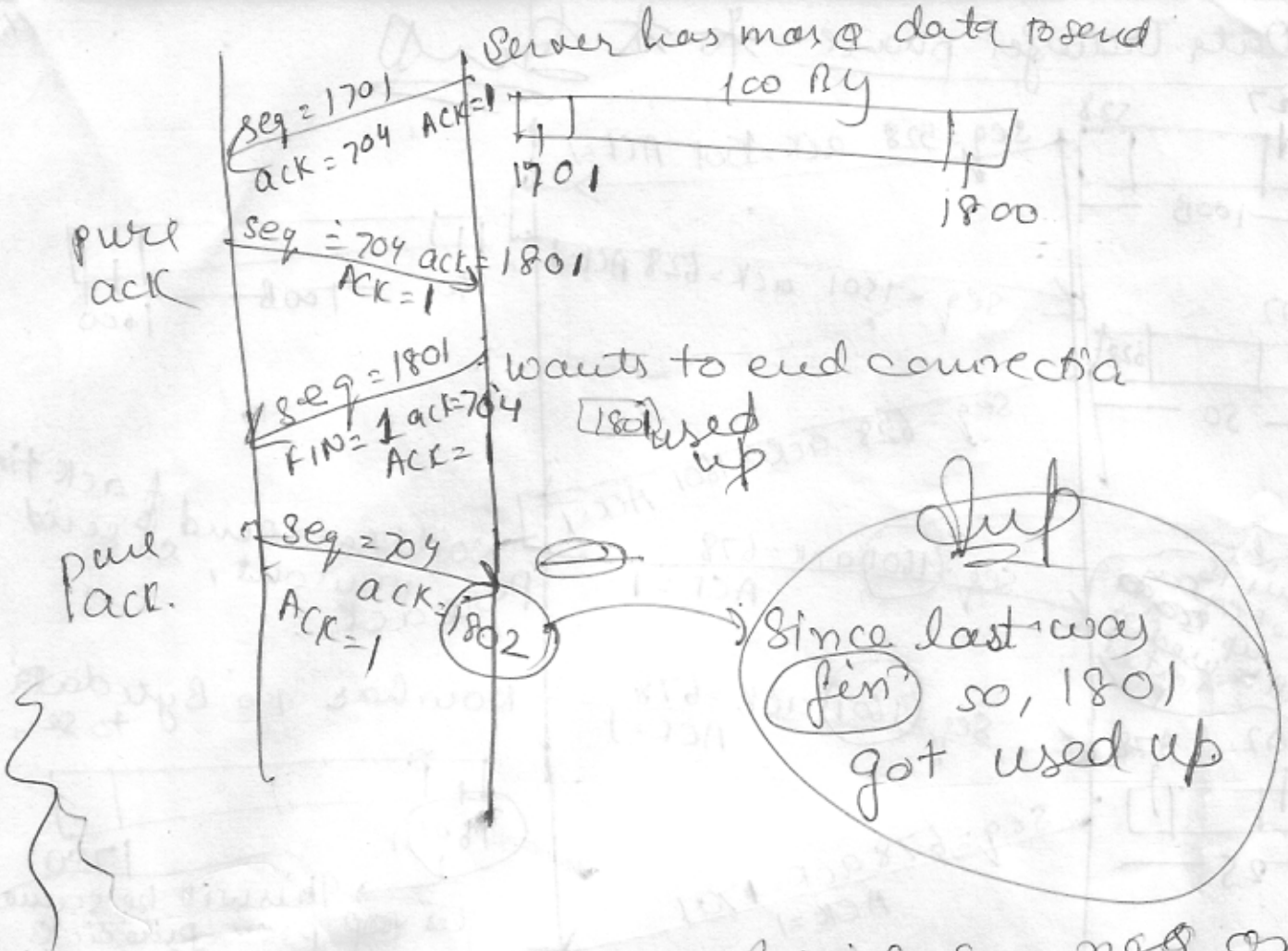
Data Transfer phase ** Imp



pure ack can be distinguished from other packets ∵ data length for pure ack = 0.

CONNECTION TERMINATION (from Client)





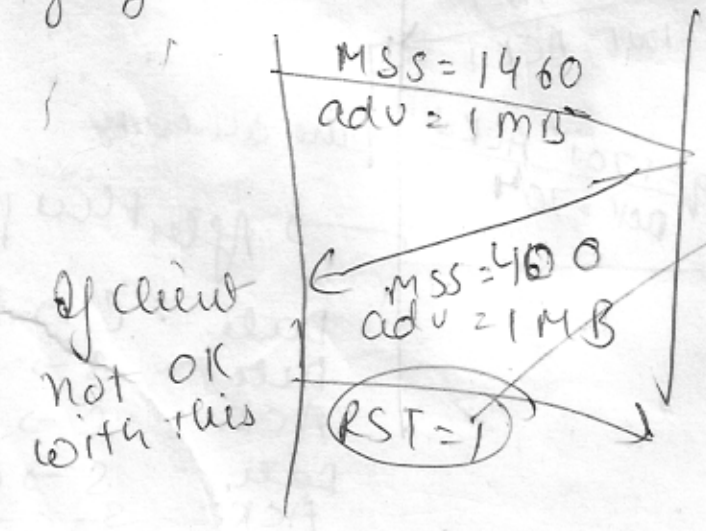
SYN : used to synchronize Initial sq nos on both sides

ACK : To acknowledge field contains a valid value.

FIN : used to terminate the connection

RST : If client is not OK with the parameter during connection establishment then it would send RST flag (MSS & MSB)

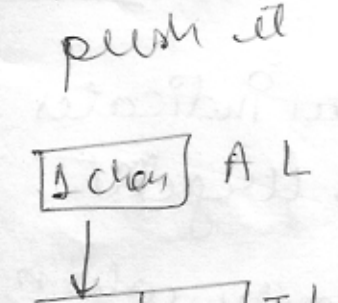
FIN → can terminate connection after establishment



Some app need Interactiveness rather than optimality. In such cases <u>push flag</u> is used to specify TCP not to buffer the data & push it

PSH

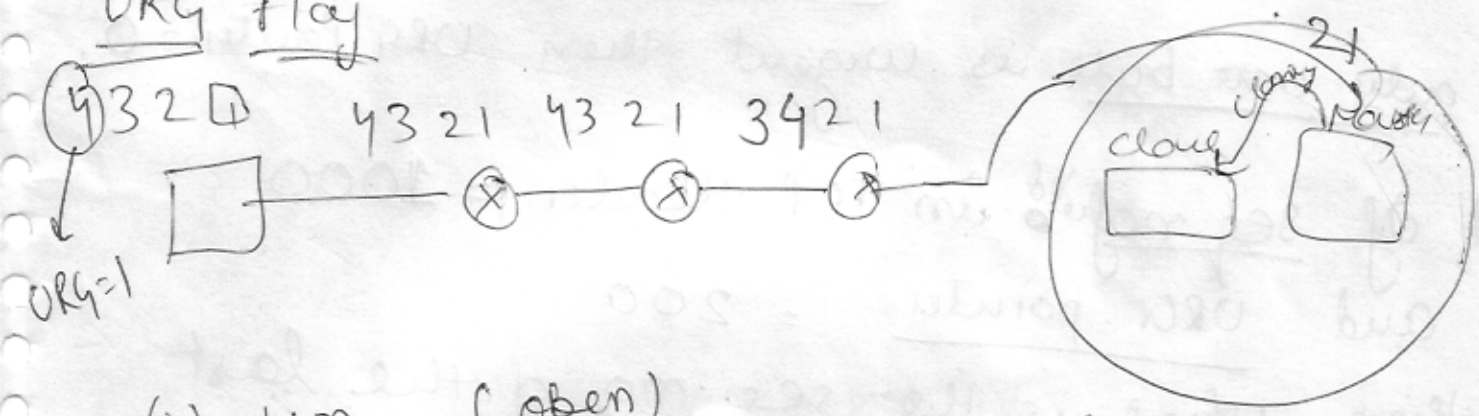
If Telnet
↓
PSH = 1



for 1 byte
80 bits
extra
↓
not optimal
but interactive

If PSH = 0 | 1 | n |
↓
waits 1459
for character
optimal.

URG flag



(1) vim (open)
(2) Modify
(3) Save
(4) ctrl + C [I don't want to save]

for closing ⇒ bK
bit keeper
of
SVN

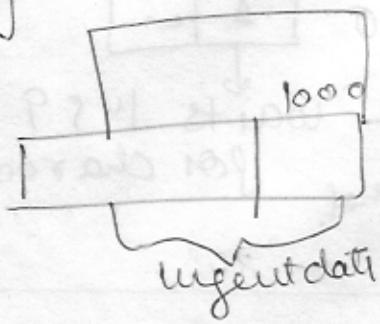
Whenever at TCP, URG = 1 ⇒ In IP, corresponding Priority = 7 increases to
At N/w layer.

∵ Routers work only till Network layer
So, router cannot see the TCP Header which contains URG so, it is necessary that something be done at IP Header to let the router know that this packet needs to be sent fast

URG flag : push & leads to OUT OF ORDER

PSH flag : pushes but INORDER

URG POINTER : This pointer indicates what part of data in the packet is urgent.



works only when urgent data is in starting.

Does not work if urgent data is in middle

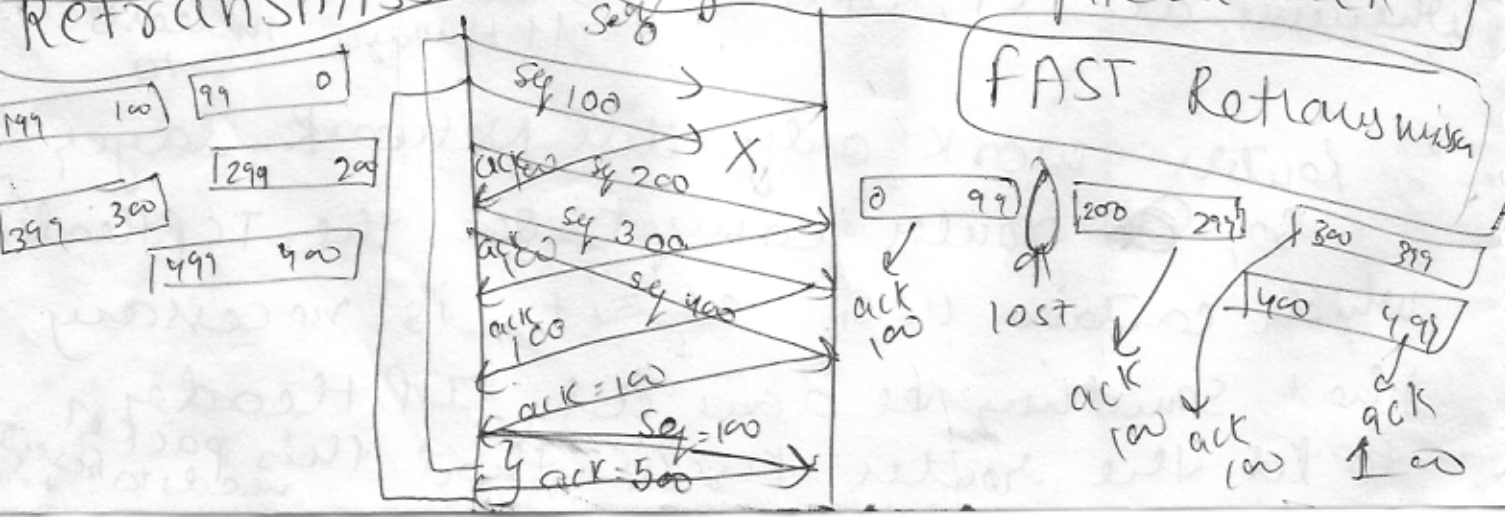
If only one byte is urgent then URG Pointer = 0.

Q) If seq no field in TCP Header = 1000
and URG pointer = 200
then what is the seq no of the last byte which is urgent

A) 1200 (seq no + urg pointer = last urgent byte no.)

Q) # bytes which are urgent = 201

Retransmissions after 3 Duplicate ACK



~~Rec. 3 dup~~

- ack No : Byte No till which data is received completely

If rec $\boxed{0 - 99}$ and $\boxed{200 \quad 299}$

ack = 100 ✓ & not 300 ✗

∴ completely rec. only till 100.

→ (experimentally)

*3 duplicate ack are enough to conclude that the packet is lost so, it should be Retransmitted. Thus it is faster Retransmission ∵ we are sending before the Time Out.

If last packet is lost, then we will have to wait for the Time out (no other option)

CONGESTION :

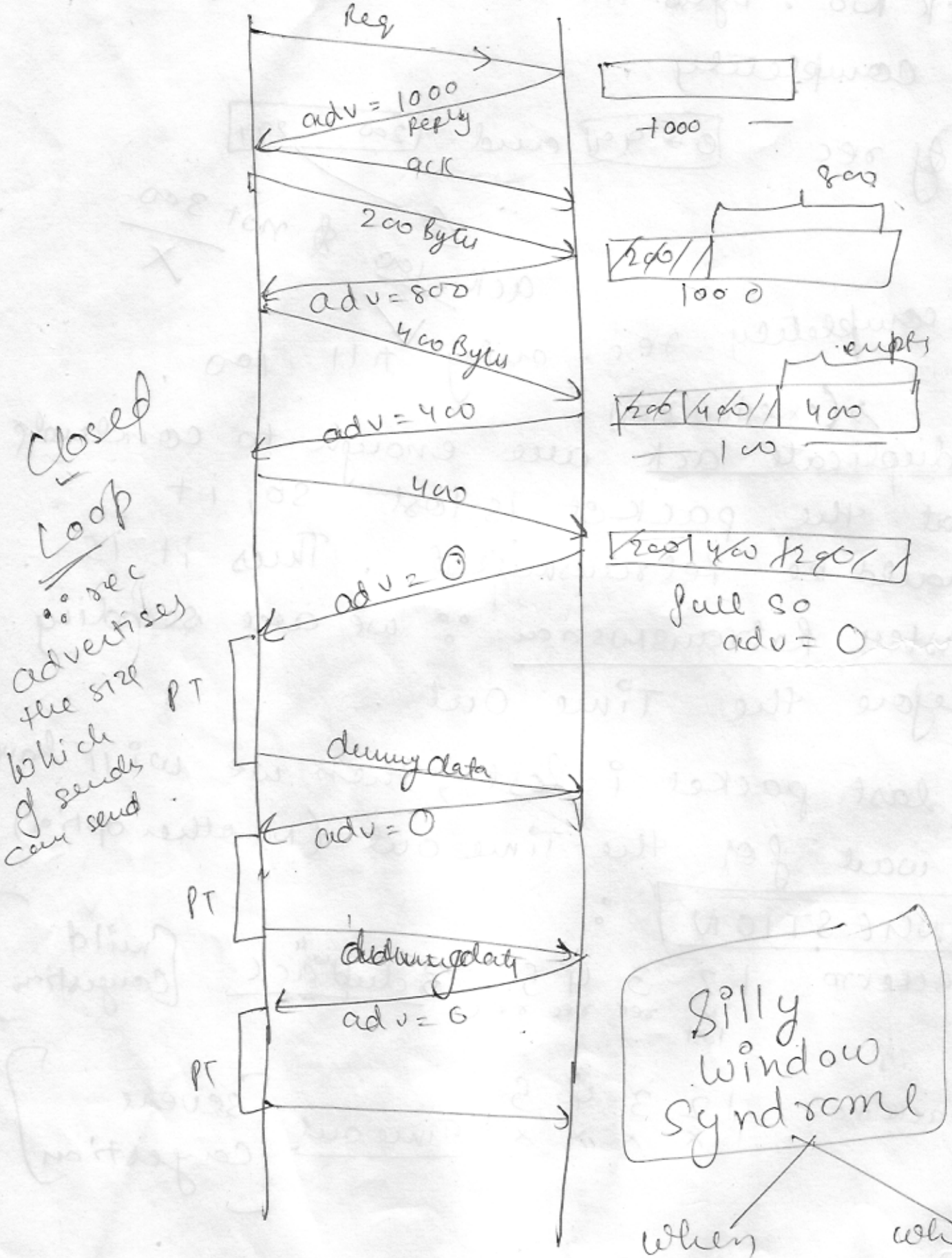
packet no 1 2 3 4 5 3 dup ack [mild congestion]

2 × lost, 3 rec, 4 rec, 5 rec

packet no 1 2 3 4 5 Time out [severe congestion]

× × × ×

FLOW CONTROL IN TCP using ADV window



3 reasons for silly window syndrome :

① whenever receiver advertises its size as 0.

② whenever sender produces only 1 byte at a time.

③ when Receiver consumes only 1 byte at a time.

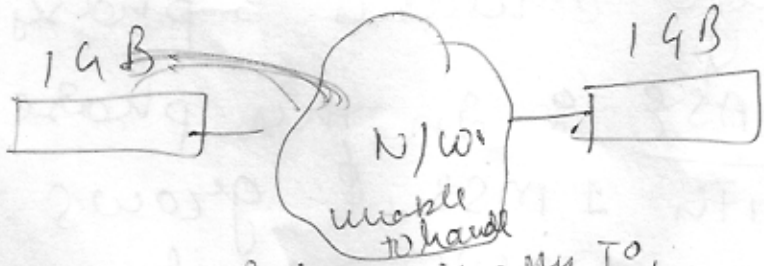
---

using 16 Bits of adv window, a receiver can advertise only $2^{16} = 64KB$ buffer. But this number is not sufficient for many servers so, 14 bits from options are added to make it 30 bit field.

$$\boxed{16 + 14 = 30}$$

adv window, options, max adv bits.

∴ Max window size that a rec can advertise $= 2^{30} = 1GB$



Retransmit after T0, again not able to handle.

So,

we must SLOW START, increase # Segments exponentially.

Suppose

C_R = 16000 Bytes

MSS = 1000 Bytes

W_S = 16 segments

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

Threshold = $\frac{W_S}{2} = \frac{16}{2} = 8$ → Till this point ↑se exponentially, after this linearly.

TCP Congestion Control Algo :

A Rec. will advertise its capacity using adv window. But sender should not send the capacity advertised ∵ the N/W may not be in a position to handle the data.

∴ one more window called CONGESTION WINDOW is used which starts from 1 MSS or 2 MSS (depending on implementation) and a sender should send $\boxed{\text{Min}\{W_{congestion}, W_{sender}\}}$

Congestion Control algo contains 3 phases:

① SLOW START PHASE : In this phase, window starts with 1 MSS & grows exponentially till we reach threshold where threshold = half of max window size

② Congestion Avoidance : This phase starts from threshold & ↑es linearly till we reach

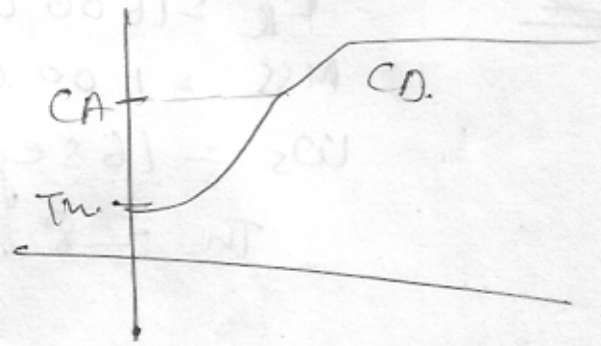
③ <u>Congestion Detection</u> : In this phase, congestion can be detected either by

a) 3 duplicate ACK

b) Time out.

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

SLOW START (grows exponential) — CA



Max R = 32000 B
MSS = 1000 B
Max WS = 32.

<u>Case A</u> : <u>3 dup ack</u> : Whenever 3 duplicate ack occur, it indicates that there is <u>mild congestion</u>. In this case, new threshold is set to <u>half of current window size</u> and new congestion will be set to <u>new threshold</u> & algo enters congestion avoidance face.

<u>Case B</u> : <u>Time Out</u> : If TO occurs then it indicates severe congestion then new threshold is set to half of the <u>current window size</u> & new window size <u>will be 1</u> & algo enters <u>Slow start</u>.

Q Maxm Rec. Capacity = 16000 B

MSS = 1000 B

~~SQ Segments~~

then after how many Round Trip Times (RTTs) Sender will be able to send maxm window of 16000 B if slow Start starts with 1 MSS.

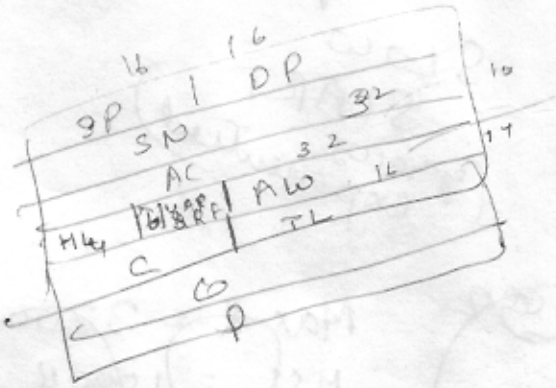
Ans

$C_R = 16000$

$MSS = 1000$

$W_S = 16$ segment

$Th = 8$. $\left(\frac{16}{2}\right)$



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

11 gaps = 11 RTTs

Ans after 11 RTTs, Sender will receive max W_S

Q $C_R = 32000$ B $\quad W_S = 32$

MSS = 1000 B $\quad Th = 16$.

then RTTs to send max W_S.

1, 2, 4, 8, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32

Ans 20 RTTs

eg3 Th = 16

newTh = 4 newTh = 5 newTh $\lfloor \frac{9}{2} \rfloor = 4$

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

TO 3dup packets TO

... 1 6 1 6 1 6 1 6 1 2 4 8 9 10 ...

TO newTh = $\frac{16}{2} = 8$

NOTE : Always sender should send minimum of advertisement window & congestion window.

TIMER MANAGEMENT

TCP Responsibilities:
- congestion control acts as protective layer for Routers
- Timers Mgmt
  - → TO
  - → ACKT
  - → PT
  - → KAT

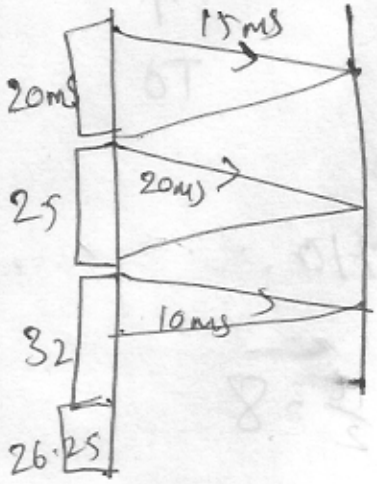
① Time Out Timer : In case of DLL, RTT over a single hop doesnot change randomly. ∴ Static TO timer works.

In case of TL, the end to end RTT may change based on the traffic. So static TO time does not work. As the traffic rises, RTT will increase & we have to wait for long time. So, we use Dynamic algorithms.

(a) Basic algorithm

$TO = 2 * RTT$

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



$IRTT = 10ms$

$TO = 20ms$ & $ARTT = 15ms$

then

$NRTT = (0.5)(10) + 0.5(15)$

$= 12.5$

$TO = 2 * NRTT$

$= 25.$

$\Rightarrow$ $IRTT = 12.5$

$ARTT = 20ms$

$NRTT = 16.25ms$

$TO = 2 * NRTT$

$= 32.5$

$\alpha = 0$ (static)

$\alpha = 1$ (Dynamic)

↳ we will a value b/w 0 & 1

$\Rightarrow$ $IRTT = 16.25$

$ARTT = 10$

$NRTT = 13.125$

$TO = 26.25$

Disadv

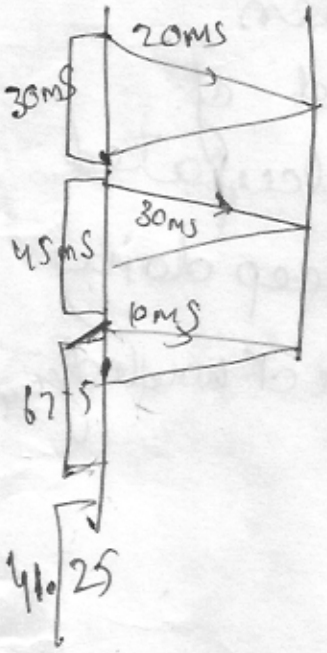
$TO = 2 * RTT$

↳ No logic

→

(b) Jacobson's algo

$TO = 4 * D + RTT$



IRTT = 10ms

ID = 5ms

TO = 4 * D + RTT

= 4 * 5 + 10

= 30ms

ARTT = 20ms

AD = 10ms

ND = α (ID) + (1-α)(AD)

= 7.5ms

NRTT = α (IRTT) + (1-α) ARTT

= 15ms

TO = 4 * ND + NRTT

= 30 + 15

= 45ms

IRTT = 15ms

ARTT = 30ms

ID = 7.5

AD = 15

ND = 11.25

NRTT = 22.5

TO = 4 * ND + NRTT

= 45 + 22.5

= 67.5

ID = 17.25
AD = 12.5
ND = 11.82 23.75
NRTT

IRTT = 22.5

ARTT = 10

ID = 11.25

AD = 12.5 23.75

ND = 11.875

It should come 41.25

153.75 (5 2.5)

NRTT = ~~38.75~~ 16.25 (77.5)

TO = 4(34.7) + ~~38.25~~ 16.25 (47.5)

= 138.8 + ~~38.75~~ 16.25

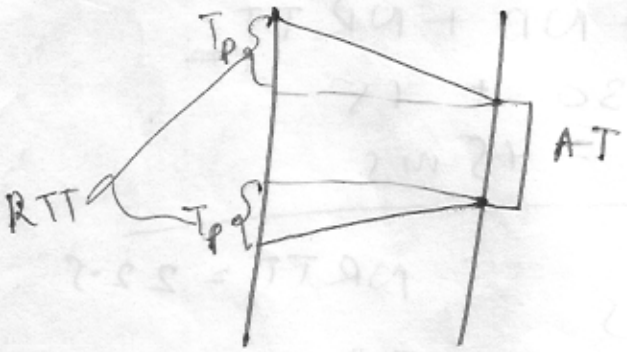
KARN'S MODIFICATION

Whenever there is a Timeout at initial stage, basic algorithm and Jacobson's algorithms fails then Karn's modification says that instead of sending the packet with new calculated TO, just double the previous TO. Keep doing this until we get the 1st acknowledgement within the TO.

② ACKNOWLEDGEMENT TIMER

Ack timer is used to implement ① cumulative ack as well as ② piggback ack,

Relation B/w TO timer and ACK Time :



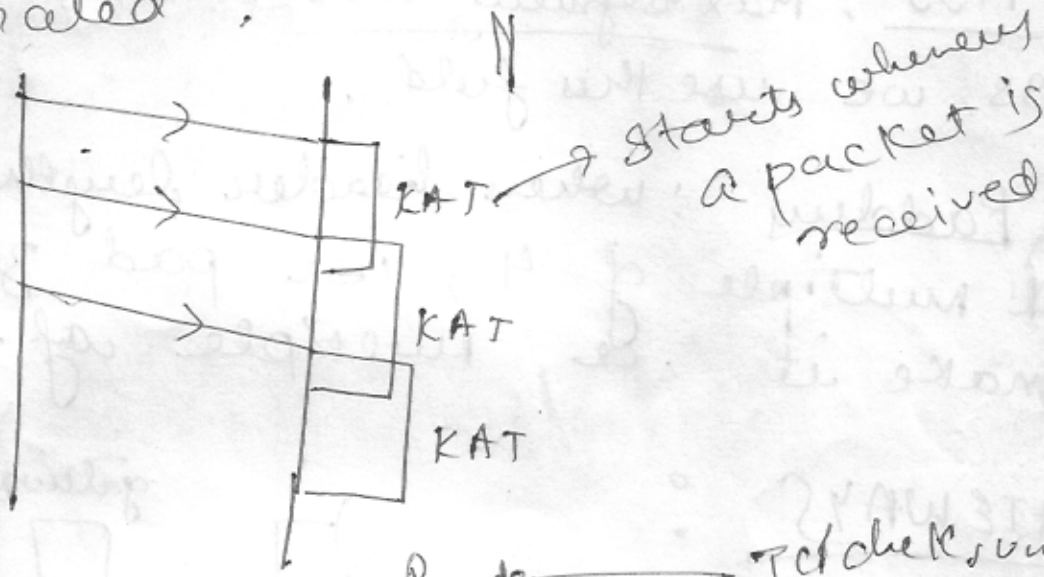
RTT + ACK Time < TO

③ Persistence Timer

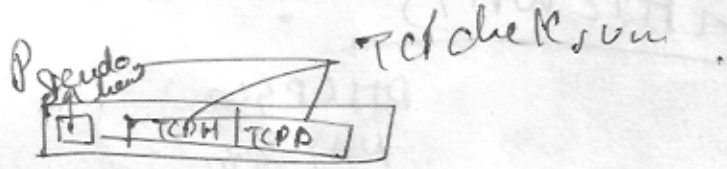
Whenever Receiver announces its windowsize = 0, then sender should poke/ping periodically to find out its buffer capacity.

∴ We use a timer called Persistence timer.

④ Keep Alive Timer : This timer is used to keep track of idle connections i.e if a server does not receive any data from a client for a specific amount of time, the connection will be terminated.



TCP Checksum :



TCP checksum is calculated on
- TCP data,
- TCP Header,
- Pseudo header from IP.

includes IPS, IPD & 2 more fields

Q Why IP header is included in calculation of TCP Checksum

A Even if packet is delivered to a wrong host, there will be 1 more checkpointing at TCP

Q Why only Pseudo header is used & not actual header

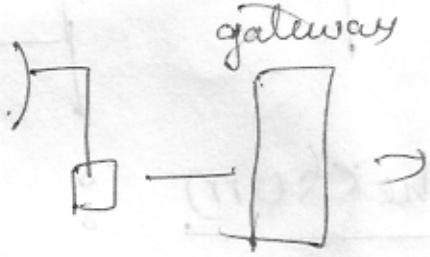
A ∵ most of the fields in IP header change

Options

- Timestamp : It is used when LT > WAT.
- Adv field (14 bits) : To make it 1 GB.
- MSS : Max Segment Size : When negotiating MSS, we use this field.
- Padding : When header length is not a multiple of 4, we pad zeros to make it a multiple of 4.

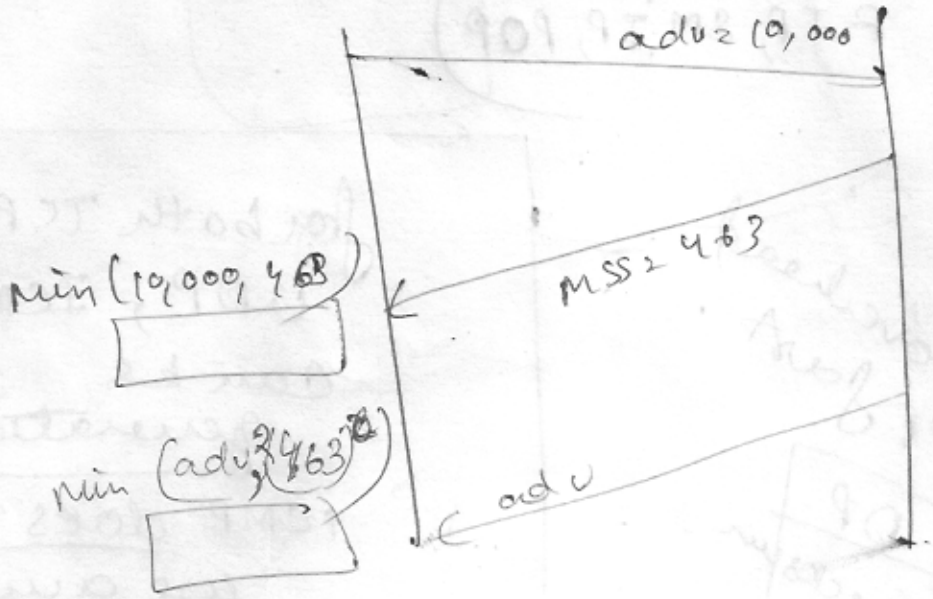
GATEWAYS :

DHCP GW
NAT GW
DPI



- gateways are connecting device which contain all the layers.
  " can establish connection with the server on behalf of client so, they also ~~called~~ act as proxies.
- Gateways can do DPI ie D packet Inspection and can look into data available at Application layer
- Gateways can buffer the data as well as reduce the speed of servers (sync b/w client & Server)

Buffer : when rec. cannot rec. more.

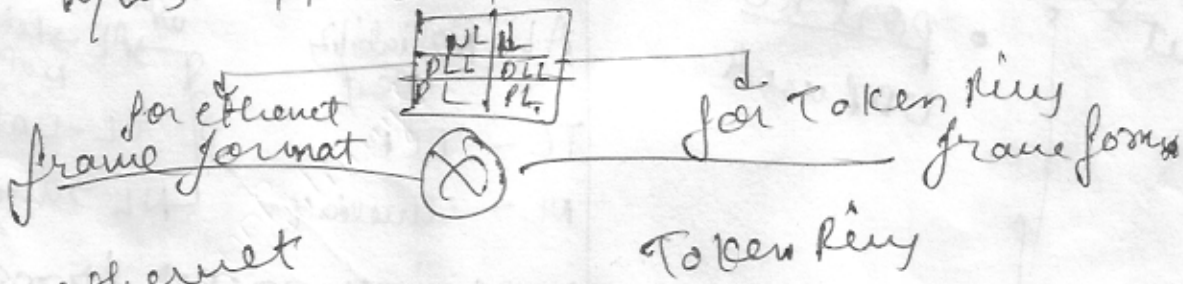


divides the message into proper sizes.

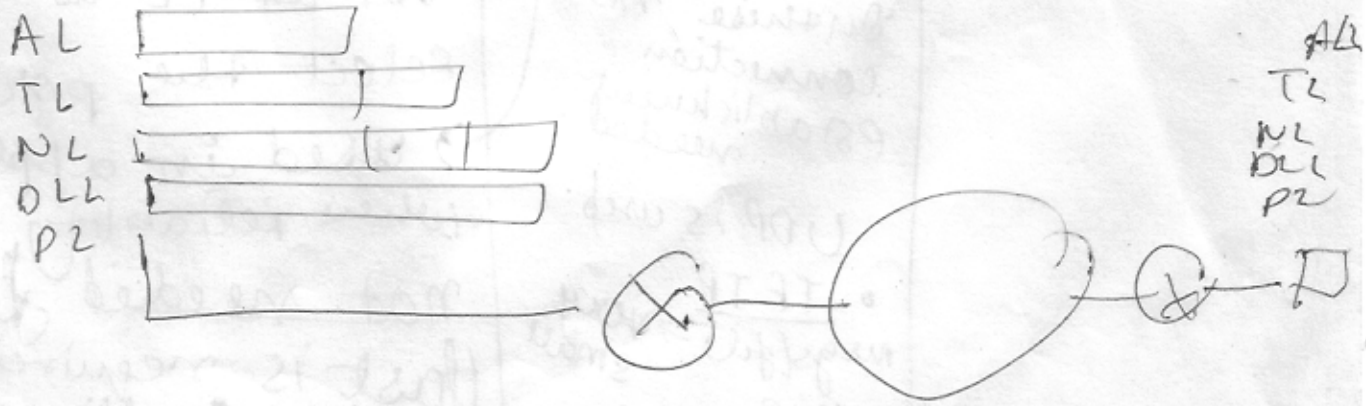
TL will never give more than IP datagram can hold.

No fragmentation at sender or N2.

A router can have more than 1 DLL depending on Type of N/ws if connected



Router : Different protocols can be implemented at one layer.



Application Layer

(DNS (UDP), HTTP (TCP), FTP (TCP), SMTP (TCP), POP (TCP))

UDP → less overhead so, fast.

SP	DP
length	checksum

Tree with more Breadth & less Depth ⇓ efficient search / faster.

- port no UDP used

- NTP, DNS ↓ 1 req, 1 reply. In these no connection establishment needed. UDP is used
- TFTP msgs/file → very small. UDP used

for both TCP & UDP, ICMP can be generated

ICMP does not use any TL protocol

TCP
- Reliability
- Slow (establish connection etc.)

AL → Reliability need
TL — TCP
NL → Unreliable

UDP: AL → Reliability not needed
TL — UDP
NL → Unreliable

UDP: gives packet from NL to AL (user). Only functionality by UDP in TL is to select the port at AL

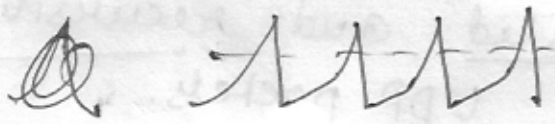
→ used in applications where Reliability is not needed but fast is requirement eg. Multimedia, Skype, online game

User Datagram Protocol :

- whenever app needs fastness rather than reliability or
- whenever 1 req, 1 reply are enough then, UDP is used at Transport layer.

ex
* multimedia protocols (online games, Skype)
* DNS (1 req, 1 rep)
* NTP (N/w Time protocol
* TFTP (Trivial file transfer Protocol)
* Quote of the day
* N/w News protocol.
* Broadcasting applications like weather reports, Stock market uses UDP

- UDP is used to overcome the disadvantages of TCP, like connection establishment, connection termination, congestion control, flow control etc.
  ↳ results in variable data rate.
- UDP can be used for applications which need constant Data rate



Application Layer Protocols

- Only DNS uses UDP.

TCP
↓
Search time is less.

DNS [Domain Name Server]

In DNS we have 3 domains :

(1) Generic
.com , .edu , .mil , .org

(2) Countries
.in , .us , .uk

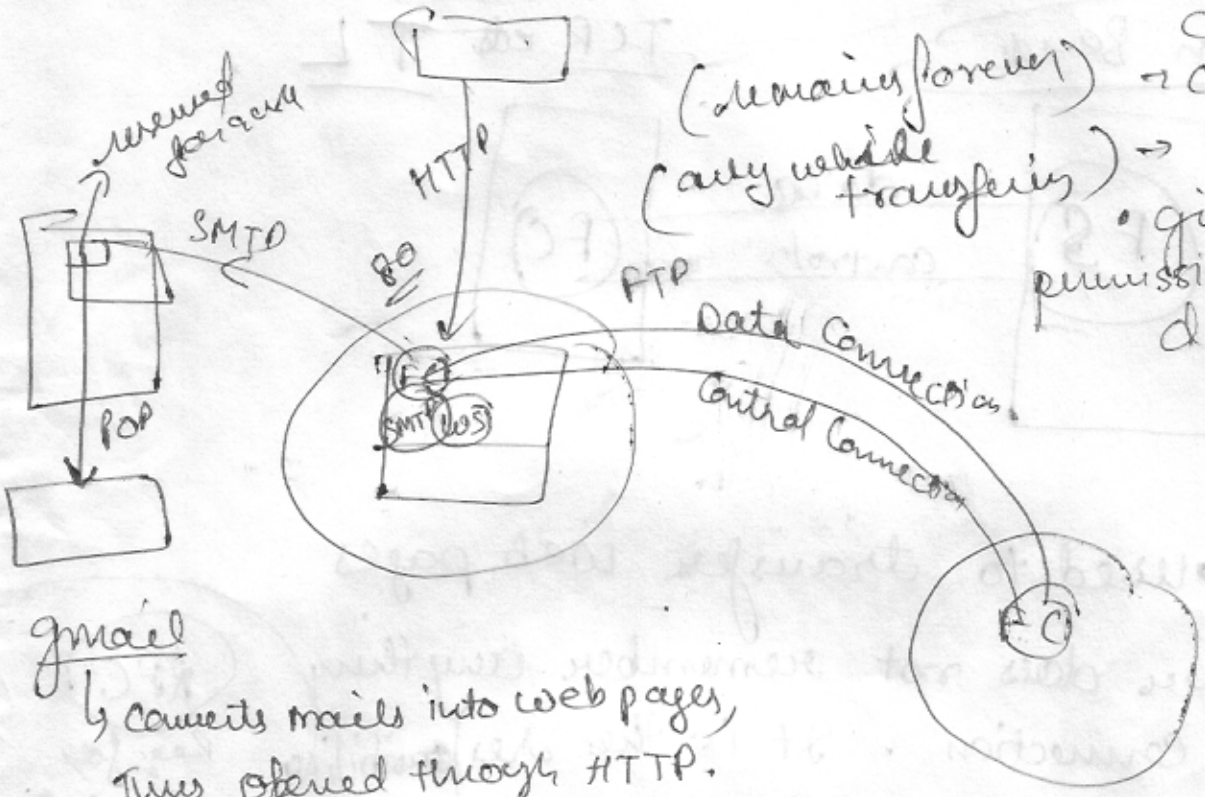
(3) Inverse Domain

given IP reaches to Domain not IP to Domain name

crazy 11.1.2.3
↑ 11-1-2-3

HTTP

FTP
- No hyperlink facility
- Has 2 connections
  - (remains forever) → control
  - (only while transferring) → Data
- gives the permission to browse directories



gmail
↳ converts mails into web pages, Thus opened through HTTP.

FTP : In FTP, 2 connections are used, in which 1 connection is used for CONTROL.

In Control connection, commands will be send like
→ open a directory
→ Remove " "
→ Statistics of a a directory
→ Create a directory etc.

nslookup → to get IP

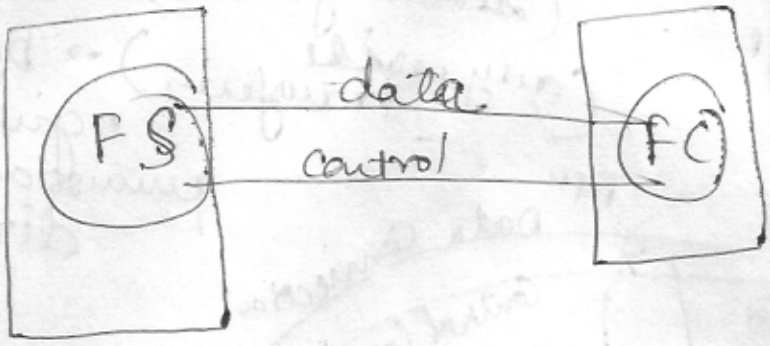
2nd connection is DATA
whenever data has to be transferred, this connection will be used. This connection will be closed after the data is transferred

Since data & Control travel in 2 different connections, It is also called as

OUT OF BAND protocol.

FTP is a protocol that needs Reliability ∴ it uses TCP at TL.

HTTP → In Band.



HTTP : used to transfer web pages.

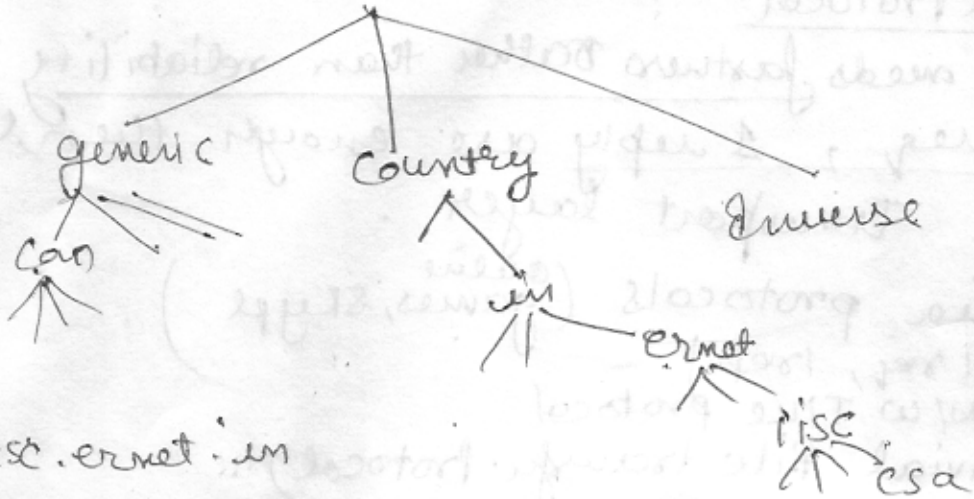
- HTTP server does not remember anything about the connection. It is the responsibility of client to save the data about the connection. It is generally called Cookie
  ↳ storing info on client itself.

RFC
Req for comments.
rfc.org

Server → Stateless

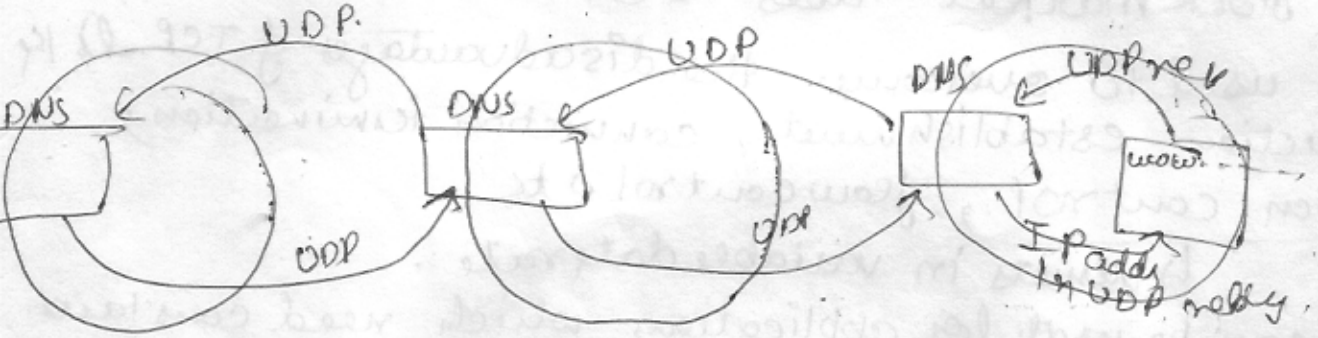
- HTTP needs reliability. ∴ we use TCP at TL

→ get
→ Status numbers 300 onwards



csa.iisc.ernet.in

In order to make the searching easy, entire database is represented as tree.



DNS DB is distributed and recursive searching will be used using UDP packets.

If our N/w DNS does not contain the entry, then DNS sends UDP req to another till we have a DNS which has that IP addr.

This is DNS overhead.

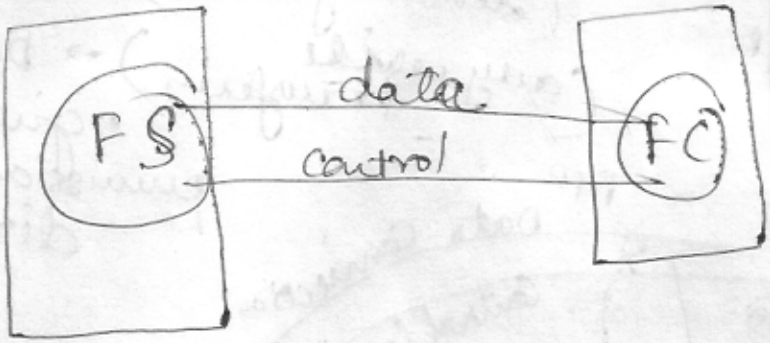
DNS entry | DN | IP | → uploaded at every DNS, if not uploaded then on req pacc, it will get updated from requests to ISP.

Since data & Control travel in 2 different connections, It is also called as

OUT OF BAND protocol.

FTP is a protocol that needs Reliability ∴ it uses TCP at TL.

HTTP → In Band.



HTTP : used to transfer web pages.

- HTTP server does not remember anything about the connection. It is the responsibility of client to save the data about the connection. It is generally called Cookie
  ↳ storing info on client itself.

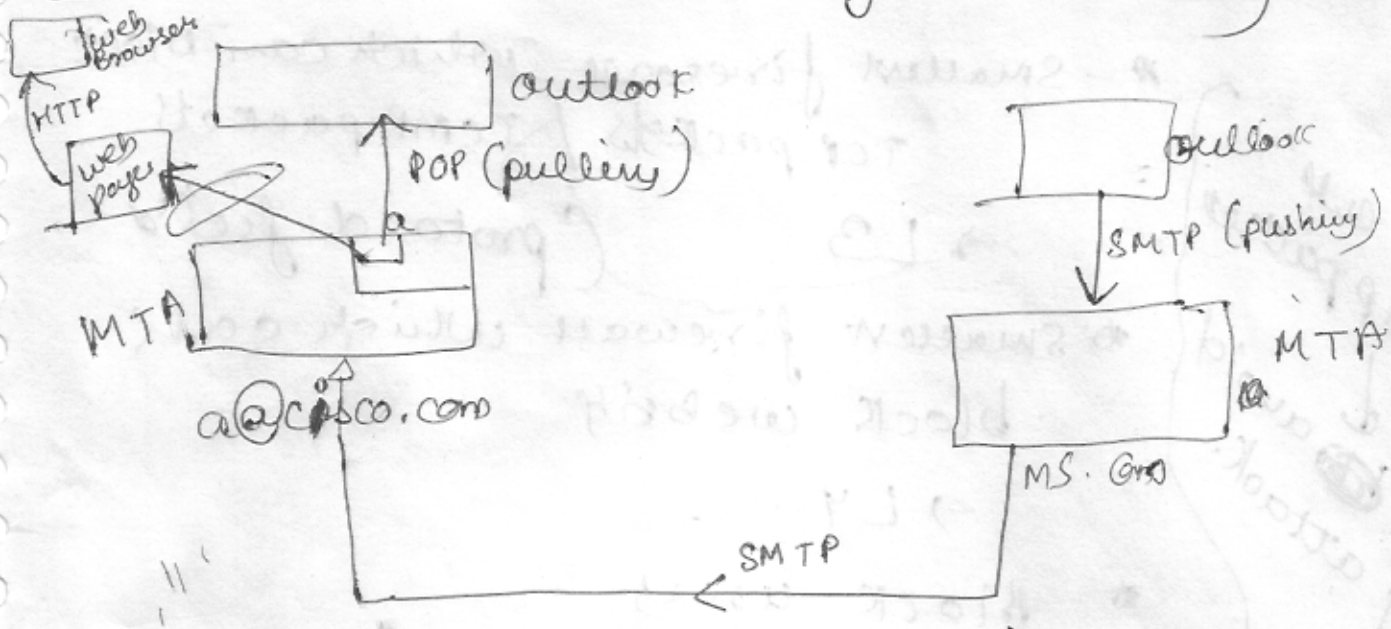
RFC
Req for comments.
rfc.org

Server → Stateless

- HTTP needs reliability. ∴ we use TCP at TL

→ get
→ Status numbers 300 onwards

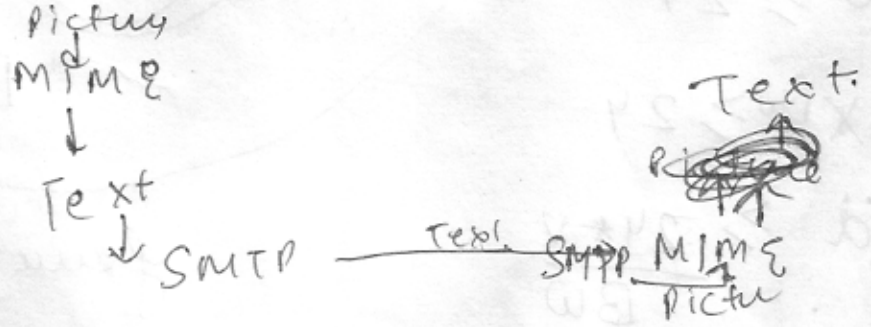
SMTP (Simple Mail Transfer Protocol)



SMTP is used to push the email and POP is used to pull the email. Both the protocols are purely TEXT BASED protocols.

∴ we cannot send pictures or .exe files which is not a text.

MIME : multimedia Internet mail extension.



MIME is a set of s/w tools which is used as a supplement to SMTP & POP to convert picture (non text) to text and Text to Picture.

So, with the help of MIME, SMTP can send pictures.

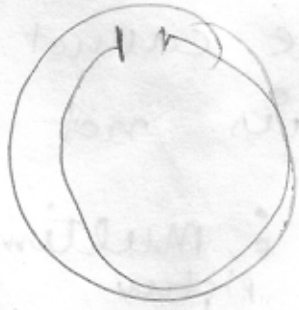
FIREWALL :

* Smallest firewall which can block TCP packets / ICMP packets
  → L3 (protocol field
* Smallest firewall which can block websites
  → L4
* Block users
  → L5

Block all ICMP packets to avoid ping attack.

Token Ring

Capacity of wire = $BW \times T_p$

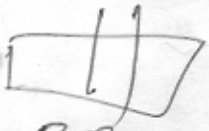


$T_p \times BW \geq 24$

$\frac{d}{v} \times BW \geq 24$

$d \geq \frac{24 * v}{BW}$

Minm length of Token Ring



3 Bytes
⇓
frame format

$L_{min} \geq 2 * T_p * BW$

A Token ring should atleast be able to hold the smallest frame ie token which is 24 bits ∴ capacity of token ring must be atleast 24 bits.

Q) If BW in a Token Ring is 4mbps and velocity of signal is $2 * 10^8$ m/s then what is the minm length of the tokenring

A)

$$d \geq \frac{\cancel{24} * 2 * 10^8}{\cancel{2 \cdot 4 * 10^8}}$$

$$= 1200 m$$

D Flip Flop (Delay flip flop):

Q) How many bits of artificial delay should be introduced so that token ring can still be constructed with 1 km in abap ring.

Ans) we need a delay of 200m at a node. So,

200m → sec → bits

$$\frac{200 m}{V} = \frac{200}{2 * 10^8} sec \rightarrow \frac{200}{2 \times 10^8} * BW \text{ bits}$$

$$\rightarrow \frac{\cancel{200}\,100}{\cancel{2} \times 10^8} * 4 \times 10^6$$

$$\rightarrow 4 \text{ bits}$$ minimum

Q If every station introduces 1 bit delay then how many minm stations are required

So that we can introduce 1 km Token Ring 142

Ans) 4 (minimum)

More than 4 stations also works.

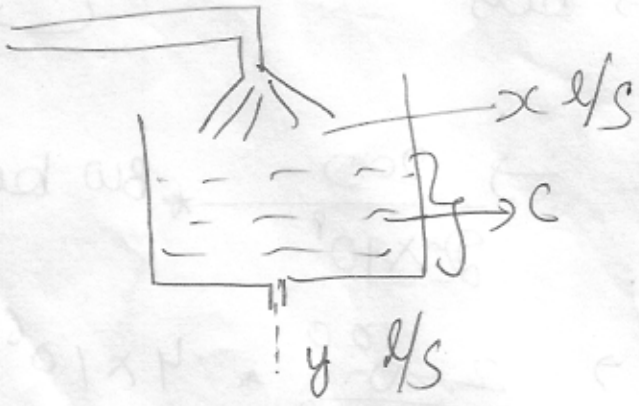
In Token Ring, "n" stations & "d" length b/w 2 stations then (nd) is the total length of wire.

Calculation of Minm No. of stations;

If every station separated by a distance of 100m & K bit delay

$$\left(\frac{n * 100}{V} * BW\right) + (n * K) \geq 24$$

TOKEN BUCKET / LEAKY BUCKET



$C + xt = yt$

$t = \frac{C}{y-x}$

If $y < x$ then overflow

Let initial capacity of buffer is B bits

Inflow = x bits per sec

Outflow = y bps then

time taken to empty the bucket

$\Rightarrow$ outflow = capacity + inflow

$$yt = C + xt$$

$$t = \frac{C}{y-x}$$

FIREWALLS :

L5 → most costly.

There are 3 types of firewalls :

1) Layer 3 : It contains physical layer, DLL & NL.

- It can block/filter packets based on IP address.
- It blocks a Host completely.
- It is also called packet filtering firewall.

So, the packets can be filtered based on IP Header

IP header contains "Protocol" field !

So, L3 firewalls can block ICMP, IGMP, TCP, UDP traffic.

Smallest firewall that can block ICMP packets $\Rightarrow$ L3.

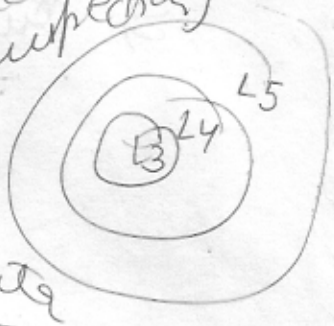
2) Layer 4 : It contains PL, DLL, NL, TL.

- It can block a host completely and also a port/service or a particular service on particular host.
- We can block HTTP, FTP, SMTP etc.

3) Layer 5 : • Proxy firewall

• It contains all layers.

This firewall is capable of DPI (Deep packet inspection)

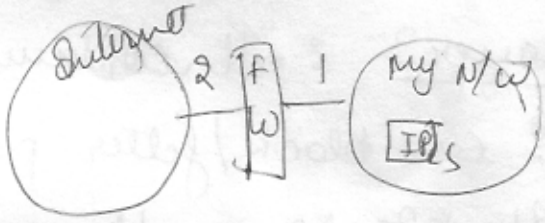


So, a packet can be blocked by IP, or port no or application data

So, some users will be blocked using username & password.

• avi
• mov
in app data

Router : multiple interfaces



Firewall ≥ 2.

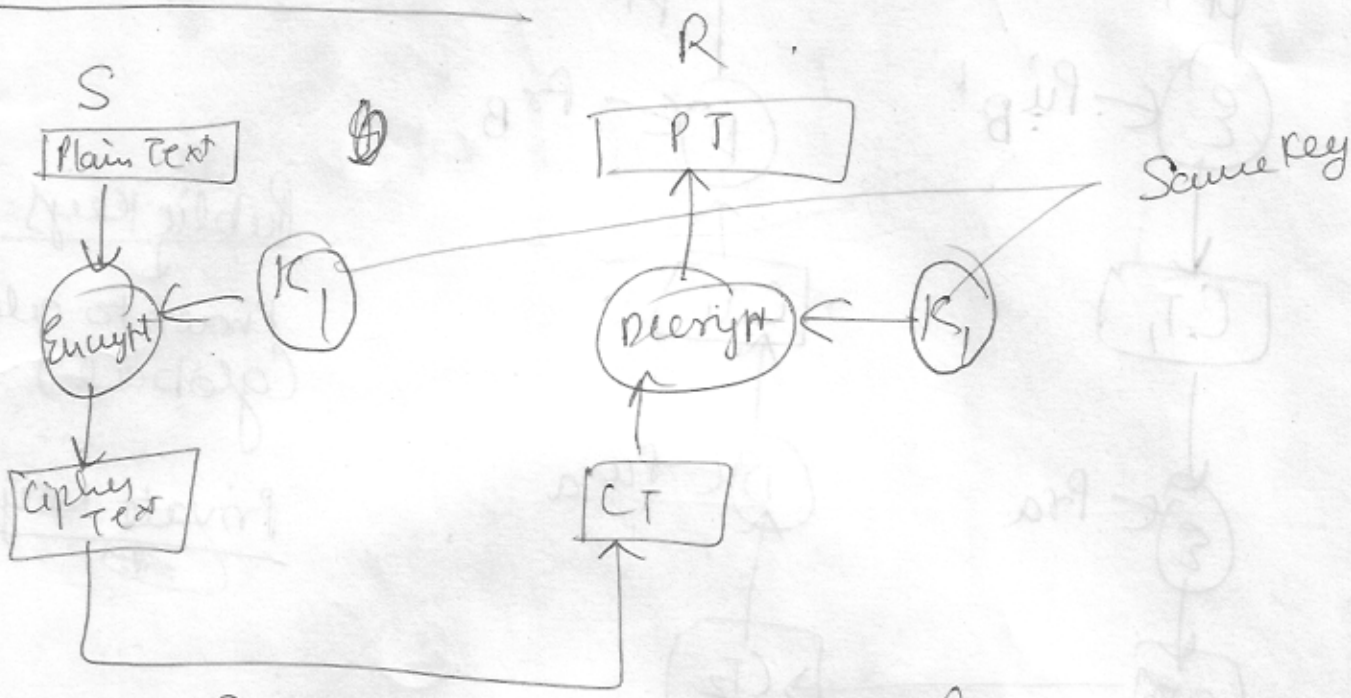
Table at firewall:

	SIP	DIP	SP	DP	Interface	User Type
1	—	—	—	80	1	—
2	—	IPcs	—	23	2	non emp
3	IPcs	—	—	—	1	—

Priority table.

1. Block website browsing from my N/W
2. Block login from outside emp on IPcs via telnet.
3. No data should be sent out of IPcs

CRYPTOGRAPHY



SYMMETRIC KEY Cryptography.

I/ Symmetric key Encryption :

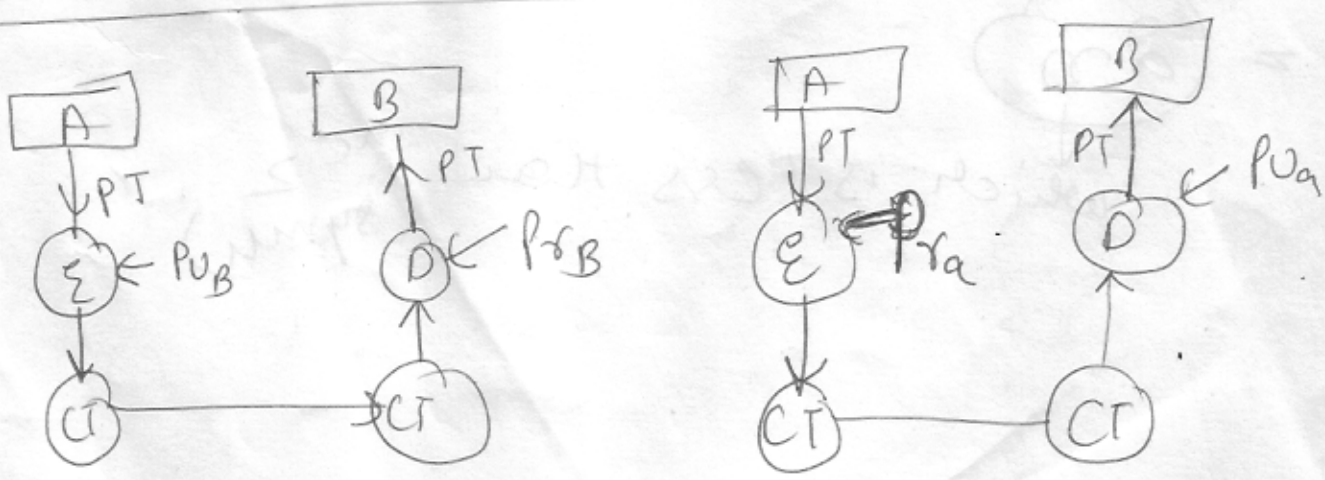
Uses same Key

Disadv :

① Key exchange.

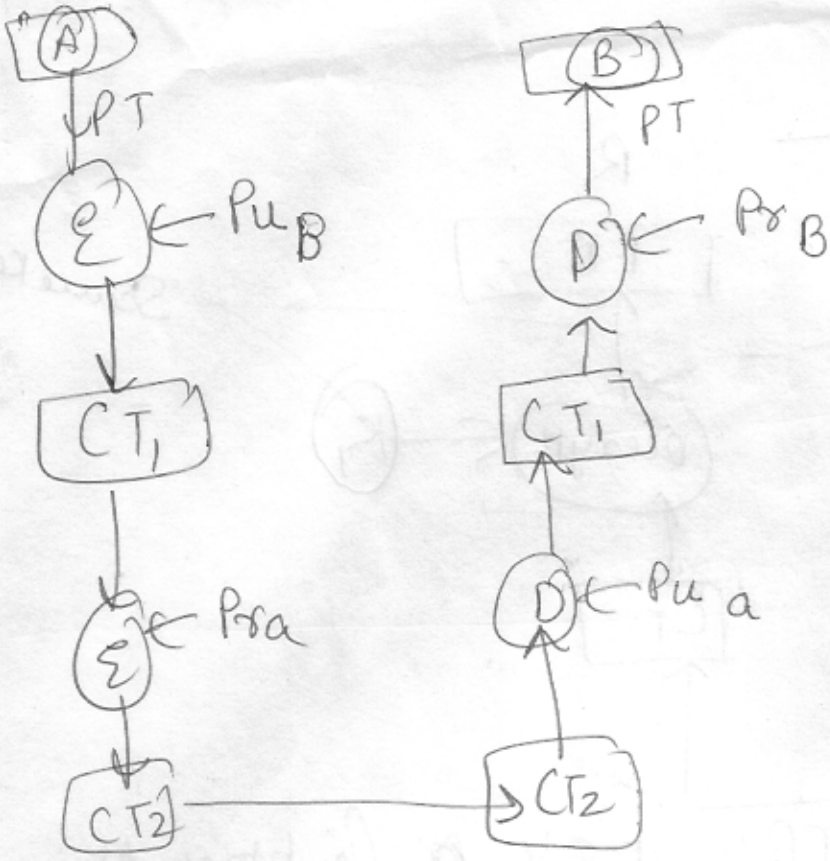
② If 'n' systems want to communicate with each other securely then we need $\boxed{{}^nC_2}$ keys.

II PUBLIC/PRIVATE KEY Encryption



on Encryption

only Authentication & Digital Signature



Public Keys
↓
Known to all
(global)

Private Keys

Encryption + authentication + Digital signature

Q If 'n' stations want to communicate securely using Public key, Private key Encryption then how many keys are req

$= 2n$

which is less than ${}^{n}C_2$ (Symm)

① Fermet's Theorm :

If "p" is a prime no and 'a' is positive integer less than P then,

$$\boxed{a^{p-1} \cong 1 \bmod P}$$

eg $P = 5$

$2^4/5 = \text{Rem } 1.$

② Euler's Totient function :

$(\phi(n))$

It is the no. of +ve integers less 'n' which are relatively prime to n.

eg $\phi(10)$ contains 1, 3, 7, 9

So, $\phi(10) = 4$ → count of these nos

eg $\phi(5) = 4$

(1, 2, 3, 4)

$\phi(\text{prime no}) = \text{prime no} - 1$

$\phi(p) = p - 1$ where p is prime no.

Imp $\phi(35) = \phi(5 \cdot 7)$ ← prime factorization

$= \phi(5) \cdot \phi(7)$

$= 4 \cdot 6$

$= 24$

$\phi(25) \neq 16$.

Relatively prime: gcd = 1

If $N = p * q$ and $p \neq q$

only then $\phi(N) = \phi(p) * \phi(q)$

no = product of 2 DISTINCT PRIME numbers.

3) CHINESE REMAINDER ~~NUMBERS~~ THEORM

It states that all the integers in certain range can be expressed as residue a set of pairwise relatively prime moduli

	2	5
0	0	0
1	1	1
2	0	2
3	1	3
4	0	4
5	1	0
6	0	1
7	1	2
8	0	3
9	1	4

1) DISCRETE LOGARITHM (order)

If a & b are relatively prime nos then there exists atleast one integer 'm' in such a way that $\boxed{a^m \equiv 1 \bmod b}$

The smallest no 'm' is called Discrete Logarithm.

If $a^{m} \bmod b$

Base

$2^1 \bmod 7 = 2$
$2^2 \bmod 7 = 4$
$2^3 \bmod 7 = 1$
$2^4 \bmod 7 = 2$
$2^5 \bmod 7 = 4$
1

order = length of period

$= 3$.

order $3 \bmod 7 = 6$

$3^1 \bmod 7 =$
$3^2 \bmod 7 = 2$
$3^3 \bmod 7 = 6$
$3^4 \bmod 7 = 4$
$3^5 \bmod 7 = 5$
$3^6 \bmod 7 = 1$

(or) order 6

order $= \phi(7)$
$6 = 6$

Thus, 3 is primitive root of 7.

If in $a \bmod b$ order $= \phi(b)$ then 'a' is called Primitive Root of 'b'.

Always order divides $\phi(b)$

$a^{m} \equiv 1 \bmod b$

RSA algorithm

RSA is used to generate Public & Private Key

Step 1

Choose 2 prime nos p & q.

Step 2

Find $n = p \times q$

Step 3

Compute $\phi(n) = (p-1)(q-1)$

Step 4

Choose a no. 'e' such that

$e < \phi(n)$ & relatively prime to $\phi(n)$

ie $GCD(e, \phi(n)) = 1$.

Step 5 Compute 'd' such that

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

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

$$\boxed{\begin{aligned} \text{public key} &= (d, n) \\ \text{private key} &= (e, n) \end{aligned}}$$

Given (d, n), computation of (e, n) is infeasible

Encryption of message - 'm' using Public key of Reciever (d, n)

$$\boxed{m^d \bmod n = \text{Cipher Text}}$$

Decryption using private key

$$\boxed{C^e \bmod n = m}$$

using RSA, Key (K) is safely transmitted & message transmission is done through symmetric encryption using the shared Key (K)

En $\quad K^d \bmod n = C$

De $\quad C^e \bmod n = K$.

RSA
↳ only 1 system involved
↳ can ~~used to~~ share data & key both

Diffie Hellman

Main purpose : To exchange the Key.

- Let us assume 'A' & 'B' are 2 stations which want to share a common key.
- Let there be 2 nos, a prime no (q) & an integer (α) in such way that α is primitive root of q.

ie $\text{order}(\alpha \bmod q) = q - 1$

p & q
↳ globally known.

At A:

'A' will choose a no. 'X_A' & generates a no 'Y_A' in such a way that

$$\boxed{\alpha^{Y_A} \cong X_A \bmod q}$$

At B:

B will ~~generate~~ choose a no 'X_B' ~~in such~~ & generates 'Y_B' in such a way that

$$\boxed{\alpha^{Y_B} \cong X_B \bmod q}$$

Both the parties will keep 'x' as secret and announce 'y' as public.

Y_A & Y_B → all (Knowns)

X_A → only A

X_B → only B

Both the parties will calculate secret key (K) as:

$$K = X_A^{Y_B} \bmod q = X_B^{Y_A} \bmod q$$

$\Downarrow$ At A $\qquad$ $\Downarrow$ At B

Thus Both have same key. [Key-exchange alg.]

numericals on minimum packet size efficiency. 2 min

① RIP : Implementation of DVR

- In this protocol, "16" is considered as ∞ and matrix are "no. of hops"

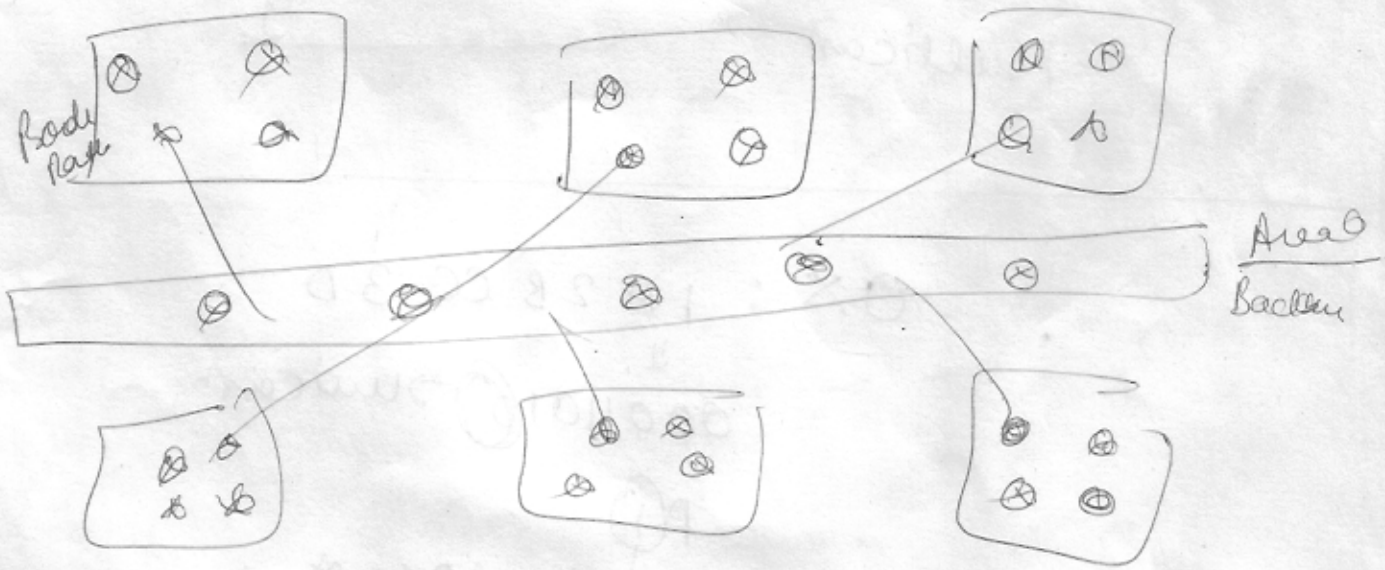
In RIP, NIDs are IDs of the N/Ws to which the router is connected to.

NextHops are IP addresses of the routers to which router is connected to

NID	Distance	Next Hop

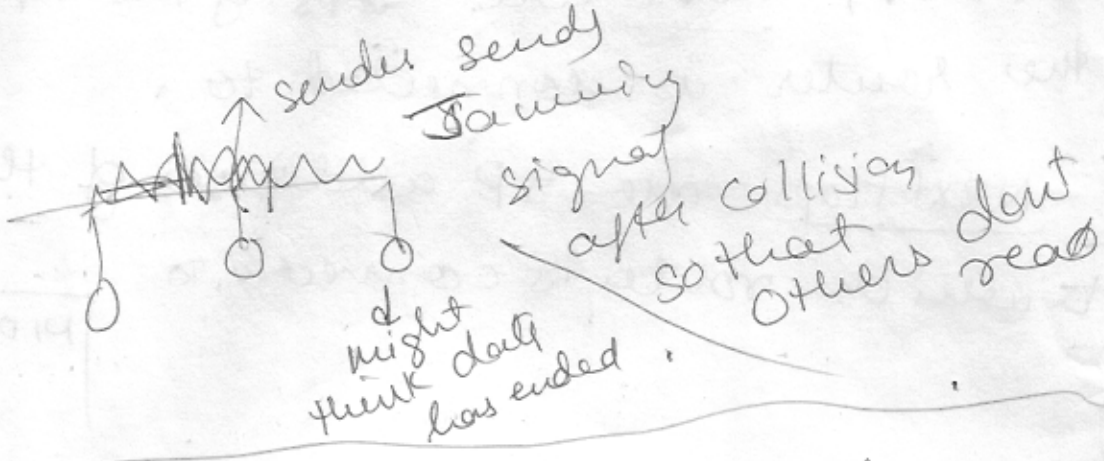
② OSPF (Open Shortest Path first)

→ Implementation of LSR



All the routers are divided into Regions, and within a Region "flooding" is used. There's a Border Router which summarize the entire data & send out the packet. Hence reduces the flooding

Jamming signal is used to indicate that there is collision and it is not used for calculation of minimum frame size



Broadcast Ethernet (all 1s)

Unicast :) _____ 0 last bit of first byte

Multicast " _____ 1

0X : 1A 2B CC 3D

⇓

00011010 → unicast

A1 → multicast