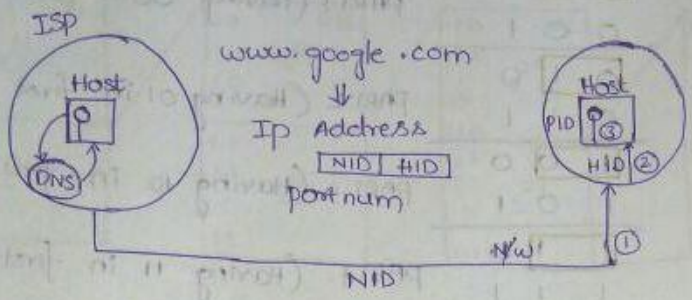
COMPUTER NETWORKS

INTRODUCTION TO CN AND IP ADDRESS



⇒ The service that is used to convert the Domain name to Ip Address is called Domain Name service

⇒ port number is used to identify a particular process in the host, for well known services the port num are already predefined and fixed

http → port num = 80 SMTP port num = 25

Ftp → port num = 21

⇒ Eventhough your intention is to reach gogle.com you are visiting DNS first and then getting the Ip Address of google.com and then visiting the google home page. This is actually a overhead which is also called "DNS overhead" so this problem of overhead can be rectified by, when you get the Ip Address of google.com you store it actually on your computer for some time, and if you again want to visit google you can directly get Ip Address from your computer. If your Ip Address in computer expires, there is no another alternative, you have to go to DNS.

Unary = 0

Binary = 0,1

octal = 0,1,2 ... 7

Decimal = 0,1,2 ... 9

Hexa decimal = 0,1,2,...9

A B ... F

10 11 15

$2^1 = 2$

$2^2 = 4$

$2^{10} = 1024 = K$

$M = 2^{20}$

$G = 2^{30}$

$T = 2^{40}$

$K = 2^{10}$

$M = 2^{20}$

$G = 2^{30}$

$T = 2^{40}$

Now, the possible Binary numbers with 1 Bit, 2 Bits, 3 Bits are

1 Bit

0
1

2 Bits

0 0
0 1
1 0
1 1

3 Bits

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

PART 1 (Having 00 in first 2 digits)

PART 2 (Having 01 in first 2 digits)

PART 3 (Having 10 in first 2 digits)

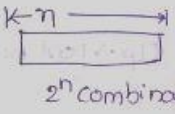
PART 4 (Having 11 in first 2 digits)

⇒ If I choose 1 bit I am going to divide the number space into 2 parts $= 2^1$

⇒ If I choose 2 bits I am going to get 4 parts $= 2^2$

⇒ If I choose K bits the number space / Address space will be divided into 2^K parts

⇒ So, if I have 'n' bits the no of possible ways are 2^n

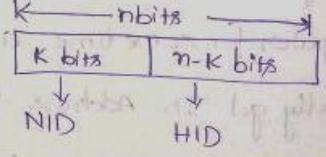


Now, if I choose 'K' bits the entire space will be divided into 2^K parts

⇒ 2^K parts $= 2^n$ numbers

⇒ 1 part $= 2^n / 2^K$ numbers

⇒ 1 part $= 2^{n-K}$ numbers



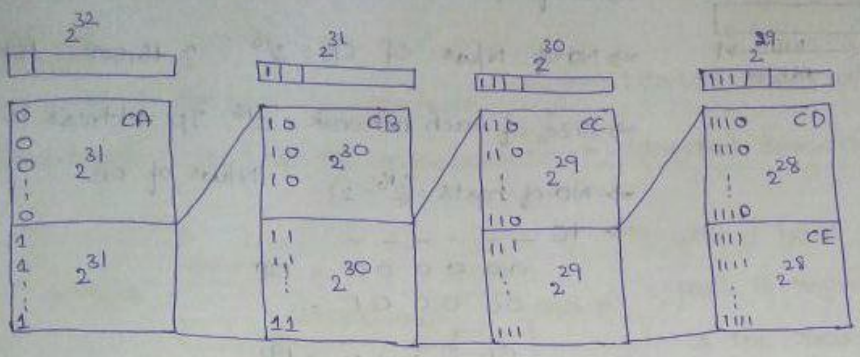
⇒ Size of Each N/w $= 2^{n-K}$

⇒ In operating system 'n' bits = Address, K = page offset / Blk no / Segment no

⇒ In computer Organization 'K' = TAG (n-K) = Block SIZE.

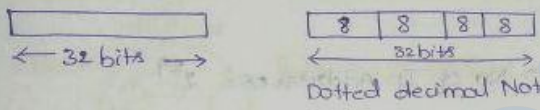
⇒ Ip Address size = 32 bits in CN. ⇒ 2^{32} Ip Address are possible

CLASS FULL IP ADDRESS CLASSIFICATION



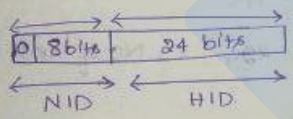
The no. of Ip Addresses possible in a N/w of class A = 2^{31} class C = 2^{29}
class B = 2^{30} class D = 2^{28}

Now, the popular Representation of Ip Address = Binary Notation, Dotted decimal Notation (4 octets)



CA = start with '0'
CB = start with 10
CC = start with 110
CD = starts with 1110
CE = start with 1111

CLASS A



⇒ The No. of possible N/ws of classA = $2^7 = 128$

⇒ Size of each Network = 2^{24} = 16M (NASA, PENTAGON USES CA)

⇒ No. of hosts = $(2^{24} - 2)$

⇒ But practically 126 N/ws are possible in CA

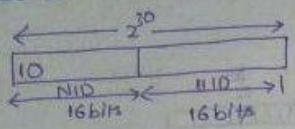
0 _ _ _ _ _ _ _
0 0 0 0 0 0 0 0 = 0
0 0 0 0 0 0 0 1 = 1
⋮
1 1 1 1 1 1 1 = 127

⇒ we donts use the starting Address (All 0's) and Last Address (all ones)

⇒ ∴ No. of N/w's in CA = 128 - 2 = 126 practically.

∴ RANGE = 0-126

CLASS B



⇒ No of Ip Addresses = 2^{30}

⇒ No of N/ws of CB = 2^{14} ≅ 16,000 = 16K N/ws

⇒ Size of each network = 2^{16} Ip Address in one N/ws of CB.

⇒ No of hosts = $(2^{16} - 2)$

⇒ 10 _ _ _ _ _ _

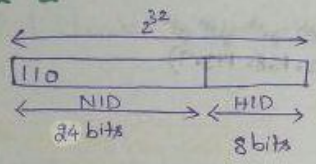
00 0 0 0 0 0 = 128

0 0 0 0 0 0 1 =

1 1 1 1 1 1 1 = 191

⇒ RANGE = 128 - 191

CLASS C



⇒ No of Ip Addresses = 2^{29}

⇒ No of N/ws of CC = 2^{21} N/ws = 2million

⇒ Size of Each N/w = 2^8 Ip Address are in one N/w of CC

⇒ 110 _ _ _ _ _

0 0 0 0 0 = 192

0 0 0 0 1 = 193

1 1 1 1 1 = 223 ⇒ No of hosts = $(2^8 - 2)$

⇒ RANGE = 192 - 223

CLASS D

There is nothing called as N/w ID and HID in CD and CE

CD

1110 _ _ _ _

0 0 0 0 = 224

1 1 1 1 = 239

RANGE = 224 - 239

1) Used for Multicasting

2) Group Emailing, Group Broadcasting

CE

1111 _ _ _ _

0 0 0 0 =

1 1 1 1 = 255

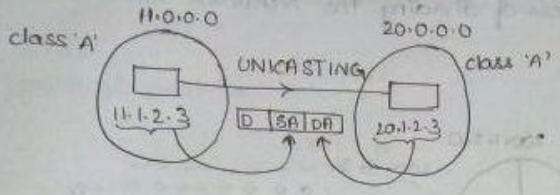
RANGE = 240 - 255

1) Used for Military Applications

2. TYPES OF CASTING - UNICAST, LIMITED BROADCAST, DIRECTED BROADCAST

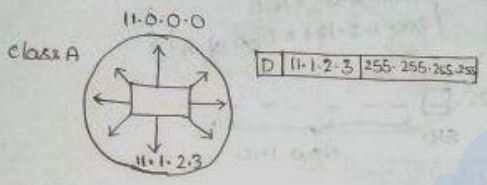
CASTING

1. Unicast - one host to one host
2. Broadcast - one host to many hosts → Limited Broadcasting / Directed Broadcasting



⇒ when i have All 0s in the HID part it represents NID that is the reason we don't use 1st Ip Address as the valid Ip Address to any host

LIMITED BROADCASTING



⇒ If Destination Address = DA = 255.255.255.255 then the packet will be sent to all the hosts in the N/w

LBA = 255.255.255.255

DIRECTED BROADCASTING

⇒ we are not going to use the Ip Address containing all 1's in HID part, it is used for Direct BROADCASTING (DBA)

DBA : NID, HID = all 1's

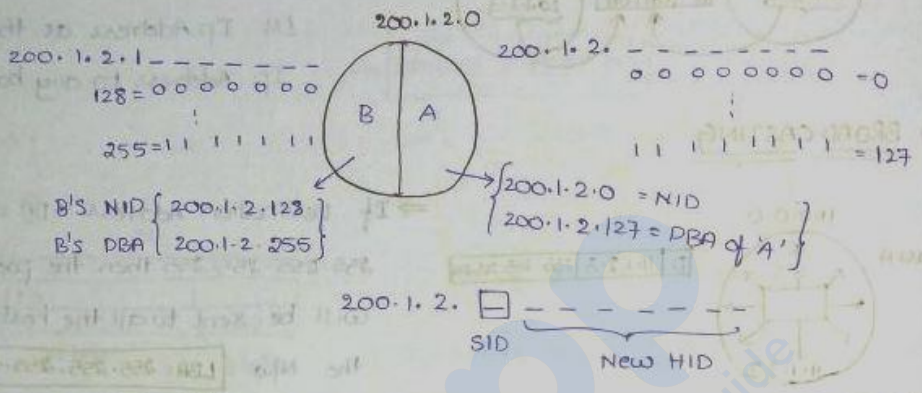
Ip Address	NID	DBA	LBA
1.2.3.4	1.0.0.0	1.255.255.255	255.255.255.255
10.15.20.60	10.0.0.0	10.255.255.255	255.255.255.255
130.1.2.3	130.1.0.0	130.1.255.255	255.255.255.255
150.0.150.150	150.0.0.0	150.0.255.255	255.255.255.255
200.1.10.100	200.1.10.0	200.1.10.255	255.255.255.255
220.15.1.10	200.15.1.0	200.15.1.255	255.255.255.255
250.0.1.2	X	X	X
300.1.2.3	X	X	X

3 SUBNETS, SUBNET MASK, ROUTING

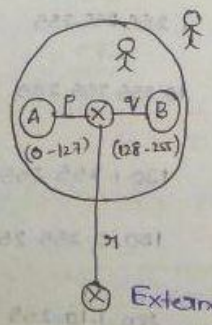
⇒ when the size of the N/w is Big then the maintainence will be difficult.

⇒ Lack of security when the size of N/w is Big, so we divide the Network into small parts and this process is called Subnetting.

⇒ "SUBNETTING" is the process of dividing the Network into Smaller Networks

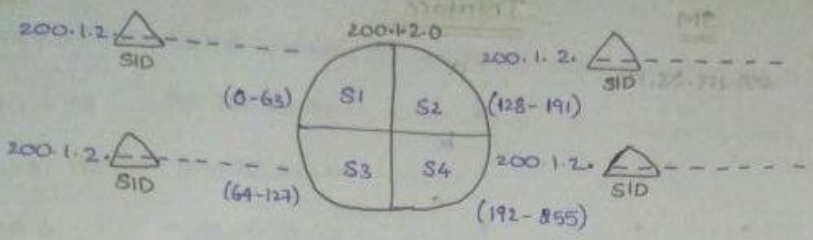


Now, ambiguity arises in the above subnetting, if is say 200.1.2.0 i mean the whole network or only the first subnet and if i say 200.1.2.255 then there is a dilema whether to transfer the packet to all the hosts in the Network or only to the hosts present in the subnet A. so this depens upon where we are standing if we are outside the network then we assume that the packet should be transmitted to all the hosts in the network, if I am standing inside the packet will be transmitted to the hosts in subnet 'A'.

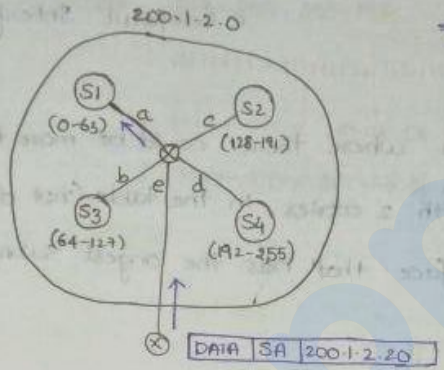


*) There will be loss of ipAddresses due to "SUBNETTING"

FOUR SUBNETS



The, practical view is.



⇒ Here the challenge is you should identify the Nlw or Subnet for which a particular Ip Address belongs to. Here this can be done using "SUBNET MASK"

SUBNET MASK

SM: 32-bit Number

1's: SID, NID part

0's: HID part

Now, the SubNet mask for the above Nlw will be

NID.
200.1.2. __ HID
SID

⇒ 11111111.11111111.11111111.11000000

∴ SM = 255.255.255.192

⇒ Given an Ip Address, if you bitwise AND with SM you will get NID for which the Ip belongs to

SM && Ip = NID for which the given Ip Belong

Ip: 11001000.00000001.00000010.10000020 = 200.1.2.130

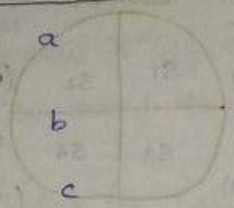
SM: 11111111.11111111.11111111.11000000 = 255.255.255.192

NID: 11001000.00000001.00000010.10000000

200.1.2.128

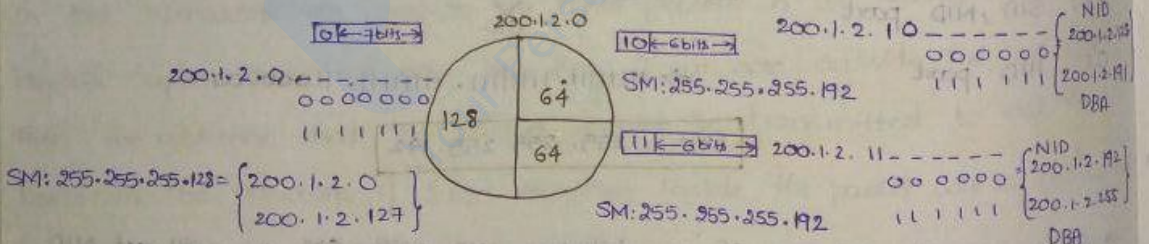
ROUTING TABLE

NID	SM	Interface
200.1.2.0	255.255.255.192	a
200.1.2.64	" "	b
200.1.2.128	" "	c
200.1.2.192	" "	d
0.0.0.0	0.0.0.0	e (default Interface)



⇒ There may be some cases where there could be more than one matches (our packet matches with 2 entries in the table (not default)), in that case choose the interface that has the longest subnet mask (more no of 1's)

4. VARIABLE LENGTH SUBNET MASKING



Step 1: Divide the entire N/w into 2 parts choosing 1 bit

Step 2: Divide the subnet (your choice) into 2 parts by choosing another bit then you will get 3 subnets

⇒ The N/ws that are having same sizes have the same "subnet mask".

⇒ when the N/w size is Big then the SM will be small
when the N/w size is small then the SM will be Big.

ROUTING TABLE

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

Now, Given Subnet mask : 255.255.255.192

: 11111111 . 11111111 . 11111111 . 11000000 = (26 1's, 6 0's)

: NID + SID = No of 1's

: NID + SID = 26 → If classA is subnetted then,

8 + SID = 26 ⇒ SID = 18

= No of Subnets = 2^{18}

→ If CB, NID + SID = 26

SID = 26 − 16 = 10 = 2^{10} Subnets

→ If CC, NID + SID = 26

SID = 26 − 24 = 2 = 2^2 subnets

0 - 00000000

128 - 10000000

192 - 11000000

224 - 11100000

240 - 11110000

248 - 11111000

252 - 11111100

254 - 11111110

255 - 11111111

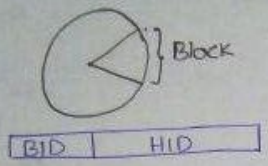
5. SUBNET MASKING QUESTION

	SUBNET MASK	No of HOSTS	SUBNETS IN CA	CB	CC
CA	255.0.0.0	$2^{24}-2$	1	–	–
	255.128.0.0	$2^{23}-2$	2^1	–	–
	255.192.0.0	$2^{22}-2$	2^2	–	–
	255.240.0.0	$2^{20}-2$	2^4	–	–
CB	255.255.0.0	$2^{16}-2$	2^8	1	–
	255.255.254.0	2^9-2	2^{15}	2^7	–
CC	255.255.255.0	2^8-2	2^{16}	2^8	1
	255.255.255.224	2^5-2	2^{19}	2^{11}	2^3
	255.255.255.240	2^4-2	2^{20}	2^{12}	2^4

CLASSLESS INTER DOMAIN ROUTING (L-6)

IANA = Internet Assigned Number Authority

⇒ Generally the Ip Addresses in CIDR Notation is represented as a.b.c.d/(n)



⇒ n = NID bits ⇒ if n = 20, NID = 20 bits, HID = 12 bits, 32 bits

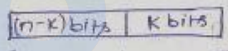
⇒ n = Slash Number

FORMATION OF CIDR BLOCKS

These are the Rules for forming CIDR Blocks

1) All the Ip Address should be contigious

2) The Block size should be a power of 2



3) The starting Address should be divisible by size of the Block evenly

$[n/2^k]$ ⇒ last k bits = Remainder
last (n-k) bits = Quotient

1)
100.1.2.32
100.1.2.33
⋮
100.1.2.47

Whether these Ip Addresses form a CIDR Block?

Rule 1 ✓

Rule 2: No. of Ip Address = 16 = 2^4 ✓ ⇒ HID = 4, BID = 28

Rule 3: 100.1.2.00100000 ÷ 2^4 = LSB (4bits) {Least significant Bits}

= 0000 ✓

∴ They form CIDR Block

2) 20.10.30.32
20.10.30.33
20.10.30.34
⋮
20.10.30.63

Rule 1 = ✓

Rule 2: (63-32)+1 = 32 = 2^5 ✓ (In powers of 2) HID = 5 bits, NID = 27 bits

Rule 3: 20.10.30.001 00000 ÷ 2^5 = last 5 bits

= 00000 ✓

∴ They form CIDR Block

3>

150.10.20.64
150.10.20.65
150.10.20.66
!
150.10.20.127

Rule 1 ✓

Rule 2: $(127-64)+1 = 64 = 2^6$ ✓ ⇒ HID=6, BID=26 Bits

Rule 3: 150.10.20.01000000 ÷ 2^6 = 000000 ✓

FORM CIDR BLOCK ✓

4>

20.10.30.35/27 Derive the Range of CIDR Block?

⇒ BID = 27 bits HID = 5 bits

BID: 20.10.30.001 | HID: 00011

0000 = 32
0001 = 33
⋮
11111 = 63

⇒ Block = 20.10.30.32
20.10.30.33
⋮
20.10.30.63

5>

100.1.2.35/28

BID = 27 bits HID = 5 bits

BIK NO = BIKID

100.1.2.0010 | 0011

0000 = 32
⋮
1111 = 47

100.1.2.32
100.1.2.33
⋮
100.1.2.47

6> 100.1.2.35/20

BID

100.1.0000 | 0010.00100011

0000.00000000 = 0 ⇒ 100.1.0.0

1111.11111111 ⇒ 100.1.15.255

15

7. SUBNETTING IN CIDR, VLSM IN CIDR

20.30.40.10/25

BID | SID | HID

20.30.40.0 | 0 | 001010

000000 = 0
⋮
111111 = 63

20.30.40.0/26
20.30.40.63/26

BID | SID | HID

20.30.40.0 | 1 | 001010

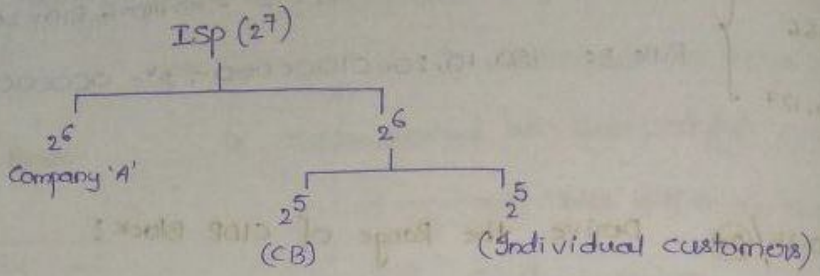
000000 = 64
⋮
111111 = 127

20.30.40.64/26
20.30.40.127/26

A B

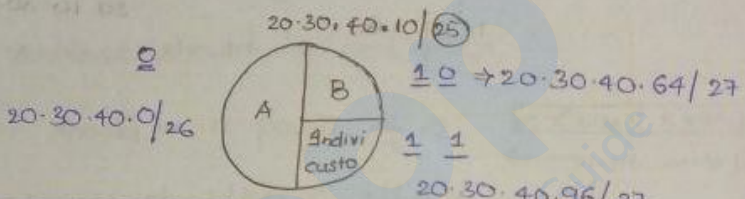
VLSM IN CIDR BLOCKS

1) 20.30.40.10/25



Now, BLK ID
20.30.40.0000 1010

0 0 0 0 0 0 0 0 = 0
1 1 1 1 1 1 1 1 = 127



∴ CA = 20.30.40.0/26
CB = 20.30.40.64/27
Customers = 20.30.40.96/27

8. SOME INTRESTING PROBLEM ON SUBNET MASK

⇒ Subnet mask is sometimes called as "Network mask".

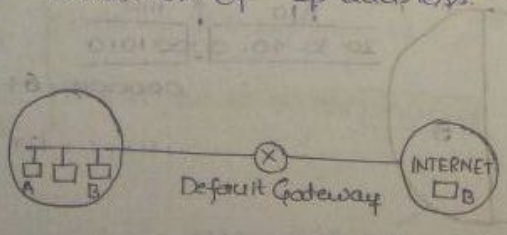
⇒ whenever you open "ipconfig" in your computer you will see these things

1> IpV4 Address: provided by ISp

2> Default Gateway: Default Router connected to your n/w

3> Subnet Mask: Subnet mask which you should use

4) DNS: Convertion of Ip addriess



Let I_A = IpAddress of 'A' I_B = Ip Address of B. (1)

S_A = Subnetmask of 'A' S_B = Subnetmask of B.

⇒ Now if you want send a packet from 'A' to 'B' there are 2 cases possible

1: 'B' may be in different network and you want to send first to DGw and DGw will forward the packet to 'B'

2. 'B' may be in same Network so that you can send the packet directly

Now what 'A' does is A: I_A I_B

Bitwise AND = S_A S_A

$((NID)A)_{Acc\ to\ A}$ $((NID)B)_{Acc\ to\ A}$

If $((NID)A)_{Acc\ to\ A} = ((NID)B)_{Acc\ to\ A}$ then A&B are in same N/w

If

I_A: 200.1.2.10 I_B: 200.1.2.130

S_A = 255.255.255.128

Now, I_A : 11001000.00000001.00000010.00001010

S_A : 11111111.11111111.11111111.10000000

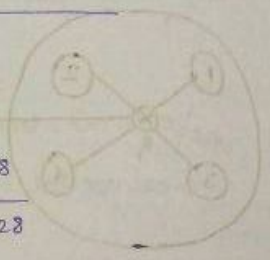
200 . 1 . 2 . 0

$((NID)A)_{Acc\ to\ A}$ = 200.1.2.0

I_B : 200.1.2.130

S_A : 255.255.255.128

$((NID)B)_{Acc\ to\ A}$: 200.1.2.128



$((NID)A)_{Acc\ to\ A} \neq ((NID)B)_{Acc\ to\ A}$ ∴ 'A' assumes that 'B' is in diff Network.

(2) I_A : 200.1.2.10 I_B : 200.1.2.69

S_A : 255.255.255.128 S_B : 255.255.255.192

I_A I_B

S_A S_A

200.1.2.0 ← $((NID)A)_{Acc\ to\ A}$ $((NID)B)_{Acc\ to\ A}$ → 200.1.2.0

I_B I_A

S_B S_B

$((NID)B)_{Acc\ to\ B}$ $((NID)A)_{Acc\ to\ B}$

= 200.1.2.64 200.1.2.0

If $((NID)A)_{Acc\ to\ A} = ((NID)B)_{Acc\ to\ A}$ ⇒ 'A' think B is in same N/w.

⇒ 'B' think 'A' is in another N/w

9. SUPERNETTING OR AGGREGATION

⇒ If we look at the Routing table, it contains a single entry for each and every N/w, now if the networks are large then the size of the routing table may grow exponentially, so that router takes lot of time to process the Routing table, hence we need to Aggregate / combine / supernet the Networks.

RULES FOR AGGREGATION

1) All the N/ws should be contigious. (N/w Id's)

2) Size of each N/w should be same and inturn they should be in powers of 2.

3) 1st Address should be divisible by size of Block

Aggregate the following N/w's

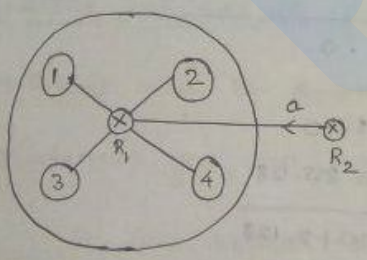
1) 200.1.0.0/24
2) 200.1.1.0/24
3) 200.1.2.0/24
4) 200.1.3.0/24

Rule 1 ✓

Total size of N/w = 4×2^8
$= 2^{10}$

Rule 2 ✓

Rule 3 ✓ 200.1.00000000.00000000 ÷ 2^{10}
= Divisible



The Routing table at R_2 looks like

NID	SM	INTERFACE
200.1.0.0	255.255.255.0	a
200.1.1.0	255.255.255.0	a
200.1.2.0	255.255.255.0	a
200.1.3.0	255.255.255.0	a

SUPERNET MASK

Supernet mask : 32 bits

: No of 1's = Fixed part

: No of 0's = Variable part

FIXED PART	VARIABLE PART
200.1.000000	00.00000000
200.1.000000	01.00000000
200.1.000000	10.00000000
200.1.000000	11.00000000
255.255.111111	00.00000000

Supernet mask = 255.255.252.0

⇒ Now, the N/w id of the Aggregated N/w is same as the starting IpAddress
= 200.1.0.0/24 (8)

SHORTCUT TO FIND FIXED AND VARIABLE PARTS

⇒ Size of all N/ws = $2^8 + 2^8 + 2^8 + 2^8 = 2^{10}$ ⇒ Host id part should contain 10 bits

⇒ BID part contain '22' bits

⇒ Now, the Routing table at R_2 look like

NID	SM	INTERFACE
200.1.0.0	255.255.252.0	a

2)

100.1.2.0/25
100.1.2.128/26
100.1.2.192/26
} Aggregate them!

⇒ All the Addresses are contigious ✓, but of diff sizes

⇒ The N/ws 100.1.2.128/26, 100.1.2.192/26 } These can be Aggregated first ← Rule 1 ✓, Rule 2 ✓, Rule 3 ✓

⇒ Total size of N/w = $2^6 \times 2 = 2^7$ // ⇒ HID = 7 ⇒ NID = 25 bits

⇒ The N/w of the above Aggregated N/w = 100.1.2.128/25 and now Combine with 1st N/w 100.1.2.0/25

⇒ 100.1.2.0/25, 100.1.2.128/25 } Rule 1 ✓, Rule 2 ✓, Rule 3 ✓

N/w id of Supernet = 100.1.2.0/24

Supernet mask = 255.255.255.0

Now size of N/w = 2×2^7

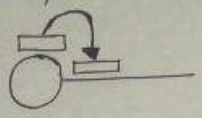
$= 2^8$ ⇒ NID = 24, HID = 8

2. FLOW CONTROL METHODS

DELAYS IN CN (L-1)

TRANSMISSION DELAY (T_t):

The time taken to transmit the packet from a host to the outgoing line is called "transmission delay"



Bandwidth = 1bps (In one second we can transmit 1bit to the outgoing delay)

Data = 10bits

$\Rightarrow$ Transmission delay = 10sec (for 10bits)

So, If size of datapacket is 'L' bits and Bandwidth is 'B' bps then the

Transmission delay = L/B sec

$$T_t = L/B \text{ sec}$$

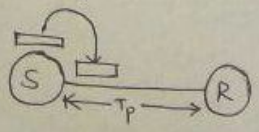
1) L = 1000 bits, Bw = 1Kbps $\Rightarrow T_t = L/B = \frac{1000}{1000} = 1$ sec

2) L = 1Kb, Bw = 1Kbps $\Rightarrow T_t = L/B = \frac{1024}{1000} = 1.024$ sec

DATA	BANDWIDTH
K = 1024 bits	K = 1000
M = 1024 × 1024	M = 1000 × 1000 = 10^6
G = 1024 × 1024 × 1024	G = 1000 × 1000 × 1000 = 10^9

PROPOGATION DELAY (T_p):

The time taken by a bit to reach from one end of the link to otherend of the link is called "propogation delay"



The propogation delay depends upon 1) distance

2) velocity

$$T_p = d/v$$

⇒ In case of optical fibers the speed of the signal is approximately 70% of speed of Light ⇒ $v = \frac{70}{100} \times 3\times10^8$ m/s

⇒ $\boxed{v = 2.1\times10^8 \text{ m/s}}$

1) d = 2.1Km

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

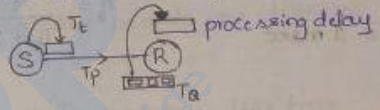
$T_p = d/v = \frac{2.1\times10^3 \text{ m}}{2.1\times10^8 \text{ m/s}} = 10^{-5}$ sec $= 10^{-5} \times 10^{-3} \times 10^{3}$

$\boxed{T_p = 10^{-2} \text{ msec}}$

Now, the total time taken to send a packet from source to Destination = $T_t + T_p$

QUEUING DELAY (T_Q):

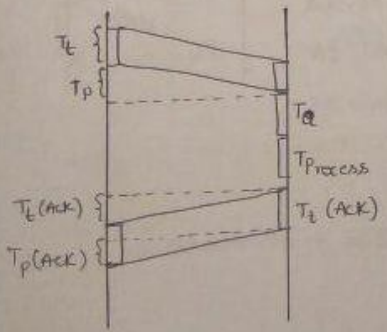
⇒ The amount of time the packet sits in the Queue before it gets processed is called Queuing delay(T_Q).



⇒ Whenever the packet is received by the Receiver, it may or may not get immediately processed they are going to sit in Buffer (Queue).

2. FLOW CONTROL STOP AND WAIT

⇒ Stop and wait is the simplest flow control mechanism in which the sender sends a packet and then stop and wait for the acknowledge from the Receiver, before sending the next packet.



Total time = T_t (data) + T_p (data) + ~~T_Q~~ + ~~$T_{process}$~~ + T_t (Ack) + T_p (Ack)

= T_t (data) + 2* T_p + ~~T_t (Ack)~~

$\boxed{\text{Total time} = T_t + 2*T_p}$ for sending 1 packet

$\eta = \frac{\text{useful time}}{\text{Total cycle time}} = \frac{T_t}{T_t + 2*T_p} = \frac{T_t}{T_t(1+2\,T_p/T_t)} = \frac{1}{1+2a}$ (where $\frac{T_p}{T_t} = a$)

Throughput = No. of bits we can send a second using this protocol

$\Rightarrow$ Throughput / Effective Bandwidth / Bandwidth utilisation / Link utilisation $= \frac{L}{T_t + 2*T_p} \Rightarrow \frac{L \times B \times (1/B)}{T_t + 2T_p} = \frac{T_t}{T_t + 2T_p} * Bw$

$\Rightarrow \frac{1}{1 + 2a} * Bw = \eta \times Bw$

$\therefore$ **Throughput = $\eta \times Bw$**

1>

T_t = 1 msec

T_p = 1 msec

Efficiency $(\eta) = \frac{T_t}{T_t + 2*T_p} = \frac{1}{1+2} = 1/3$

2>

T_t = 2 msec

T_p = 1 msec

$\eta = \frac{T_t}{T_t + 2*T_p} = \frac{2}{2 + 2(1)} = 2/4 = 1/2 = 0.5 = 50\%$

3> Now if the efficiency has to be 50% what is the relation between T_t & T_p?

$\eta = 50\% = 1/2 \Rightarrow \frac{T_t}{T_t + 2T_p} \geq 1/2$

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

$\Rightarrow$ **$T_t \geq 2T_p$** $\Rightarrow L/B \geq 2T_p$

$\Rightarrow$ **$L \geq 2*T_p*B$**

4> Bw = 4 Mbps

T_p = 1 msec

L = ? so that η = atleast 50% ?

$L \geq 2*T_p*Bw$

$L \geq 2 \times 1 \times 10^{-3} \times 4 \times 10^{6}$

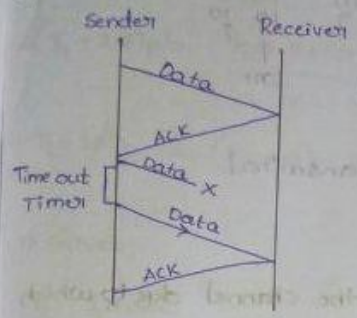
$L \geq 8 \times 10^{3}$ bits

⇒ Now, $\eta = \frac{1}{1+2a} = \frac{1}{2+2\left(\frac{T_p}{T_t}\right)} = \frac{1}{1+2*\frac{d}{v}*\frac{B}{L}} \Rightarrow \boxed{\eta \propto 1/d}$. SAW is best for LANS

$\Rightarrow \boxed{\eta \propto L}$

(10)

5)

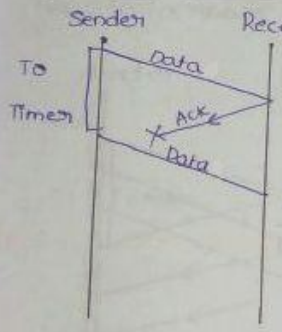


∴ STOP AND WAIT + TIMEOUT TIMER

⇒ STOP AND WAIT ARQ {Automatic Repeat Request}

DATA PACKET LOSS PROBLEM ⇒ {overcome by TO timer}

6) ACK LOST PROBLEM

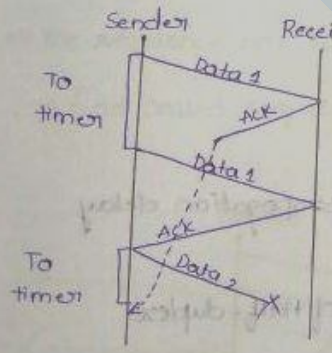


⇒ Acc to sender both the data packets are same but Acc to Receiver both the packets are different

DUPLICATE PACKET PROBLEM ⇒ {To overcome this have sequence numbers}

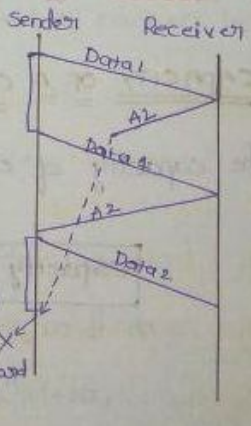
SAW + TO + SEQ. NO TO DATA PACKET

7) DELAYED ACKNOWLEDGEMENT



⇒ This problem can be overcomed by having seqno on Ack also.

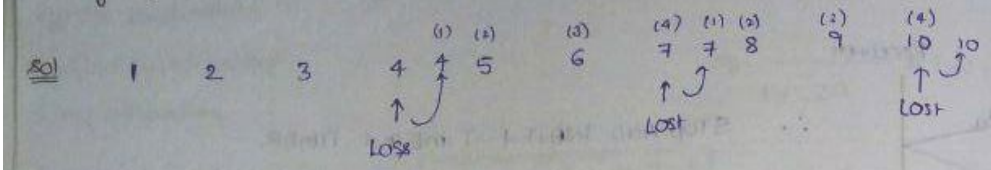
⇒ If the Receiver receives 'D1' sucessfully then it sends Ack A2 which means {"I have sucessfully received D1, please send D2}



⇒ SAW + TO + SEQ NO TO DATA AND ACK

8> Using SAW we have to send 10 packets from Sender to Receiver of which every 4th packet is lost then how many packets are we going to send totally?

Sol



∴ Total = 13 ; (4,7,8) ⇒ packets are Retransmited

9> (S) ——— (R) , Now there are some problems in the channel due to which some of bits are lost, Let us say error probability of channel is 0.2 = 20%, Now if i send 400 packets, then how many packets are transmited totally

Sol ⇒ $400 + 400(0.2) + \{400(0.2)\}(0.2) + \cdots$

⇒ $n + np + np^2 + \cdots$ ⇒ $n(1+p+p^2+p^3+\cdots)$

⇒ $n\left(\frac{1}{1-p}\right) = \frac{n}{1-p}$

Here $n = 400$, $\frac{400}{1-0.2} = \frac{400}{0.8} = \frac{4000}{8} = 500$ packets totally.

3. CAPACITY OF PIPE AND PIPELINING

CAPACITY OF A CHANNEL / WIRE / LINK

The capacity of channel depends on Bandwidth and propogation delay

Capacity of the channel = Bw * Tp ⇒ Incase of Half-duplex

Capacity = 2 × Bw × Tp ⇒ Incase of Full duplex

PIPELINING

T_t sec – 1 packet
1 sec – $1/T_t$ packets

Now time taken to transmit one packet of data in stop and wait protocol is $T_t + 2*T_p$. Now,

$\Rightarrow (T_t + 2*T_p) = \frac{T_t + 2*T_p}{T_t}$ packets

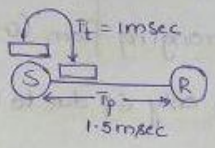
$\Rightarrow T_t + 2*T_p = (1 + 2a)$ packets $\left(a = T_p/T_t\right)$

WINDOW OF PACKETS

1) Given,

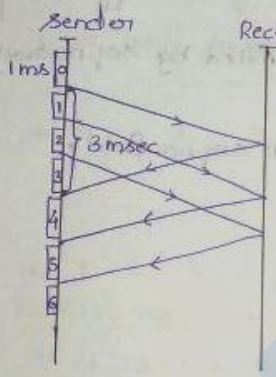
T_t = 1 msec

T_p = 1.5 msec



$\Rightarrow \eta = \frac{T_t}{T_t + 2*T_p} = \frac{1}{1+2(1.5)} = \frac{1}{4} = 25\%$

Now to increase the efficiency of stop and wait protocol,



Round trip time = $2*T_p$

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

9 8 7 → Yet to be Transmitted

2 1 0 → Already Transmitted and Acknowledged

$\Rightarrow$ The sender window size in sliding window protocol = $w_s = 1 + 2a$

$\Rightarrow$ The sequence nos have to be stored in the header field of packets in a field called sequence no of field.

$\Rightarrow$ Min no of sequence nos = $1 + 2a$

$\Rightarrow$ Min no of bits in seq no fields = n

$2^n = 1 + 2a$

$\Rightarrow n \log 2 = \log(1 + 2a)$

$\Rightarrow n = \lceil \log_2(1 + 2a) \rceil$

2) $T_t = 1\,ms$, $T_p = 49.5\,ms$ } what is the sender window size to get max efficiency?

So $w_s = 1+2a \Rightarrow 1 + 2\left(\frac{T_p}{T_t}\right) = 1+2(49.5) = 100$

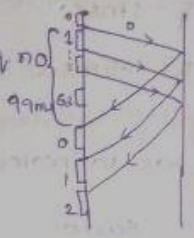
Min. no. of sequence no's = $100 = (1+2a)$

Min no. of bits in seq no field = $\lceil \log_2(1+2a) \rceil = \lceil \log_2(100) \rceil = 7$

⇒ Now, in the above problem if the min. no. of bits in seq no field is "6", then we can get 64 seq. no (2^6) ranging from (0-63)

⇒ $\eta = \frac{64}{100} = 0.64$ → But I am sending only 64 due to lack of seq no

↳ I can send 100 packets in the time available



∴ The window size in the stop and wait protocol is limited by seq. no available

∴ $\boxed{w_s = \min(1+2a, 2^n)}$ {n = no. of bits in seq no field}

GO BACK - N (L-4)

Sliding window protocol

Go-Back N (cumulative ACK) | Selective Repeat

GOBACK - N (N>1)

1) Sender window size in GBN is "N"

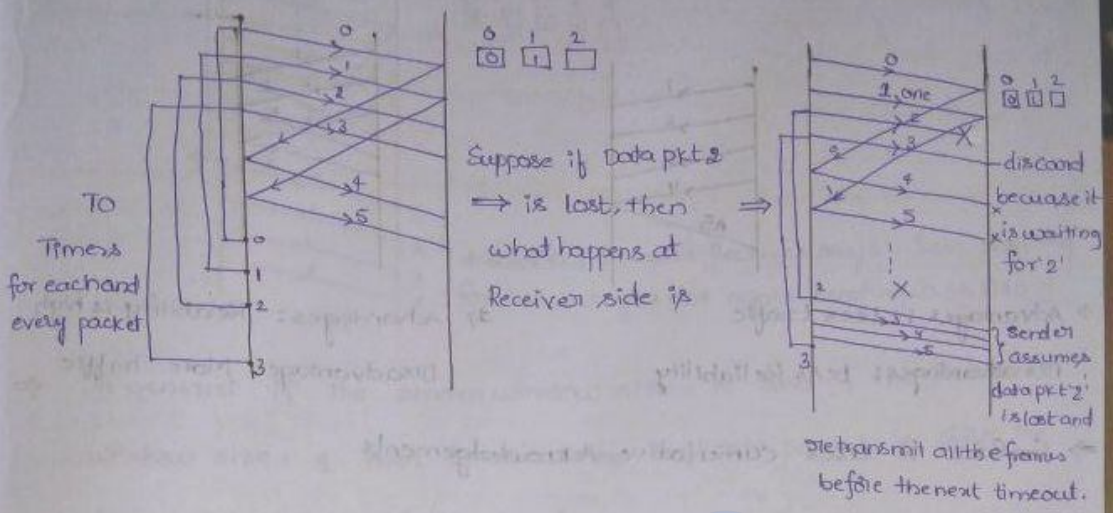
T_t = 1msec
T_p = 49.5msec
GOBACK - 10
} η = ?

Max window size = $(1+2a)$
$= 1 + 2(49.5)$
$= 100$

$\eta = \frac{10}{100} = 10\%$

BW = 40mbps
Throughput = ?
$\boxed{Th \Rightarrow \eta \times BW}$
$= 40 \times \frac{10}{100}$
$\boxed{Th = 4mbps}$

2) Receiver window size = 1



3) In GoBack-3 if every 5th packet that is being transmitted is lost and if we have to send 10 packets, then how many transmissions are Required?

⇒ 1 2 3 4 [5 6 7] 8 9 10
↑ is loss

⇒ 1 2 3 4 [5 6 7] 5 6 [7 8 9] 10

= 1 2 3 4 5 6 7 5 6 7 8 9 7 8 [9 10]

= 1 2 3 4 5 6 7 5 6 7 8 9 7 8 9 10 9 10 = 18 packets are Transmited

4) GB-4, Every 6th packet is lost, I should send 10 packets, How many transmissions

Sol = 1 2 3 4 5 [6 7 8 9] 10

= 1 2 3 4 5 6 7 8 9 6 7 [8 9 10]

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

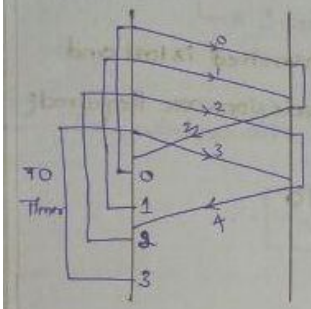
= 17 transmissions

5) Acknowledgements: There are 2 kinds of Ack 1) Cumulative Ack
2) Independent Ack

1) Advantages: Less traffic
Disadvantages: Less Reliability

2) Advantages: Reliability is high
Disadvantage: More Traffic

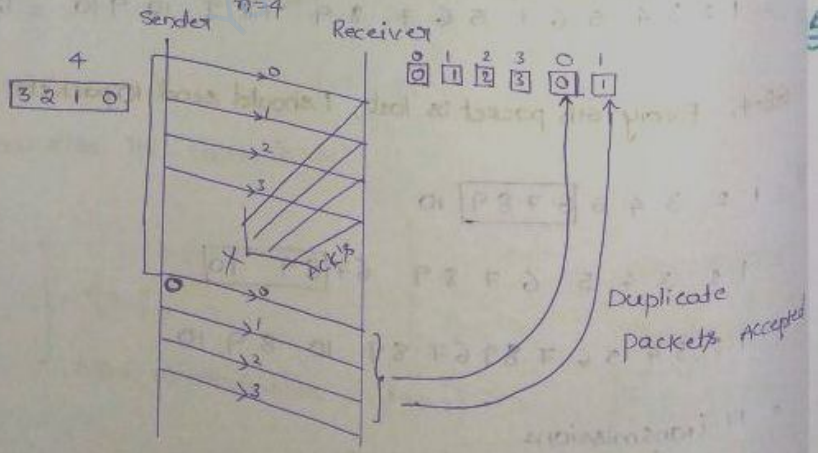
⇒ GoBack - N uses cumulative Acknowledgements.



⇒ TO Timers > Ack Timers

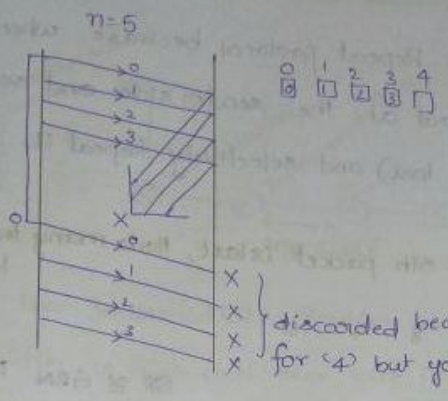
RELATION SHIP BETWEEN WINDOW SIZES AND SEQUENCE No's IN GBN

GB-4 ⇒ Sender window size = 4 ⇒ No of sequence no.



∴ This scenario ⇒ (having the sequence nos = window size) don't work

2) GB-4 , N=5 (Seq. No's)



⇒ In general if the sender window size = 'N' and if the Receiver window size = 1 then to detect the Duplicate packets in GBN the No. of Seq nos should be (N+1).

⇒ In any sliding window protocol if it has to work without any problem the condition is $(w_s + w_R) \leq$ ASN (Available Sequence nos)

⇒ If $w_s = N$, $w_R = 1$ ⇒ Seq Nos = N+1 ⇒ No. of bits in Seq no field = $\lceil \log_2(N+1) \rceil$

⇒ If Seq Nos = N then wat is the max value of $w_s = N-1$ $w_R = 1$

⇒ No. of bits in Seq no field = K then $w_s = 2^K - 1$, $w_R = 1$, No. of Seq Nos = 2^K

5. SELECTIVE REPEAT AND COMPARISION BETWEEN ALL SLIDING WINDOW PROTOCOLS

1) $w_s > 1$

$T_t = 1$ ms $T_p = 49.5$ msec, $w_s = 50$, In SR-protocol η = ?

Sol) $\eta = \frac{50}{100}$ ⇒ $\eta = 50\%$

Max window size of sender = $1 + 2a$
$= 1 + 2(49.5)$
$= 100$

Bw = 4mbps ⇒ Throughput = $\eta \times Bw$
$= 1/2 \times 4 = 2$mbps

2) $W_R = W_S$

⇒ This protocol is called selective Repeat protocol because whenever a packet is lost there will be a timeout at the sender side and sender is going to select that packet (which is lost) and selectively Repeat it.

$W_S = 3$, 10 packets are sent, every 5th packet is lost, How many transmissions in SR protocol

sol: = 1 2 3 4 5 6 7 8 9 10

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

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

= 12 transmissions

SR ≅ GBN interms of efficiency

SR ≅ SAW interms of Retransmissions

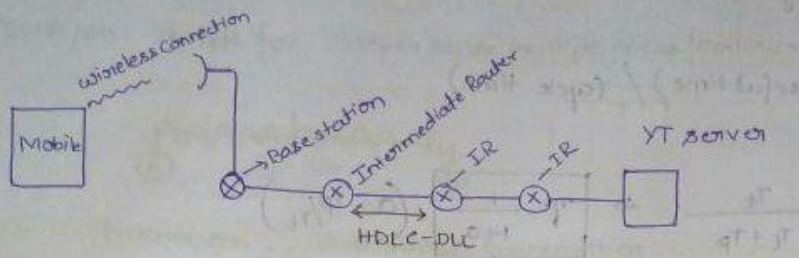
3) The Ack are independents in SR-protocol

⇒ In case of packet loss both GBN and SR protocol behaves same way but in case of corrupted Data packets SR is going to send "NACK" (Negative Acknowledgement)

	SAW	GBN	SR
Efficiency	$\frac{1}{(1+2a)}$	$\frac{N}{(1+2a)}$	$\frac{N \rightarrow \text{sender window size}}{(1+2a)}$
Buffers	1+1 = 2	N+1	N+N = 2N
Sequence Numbers	1+1 = 2	(N+1)	(N+N) = 2N
Retransmissions (1 packet is lost)	1	N	1
Bandwidth	Low	High (More Retrans)	Moderate
cpu	Low	Moderate	High
Implementation	Easy	Moderate	Difficult/complex

⇒ The maximum window size for data transmission using Selective Reject protocol with n-bit frame sequence numbers is 2^{n-1}

PRACTICAL SCENARIO WHERE SR AND GBN ARE USED



⇒ Generally the Intermediate Routers are connected with Thick wires.

⇒ Bandwidth is high and error rate is low, out of order are not possible, cpu's are powerful and there are always Busy in processing the pkts : The availability of cpu is very less.

⇒ For all the above scenarios GBN is used.

⇒ When we consider near our mobile phone / Laptop, Error rate ↑, Bw↓ and out of order packets are possible, cpu is available : cpu ↑

⇒ SR protocol is used.

∴ Link to Link protocols like HDLC -(DLL) uses "GBN"

End to End protocols like Tcp - uses - "SR" Tcp-(Transport layer)

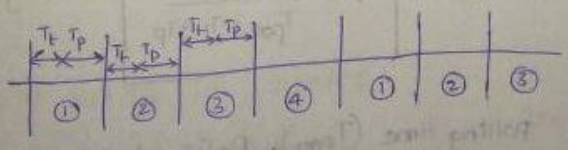
6. INTRODUCTION TO ACESS CONTROL METHODS, TDM, POLLING

The are two types of Links
- Broad cast Link
- point - point Link

TDM

⇒ TDM stands for Time Division Multiplexing.

⇒ Divide the timeline into slots and allot a station to each slot in Round Robin manner.



η = (cycle time)$^{-1}$ × (use ful time)

η = (useful time) / (cycle time)

$\eta = \frac{T_t}{T_t + T_p} \Rightarrow \boxed{\eta = \frac{1}{1+a}} \quad \left(a = T_p/T_t\right)$

If T_t = 1msec, T_p = 1ms and η of TDM? $\eta = \frac{1}{1+a} = \frac{1}{1+1} = \frac{1}{2} = 50\%$

Bw = 4mbps Throughput = $\eta \times Bw = \frac{1}{2} \times 4 = 2$mbps

Now, if 'N' stations are connected to the channel and each station Require 2kbps Bw then what is the max value of N?

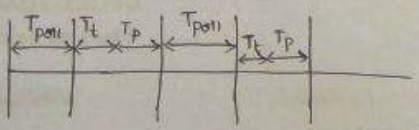
$\Rightarrow N * 2kbps = 2mbps$

$\Rightarrow \boxed{N = 1000}$

Disadvantages

1. The problem with Reservation method in TDM is, whenever you reserve a slot for a station, the station might not use it completely the reason is it is not always true that every station might have data to transmit then the slot alloted will be wasted.

POLLING



η = (useful time) / cycle time

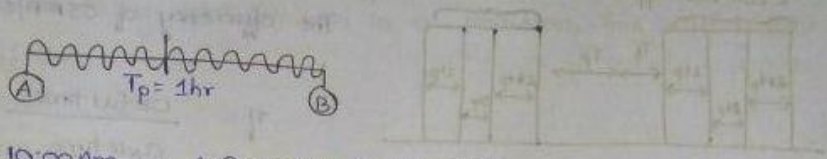
$\eta = \frac{T_t}{T_{poll} + T_t + T_p}$

$\boxed{\eta = \frac{T_t}{T_{poll} + T_t + T_p}}$ Throughput = $\eta \times Bw$

$\Rightarrow$ The disadvantage is the polling time (T_{poll}). Before transmiting we should actually construct polling.

7. CSMA/CD

⇒ The CSMA/CD stands for carrier sense multiple Access/collision Detection.



t = 10:00 Am , A, B starts transmitting

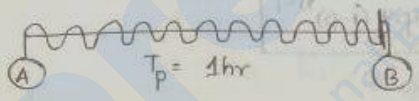
t = 10:30 Am, collision

t = 11:00 AM, collision signals observed by A, B

⇒ Now, if A, B must be shoure about their data (i mean whether the data that they are transmitting, got collided or others data) the condition is

$$T_t > T_p$$

⇒ At worst case, if A starts transmiting at 10:00 Am after how much time will 'A' get the collision signal back.



10:00.00 Am, A start transmitting

10:59:59 Am, B starts transmiting

11:00:00 Am, collision

12:00:00 Am, 'A' sees the collision signal.

∴ The time at which 'A' sees the collision signal is 2hrs = $2*T_p$

∴ If 'A' has to detect collision (be shure about whether its own data) got collided or others data) it should still be transmiting the data in worst case also

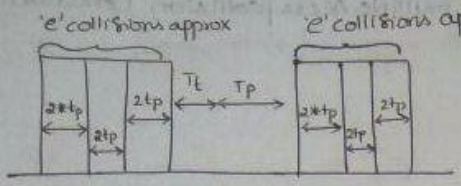
$$\Rightarrow T_t \geq 2*T_p \Rightarrow L/B \geq 2*T_p$$

$$\Rightarrow L \geq 2*T_p*B$$ = This is the min size of packet to detect collision

⇒ There are NO Ack in CSMA/CD

⇒ CSMA/CD is used in Ethernet

EFFICIENCY OF CSMA/CD/ETHERNET



The efficiency of CSMA/CD is

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

$$\eta = \frac{T_t}{[c * 2 * t_p] + T_t + T_p} \Rightarrow \eta = \frac{\text{Trans mission delay}}{[(\text{No. of contention slots}) * 2 * PD] + TD + PD}$$

1) If there are 'n' stations connected to CSMA/CD and every station wants to send the data with probability 'p'.

Now, There will be a successful trans mission if only one station transmits the data and other stations Refrain = ploty for successful trans =

$$P_{sucess} = {}^{n}C_1\, p * (1-p)^{n-1}$$

$$\boxed{P_{sucess} = np * (1-p)^{n-1}}$$

Now, $\frac{dP_{sucess}}{dp} = 0 \Rightarrow \boxed{p = 1/n}$ At value of $p = 1/n$ the sucess probability will be maximum.

$$\Rightarrow \boxed{P_{max} = (1 - 1/n)^{n-1}}$$ Now, $\boxed{\lim_{n \to \infty} (1 - 1/n)^{n-1} = 1/e}$

2) No. of tries or) times we should try before getting first sucess = $1/P_{max} = e$

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

$$\boxed{\eta = \frac{1}{1 + 6.44a}} \quad \left(a = T_p/T_t\right) \Rightarrow \eta = \frac{1}{1 + 6.44\left(\frac{d}{v} \times \frac{B}{L}\right)}$$

$\Rightarrow \eta \downarrow$ as $d \uparrow$ {CSMA/CD NOT SUITABLE FOR WANS}

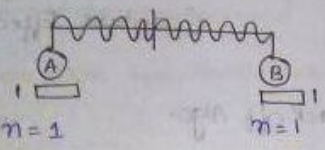
$\Rightarrow \eta \uparrow$ as $L \uparrow$ {send Bigger pkts for good efficiency}

$\Rightarrow$ The max amount of data that can be sent through Ethernet = 1500 Bytes

8. BACK OFF ALGORITHM FOR CSMA-CD

⇒ The Back-off Algorithm is used to give waiting time of a station before it starts Retransmiting after involved in a collision. This waiting time is called Backoff time

1)



n = 1 n = 1

⇒ $(0, 2^n - 1)$ ⇒ $(0, 2^n - 1)$

⇒ (0,1) ⇒ (0,1)

n = collision Number

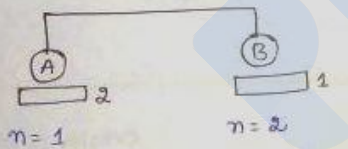
⇒ This Algo says both the stations 'A' and 'B' should randomly choose a no between $(0 \to 2^n - 1)$

∴

A	B	
0	1	⇒ A's waiting time = 0 * T_{slot} = 0 } A' won Backoff race
1	0	B's waiting time = 1 * T_{slot} = T_{slot} }
0	0	} - collisions
1	1	

p(A) = 1/4 p(B) = 1/4 p(collision) = 1/2

Now, Suppose let us Assume 'A' wins the Backoff Race ⇒ 'A' has successfully transmited its 1st Data packet and is ready with 2nd Data packet



n = 1 n = 2

$= (0, 2^n - 1)$ $= 0(0, 2^n - 1)$

$= (0,1)$ $(0,3)$

$= (0,1)$ $(0,1,2,3)$

A	B	
0	0	⇒ collision
0	1	⇒ 'A' win
0	2	⇒ 'A' win
0	3	⇒ 'A' win
1	0	⇒ 'B' win
1	1	⇒ collision
1	2	⇒ 'A' win
1	3	⇒ 'A' win

p(A) = 5/8 p(B) = 1/8 p(collision) = 2/8

⇒ Waiting Time = K × T_{slot} $K \in (0, 2^n - 1)$ n: collision number

⇒ collision probability is decreasing exponentially.

⇒ If 'A' has won first collision the pbty that it wins 2nd collision is high (Capture Effect)

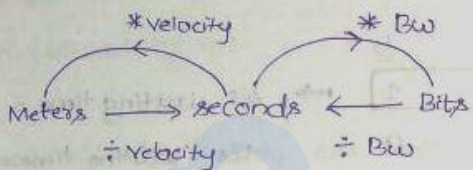
⇒ Applicable only for 2 stations.

⇒ Binary Backoff Algo or Binary exponential Backoff Algo.

9. TOKEN PASSING ACESS CONTROL METHOD

Bits ⟶ seconds
÷Bw

Meters ⟶ seconds
÷velocity



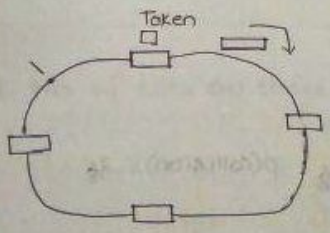
How many meters 10bits Equalent to, given that Bw 4mbps, $V = 2 \times 10^8$ m/s

Sol Bits = 10bits

$$\frac{Bits}{Bw} = \frac{10}{4} = \frac{2.5 \text{ sec}}{10^6} * 2 \times 10^8 \text{ m/s}$$

$$= \frac{5 \times 10^8 \text{ m}}{10^6}$$

Ans = 500m



1. Unidirectional flow of data
2. only one station can transmit at a time
3. If i leave a bit at some point in the ring the time taken by it to take one complete Rotation is called "RING LATENCY"

Ring latency = $\left(\frac{d}{v}\right)_{sec} + (N * b)_{Bits}$

{d = length of Ring
V = Velocity

Ring Latency = $\frac{d}{v} + \frac{(N*b)}{Bw}$ seconds (or) $RL = \frac{d}{v} * Bw + N*b$ Bits

THT = Token Holding time → THT is time for which each station holds the token

Cycle time = $\frac{d}{v} + (N * THT) = T_p + (N * THT)$

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

There are two strategies that are present in token passing mechanism they are

1) Delayed Token Reinsertion ⇒ $THT = T_t + T_p$ ⇒ $\eta = \frac{1}{1 + \frac{(N+1)}{N} a}$ $(a = T_p/T_t)$

2) Early token Reinsertion ⇒ $THT = T_t$ ⇒ $\eta = \frac{1}{1 + (a/N)}$

10. ALOHA AND DIFFERENCES BETWEEN FLOW AND ACESS CONTROL

1) Any station could transmit data at any time - No carrier sensing.
2) collisions are possible, Acknowledgements are there ⇒ Collision detection ✗
3) Retransmissions ✓ after some time (Backoff time)

ALOHA
- Pure Aloha
- Slotted Alha

Pure Aloha

1) Vulnerable time = $2T_t$
2) $\eta = G * e^{-2G}$ (G = No. of stations who wants to transmit in T_t slot)

$\Rightarrow \frac{d\eta}{dG} = 0 \Rightarrow G = 1/2$

if $G = 1/2$, η = max

$\Rightarrow \eta_{max} = 1/2 \times e^{-1}$

$\Rightarrow \eta_{max} = 0.184$

$\Rightarrow \eta_{max} = 18.4\%$

Slotted Aloha

1) Vulnerable time = T_t
2) $\eta = G * e^{-G} \Rightarrow \frac{d\eta}{dG} = 0 \Rightarrow G = 1$

$\Rightarrow \eta = 0.368$

$\eta_{max} = 36.8\%$

FC	AC
SAW: $\frac{1}{(1+2a)}$, SWP: $\frac{N}{(1+2a)}$	TDM: $\frac{1}{1+a}$, polling: $\frac{T_t}{T_t + T_{poll}}$
GBN: $\frac{N}{(1+2a)}$	CSMA/CD: $\frac{1}{1+6.44a}$
SR: $\frac{N}{(1+2a)}$	Aloha → Pure: $G * e^{-2G} = 18.4\%$; Slotted: $G * e^{-G} = 36.8\%$

3. ERROR CONTROL METHODS

1. ERROR CONTROL AND CRC

⇒ The main reason for packet loss is "Congestion"

⇒ Error handling is mainly of 2 types → Error Detection → (D+D), parity check, Checksum, CRC
→ Error Correction → Hamming code (NOT used in CN)

CRC (CYCLIC REDUNDANCY CHECK)

⇒ CRC is mainly used at Hardware Level

Sender: 1011011 CRCG: 1101 Receiver:

⇒ If CRC Generator is 'n' bits we are going to append (n-1) bits (All 0's) to the data.

Appended Bits

Appended data = 1011011 000

⇒ ⊕ = Exclusive OR (or) modulo 2 sum

1⊕1 = 0 1⊕0 = 1
0⊕0 = 0 0⊕1 = 1

Now,

```
1101) 1011011000 (
      1101
      ----
      0110011000
       1101
      ----
      0001111000
        1101
      ----
      0011100
        1101
      ----
      0001
```

⇒ If CRC G = n bits then CRC = (n-1) bits

⇒ Always start Applying XOR from the Leading 1 (Starting 1)

Now the least significant 3 bits will be the "CRC" = 001

∴ The Zeros that you appended at the begining will be replaced by the CRC and gets transmited.

∴ The data that will be sent to Receiver is = Actual data + CRC

= 1011011 001
AD CRC

At Receiver side, Whether the data that is transmited is Right or wrong

```
1101)1011011001(
     1101
     ----
     01100011001
      1101
      ----
      00011001
       1101
       ----
       001101
        1101
        ----
        0000
        =CRC
```

If you get all zeros as CRC then the data that is transfered is error free

2. CRC EXAMPLE

CRC: $x^3+x+1 = 1\cdot x^3+0\cdot x^2+1\cdot x+1\cdot x^0 = 1011$ is the CRC

If the degree of the polynomial = 3 the no.of bits in CRC Generator are 4 and we append '3' 0's ∴ Degree of the CRC polynomial = Size of CRC generated.

At sender

Data: 11010

CRC: $x^3+x+1 = 1011$

∴ Appended data = 11010 000

```
1011)11010000(
     1011
     ----
     01100000
      1011
      ----
      0111000
       1011
       ----
       010100
        1011
        ----
        00010

CRC = 010
```

At Receiver side

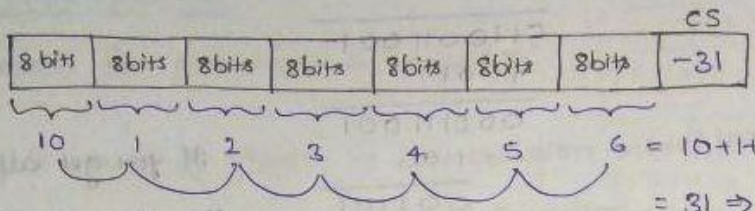
```
1011)11010010(
     1011
     ----
     01100010
      1011
      ----
      0111010
       1011
       ----
       010110
        1011
        ----
        00000  = All 0's ✓
```

3. CHECKSUM

⇒ Tcp, Ip, UDP uses 16 bit checksum, and checksum is used at Software level.

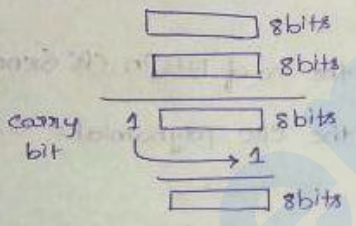
⇒ The entire stream of data (bits) are divided into many parts of equal sizes which is equal to the size of checksum that you need i,e If you are using 8 bit checksum then divide the entire stream of data into equal parts and size of each part should be 8.

⇒ Now encode the decimal values of all the parts and add them all, finally than ans you get must be Negotiated (if we get 34 you should write -34) and add this value (-34) to the checksum field.



6 = 10+1+2+3+4+5+6
= 31 ⇒ CS = -31
Ones Compliment

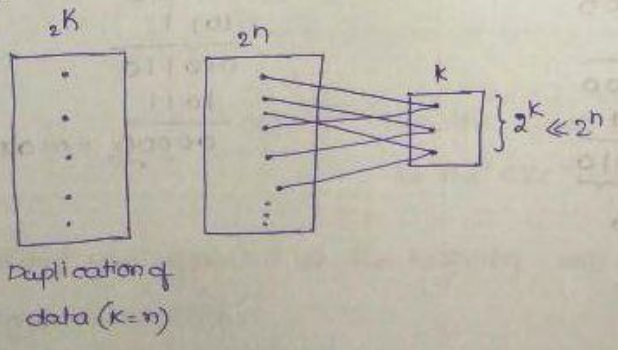
⇒ Suppose if i add two 8 bit no's in the above checksum then i may get a 9 digit Binary number then you take the LSB and add it to the obtained number, This is called "wrap Around."



4. CN - SUMMARY

⇒ No method is actually Reliable, the data may be corrupted (or) the Error handling bits may be corrupted.

⇒ Meaningful errors are nothing but, data is being corrupted in such a way that no one is able to detect it because error handling bits are also changed.



4. ISO/OSI STACK

1. ISO - OSI LAYERS

Functions

Mandatory	optional
1. Error Control	1. Encryption and Decryption
2. flow control	2. Checkpointing
3. Access control	3. Routing
4. Multiplexing and Demultiplexing	⋮
5. Addressing	70 functions (Both Mandatory and optional)
⋮	

⇒ To implement all the above functionalities there are various Reference models which classify all the above functionalities and define what functions are carried out at a particular layer. one of the Reference model is "ISO-OSI STACK".

1) ISO - OSI
2) TCP/IP
3) ATM
4) X.25
5) IEEE (Mainly deals with LAN Technologies)

} various Reference models

⇒ ISO - OSI stands for "International standard organisation - open system interconnection"

⇒ The various layers in the OSI model are:

User Interactiveness
- ← 1) Application layer
- ← 2) Presentation layer
- ← 3) session Layer

Thick layer ← 4) Transport Layer

complex ← 5) Network layer

Both hw/sw ← 6) DataLink layer

hw ← 7) physical layer

1) 1 Advantage of Layering is
1. Divide and conquer
2. Encapsulation is possible
3. Abstraction.
4. Testing is made easy

{ Google out: RFC (Request for comments) for learning CN }

2. PHYSICAL LAYER

⇒ physical layer deals with Electrical, Mechanical, Functional, procedural characteristics of physical Links.

⇒ Transmission modes: (S) ⟶ (R) (simplex / half duplex)

(S) ⇄ (R) (walkie talkies)

(S) ⇄ (R) (Full duplex)

⇒ Deals with Topologies.

⇒ Deals with Encoding (Bits → Signals, waves) 1010

MANCHESTER ENCODING

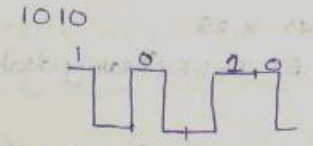
1 - Represented by ⌐¬

0 - Represented by ¬⌐

DIFFERENTIAL MANCHESTER ENCODING

0 - Represented by ⊓ .. ⌐¬

1 - Represented by ⌐ ¬



In both the Encodings (Manchester and Differential manchester)

Baud Rate = 2* Bit rate { Baud = a letter with 4 letters
Bit = a letter with 3 letters }

Baud Rate = No. of voltages that are being sent per second

Bit Rate = No of Bits that are sent per second.

3. INTRODUCTION TO DATA LINK LAYER

The main functions of DLL are 1> flow control (SAW, GBN, SR)

2> Error control (CRC, checksum)
↓ DLL ↓ TCP/IP

3> Framing

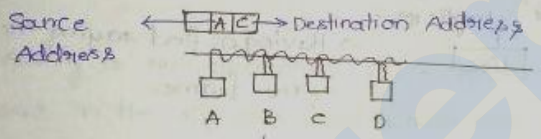
4> physical Addressing

5> Access control (CSMA/CD, Token passing..)

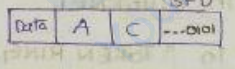
4. FRAMING AT DLL

⇒ At DLL we are mainly talking about LANS and all the LANS are multiaccess channels (All the stations are connected to a single Broadcast medium)

⇒ Framing is nothing but taking the data from the above layer (NL) and putting it in a frame and sending it



⇒ The starting of the packet is seen by B and it will discard it as the destination Address is 'C', so every station should know that a frame is Begining so it has a special field called "SFD" (Starting Frame Delimiter)



⇒ Now, SFD Represents that the frame is comming and Alerts all the stations, generally "SFD" is a sequence of bits that is entirely different from the data

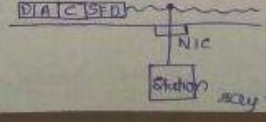
The bits in the "SFD" can be represented with the help of Regular Expression

$(10101010\cdots11)(0+1)^*$

SFD — The data that is present

Now we can convert Regular Expression → DFA → Sequential Circuit (Built using Flipflops)

Now, this sequential circuit has the capability of Recognising the Bits/RE ie $(10101010\cdots11)(0+1)^*$ so,



At this point the sequential circuit alerts the station and say that see a frame is going if it is for

⇒ It is not just enough that you identify the start of the frame (SFD) it is important to find where the frame ends.

⇒ Traditionally framing has been divided into two types — Fixed Length, Variable Length

⇒ The disadvantage of Fixed Length framing is it has Internal Fragmentation

Ex: L = 1000 Bytes ⇒ The min and max amount of data that can be sent is 1000 Bytes, if i have to send 100B then ??

100B + [900 Bytes] Dummy bits Added [This procedure is called padding].

⇒ To identify the ending of frame there are two methods they are,

1. Having a field saying the Length of the frame
2. Having an Ending Delimiter

a) Length = 10 | length = 1000 | Length = 100

[10] [1000] [100]

⇒ Having a field saying the length of the frame.

b) Having the End Delimeter: [ED | D | A | C | SFD]

ED — End of the frame; D — Data transmitted; A — Source Address; C — Destiny Address; SFD — Starting of frame

⇒ Fixed Length frames are used in "ETHERNET."

⇒ variable Length frames are used in "TOKEN RING."

⇒ The problem with having the length field is, if it gets corrupted then Receiver reads only the data upto the modified value of Length field.

⇒ The problem with End delimeter is if ED pattern matches with Data (part) then the receiver cannot scan the entire data available

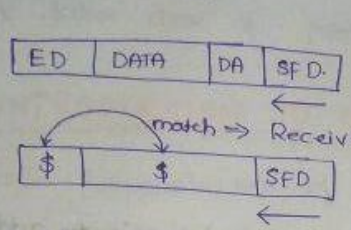
[ED | DATA | SFD]

pattern 2

pattern 1 if p1 = p2 then Receiver scans only a part of data

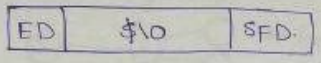
⇒ End Delimeter can be dealt in 2 ways they are : charecter stuffing
Bit stuffing.

CHARECTAR STUFFING

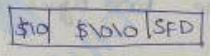


⇒ OBSELETE TECHNOLOGY

⇒ so, add a null charectar after $ so that the Receiver thinks / sees a '\0' followed by a '$ then it assumes it is not the end of data and it discard the Null charectar and scans the entire data.

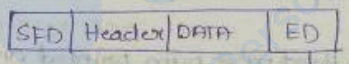


Now, if ED = $\0 and the data is $\0 then there is problem so add again null charectar $\0\0 to the data. ⇒



⇒ so Adding a Null charectar for each and every match with the charectar present in the ED is called "charectar stuffing."

BIT STUFFING (V. Imp FOR GATE)



Sender

D: 01111
⇓
D: 011101

Receiver

01111
⇑
0111 ⊗ 1

D: 011101 ⟶ 0111 0 ⇒ 01111

→ The rule at sender side is when you see 0,31's add an Zero

If ED = 011111

Sender side

D: 011110
SD: 0111100 (After four ones add a zero)

D: 011111
SD: 0111101

Receiver side

D: 01111⊗0
01110

SD = Stuffed Data

1) ED = 01111
D = 0111 000 11110 } Now, what is the data after Bit stuffing?

Sol Data = 0 111 000 11110

Stuffed Data = 0111 0 000 111010

2) ED = 0111111

Data ⇒ At the sender after the Run of 6 ones im going to add zero

= 0111111 0 1

⇒ At the Receiver side after a '0' followed by 6 ones im going to delete a zero.

4. PHYSICAL ADDRESSING

There are two types of Addresses — physical address - static, constant
— Logical Address.

⇒ Now physical Address should be unique within the Network

⇒ Logical Address should be unique in the entire World wide web.

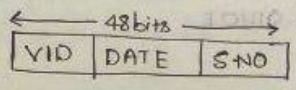
⇒ Ip Address = 32bit Number, software num.

⇒ MAC Address = 48bit Number, Hardware num printed on our "NIC" → ROM → MAC

The MAC Address is divided into three parts (unequal lengths) they are

1. The Manufacturer ID or VendorID
2. What is the date of manufactoring
3. Serial NO. of the device on that day

} UNIQUE GLOBALLY



⇒ Conceptually Both Ip, MAC Addresses can be logical but only Ip Addresses is used for Logical Address because the info in the Ip Address (NID, HID) are sufficient for Routing the packets. where as this is not the case in MAC Addressing.

⇒ MAC is a physical Address.

⇒ "APPLE TALK" is the N/w in the world that doesnot use the MAC Address it artificially generates a random Number and assigns to the users.

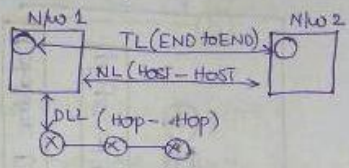
⇒ Data Link Layer is divided into two parts they are: 1> Logical Link Control
2> Medium Access control

⇒ LLC takes care of Error control, Flow control, (12)
⇒ MAC takes care of Framing, Access control, Error control, physical Address

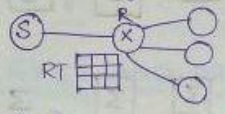
L-6: INTRODUCTION TO NETWORK LAYER

The main Responsibilities of Network layers is

1> Host to Host connectivity

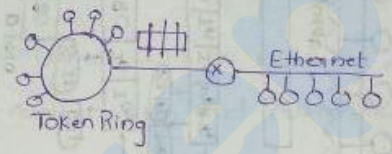


2> Logical Addressing
3> Switching (connecting various Networks together)
4> Routing



⇒ Building the Routing table = Routing
⇒ Using the Routing table = Switching

5> Congestion Control
6> Fragmentation.

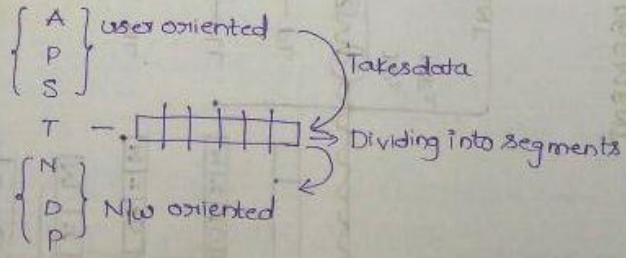


Max frame size = 1500B

7. TRASPORT LAYER

The main Responsibilities are

1> End to End communication using port numbers (port nos are also called Service point Addressing)
2> Flow control (SR)
3> Error control (checksum)
4> Segmentation ⇒



5> Multiplexing and Demultiplexing.
6> Congestion control

8. HOW ALL THE LAYERS WORK TOGETHER

⇒ The protocol used to convert IP Address to MAC address is 'ARP'.

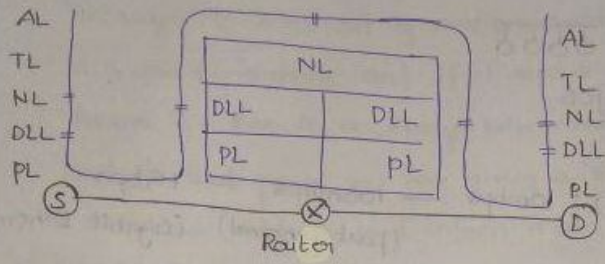
⇒ Router will have only three layers (NL, DLL, PL)

⇒ In general the physical layer and Datalink layer are present on NIC and Network layer is present in the operating system.

⇒ In IP there are no Acknowledgements. (Connectionless datagram service)

Imp points for Gate

⇒ Routers have only 3 layers 1.PL 2.DL 3.NL



= The no of times a packet hit the DataLink layer = 4

= The no of times a packet hit the Network Layer = 3.

= Depending on No of Routers we have the no of time each layer hits depend.

9. SESSION LAYER AND PRESENTATION LAYER

SESSION LAYER

1) passwords and usernames will be dealt by session layer

2) Main Responsibility Authorization and Authentication (using Digital signatures)

3) check pointing (Torrentz example).

4) Synchonization.

5) Dialog control

6) Logical Grouping of operations

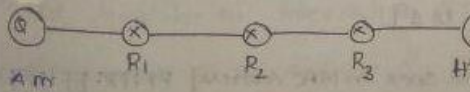
PRESENTATION LAYER

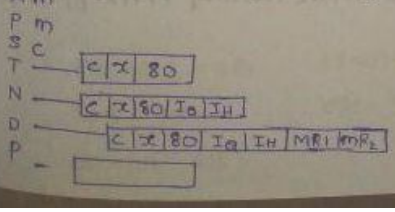
1) character translation

2) Encryption and Decryption.

3) Compression (.zip)

GATE - 2014

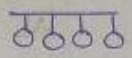




} These are the situations when the packet reaches "R2." ⇒ Intruder could find Ip's of S,H and port numbers of S,H

5. LAN - TECHNOLOGIES

ETHERNET (IEE 802.3)

1> Topology: Bus topology 

2> Acess control method: CSMA/CD

3> NO Acknowledgement

4> Data Rates (Bandwidth): 10mbps — 100mbps — 1Gbps
(Fast Ethernet) (Gigabit Ethernet)

5> Encoding technique: Manchestar Encoding (Baud Rate = 2* Bit Rate)

6> "Ethernet" operates at Data Link Layer. (LAN technologies are dealt at DLL)

AL = Message

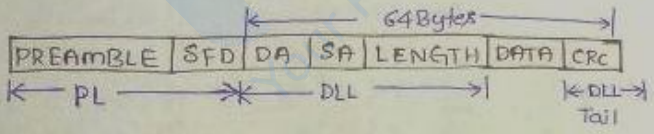
TL = Segment

NL = Datagram

DLL = Frame

PL = Single protocol Data unit. (I-PDU)

FRAME FORMAT OF ETHERNET (OR) IEEE 802.3 FRAME FORMAT



PREAMBLE: 7bytes : 10 10 1010 10 ----- 10 ⇒ used to alert all the stations

SFD: [Start frame Delimeter]: 1Byte : 10 10 10 11 ⇒ Indicate Start of frame, Synchronization.

DA: 6Bytes } MAC Address
SA: 6Bytes }

CRC: 4 Bytes

TYPES OF MAC ADDRESS

Ex: 1A:2B:34:48:56:6F

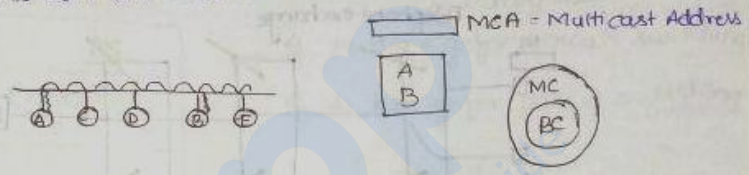
1. UNICAST MAC ADDRESS: [LSB of 1st byte is 0] 00011010 : 00101011 :

2. MULTICAST MAC ADDRESS: [1st Byte's LSB is 1]

3. BROADCAST MAC ADDRESS: [All the bits are ones in MAC Address] FF:FF:FF:FF:FF:FF

⇒ The multicasting MAC Address is implemented in this way

⇒ Consider a LAN containing many stations, Now, i want to send a message to two stations A and B (or) Repeatedly i want to send messages to some set of stations inside my LAN, Let us Assume that A, B are my stations and if I send a message both of them should Receive it. Then it is always better that you create a group for them and for that group you are giving a Group ID and that ID should be a multicast Address, and inform A and B that see you both are present in a Multicast Group with some Address then Both A/B will configure their NIC's in such a way that whenever any packet is sent to that particular Multicast Address both of them will automatically Read and others will discard it.



LENGTH : 2Bytes = 16bits

: The main reason for including Length field is ETHERNET follows variable length frames

: In CSMA/CD : $L \geq 2 * Tp * Bw$ substituting the standard values of Tp, Bw for Ethernet we get

$L \geq 64$ Bytes

: Max Length : 1500 Bytes

	MIN	MAX
DATA	46B	1500B
FRAME	64B	1518B

DISADVANTAGES

1) Not applicable to Real time Applications
2) Not Applicable to Interactive Applications [chatting].
3) No priorities so not suitable for client Server Applications.

∵ PREABLE = 7B
SfB = 1B
SA = DA = 6B
LENGTH = 2B
CRC = 4B

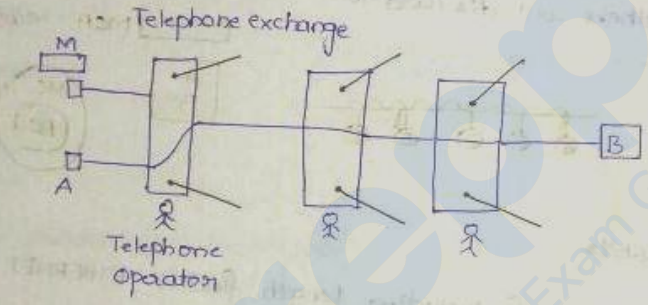
6. SWITCHING

1. INTRODUCTION TO SWITCHING

Switching
- Circuit switching ⇒ Applied at physical Layer, Obsolete technology.
- Message switching
- Packet switching
  - Virtual circuits switching
  - Datagram switching.

⇒ Switching is done at Network Layer

2. COMPARISION BETWEEN CIRCUIT SWITCHING AND PACKET SWITCHING



Let M = size of total message that 'A' wants to send to 'B'.

B = Bandwidth of the entire channel

X = No. of Hops

d = Length of Each hop

V = Velocity of signal

Total time taken to send message 'm' to 'B' is = Setup time (connection establish time) $+ T_t + \frac{xd}{V}$

$= \text{Setuptime} + \frac{M}{B} + \frac{xd}{V} + TD$ { T_p btn two hops $= d/V$ ∴ In btn x hops $= x \cdot d/V$ } → Teardown the link

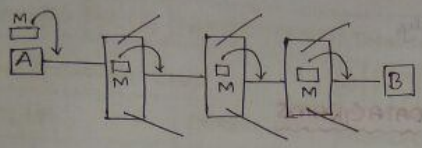
$$TT = \text{Setuptime} + M/B + \frac{xd}{V} + TD$$

⇒ In the above scenario Headers are not Required. (25)

⇒ Inorder ✓ (No Reordering of packets) ⇒ No sequence nos are Required.

PACKET SWITCHING

⇒ Multiplexers are used.



⇒ store and forward N/w.

Total time = $\times \frac{M}{B} + \times \frac{d}{v}$

No setup and Teardown time in P.S.

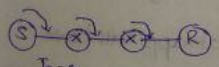
CS	PS
Setup time + Teardown tym	$(x-1)\ M/B$

If M = Big (or Bursty data) = go for circuit switching

If M = less go for packet switching

3. PACKETIZATION AND PACKET SWITCHING

Data = 1000 Bytes
Bw = 1MBps = 10^6 Bps
Header = 100 Bytes
No of packets = 1
packet size = $\left(\frac{1000}{1}\right)$ + Header
= 1000 + 100
= 1100 B.



$T_p = 0$

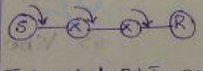
$T_t = L/B = \frac{1100}{10^6} = 1.1$ msec

$TT = 3 * T_t = 3.3$ msec

TT = 3.3ms

Data = 1000 Bytes
Bw = 1MBps = 10^6 Bps
Header = 100 Bytes
No of packets = 5
packet size = $\frac{1000}{5}$ + 100

PS = 300



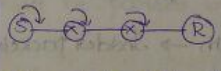
Ist packet $3 * T_t = 3 * \frac{300}{10^6}$
= 0.9ms

Remaining 4 packets = $4 * T_t$
= 4 × 0.3
= 1.2ms

TT = 2.1ms

Data = 1000 Bytes
Bw = 10^6 Bps = 1mbps
Header = 100 Bytes
No of packets = 10
Each packet size = $\frac{1000}{10}$ + 100

PS = 200



Ist packet = $3 * T_t = 3 * 0.2$
= 0.6ms

Remaining 9 packets = 9 * 0.2
= 1.8ms

TT = 2.4ms

∴ packetization is the process of dividing the data into packets and transmitting.

⇒ The packetisation process helps in reducing the total transmission time and the size of the packet should not be so small if it is the case the total time taken to transmit the data will increase. so the packet size must be choosen Appropriately.

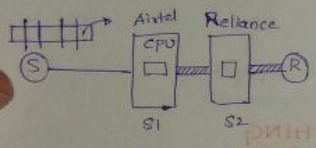
4. VIRTUAL CIRCUITS AND DATAGRAMS

There are two types of packet switching they are Virtual circuits, Datagrams

⇒ our phone call is going through Virtual circuit.

⇒ our Data usage is going through Datagram circuit

Virtual circuit



1> Headers are not Required

2> for the 1st packet, Global Header for other packets, local header

3> Connection oriented (Resource Reservation) if Resources are Reserved then it is connection Oriented. (Buffer, CPU, Bw)

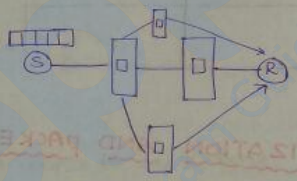
4> Same path → Inorder packets

5> Highly Reliable

6> Costly

7> ATM (Asynchronous Transfer mode) Network uses ATM. VC.

Datagram Circuit



1> Headers are Rd for all packets

2> Connectionless oriented

3> NoGuarantee, they may follow diff path, they may appear out of order.

4> Not Reliable

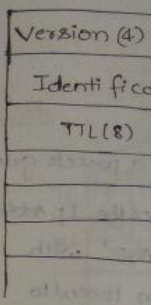
5> Not very costly.

6> Viber, whatsapp call have VoIp (VOICE OVER INTERNET PROTOCOL)

7> Ip Network uses Datagram

1. INTRODUCTIO

IpV4 Header



HL:

Here the min
HLF is of 4
so we use th
∴ If H

VERSION : The
if version=
appropriately.

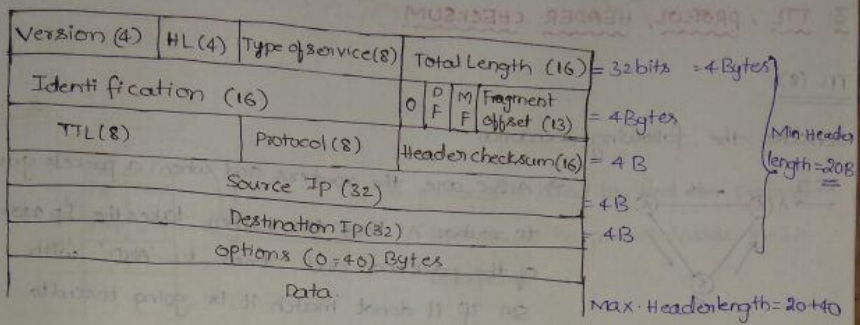
2. IDENTIFICA

⇒ Identification
out of a host.
we do fragm
number.

7. INTERNET PROTOCOL

1. INTRODUCTION TO IP HEADER

IpV4 Header



HL:

Here the min size of the Header is 20B but the Header Length field HLF is of 4 bits ∴ only the numbers ranging from (0-15) can be represented so we use the process called Scaled Arithematic (divide by 4)

∴ If Header Length = 20B ⇒ HLF contain (5)

HL	HLF
20B ⟶	5
32B ⟶	8
40B ⟵	10

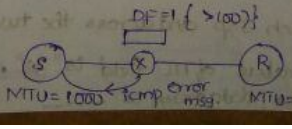
⇒ In exam if a number is given, and if you want to identify it whether it is a Header Length or Header Length field the process is, convert it to decimal and check the Ranges HL (20B - 60B)
HLF (5 - 15).

VERSION : The version (of the Ip Address IPv4/IPv6) If version = 0100 = IPV_4 if version = 0110 ⇒ IpV_6 ⇒ the packet must be parsed by the parsers appropriately.

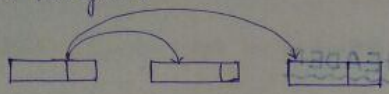
2. IDENTIFICATION, MF, DF AND FRAGMENT OFFSET

⇒ Identification number is used to number every datagram that is going out of a host. This field is mainly used when we do Fragmentation, when we do fragmentation all the fragments are going to get same identification number.

⇒ DF: Donot fragment



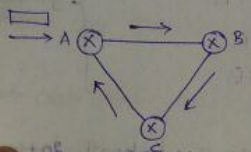
⇒ Fragment offset is no of data Bytes ahead of this particular fragment in this particular Datagram.



3. TTL, PROTCOL, HEADER CHECKSUM

TTL (8):

consider the following scenario,



⇒ A, B, C are the routers and when a packets gets to router 'A' then the router takes the Ip Address of the packend and tries to 'AND' with SM if it donot match it is going to send to default entry.

⇒ Let us say, Router A default entry = Router B and Router B's default entry goes/leads the packet go to Router 'C' and Router 'C' default entry = Router 'A'. then the packet will fall in an infinite loop and will be entirely circulating and there may be some thousands of such packets, then as a results the Buffers are full, Routers will be Busy.

⇒ In order to overcome the above problem there is a Concept called "TTL (TIME TO LIVE)" in which a packet is made to circulate only upto certain no of Hops after that many hops the packet will be discarded.

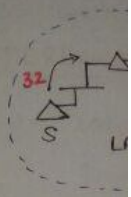
⇒ The main purpose of TTL is to discard the packet that falls in Infinit cloop

⇒ THE TTL VALUE WILL BE DECREMENTED ONLY AT THE INFERMEDIATE ROUTERS AND THE DESTINATION.

GATE -14 paper 02

In the diagram shown below, L1 is an Ethernet LAN and L2 is a Token Ring LAN. An Ip packet originates from sender 'S' and traverses to 'R', as shown. The Links within each ISP and across the two ISP, are all point-point optical Links. The initial value of TTL field is "32". The max possible value of TTL when R Receives the datagram is ____

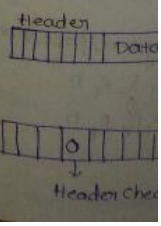
Ans: 26



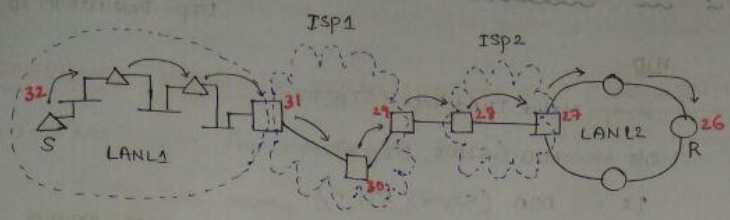
PROTCOL

⇒ Now consider
Buffers at t
at their fu
Buffers are
it is better
packet is T
is a Reliable
instead ch
ICMP, IGM
of more
in which the

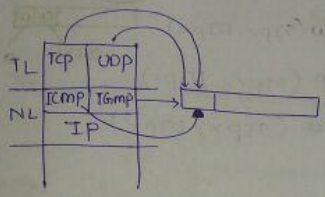
HEADER CHECKS



Ans 26

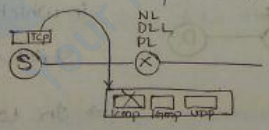


PROTOCOL

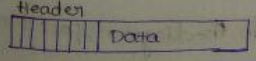


⇒ protocol field is used to find the protocol that could be present inside the IP Datagram.

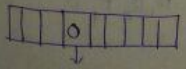
⇒ Now consider the following scenario where Tcp packet is sent, and the Buffers at the Routers are full which means the Buffers are operating at their full capacity. Now if you send a packet to a router whoose Buffers are full, then the router discards it. But before discarding it it is better that what kind of data is present in the datagram, if the packet is Tcp, the sender is going to send the same packet again as Tcp is a Reliable protocol. so if the packet is Tcp dont discard the packet, instead check the Buffers if you find the packets with the protocols ICMP, IGMP discard them and make room for the Tcp packet as it is of more Importance. so that is why we need protocol field. The order in which the routers discards the packets is 1) ICMP 2) IGmp 3) UDP 4) Tcp



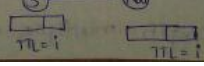
HEADER CHECKSUM(16):



⇒ 16Bit ⇒ 16Bit checksum ⇒ The Header is divided into no of parts each of size = 2Bytes = 16 Bits



Header Checksum (Initially Zero)

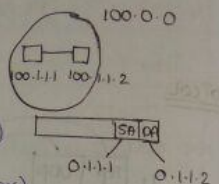


1) FO, MF, TTL
2) Options
3) HL

SOURCE IP AND DESTINATION IP

SIp = Source Ip Add
DIp = Destination Ip Add

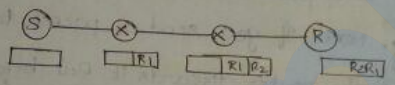
NID	HID	
✓	✓	Valid IpAddress (SIp, DIp)
✓	0's	NID (SIpx, DIpx)
✓	1's	DBA (SIpx, DIp✓)
1's	1's	LBA (SIpx, DIp✓)
1's	0's	SM or NM (SIpx, DIpx)
0's	✓	Host within a N|w (SIp✓, DIp✓)
0's	0's	I dont have Ip (SIp✓, DIpx)
127	✓	Loop back Address (SIpx, DIp✓)



5. OPTIONS

⇒ This field is optinal

Record Route:



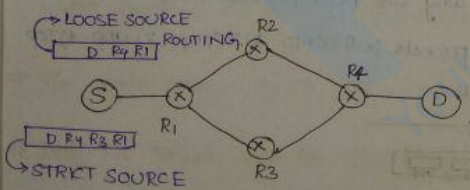
The max no of Ip Address that could be recorded ona packet is "9".

⇒ Ip Address length = 4B

⇒ optional field max length = 40B ⇒ Max no of Ip Add = 10 Ips

But one field is used to indicate the type of option: 10 − 1 = 9

SOURCE ROUTING



⇒ You will specify the entire path in which a packet must go in Strict source Routing

⇒ In Loose source routing you will indicate the Routers from which your packet must definitely go

⇒ In the diagram, packet may go as $R_1 R_2 R_4 D$ or $R_1 R_3 R_4 D$.

⇒ In addition padding is also done at "options" inorder to make Header Length multiple of '2'.

1. DIFFERENCE

MTU = Maxi
frame

⇒ The next field

∴ The max s

Message = AL
Segment = TL
Datagram = NL
Frame = DLL
I-PDU = PL

⇒ Ip is a protoc
the max size of

⇒ If we remov
payload/ datap

⇒ Here we are c
at each and eve
= 20B

⇒ Now Applicatio
now it is the
given by AL int
the process of

⇒ Now the data f
the problem is
Ethernet then

8. FRAGMENTATION

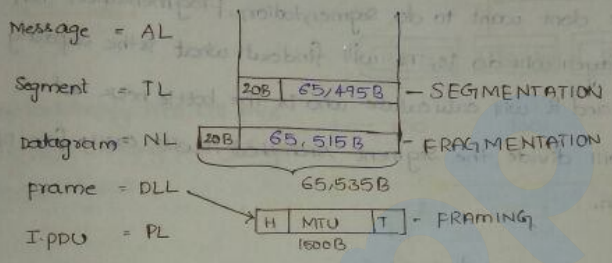
1. DIFFERENCE BETWEEN SEGMENTATION AND FRAGMENTATION

MTU = Maximum Transmitable Unit (The max amount of data that DLL frame can carry is called MTU)

⇒ The next field in the Header format = Total Length = 16b

∴ The max possible number with 16bits = $2^{16}-1$
= 65,535

∴ The max size of datagram that could be present in Ethernet in these days = 65,535



⇒ Ip is a protocol that operates at Network layer. And now, at N/w layer the max size of datagram is 65,535 ⇒ Total amount of data + Header = 65,535

⇒ If we remove the Header from the data then the remaing data is called payload/datapart at that layer.

⇒ Here we are caluculating the max amount of data that can be accomodated at each and every layer, to get max. data the Header Length should be min = 20B

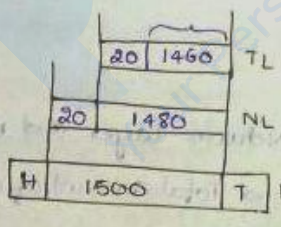
⇒ Now Application Layer could give any amount of data to transport layer, now it is the responsibility of the Transport Layer to divide that packet given by AL into parts so that each part could sit in 65,495B. And the process of dividing the packet into parts is called "Segmentation."

⇒ Now the data from the TL can easily fit into NL but here there is a problem, the problem is DLL has also some limitations, if the underlying LAN is Ethernet then the max size of data that can be fit is only 1500B.

which means NL is going to give you a data of 65,535 B, But DLL c hold a maximum of 1500 Bytes only. Now, NL should divide the datagram into parts so that each part could go and sit into 1500B and that proce is called "FRAGMENTATION".

⇒ Now, once the Datagram is received by the DLL which is already fragmented, it will add Header and Tail along with preamble, SFD-- and this process is called "FRAMING".

⇒ The problem Here is either the Network Layer is going to be Bottleneck or DLL is going to be bottleneck, Sometimes DLL can hold the data sent by NL without fragmentation, the Bottleneck will be NL. So at the same host if we dont want to do Segmentation, Fragmentation then what transport layer will do is, TL will find out what is the capacity of NL and DLL and it will calculate who is the bottle neck and accordingly TL will divide the Segment such that it will easily fit in DLL without any problem.



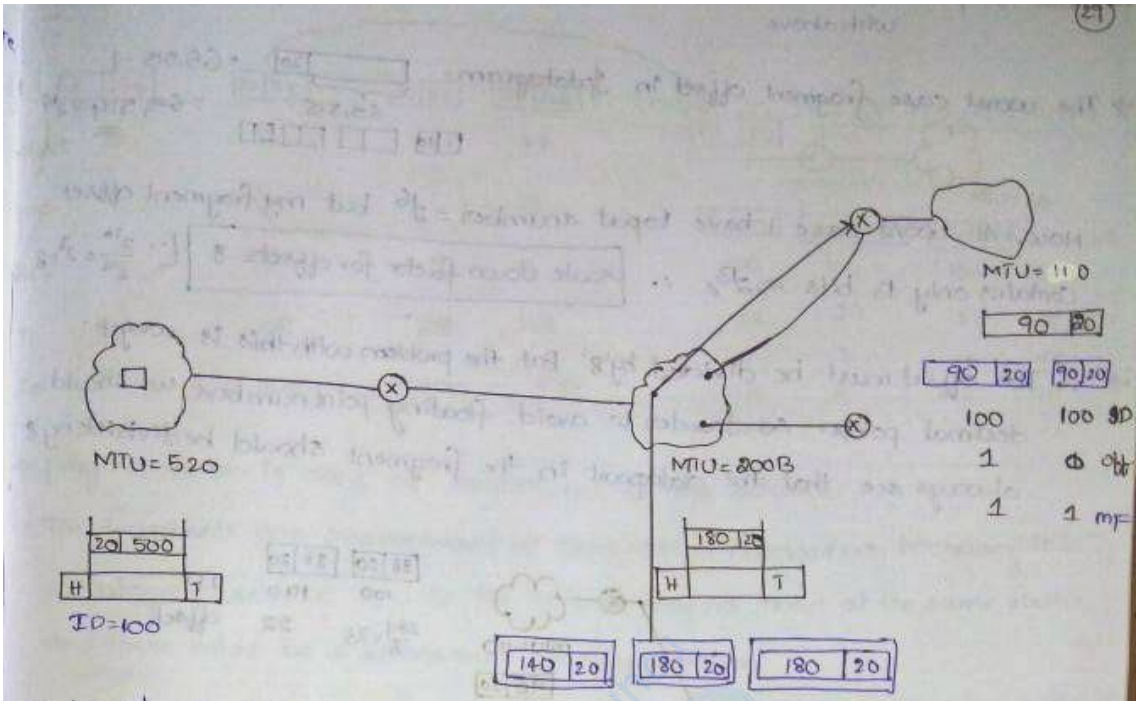
TL – SEGMENTATION ✓

NO FRAGMENTATION.

2. FRAGMENTATION EXPLAINED WITH NUMERICAL EXAMPLE

⇒ At source we do only Segmentation in such a way that there is no need of Fragmentation

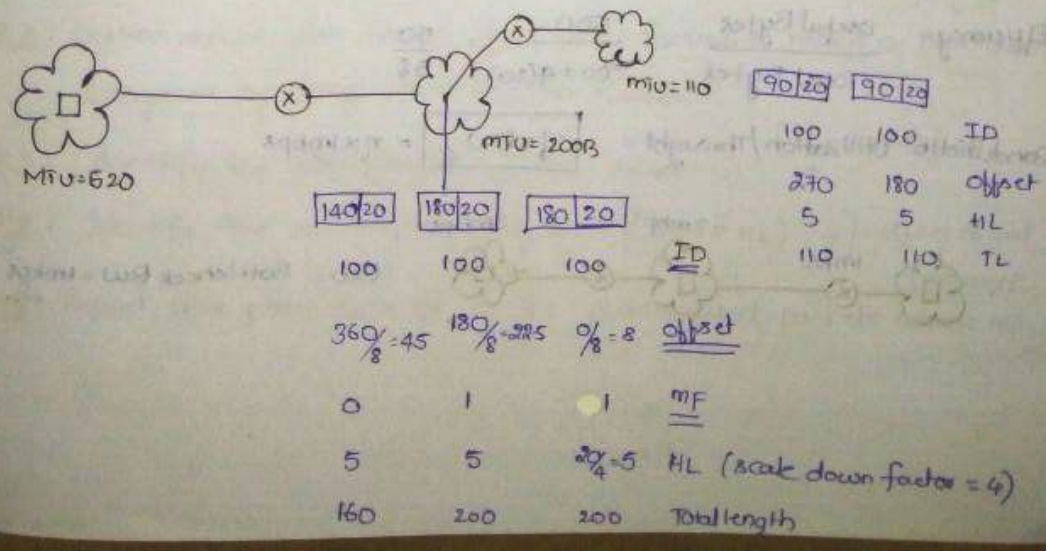
⇒ MTU= Max amount of data that can sit in DLL frame



used to identify that all this Fragments belong to a datagram whose id=100.

← 100	100	100	Identification number
2	1	0	offset (used to identify order)
0	1	1	MF (More Fragments following)

⇒ There is a problem with the above type of scenario, the offset is not actually scalable, so the offset is defined this way, offset is nothing but the no. of data bytes ahead of particular fragment. so the offset will be



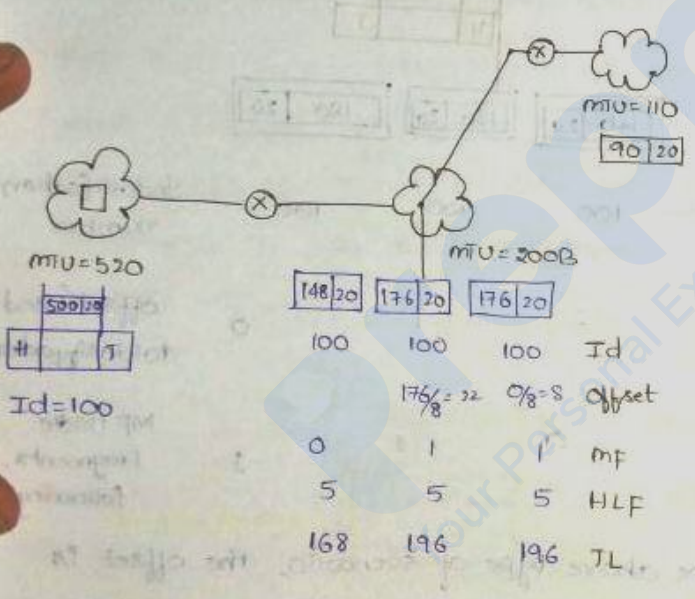
100	100	100	ID
360/8 = 45	180/8 = 22.5	0/8 = 0	offset
0	1	1	mf
5	5	20/4 = 5	HL (scale down factor = 4)
160	200	200	Total length

⇒ Now the problem scenario is, fragment offset = 13 bits ⇒ the max no in offset field = $2^{13}-1 \approx 8000$

⇒ The worst case fragment offset in 1 datagram = 20 | 65,515 = 65,515 − 1 = 65,514 ≈ 65K

Now, At worst case I have to put a number = 2^{16} but my fragment offset contains only 13 bits ⇒ 2^{13} ∴ scale down factor for offset = 8 [∵ $\frac{2^{16}}{2^{13}} = 2^3 = 8$]

⇒ ∴ The offset must be divided by '8' But the problem with this is we get decimal points. So inorder to avoid floating point numbers we should always see that the datapart in the fragment should be divisible by 8



∴ The offset for the last fragment can be decimal

⇒ Each fragment size should be divisible by 8

⇒ HLF should be divided with four.

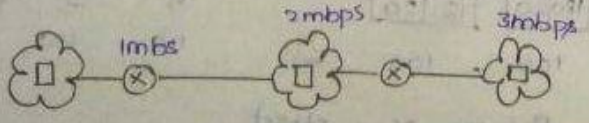
Initial data = (500B + 20B) → Split into 4 packets

Final data = 500B + (4 * 20B) ⇒ Overhead = 3 headers [(4*20) − 20 = 3*20]

(Total headers → Initial header)

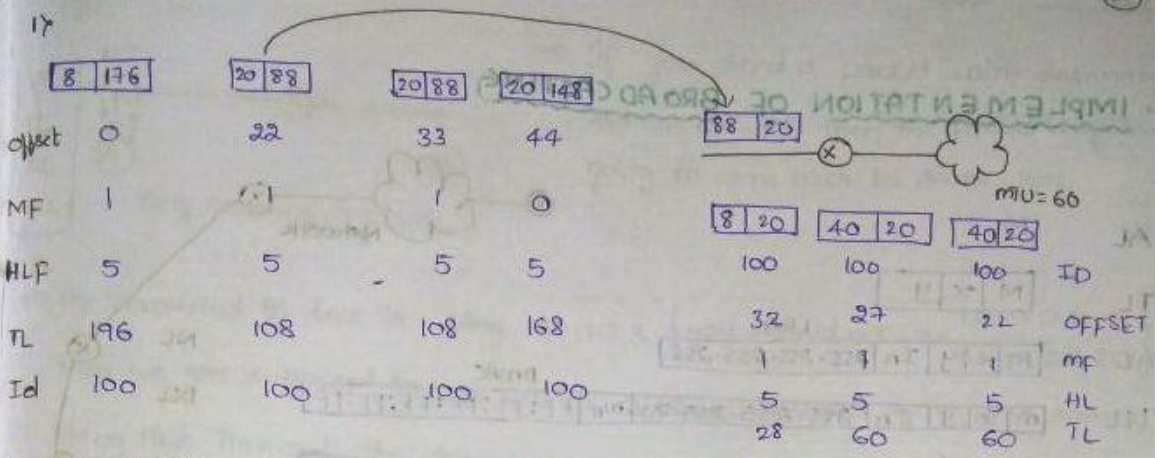
Efficiency = $\frac{\text{Useful Bytes}}{\text{Total Bytes}} = \frac{500}{500 + 4(20)} = \frac{50}{58}$

Bandwidth Utilization / Throughput = $\eta \times BW$ = $\eta \times 1$ mbps



Bw = Bottleneck BW = 1mbps

3. THEORY ABOUT FRAGMENTATION



⇒ Fragmentation is done at Router, not at the source

⇒ The fragments are reassembled at Destination not routers because, it is a Datagram service and all the routers may not meet at the same router, and there might be a further need for fragmentation.

4. REASSEMBLY ALGORITHM

⇒ Depending on MF and offset the Receiver will identify that the Datagram is fragmented.

MF	offset		
1	0	⇒ Ist fragment	FRAGMENTATION
1	!0	⇒ Intermediate fragment	
0	!0	⇒ Last fragment	
0	0	⇒ only single Datagram = No Fragmentation.	

Step-1 : Destination should identify that DataGram is fragmented (MF, OFSET)

Step-2 : Destion should also identify that what fragments belong a particular datagram by using Identification number.

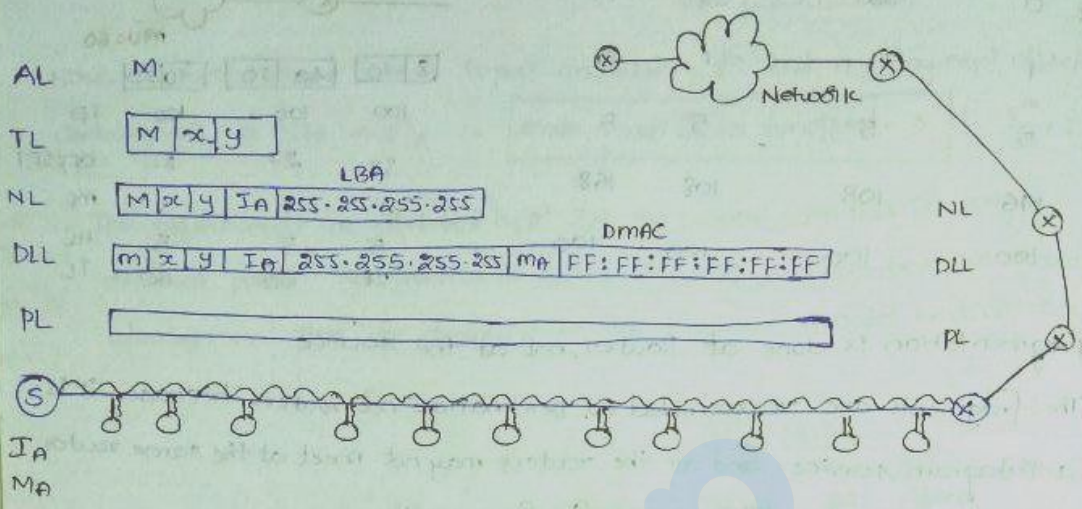
Step-3 : Identify the first fragment (MF=1, offset = 0) $[TL-(HL*4)] \div 8$

Step-4 : Identify the subsequent fragments $[(Data \div HL) \div 8]$ = offset of 2nd fragment

Step-5 : Repeat this procedure for all the subsequent packets where MF=0

9. PROTOCOLS AND CONCEPTS AT NL

1. IMPLEMENTATION OF BROADCASTING

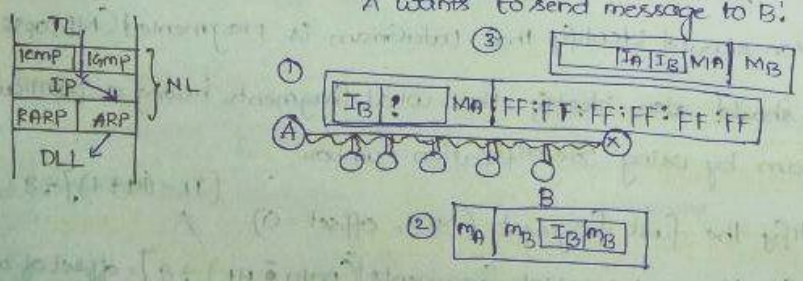


⇒ Even though Broadcasting is done at NL it cannot be done without the Broadcasting features of Data Link layer.

⇒ For Direct Broadcasting the field where LBA is present in above figure will be Replaced by DBA and DMAC will be filled by the next router and then router routes the packet as Unicast message and when the Router forwards to the Network then the DBA will be changed to LBA and DMAC will be all 1's.

2. ARP

ARP (IP Address → MAC Address)

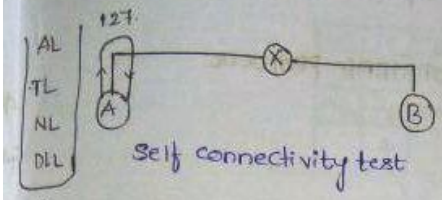


⇒ ARP Request is a Broadcasting packet

⇒ ARP Reply is a unicast packet

⇒ ARP is applied when a Router wants to find MAC Address of another Router (R-R)
host wants to find MAC Address of another host (H-H)
" Router " " " " " " router (H-R)

3 SPECIAL ADDRESS - 127



⇒ If you send a packet with destination Address as 127 then the packet is again going to come back to same host.

⇒ The command to test is ping 127.1.2.3 {you should not use 127.0.0.0 / 127.255.255.255} then we are supposed to see RTT as ≅ 15ms instead if it sees some thing like Time out the something is wrong in our NIC and you should start troubleshooting.

⇒ It is also called Loop Back Address

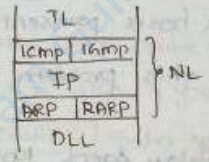
4. RARP

RARP = Reverse ARP (MAC ⟶ Ip)

NFS = Network File Server.

MAC → ROM

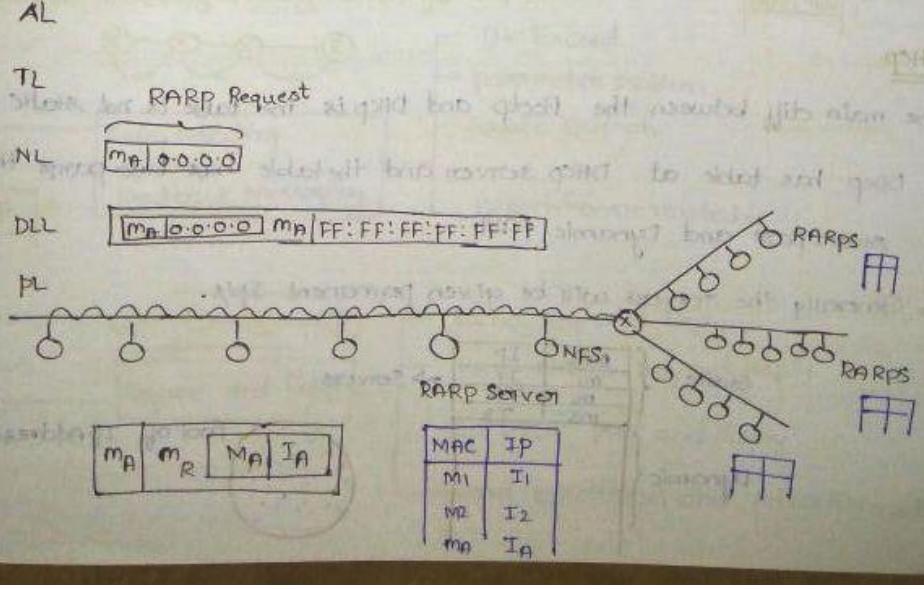
IP → RAM



1) The disadvantages of RARP are,

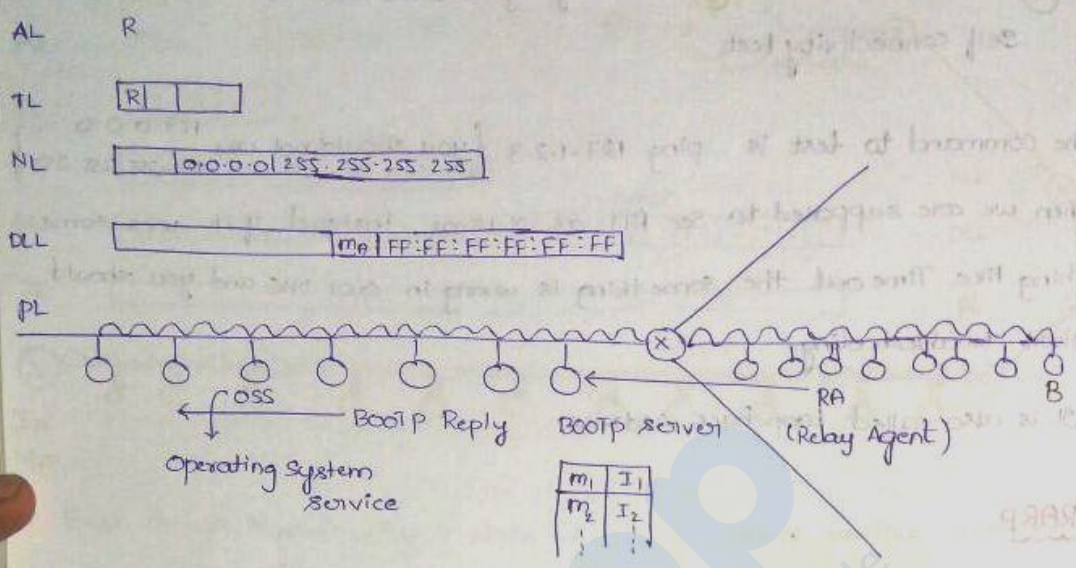
a) Every N/w should have a RARP server OBSELETE TECHNOLOGY

b) Static mappic (no.IP > no. of Hosts)



5. BOOTP AND DHCP

⇒ The main difference between Bootp and RARP is, BOOTP works at AL, and RARP works at DLL. BOOTP = BOOTSTRAP PROTOCOL



⇒ Now the hosts present in other N/w are never going to reach the BOOTP server, this problem is solved by Relay Agent.

⇒ The Relay Agent knows the Ip Address of the BOOTP server and the packet that is Broadcasted by Station B will be Read by RA and RA will ask the Bootp server by sending a Unicast packet.

⇒ only one Bootp server is Required

⇒ Mapping table is static.

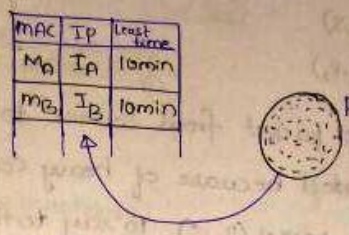
DHCP

The main diff between the Bootp and DHCP is the table is not static in DHCP

⇒ DHCP has table at DHCP server and the table has two parts they are static part and Dynamic part.

⇒ Generally the servers will be given permanent Ip's.

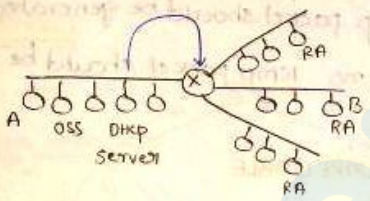
⇒ Let us say, A host M_A wants its Ip Address then the Dynamic port look like this



⇒ only one DHCP server is Enough

⇒ The Stable is a Dynamic table (No. of Ip Address = No. of stations online)

⇒ To make DHCP backward compatable with BOOTP, both DHCP, BOOTP should have same port numbers.



⇒ DHCP is operated at Application layer

⇒ DHCP cannot be implemented on Router as Router has only '3' Layers

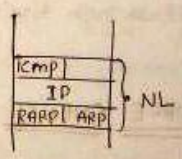
⇒ But practically DHCP can be applied on the routers also, if you Buy cisco's CASR5K Router you are going to be provided with DHCP facility at Router itself.

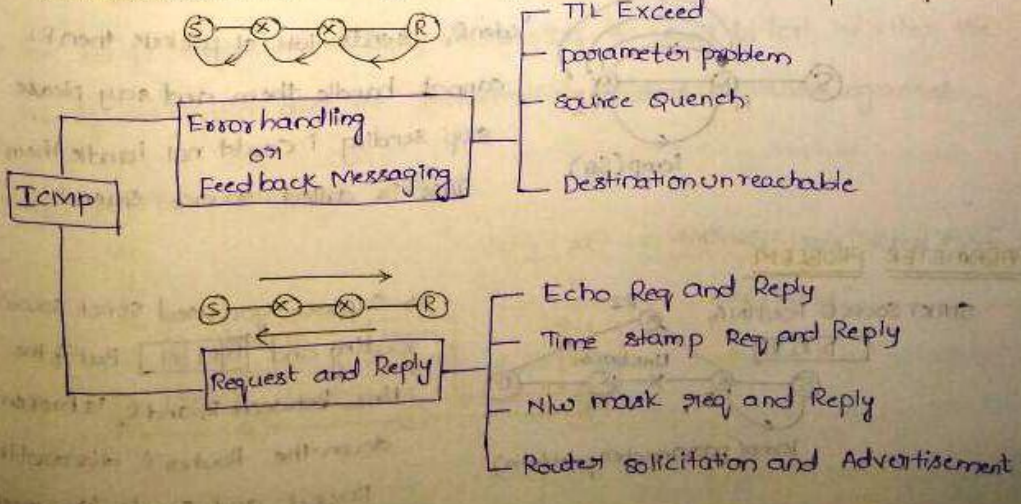
6. INTRODUCTION TO ICMP (v. Imp for GATE)

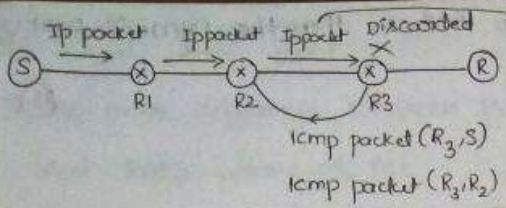
→ ICMP works at Network Layer

→ ICMP = Internet control message protocol

(whenever something is wrong then we get icmp packet)







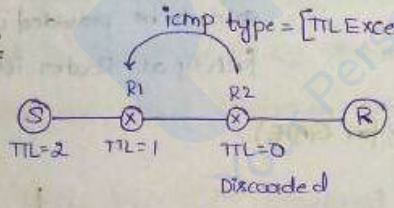
⇒ Suppose you are sending an Ip packet from Source to Receiver but on the way the Ip packet is discarded because of heavy congestion at Router R_3 Now the Router R_3 sends icmp packet (R_3,S) to say to the source that i discarded your packet. Now, the Router R_3 sends "Icmp (R_3,S)" first to Router R_2 but because of heavy congestion at R_2 the Icmp packet (R_3,S) is discarded at 'R_2' so 'R_2' again sends an Icmp packet to 'R_3' and because of heavy congestion at R_3 again the Icmp packet is discarded this scenario will lead to Infinite loop

⇒ Ip packet is lost the Icmp packet should be generated but if Icmp packet is lost then no Icmp packet should be generated

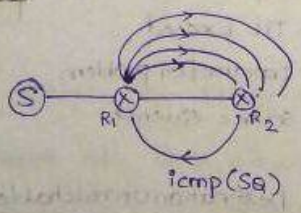
∴ (Icmp + IP) UNRELIABLE

7. ICMP FEEDBACK MESSAGING

TTL EXCEED

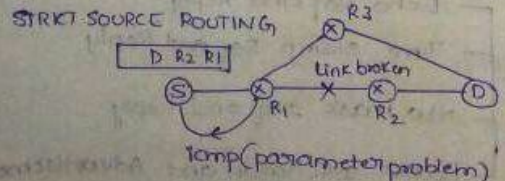


SOURCE QUENCH



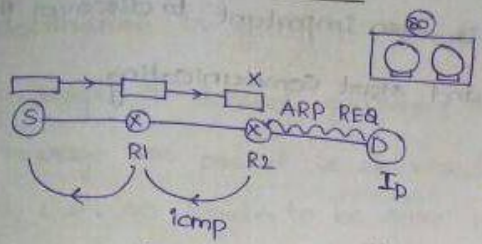
"when R_1 sends lots of packets then R_2 cannot handle them and say please stop sending i could not handle them" This is called source quench.

PARAMETER PROBLEM



⇒ Suppose you used strict source routing and [D|R2|R1] But if the link between R_1 and R_2 is broken down the Router 'R_1' discard the packet and sends icmp message

DESTINATION UNREACHABLE

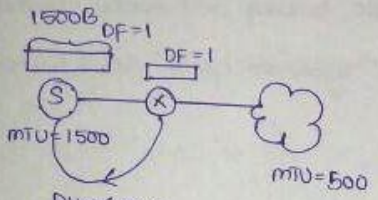


(DH unreachable)

⇒ Destination unreachable is of 2 types they are

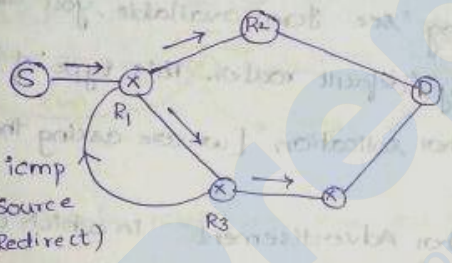
1. Destination Host unreachable
2. Destination port unreachable

⇒ R2 send ARP Req for getting MAC Addd if the Host is down then there wont be reply then the Router R2 sends icmp DH unreachable error message").



DHU = Destination Host unreachable
DF = Donot fragment = 1
FR = Fragmentation Required.

SOURCE REDIRECT

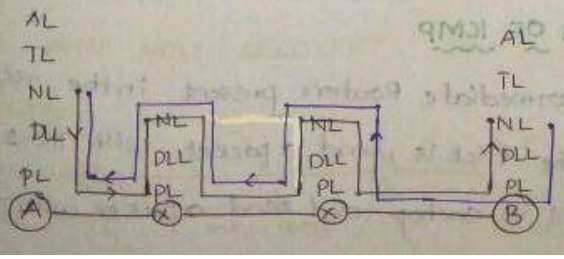


⇒ Now source sends a pacet to R_1 and at R_1 due to some errors in the Routing table if it has choosen Router R_3 (actually the best path is "R_1 R_2 D") then if there is some manual mechanism in R_3 which knows that there is a better path btn 'S' and 'D' then 'R_3' sends icmp packet saying "see there is another better path so you please redirect dont send to me next time".

8. ICMP REQUEST AND REPLY MESSAGING

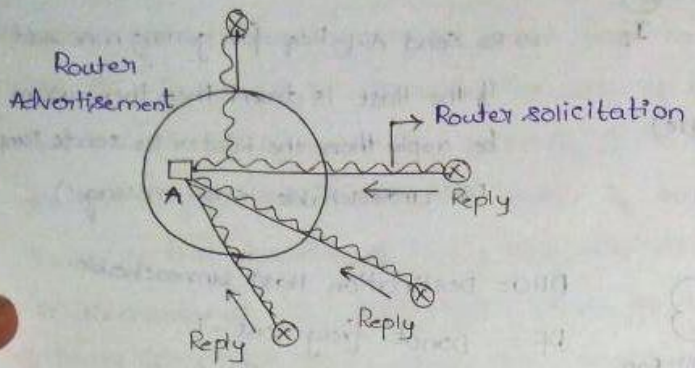
⇒ The ICMP Request and Reply messaging is used to test whether the NL of destination and the Intermediate routers are working or not

PING : PACKET INTERNET GOFER



⇒ PING uses "icmp echo Req/Reply".

⇒ It is not enough just that you get the Ip Address from DHCP when you are connected to internet, it is also important to discover the default Router for your connection and start communicating.



⇒ The station present in the Network sends a Broadcasting message to all the Routers which are connected to that particular Network then the Routers reply saying "see I am available, you can use me as your default gateway / default router. This type of Broadcasting the message is called "Router solication" [we are asking the Routers]

⇒ The other scenario is "Router Advertisement" in which a router connected to a N|w, says "see I am available you can useme as your default router.

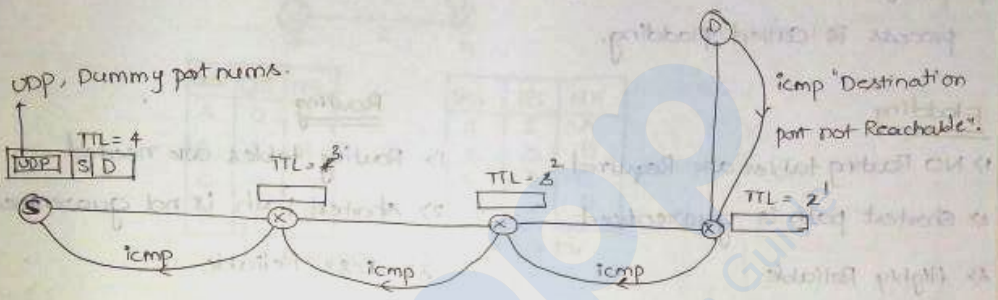
Time Stamp Req and Reply

Generally various Networking devices are placed at different parts of the world. The challenge here is synchronisation which means it is important to check whether they work at same time, inorder to that there is a special type of icmp message called "Time Stamp Req and Reply" message. This is a obselete technology and now we are using "Network time protocol"

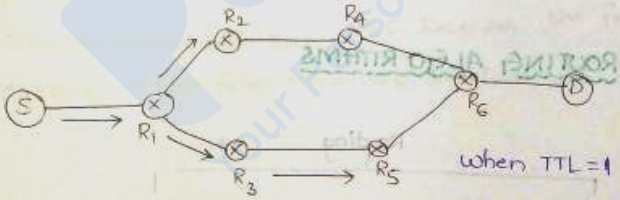
9. TRACE ROUTE APPLICATION OF ICMP

⇒ Inorder to find the intermediate Routers present in the route between source and destination the trick is, send a packet with TTL=1 inorder to get the Router present in one hop, and send a packet with TTL=2 to get the Router present at 2nd hop. when TTL=0 at the router it sends "Icmp TTL exceed" error to the source /send

⇒ when the packet reaches the destination TTL will be zero and the destination is going to accept it but it doesn't generate icmp message, it doesnot guarentee that when we dont receive a Icmp message the packet is delivered to the destination, the Icmp may be lost. so inorder to be sure whether the packet reached the destination insert a UDP packet in to the Ip packet and put dummy port numbers so that when the packet reaches destination it is going to accept it and returns an "icmp message" "Destination port un reachable."



⇒ The Trace route might not give you the actual path, but in almost majority cases we get Actual path. for, example



when TTL = 1 = R_1

when TTL = 2 = R_2

when TTL = 3 = R_5 (This is also a possibility)

when TTL = 4 = R_6

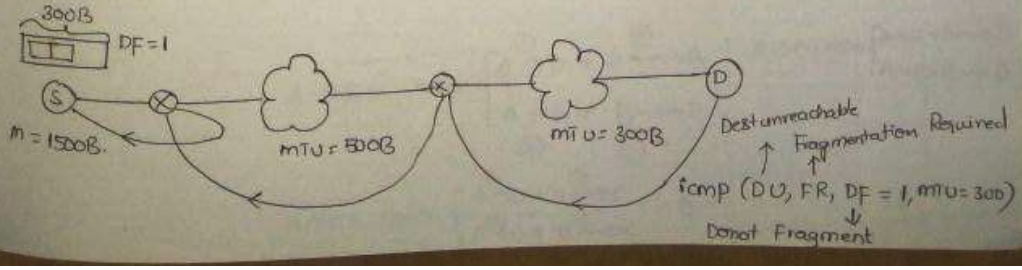
when TTL = 5 = D.

: (R1 R2 R5 R6 D)

: Not valid path.

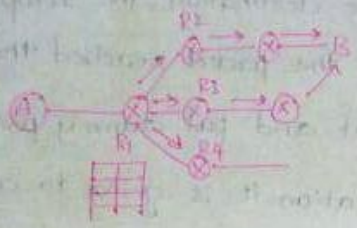
10. PMTDU APPLICATION OF ICMP

PMTDU = PATH MTU DISCOVERY



10. ROUTING

1. DIFFERENCE BETWEEN ROUTING AND FLOODDING



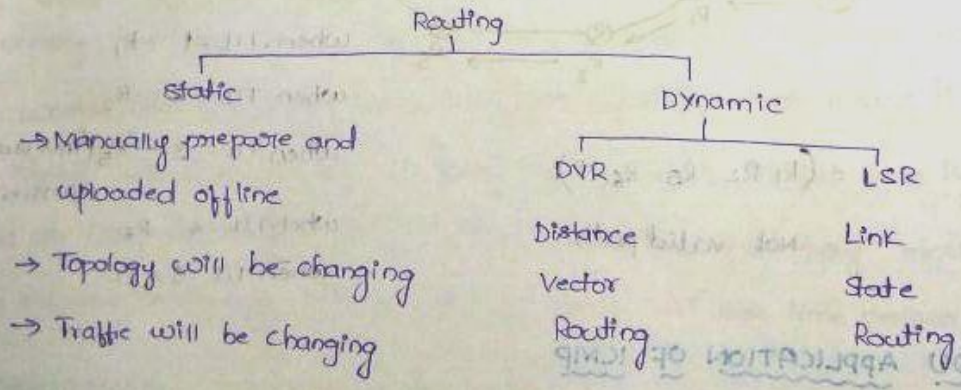
⇒ How does the Router P1 comes to know about the destination, i mean which is the path that it should send the packet so that it will reach destination.

⇒ The process of building the routing table is called "Routing"

⇒ If you dont know which way to send, send it in all possible paths, this process is called flodding.

Flodding	Routing
1> NO Routing tables are Required	1> Routing tables are needed
2> Shortest path is guarenteed.	2> Shortest path is not guarenteed
3> Highly Reliable	3> Less Reliable
Disadv	**Adv**
1> More Traffic	1> Less traffic
2> Duplicate packets are possible	2> No Duplicate packets.

2. TYPES OF ROUTING ALGORITHMS



→ Manually prepare and uploaded offline

→ Topology will be changing

→ Traffic will be changing

3. DISTANCE VECTOR ROUTING

⇒ In the DVR every node is going to know only about its Neighbours.

⇒ The Routing table at a node contain three fields Destination, Distance, Next hop

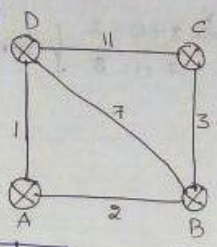
Step 1

Des	Dis	NH
A	1	A
B	7	B
C	11	C
D	0	D

↳ DV

Des	Dis	NH
A	∝	–
B	3	B
C	0	C
D	11	D

↳ DV



⇒ Building Local Routing table with knowledge of Neighbours.

Des	Dis	NH
A	0	A
B	2	B
C	∝	–
D	1	D

↳ Distance vector

Des	Dis	NH
A	2	A
B	0	B
C	3	C
D	7	D

↳ DV

Step-2:

⇒ Every Router will take their distance vectors and exchange with their Neighbours.

⇒ Now at every node the new distance vectors will be newly created.

⇒ 'A' Receives Distance Vector from B, D. ⇒ C Receives DV from B, D.

⇒ 'B' Receives Distance vector from A, C, D ⇒ D Receives DV from A, B, C

At A:

DV's from B, D

From B
2
0
3
7

From D
1
7
11
0

⇒ Using these two the new DV of Router 'A' will be

A	0	A
B	2	B
C	5	B
D	1	D

$A \to B = \min \begin{cases} A \to D + D \rightsquigarrow B & (1)+(3) \\ A \to B + B \rightsquigarrow B & (2)+(0) \end{cases}$

$A \to D = \min \begin{cases} A \to D + D \rightsquigarrow D \\ A \to B + B \rightsquigarrow D \end{cases}$

$A \to C = \min \begin{cases} A \to D + D \rightsquigarrow C & (1)+(4) \\ A \to B + B \rightsquigarrow C & (2)+(3) \end{cases} = 5$

The shortcut is

At A:

From B

2	
0	
3	
7	

add

AB: 2 add

↓ AB distance

From D

1	
7	
11	
0	add

add

AD: 1

↓ AD Distance

A	0	A
B	2	B
C	5	B
D	1	D

→ Exception

Take the min of "2"

2+0 = 2
7+1 = 8 } min = 2 from 'B' table

So NH = Next Hop = B

At B:

DVs from A, C, D

At B

From A

A	0
B	2
C	∞
D	1

add

AB: 2
BA

From C

B	∞
B	3
C	0
D	11

add

BC: 3

From D

A	1
B	7
C	11
D	0

add

BD: 7 = min{2, (3+∞), (1+7)}

= {2, ∞, 8} = {2} from A

The New Routing table At B is

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

⇒ Repeat this procedure at every node and do it for 3 times (3 Rounds) because the shortest path is of 3 edges. ⇒ no of (nodes − 1). So Repeat the rounds for (n−1) times {n = no of nodes}

The final Routing table is after everything converges is,

At A:

A	0	A
B	2	B
C	5	B
D	1	D

At B

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

At C

A	5	B
B	3	B
C	0	C
D	6	B,A

At D

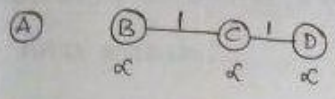
A	1	A
B	3	A
C	6	A,B
D	0	D

After the Routing table converges the edges that are unused is D→C, D→B.

4. COUNT TO INFINITY

⇒ There is a problem with DVR the problem is called count to Infinity.

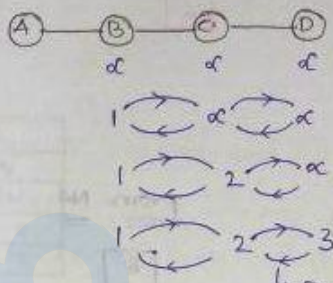
Initially



(distances to A)

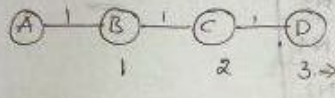
Step-1

Suppose if 'B' is connected to A



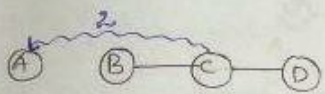
↳ Exchange of DV's

Suppose if Everything is stabilized



→ (distances to A)

Now, suddenly if the link between A, B is down, then it is a Bad news.



(Initial values before the link is broken)

∝ 2 3. (Here is the trap, we know that the Routing tables are exchanged in neighbours. Here, B says i cannot reach 'A' but 'C' says dont worry im able to reach 'A' in 2 hops").

Dist. from B→A via C:

3 4 3

5 4 5

5 6 5.

7 6 7

Distances from that node to 'A':

7 8 7

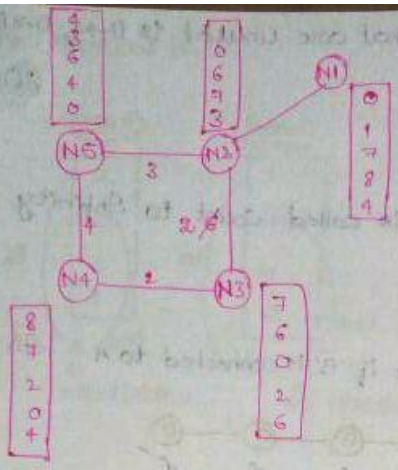
9 8 9

9 10 9

∝ ∝ ∝

⇒ This will execute till the value that you have chosen for Infinity.

Now the above explanation can be understood by the explanation given below. This question is asked in GATE



Now suppose the value of distance btn N_3 and N_2 which is '6' falls down to '2' then what is the distance vector at N3? (After 1 round of exchange).

At N3:

From N2

1
0
6
7
3

N_3N_2 : 2

From N4

8
7
2
0
4

N_3N_4 : 2

New routing table at N3 is

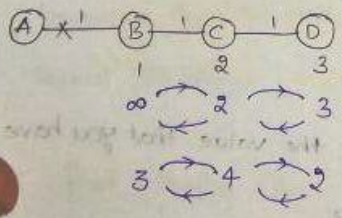
N1	3	N2
N2	2	N2
N3	0	N3
N4	2	N4
N5	5	N2

∴ Ans: 32025

5. SPLIT HORIZON

⇒ The solution to count to infinity is split horizon.

⇒ The count to infinity also creates some loops



The routing tables at A, B and C will be

A	∝	-
:	:	:

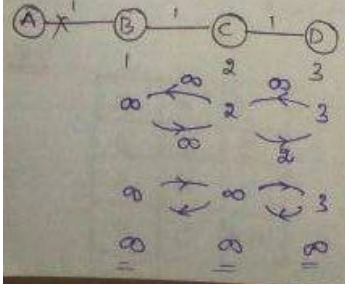
A	2	B
:	:	:

A	3	C
:	:	:

Initially

⇒ This procedure repeats until all the values are stabilized.

So this problem can be solved by sending the Next hops also and this procedure is called split horizon.



⇒ Now, At 'C', c is depending on B' inorder to goto 'A' ∴ when 'C' is already depending on 'B' to goto 'A' it should not say 'i can take 'u' in some hops", i am already depending on 'y' so please dont

⇒ By using the method of Split horizon convergence is fast, No loops are formed.

⇒ In case of DVR convergence is slow and loops will occur.

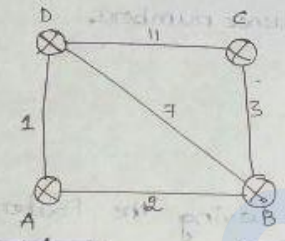
6. LINK STATE ROUTING

⇒ Every Router will create something called as Link state packets.

⇒ In the first round Every node creates a Link state packets with the help of "Hello packets".

D	
Seq	
TTL	
C	11
B	7
A	1

C	
Seq	
TTL	
D	11
B	3



A	
Seq	
TTL	
B	2
D	1

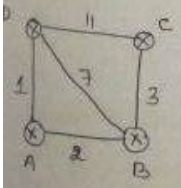
B	
Seq	
TTL	
A	2
D	7
C	3

⇒ In ROUND2 EVERY NODE IS SUPPOSED TO FLOOD THE INFO TO EVERY OTHER NODE. THERFORE WE WILL HAVE GLOBAL DATABASE AT EACH NODE.

DVR- Local Knowledge (knows only about the Neighbours)

LSR= Global Knowledge (Knows about all the nodes.

AA:



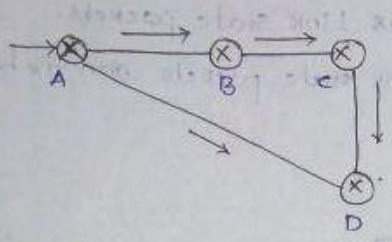
⇒ Then at 'A', 'A' will start applying single source shortest path (Dijkstra Algo) and 'A' comes to know all the shortest path to all nodes and 'A' will construct its Routing table

DES	DIS	NH
A	0	A
B	2	B
C	5	B
D	1	D

⇒ LSR converges faster compared to DVR

⇒ There are some problem in the LSR, since we are flodding, it is going to lead Heavy traffic.

⇒ Now, consider the following scenario,



⇒ Now if a packet is flodded from 'A' to all the nodes then the packet will reach 'D' in less time when compared with the other path (A-B-C-D), so the arrived packet (which has arrived Quickly) is the latest packet but depending on the time of arival we may think that the packet which came late is the latest packet and the packet arrived Quickly is the old packet. so inorder to correct that we have sequence numbers.

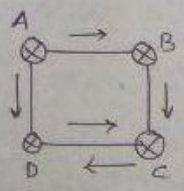
(A, 10) ✓ } Latest packet
(A, 5) × } old packet - discarded.

Therfore At A, it maintains a table having the Router name and latest sequence numbers.

Router	Latest
B	1615
C	20
D	30

(B,8) × - discarded
(B,15) ✓ - Accepted
(B,11) - discarded

Now,



⇒ The packet that is already sent to 'D' from 'A' is going to come again to the same destination via another route (A-B-C-D) so there is a problem of falling in infinite loop. This can be overcomed by having TTL field

7. DIFFERENCE BETWEEN DVR, LSR, RIP AND OSPF

DVR	LSR
1) used in 1980's	1. used in 1990's
2) Bw required is less	2. High Bandwidth
3) Local Knowledge	3. Global Knowledge
4) Bellman ford Algo	4. Dijkstra Algo
5) Less Traffic	5. High Traffic
6) periodic updates	6. periodic updates
7) Converges slowly	7. Converges fast
8) count to infinity	8) No count to Infinity
9) persistent Loops	9) Transient Looping
10) RIP	10) OSPF

RIP (ROUTING INFORMATION PROTOCOL)

⇒ Rip is the implementation of (DVR) simple to Implement

⇒ Metric = Hopcount (weights)

⇒ Infinity cannot be represented in programing language hence the value 16 is used as Infinity.

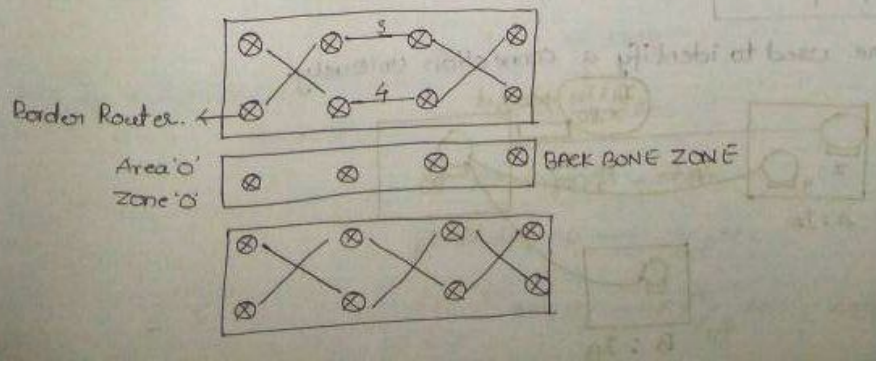
OSPF

⇒ OSPF is the Implementation of the (LSR) → computation is complex

⇒ OSPF divides all the routers into some Regions and flodding is Restricted to that Region, and there will be one router designated as "Border Router" (It will take all the flodded packet and summarize the Info).

⇒ Border Router is connected to "Area '0'" or "Zone '0'" or "Back Bone Zone".

⇒ "EIGRP" PROTOCOL INVENTED BY CISCO EIGRP = RIP + OSPF



11. TCP

1. TCP HEADER

SOURCE PORT (16)	DESTINATION PORT (16) = 4 Bytes
SEQUENCE NUMBER (32)	= 4 Bytes
ACKNOWLEDGE NUMBER (32)	= 4 Bytes
HEADER LENGTH (4 bits) / RESERVED (6 bits) / URG / ACK / PSH / RST / SYN / FIN	WINDOW SIZE (OR) ADV WINDOW (16) = 4 Bytes
CHECK SUM (16)	(16) URGENT POINTER = 4 Bytes
OPTIONS (0-40) Bytes	= 40 Bytes
DATA	

∴ Max Header Length in Tcp = 60 Bytes

∵ Min. header length in Tcp = 20 Bytes

2. SOURCE PORT, DESTINATION PORT AND SOCKET

⇒ Tcp is End-End protocol because of having port numbers.

⇒ port numbers are used for multiplexing and Demultiplexing.

⇒ SP DP

16 16

[0 to $(2^{16}-1)$] nos can be represented

$= (0, 65,535)$

Http : 80
FTP : 21
Telnet : 23
Smtp : 25

0 – 1023 } well known services port numbers

1024 – 49,151 } Reserved

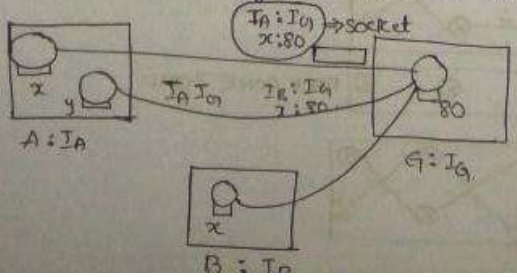
49,152 – 65,535 } General public

⇒ open your browser and type "Net stat" to see all the connections, and port numbers (source port number, Destination port number)

⇒ Tcp is connection oriented protocol (Resources are Reserved).

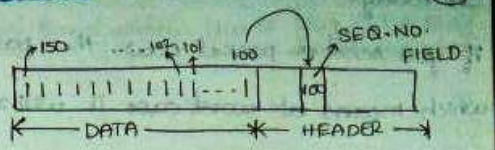
⇒ Socket = Ip + portnum = 32 + 16 = 48 bits

⇒ sockets are used to identify a connection uniquely.



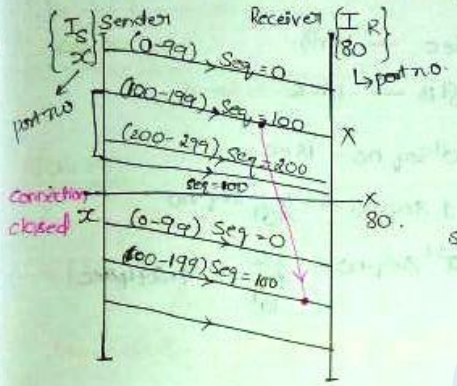
3. SEQ. NO, ACK NO., RANDOM INITIAL SEQ. NO

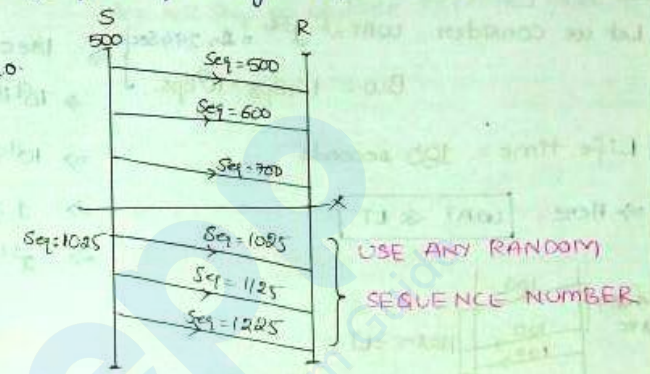
⇒ Tcp is a Byte stream protocol.
⇒ Ip is a packet Stream protocol
⇒ HDLC (DLL) is a Bit Stream protocol.



⇒ In Tcp the data Bits are numbered. the seq.no field contains the value of starting data bit

⇒ Ack field is used when the sender sends the Tcp packet and is waiting for Acknowledgement. Ack field contains the value of the data bit that is expected next (seq no of the Byte expected to come next)





4. WRAP AROUND TIME AND PROBLEMS ON WRAP AROUND TIME

⇒ wrap around is nothing but using up all the seqnos present and reusing the seq.no's that are already been used is called wraparound. From this we get "wrap Around time".

WAT

⇒ Wrap Around Time (WAT) depends on the Bandwidth.

⇒ Seq. No's field contain 32 bits ⇒ 2^{32} seq.nos are possible

⇒ Now, Letus say Bandwidth = 1 MBps

WAT of 'n' seqnos = $\frac{n}{\text{Bandwidth}}$

⇒ 1 sec — 1 MB

⇒ 10^6 B — 1 sec

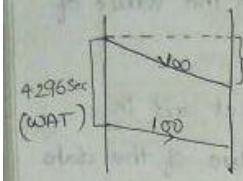
⇒ 10^6 seq.nos — 1 sec

⇒ 1 seq.no — $1/10^6$ sec

⇒ 2^{32} seq.no — $2^{32}/10^6$ sec = 4296.967296 sec

In today's internet there is a concept called Life time which means if you send a packet now the packet will be alive fore sometime (3 min) which means at worst case it will reach after 3minutes after you send it.

LT = 3min = 180sec



{practical value} ⇒ As long as WAT > LT there is no problem

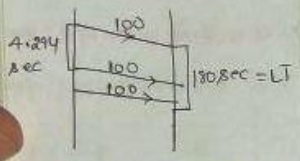
II.

Let us consider WAT = 2^{32} = 4,294sec
Bw = 1GBps = 10^9 Bps.

⇒ 1sec — 1GB
⇒ 10^9 B — 1sec
⇒ 10^9 seq.no — 1sec
⇒ 1 seq.no — $1/10^9$ sec
⇒ 2^{32} seq.no — $\frac{2^{32}}{10^9}$ = 4.294sec

Life time = 180 seconds

⇒ Here WAT ≪ LT



Now, to solve the above problem, the possible solution is Decreasing the Bandwidth, But this is not possible because we are intrested in having high Bandwidth (The speed at which the data gets transfered) so the other possible solution is increasing the ↑ bits in the sequence no field.

WAT > LT ⇒ No problem

WAT < LT ⇒ problem ⇒ so calculate how many seq.no's are transmitted in 180sec

⇒ 1 sec — 1GB
⇒ 1sec — 1G seq.no's
⇒ 180 sec — 180 × (1G seq.no's)

∴ The min seq no's required to avoid wrap Around within Life time
= 180 × 1G seq.no

∴ The No.of bits in seq.no field is = $\lceil \log_2(180 \times 1G) \rceil$ = 42

∴ The no.of additional bits needed to avoid above problem = 42 - 32 = 10

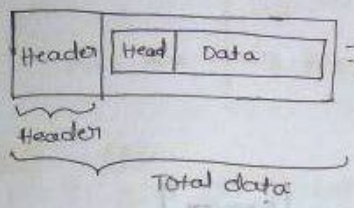
[These 10 bits are used in options and called as Time stamp]

⇒ The no of bits in the sequence no field such that there wont be any problem with WAT and LT, given at Bandwidth Bw is $\lceil \log_2(LT * Bw) \rceil$ (40)

5. HEADER LENGTH AND CALUCULATION OF ACKNOWLEDGENT NUMBERS

⇒ Header length field is 4 bits, but the min size of the header is 20B

⇒ Scaling factor of $\frac{60}{15} = 4$ is used. $\left\{\frac{\text{max size of header}}{\text{largest no by 4 bits}} = \text{scalufactor}\right\}$



SEQ NO: Seq no of 1 Byte

ACK NO: Seq no of Byte expected next.

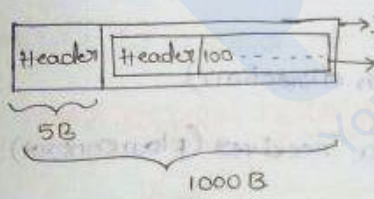
PROBLEM

TL = 1000 B, HLF = 5 } for Ip Datagram

HLF = 5, Seq no of Tcp segment = 100B } for Tcp Segment

then what is the ack number of the next Byte expected?

SOL: Tcp segment will be in the Ip Datagram

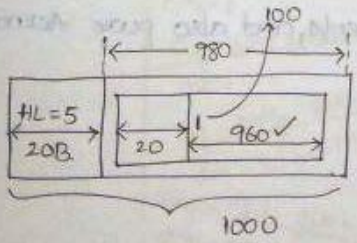


⇒ Now, the data part in the Ip datagram = 1000 − (5 * 4)
= 980B

⇒ 980B = Header + Data (in Tcp)

⇒ 980 − (5+4) = Data

⇒ 960 B = Data ⇒ Ack = (960 + 100) + 1
= 1059



⇒ last Byte seq no = 100 + 960 − 1

⇒ 1059

⇒ Next Byte seq no = 1059 + 1
= 1060

6. TCP CONNECTION ESTABLISHMENT

⇒ Flags are nothing but 1 Bit information, Tcp is connection oriented.

SYN (SYNCHRONIZATION FLAG)

ACK (ACKNOWLEDGEMENT FLAG)

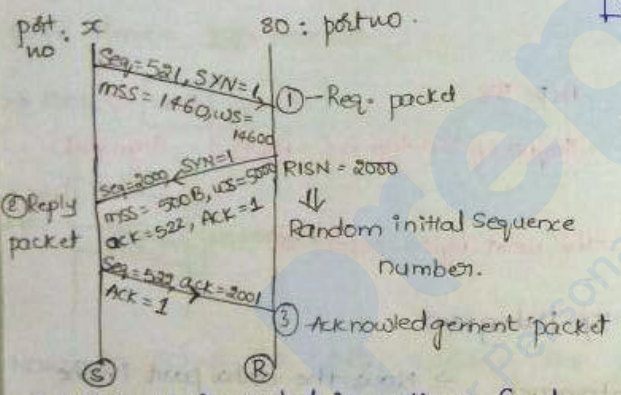
NOW, Tcp is connection oriented protocol and it has 3 phases,

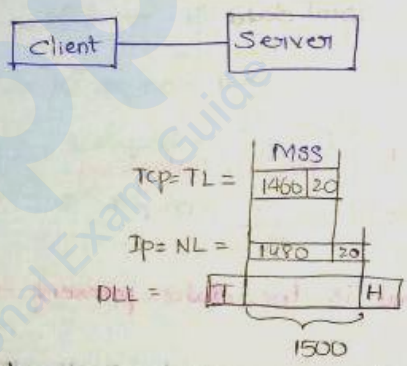
1) Connection Establishment

2) Data Transfer

3) Connection Termination.

CONNECTION ESTABLISHMENT





⇒ MSS is included in options field.

⇒ Tcp is full duplex (Data transfers in both direction)

⇒ window size : capacity of the host as a receiver (flow control)

⇒ "SYN" PACKET WILL EATUP 1 SEQ NUMBER.

⇒ Tcp uses "piggy Backing" Acknowledgements, and also pure Acknowledgement

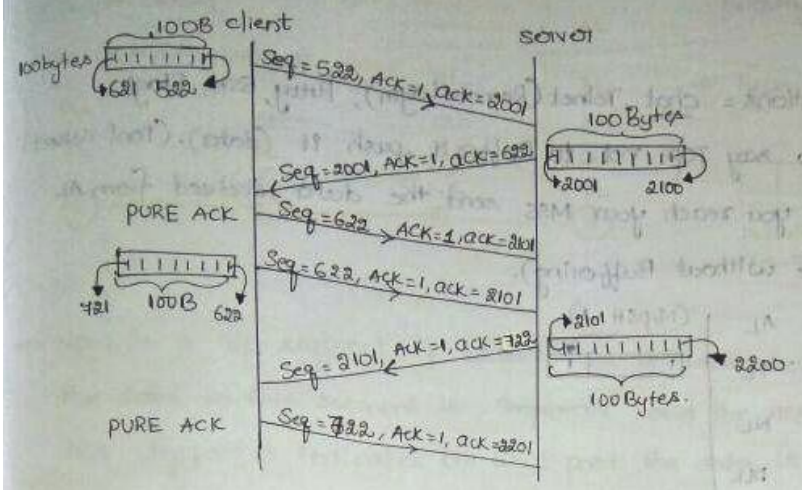
⇒ "ACK" 1 TAKE '0' NO OF SEQ NOs

⇒ "FIN=1" TAKE '1' SEQ NOs

⇒ 1 DATA BYTE TAKES '1' SEQ NOs.

⇒ THIS METHOD OF CONNECTION ESTABLISHMENT IS CALLED "3-WAY HAND SHAKE" PRINCIPLE.

7. TCP DATA TRANSFER AFTER CONNECTION ESTABLISHMENT



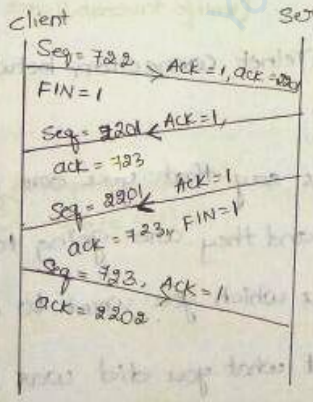
SYN	ACK	
1	0	⇒ 1st packet = 1st Segment = Request segment
1	1	⇒ 2nd Segment = Reply Segment
0	1	⇒ pure Acknowledgement (ACK is present in the header)
0	0	⇒ (Not possible)

⇒ sender is not going to send Acknowledgements immediately, he is going to wait for some time

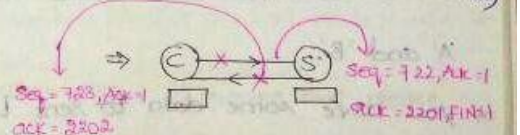
⇒ The Reciver also waits for some time before transmiting the data.

⇒ If u dont have data to send immediately we should not wait because there will be a timer at the other end. so we should send PURE ACK.

8. TCP CONNECTION TERMINATION



⇒ FIN = FINISH FLAG (used to terminate the Connection)



⇒ After client sending FIN packet the events that are possible and not possible are

⇒ Data C→S ✗
⇒ Data S→C ✓
⇒ ACK C→S ✓ (pure ACK)
⇒ ACK S→C (piggy Backed)

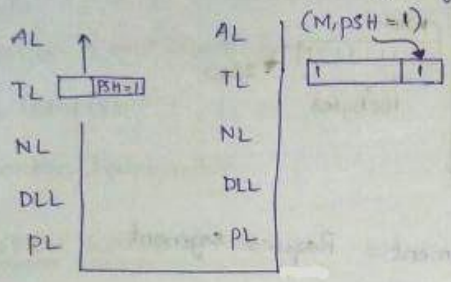
SYN: Synchronising Seq numbers
ACK: Indicate whether Ack num field is valid/not
FIN: Request for connection termination.

9. PSH FLAG

PSH = PUSH FLAG

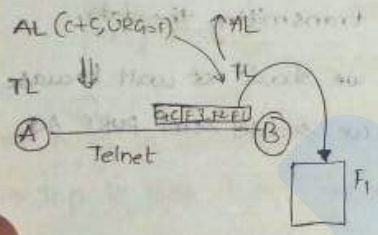
⇒ Interactive Applications = chat, Telnet (Remote Login), Putty, SSH, Rlogin.

⇒ PUSH is used to say TCP not to Buffer it, push it (data). (Don't collect the data until you reach your MSS send the data received from AL at the same time without Buffering).

⇒ 

10. URG FLAG AND URGENT POINTER

URG = URGENT FLAG



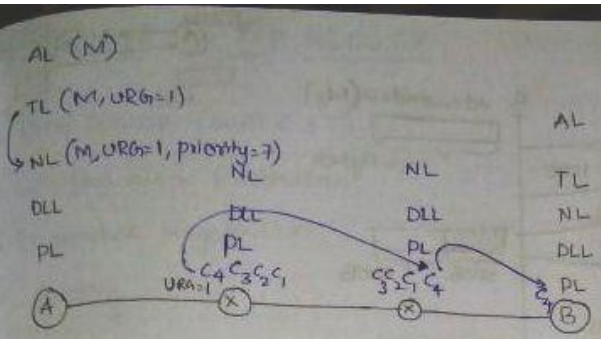
⇒ Let us assume that there is a Telnet connection between the two hosts 'A' and 'B'.

⇒ you have some data to sent Let us say that you are sending the data in the form of files F1 F2 F3 and they are going to sit in Buffer at host 'B'. All those are existing files which you want to modify

⇒ Now, suddenly you realised that what you did was wrong and you want to execute the command (Ctrl+c)(c+c) before all the files. [Ctrl+c is used for stopping]

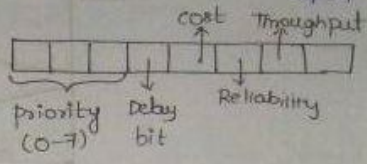
⇒ so Urgent flag is used here, which means you have to say that last command is Urgent when compared to other things.

⇒ so for "(Ctrl+c) send URG=1"

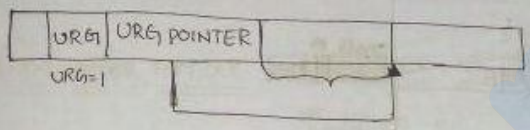


⇒ For urgent packet the priority is set to seven.

⇒ Type of service (8bits) in IPv4

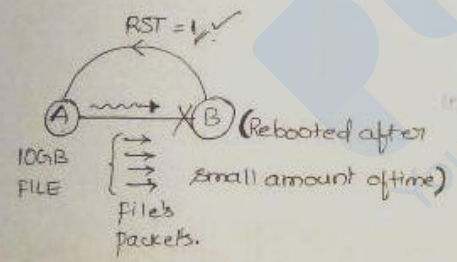


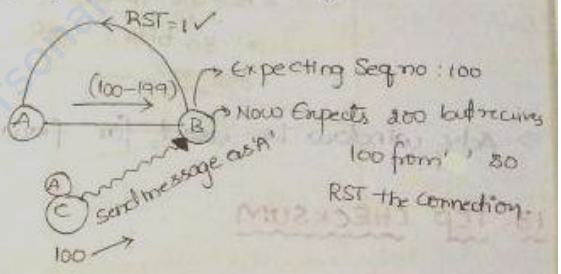
⇒ Now, In a Tcp segment if URG=1 then it indicates that some part of the data in this segment is important and the nextone is urg pointer this urg pointer indicates till what part the data is urgent.

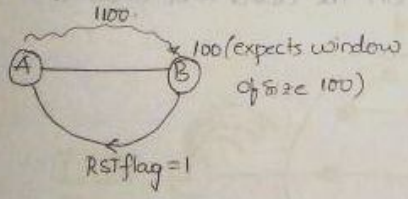


11. RST FLAG

⇒ RESET Flag is used when there is a wrong in the connection. If you see anything unexpected then you are going to use Reset flag.



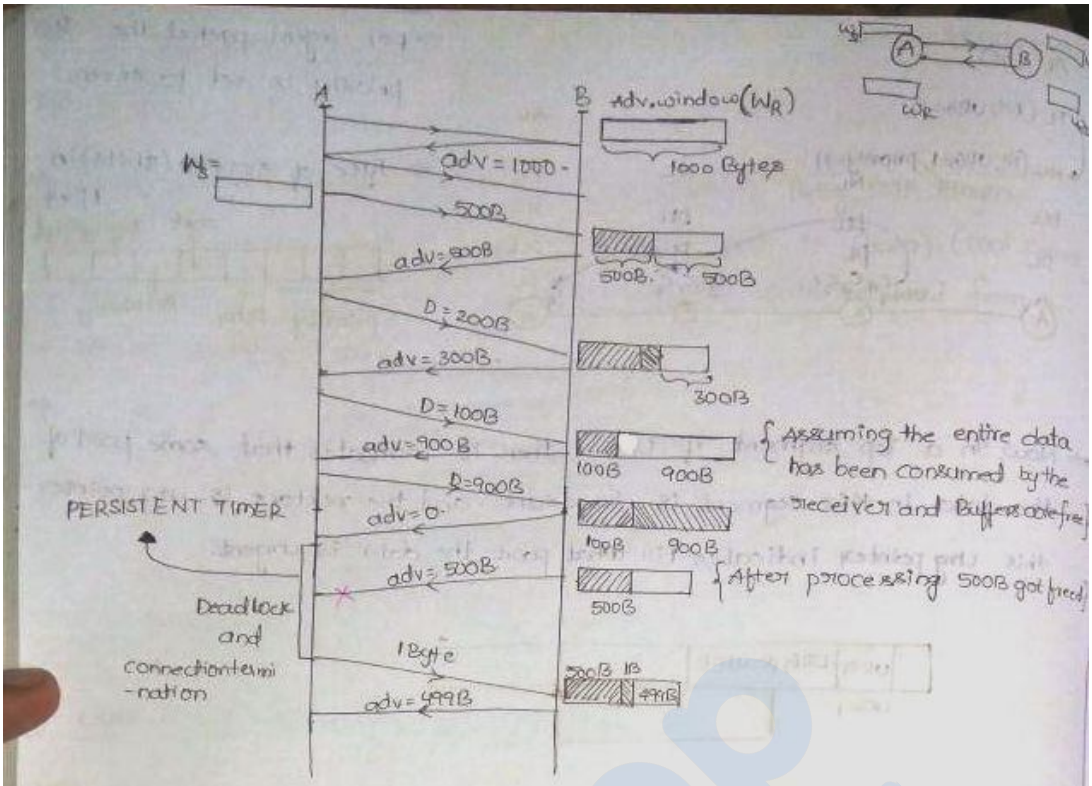




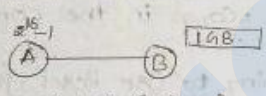
⇒ when RST=1 it means that, please terminate the connection i'm not expecting anything from you.

12. TCP FLOW CONTROL USING ADVERTISEMENT WINDOW

⇒ Flow control, a sender should never send what a receiver can't receive. For that Reason we are going to use window size.



⇒ The window size is 16bits ⇒ Max no = $2^{16}-1$ = 65,535



= 16 + 14 (options)

= 30 bits = 2^{30} = 1GB.

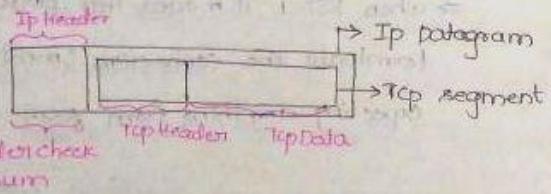
⇒ Adv window is used for flow control.

13. TCP CHECKSUM

⇒ Tcp Header has 16-bit checksum

⇒ In Tcp the checksum is calculated both on the data and Header.

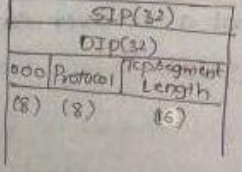
⇒



⇒ Checksum is calculated on TcpHeader, Tcpdata, IpHeader {Calculate check sum only on the fields in pseudo Ip Header} Because other fields will change when the packet is received by Receiver

In Ipv4 Header

PIpH: Pseudo Ip Header.



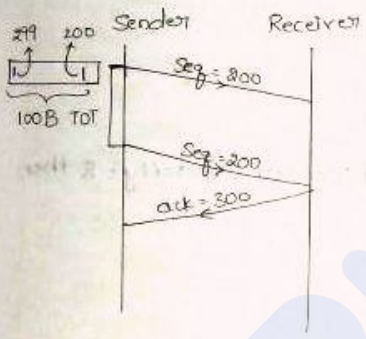
14. OPTIONS IN TCP HEADER

1) Time stamp (WAT < LT)
2) Window size Extension
3) Parameter Negotiation
4) Padding

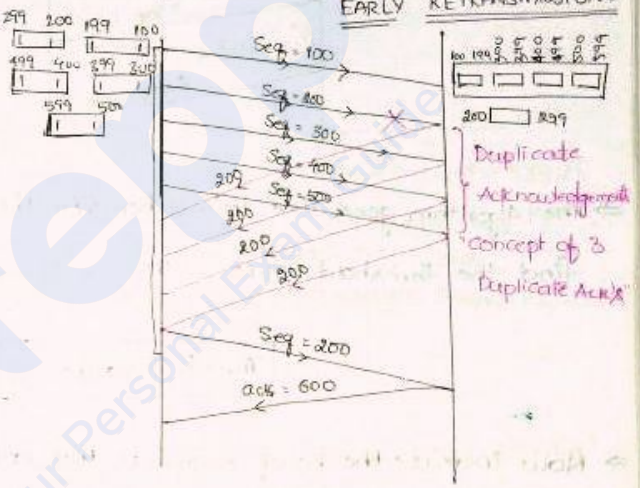
15. RETRANSMISSIONS IN TCP

⇒ TCP uses SR + GBN
⇒ $w_s = w_R$ | ⇒ Acknowledgements are cumulative
⇒ out of order packets are possible | ⇒ 25% GBN
• 75% SR

RETRANSMISSION AFTER TIMEOUT

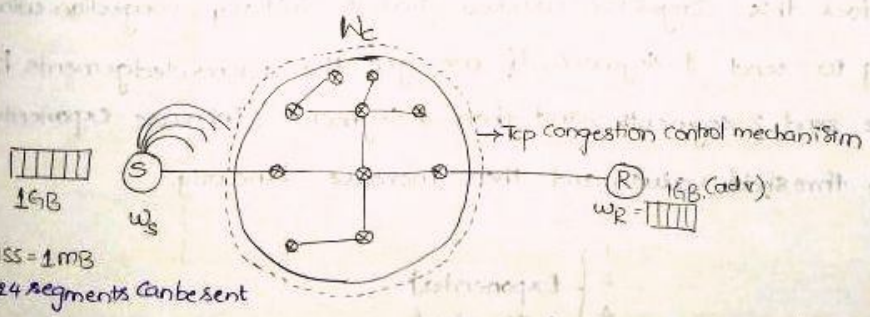


RT AFTER 3 DUPLICATE ACKNOWLEDGEMENTS



EARLY RETRANSMISSIONS

16. INTRODUCTION TO TCP CONGESTION CONTROL



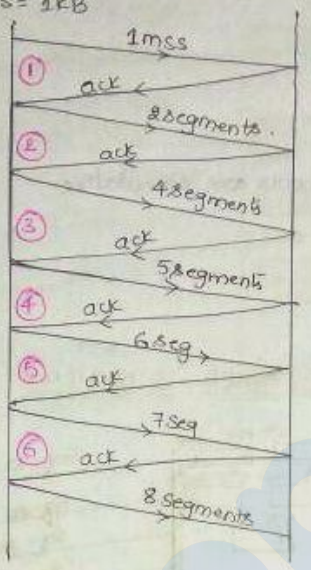
⇒ Assume that the Receiver has said that sender can send 1GB of data in the Advertising window. ⇒ w_s = 1GB, w_R = 1GB

⇒ The problem here is even though the Receiver can hold 1GB of data the underlying Network cannot hold 1024 packets.

⇒ So the sender should not dump the data on the Network without finding the capacity of the Network(W_c).

⇒ A sender should always send $\min(W_c, W_R)$ data to the Network & Receiver so we should stop the traffic at sender side.

Adv win = 8KB
MSS = 1KB



Adv win = 8KB
MSS = 1KB.
W_R = 8 segments
W_c = 1 Segment
$W_s = \min(1, 8)$
$= \min(W_c, W_R)$
$W_s = 1$

⇒ Even though you can send 8 segments at a time don't do it because the underlying N/w may not be in a position to handle the segments. So send segments one by one in some order.

⇒ The Algorithm goes like this whenever the Receiver capacity = 8 then find the threshold value = $W_R/2 = 8/2 = 4$

Threshold value = 4

⇒ Now increase the no of segments that are being sent, exponentially upto the threshold value and after that send the packets Linearly.

⇒ How does this congestion window grow is initially congestion window is going to send 1 segment, if we get the acknowledgements back then we send 2 segments, and then 4 segments (increase exponentially till the threshold value) and then increase Linearly.

1, 2, 4 → Exponential → Threshold
5, 6, 7, 8 → Linear Growth of No of Transmissions.

V. Imp

(44)

1) AFTER HOW MANY RTT's ARE WE GOING TO REACH THE MAX SENDER WINDOW SIZE (8)?

Ans 6 Roundtrip times.

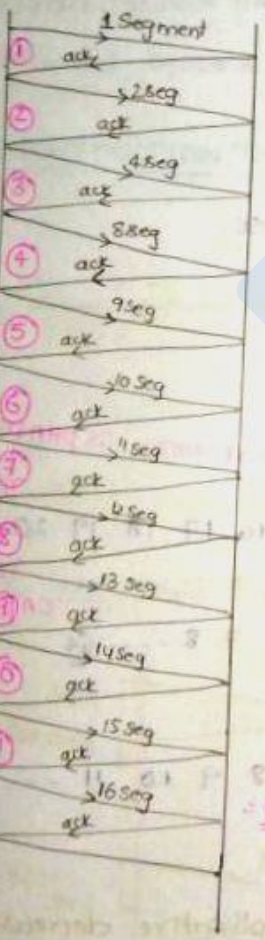
1 | 2 | 4 | 5 | 6 | 7 | 8
1RTT 2RTT 3RTT 4RTT 5RTT 6RTT

2) Adv Win = 16KB
MSS = 1KB
W_R = 16 Segments ⇒ Threshold = $16/2 = 8$
W_c = 1 Segment

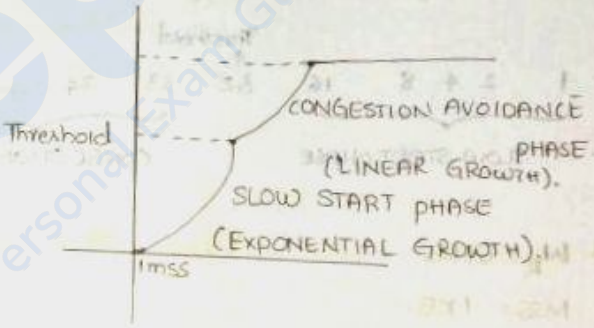
} AFTER HOW MANY RTT's WILL WE REACH TOTAL CAPACITY?

⇒ 1 | 2 | 4 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16
1RTT 2RTT 3RTT 4RTT 5RTT 6RTT 7RTT 8RTT 9RTT 10RTT 11RTT.

⇒ 11 RTT's



∴ 11 RTT's



SS PHASE = "EXPONENTIAL GROWTH".
CA PHASE = "LINEAR GROWTH".

17. TCP CONGESTION CONTROL ALGORITHM WITH EXAMPLE

⇒ The congestion control Algorithm has three steps they are

1. Slow start phase
2. Congestion Avoidance phase
3. Congestion Detection phase

⇒ There are 3 ways to detect congestion they are

1. Time out (Severe Congestion) ⇒ (SS PHASE)
2. Three Duplicate Acknowledgements (Mild congestion) = (CA PHASE)
3. ICMP Messaging (source Quench). (Most of the Implementations dont consider it)

1) W_R = 64 KB

MSS = 1 KB

W_R = 64 Segments ⇒ Th = $64/2$ = 32 MSS

⇒ 1 2 4 8 16 | 32 (Threshold) | 33 34 ----- 64

SLOW START PHASE — CONGESTION AVOIDANCE PHASE.

2) W_R = 64 KB

MSS = 1 KB.

W_R = 64 MSS ⇒ Th = 32 mss.

⇒ 1 2 4 8 16 32 33 34 | 1 2 4 8 16 17 18 19 20

(Th = 17 = (34/2)+ AND ALGO ENTERS SS PHASE; Th = 20/2 = 11; 34 TO; 20 3 DupAck 'CA PHASE')

10 11 12 | 1 2 4 6 7 8 | 4 5 6 7 8 ---- 64

(Th = 12/2 = 6; TO; Th = 8/2 = 4; 3DA)

64 | 1 2 4 8 16 32 | 16 17 | 1 2 4 8 9 10 11 ------

(Th = 32; TO; Th = 16; 3DA; Th = 8; TO)

GATE

Consider an instance of Tcp Additive Increase Multiplicative decrease (AIMD) Algorithm where the window size at the start of ssphase is '2' MSS and the Threshold at the start of 1st transmission is '8' mss. Assume timeout occurs during 5th transmission, find the Congestion window size at the end of 10th transmission?

(A) 8 Mss (B) 14 mss (C) 7 mss (D) 12 mss

Sol

⇒ Initial value = 2

⇒ Threshold = 8 ⇒ 2 (1st Trans) 4 (2nd Trans) 8 (3rd Trans) 9 (4 Trans) 10 (5th Trans) Timeout, Th=5 → 2 (6th Tr) 4 (7th Trans) 8 (8th Trans) 9 (9th Trans) 6 (10th Trans) 7

18. TCP TIMER MANAGEMENT

1) Time-wait timer ⇒ Dont close the connection immediately wait for 2*LT

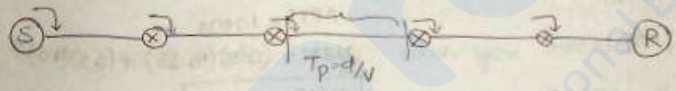
2) Keep-Alive timer ⇒ close All the idle connections

3) persistent timer ⇒ For finding the capacity of Receiver

4) Acknowledgement timer ⇒ cumulative Acknowledgement and Piggy Backing

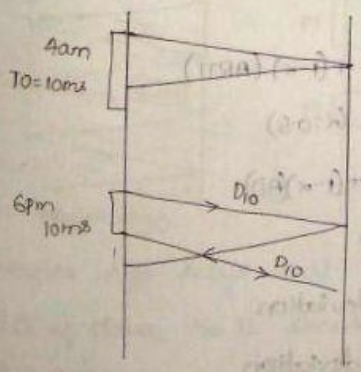
5) Time-out timer ⇒

19. INTRODUCTION TO TIMEOUT TIMER

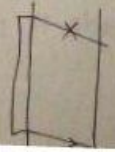


RTT = 2*Tp
TO = 2*RTT
= 4*Tp

⇒ At tcp static time out timeouts cannot be used



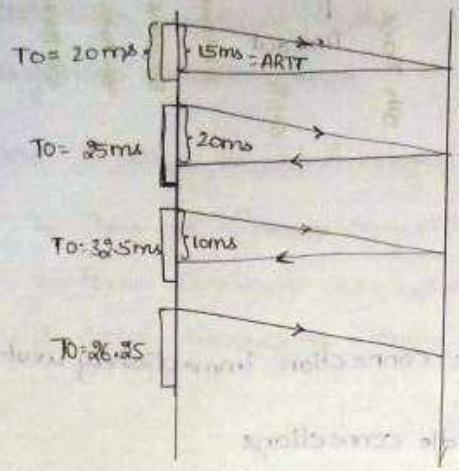
⇒ Initially there is no congestion but after that because of your timeout timer settings it might lead to congestion so have a large timeout timer, but the disadvantage of having Large TOT is if a packet is lost you have to wait for a long time before Retransmitting it.



20. BASIC ALGORITHM FOR TOT COMPUTATION

⇒ The first Algorithm used to set timeout timer dynamically at Tcp is

"BASIC ALGORITHM"



IRTT= 10ms. (Guessing)

TO= 2 * IRTT = 20ms

ARTT= 15ms.

NRTT= α IRTT + $(1-\alpha)$ ARTT

α= Smoothening Factor

$0 \le \alpha \le 1$ α=0.5

NRTT = 0.5 ×10 + (0.5) 15

NRTT= 12.5ms ⇒TO=25ms

⇒This Algorithm is Dynamic Nature

⇒ The disadvantage is TO= (2) * RTT

No Logic why we are using "2"

IRTT= INITIAL ROUNDTRIP TIME

ARTT= ACTUAL ROUNDTRIPTIME

NRTT= NEXT ROUNDTRIP TIME

∴ For Next packet IRTT=12.5ms

TO= 2 * IRTT = 25ms.

ARTT= 20ms

NRTT= (0.5)(12.5)+(0.5)(20)

NRTT = 16.25ms

For Next packet IRTT=16.25

TO= 32.5ms

ARTT= 10ms

NRTT= (0.5)(16.25) + (0.5)(10)

NRTT = 13.125

21. JACOBSON'S ALGORITHM FOR TIMEOUT COMPUTATION AND KARN'S MODIFICATION

⇒ The Jacobsons Algorithm is used to set the time out timer.

①

IRTT= 10ms

ID= 5ms

TO= 4 * ID + IRTT

= 4 * 5 + 10

= 30ms

ARTT= 20ms

AD= 10ms (|IRTT − ARTT|)

NRTT= α(IRTT) + $(1-\alpha)$ (ARTT)

= 15ms (α=0.5)

ND= α(ID) + $(1-\alpha)$(AD)

= 7.5ms

ID= Initial deviation

AD= deviation.

For the 2nd segment that you are sending

IRTT = 15ms

ID = 7.5ms

TO = 4 * ID + IRTT

= 4 * 7.5 + 15

TO = 45ms

ARTT = 30ms

AD = (30ms - 15ms) = 15ms

NRTT = α(IRTT) + (1-α)(ARTT) {α = 0.5}

= (0.5)(15) + (0.5)(30)

= 7.5 + 15 = 22.5ms

ND = α(ID) + (1-α)(AD)

= (0.5)(7.5) + (0.5)(15)

= 11.25 ms

For the 3rd segment that you are sending

IRTT = 22.5 ms

ID = 11.25ms

TO = 4 * ID + IRTT

= 4 * 11.25 + 22.5

= 67.5

ARTT = 10 ms

AD = 12.5

NRTT = α(IRTT) + (1-α)(ARTT) {α = 0.5}

= (0.5)(22.5) + (0.5)(10)

= 16.25

ND = α(ID) + (1-α)(AD)

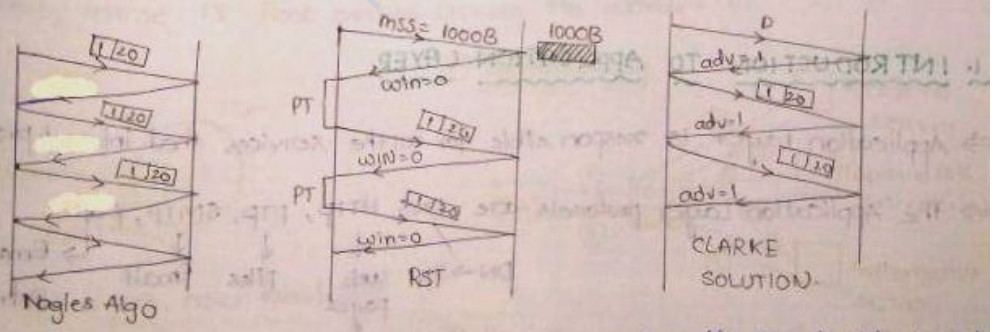
= 0.5(11.25) + (0.5)(12.5)

= 11.875

TO = 4 * D + RTT = 63.75

⇒ Karns modification says when you receive the Acknowledgement for the packet after the time out, keep doubling the TOT for the next packets.

21. SILLY WINDOW SYNDROME



⇒ Nagles Algo says that if sender is very slow the TL should not send 1B of data, the TL should wait for 1RTT and collect the data that has come in 1RTT time and send that data in one Segment.

⇒ Clarke Algo says that Receiver should not advertise 1 it should wait until it gets a size of 1mss or half of the Buffer.

12. UDP

1. NEED FOR UDP

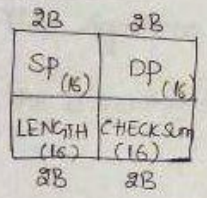
⇒ Tcp is going to be disadvantageous fore some applications they are

1> If Application needs 1 Request and 1 Reply Ex: DNS, BOOTP, DHCP, NTP (Network Time protocol), NNP (Network News protocol), "Quote of day" protocol, TFTP, RIP, OSPF

2> Broadcast or Multicast

3> fastness rather than Reliability (Multimedia Applications) (Online game)

UDP HEADER



⇒ Check sum is calculated on UDP Header + UDP Data + pseudo Header from Ip

⇒ UDP should communicate the options to the Ip Datagram, the various options are 1> Trace route 2> Record Route 3> Time Stamp

⇒ Additional Responsibilities of UDP is, if we get any icmp error packets UDP should inform Application Layer

14. APPLICATION PROTOCOLS

1. INTRODUCTION TO APPLICATION LAYER

⇒ Application Layer is responsible for all the services that internet provide

⇒ The Application Layer protocols are DNS, HTTP, FTP, SMTP, POP

DNS → Ip
HTTP → web pages
FTP → files
SMTP → Email
POP → Email Retreiving

DNS

DNS PORT NO: 53

⇒ DNS = DOMAIN NAME SERVICE (DNS uses UDP at Transport Layer).

⇒ To convert Domain name to Ip Address we use DNS.

⇒ The various domains are 1> Generic Domains (.com, .edu, .mil, .org, .net)

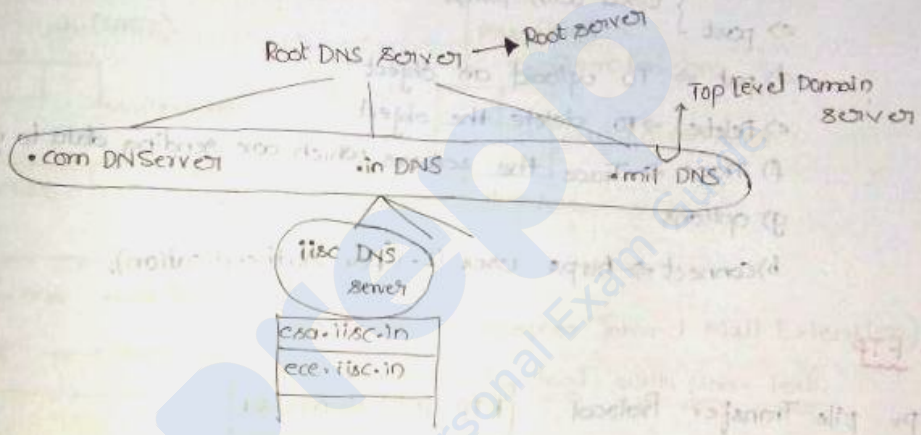
2> country Domain (.in, .us, .uk...)

3> Inverse Domain (Given Ip Address → find Domain Name)

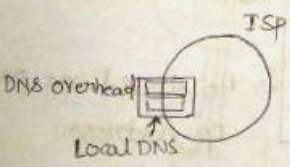
⇒ To find the ip Address of a website type "NSlookup www.google.com"

⇒ DNS is also used for Load Balancing.

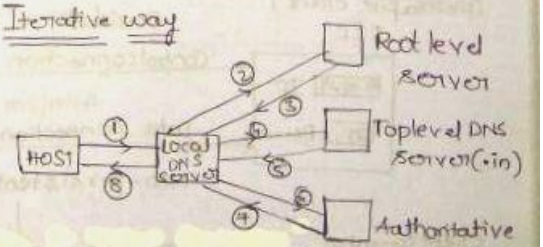
⇒ DNS Database is organised in this way (Distributed Database)



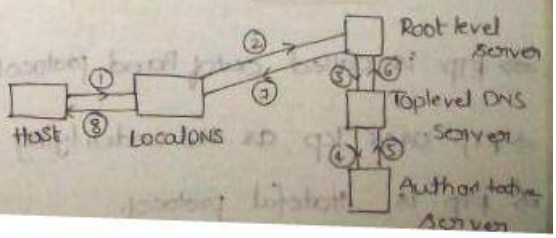
⇒ what if the Root DNS server fails, so IETF (INTERNET ENGINES TASK FORCE) mange 13 Root serves across the world.



① Iterative way



② Recursive way



⇒ Now, let us see how local DNS is going to contact Root DNS.

1) Iterative way

2) Recursive way

3. HTTP

HTTP => PORT NO = 80

⇒ Hyper text transfer protocol (For getting web pages).

⇒ Http always need Reliability.

⇒ Http uses "Tcp" at Transport layer

⇒ Http is Inband protocol (Both data and commands goin one connection)

⇒ Http is stateless protocol.

⇒ The two popular versions of Http are: Http1.0 (Non-persistent connection)
Http1.1 (persistent connection)

⇒ The popular methods that are used by Http are

a) Head ⇒ Get the Header of webpage (metadata).

b) Get } used with forms
c) post }

d) put ⇒ To upload an object

e) Delete ⇒ To delete the object

f) Trace ⇒ Trace the servers which are sending data to you.

g) options

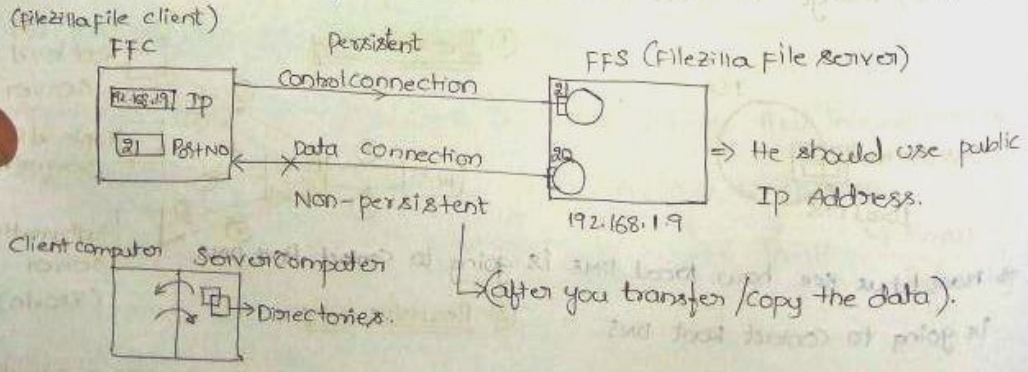
h) Connect ⇒ https uses it. (For Authentication).

4. FTP

FTP = File Transfer Protocol. FTP PORT Num = 21

⇒ Tecfia, filezilla are the two most popular FTP's.

⇒ FTP is used to transfer files.



⇒ FTP is called out of Band protocol and requires Reliability.

⇒ FTP uses Tcp as the underlying protocol

⇒ FTP is stateful protocol.

5. SMTP AND POP

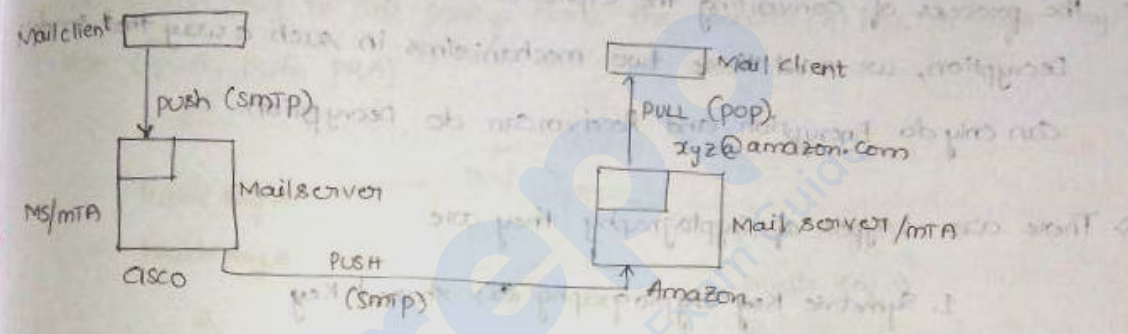
SMTP: Simple Mail Transfer Protocol, POP: post office protocol.

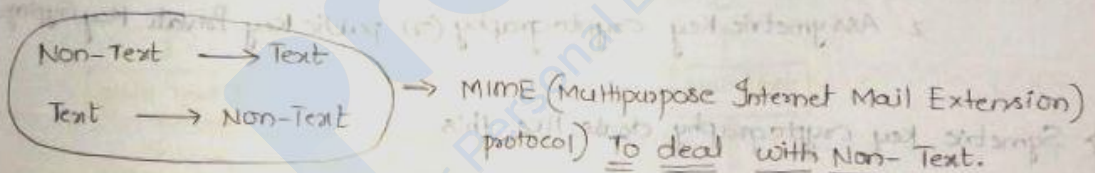
⇒ E-mails are transfered using SMTP

⇒ we cannot transfer the files using FTP because to transfer the files using FTP both server and client should be online.

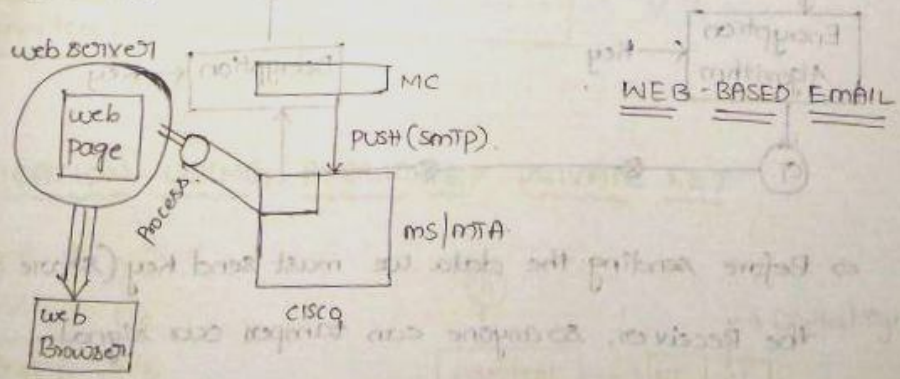
⇒ Gmail is the web Based mail

⇒ MTA: Mail Transfer Agent





What Gmail does is,



⇒ SMTP and pop are In band protocol.

19. NETWORK SECURITY

1. INTRODUCTION TO CRYPTOGRAPHY

⇒ The main aim of cryptography is to convert a message from plain text into some unreadable form before you send it over the wires.

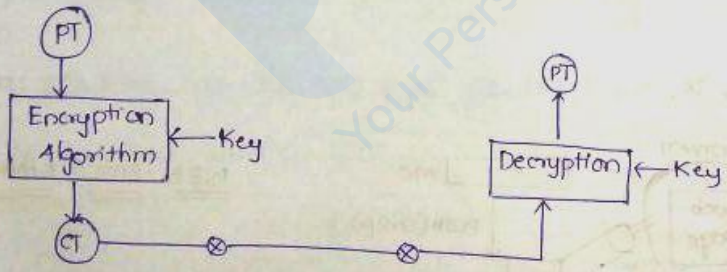
⇒ The process of converting the plain text to cipher text is called Encryption.

⇒ The process of converting the cipher text to plain text is called Decryption, we need these two mechanisims in such a way that sender can only do Encryption and Receiver can do Decryption.

⇒ There are 2 types of Cryptography they are

1. Symetric Key Cryptography (or) shared Key
2. Assymetric Key cryptography (or) public Key Privati Key Cryptography

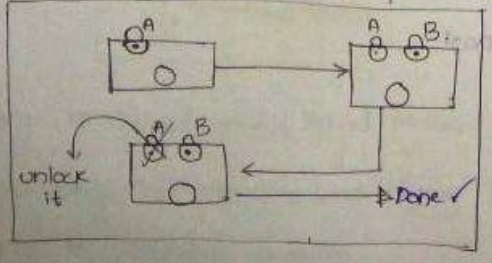
⇒ Symetric key cryptography deals like this



KEY- EXCHANGE PROBLEM.

⇒ Before sending the data we must send Key (share the Key with the Receiver, so anyone can tamper our signal.

⇒ The solution to the above problem comes in form of public, private keys



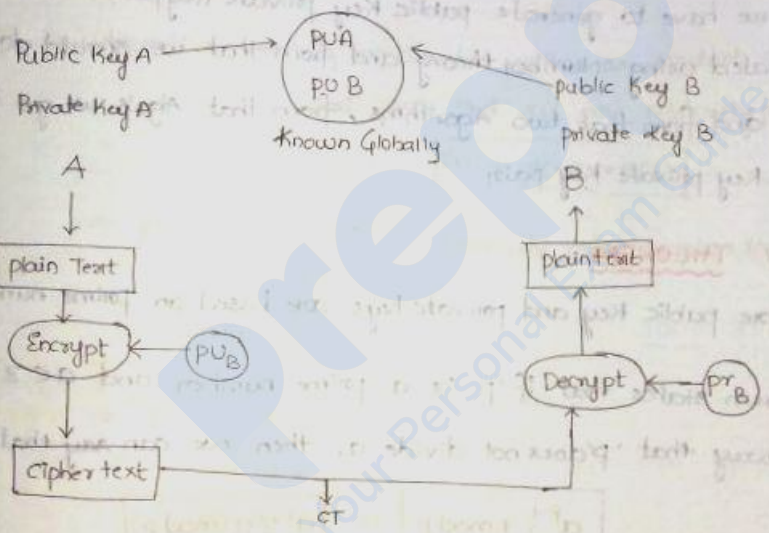
public, private Keys works Analogus this way.

2. ENCRYPTION USING PUBLIC KEY PRIVATE KEY

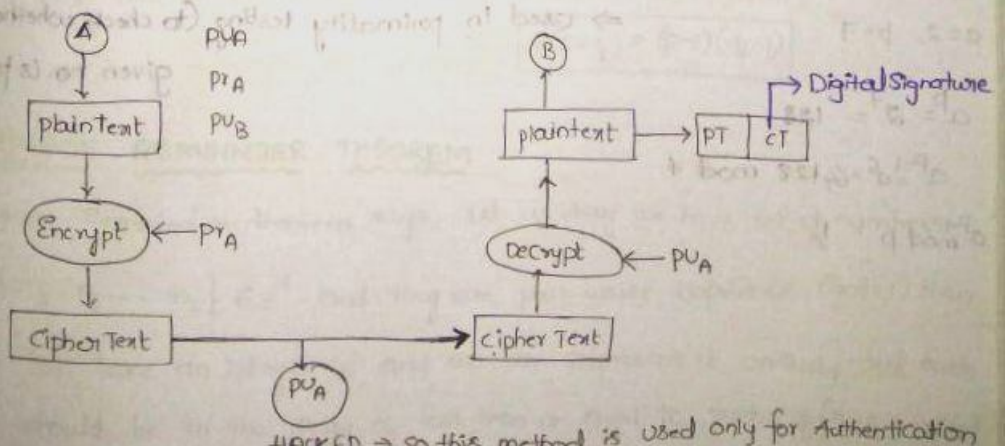
⇒ Before transmission of Data both sender and Receiver should generate a pair of keys called public Key and private Key. These Keys work this way,

⇒ If you encrypt anything with public Key of A, you can Decrypt only with private key of 'A' and vice versa.

⇒ Now, after generating pair of keys, Now both public Key of A, public Key of B should be known Globally. and the private key will only be known only to that party. Now, the sender (A) has 3 keys they are (PUA, PUB, PRA).

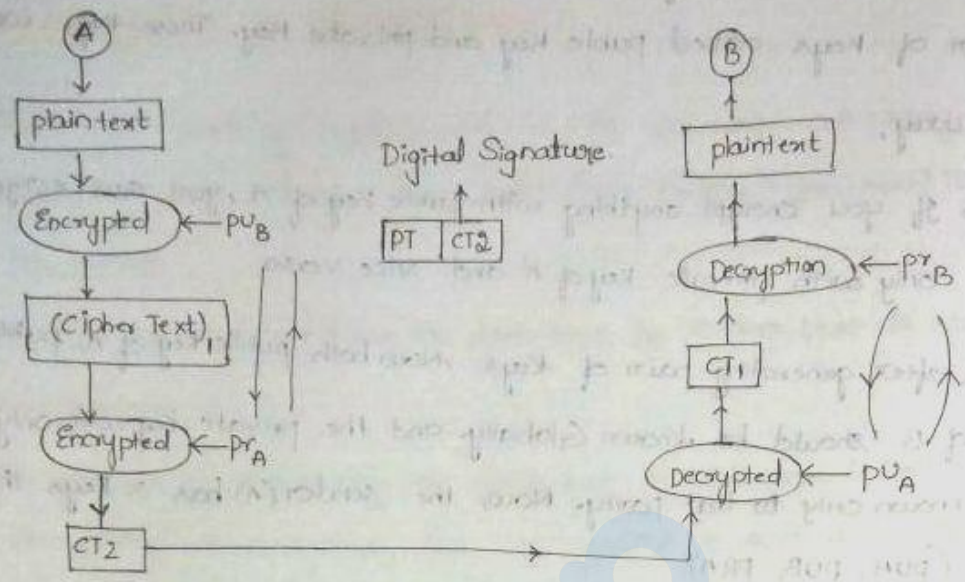


3. AUTHENTICATION USING PUBLIC KEY PRIVATE KEY



HACKED ⇒ so this method is used only for Authentication not for security.

4. ENCRYPTION AND AUTHENTICATION TOGETHER



⇒ Initially we have to generate public Key private Key pairs this can be generated using Number theory and from that we should do some Basics and from that two Algorithms, from that Algo's we get the public Key private Key pairs

5. FERMAT THEOREM

⇒ The entire public Key and private Keys are based on prime numbers

⇒ This theorem states that if 'p' is a prime number and $a \in z^+$ in such a way that 'p' does not divide 'a'; then we can say that $a^{p-1} \cong 1 \bmod p$

$$a^{p-1} \equiv 1 \bmod p \equiv a^p \cong a \pmod p$$

Ex:

$a = 2, \; p = 7$

$a^p = 2^7 = 128$

$a^{p-1} = 2^6 = 64$

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

⇒ used in primality testing (to check whether a given no. is prime)

6. EULER'S TOTIENT THEOREM

⇒ Euler Totient function says that if we have a number n, then the Euler totient function $\phi(n)$ is defined as the no of integers 'k' in the range $1 \le k \le n$ which are relatively prime to 'n' [gcd (k,n) = 1].

$n=9, \Rightarrow 1,2,4,5,7,8 \Rightarrow \phi(n) = \boxed{\phi(9)=6}$ {(1,2,4,6,7,8) are Relatively prime to 9. ⇒ gcd(1,9) = gcd(2,9) = gcd(4,9) = gcd(5,9) = gcd(7,9) = gcd(8,9) = 1

$n=5 \Rightarrow 1,2,3,4, \Rightarrow \boxed{\phi(5)=4}$

$n=7 \Rightarrow 1,2,3,4,5,6 \Rightarrow \boxed{\phi(7)=6}$

∴ {If 'n' is prime nos then $\phi(n) = (n-1)$}

$n=10 \Rightarrow 1,3,7,9 = \boxed{\phi(10)=4}$

⇒ $\phi(n)$ is Multiplicative ⇒ If 'n' can be written as product of two nos m×l such that m,l are relatively prime then

$$\boxed{\phi(n) = \phi(m) \times \phi(l)}$$

$\Rightarrow \phi(10) = \phi(2) \times \phi(5)$ — Relatively prime numbers to each other.

$= 1 \times 4$

$\Rightarrow \boxed{\phi(10)=4}$

$\Rightarrow \phi(35) = \phi(5) \times \phi(7)$

$= 4 \times 6$

$\boxed{\phi(35) = 24}$

⇒ If p,q are prime numbers then $\phi(p \times q) = \phi(p) \times \phi(q)$

$$\boxed{\phi(p \times q) = (p-1)(q-1)}$$

7. CHINESE REMAINDER THEOREM

⇒ Chinese Remainder theorem says, Let us say we have set of numbers $\{n_1, n_2, n_3 \cdots n_k\} \in Z^+$ and they are pair wise coprimes (gcd = 1) then we can take an integer 'x' and we can represent it uniquely and such 'x' should be in the range of less than or equal to product of $\{n_1 \times n_2 \cdots \times n_k\}$

$\Rightarrow x \cong (1 \cdots n_1 \times n_2 \times n_3 \cdots \times n_k)$

$x \cong a_1 \bmod (n_1)$

$x \cong a_k \bmod (n_k)$

⇒ Now let us say I have two relatively prime numbers (2), (5) Now I come take $2 \times 5 = 10$ and I can write all the nos between 1 to 10 uniquely as set of Remainders when divided with 2 and with 5.

	÷ by 2 (rem)	÷ by 5 (rem)
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
10 –	0	0

Remainder

⇒ Mainly used in Encryption.

8. PRIMITIVE ROOT

⇒ If 'a' and 'n' are Relatively prime numbers then there exists atleast one number 'm' such that $\boxed{a^m \cong 1 \bmod n}$.

If $a=7$, $n=19$ then $m=$? (Take the smallest value)

⇒ $7^m \cong 1 \bmod 19$

⇒ $m=3$ ⇒ $7^3/19 = 1$ ⇒ $\boxed{m=3}$ {option verification method}

(5) If $a=3$, $n=7$, $m=$?

⇒ $3^m \cong 1 \bmod 7$.

⇒ $\boxed{m=6}$

→ Now, 3 mod 7, ⇒ $3^1 \bmod 7 = 3$, $3^2 \bmod 7 = 2$, $3^3 \bmod 7 = 6$, $3^4 \bmod 7 = 4$, $3^5 \bmod 7 = 5$, $3^6 \bmod 7 = 1$, $3^7 \bmod 7 = 3$ ...

I got all numbers from (1 to 6).

∴ The repeating sequence (3, 2, 6, 4, 5, 1) is called period = '6' (total nos).

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

The Repeating sequence is (2, 4, 1) ⇒ period = 3.

⇒ Now, If i find (a mod b) to all powers of upto 'b' and the period is $\phi(b)$ then 'a' is called as primitive root of 'b'.

9. RSA ALGORITHM (RIVEST SHAMIR ADLEMANN ALGORITHM)

⇒ The most popular Algorithm to generate public key, private key ⇒ 1977 proposed.

Step-1 : CHOOSE TWO DISTINCT PRIME NUMBERS 'P' AND 'Q' (VERY BIG NUMBERS)

Step-2 : COMPUTE $N = P \times Q$

Step-3 : FIND $\boxed{\phi(n) = (p-1) * (q-1)} = (p*q) - (p+q-1) = N - (p+q-1)$.

Step-4 : CHOOSE AN INTEGER 'e' SUCH THAT $1 < e < \phi(n)$ && $\gcd(e, \phi(n)) = 1$ (↓ prime no.)

$\boxed{(e, N) \Rightarrow \text{PUBLIC KEY}}$

Step 5: DETERMINE 'd' as $\boxed{d \cong e^{-1} \pmod{\phi(n)}} \Rightarrow \boxed{ed \cong 1 \bmod \phi(n)}$

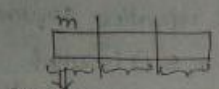
$\boxed{(d, n) \rightarrow \text{private key}}$

EXAMPLE

1) $p = 61$, $q = 53$
2) $n = 61 \times 53 = 3233$
3) $\phi(n) = 60 \times 52 = 3120$
4) e ⇒ $1 < e < 3120$ ⇒ $\boxed{e = 17}$ (choosen) ⇒ $\boxed{\text{PUBLIC KEY} = (17, 3233)}$
5) d ⇒ $ed \bmod \phi(n) = 1$ → $[(ex d) \bmod \phi(n)] = 1$ ⇒ $d = 2753$ ⇒ $\boxed{(2753, 3233) = \text{PRIVATE KEY}}$

$PU_A = (17, 3233)$

$PR_A = (2753, 3233)$



(A)

m = 65

$CT \Rightarrow (65)^{17} \bmod 3233$ } Encryption

CT = 2790

Cipher Text = $(m)^e \bmod n \Rightarrow$ To Encrypt

plain Text = $(CT)^d \bmod n \Rightarrow$ At Decryption

⇒ At Receiver $m = (2790)^{2753} \bmod 3233$

m = 65

10. DIFFIE HELLMANN ALGORITHM

⇒ This Algorithm is based on the concept of primitive Root

⇒ This Algo is used for generated public and private keys

⇒ This Algo is used to share the key securely. Key Exchange

Alice Bob

1) Both Alice and Bob Agreed to use a prime number 'p' and base 'g'.

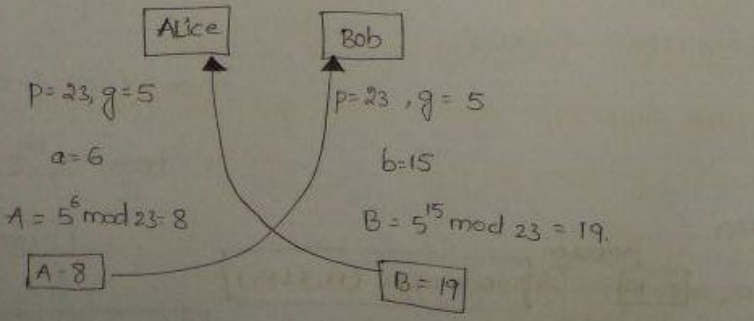
2) Alice choses a secret key (secret integer) 'a', then sends $A = g^a \bmod p$ to Bob

3) Bob also choses a secret integer 'b' then he too sends $B = g^b \bmod p$ to Alice

4) Both Alice and Bob compute the secret key like this.

5) Alice computes $S = B^a \bmod p = 19^6 \bmod 23 = 2$

   Bob computes as $S = A^b \bmod p = 8^{15} \bmod 23 = 2$



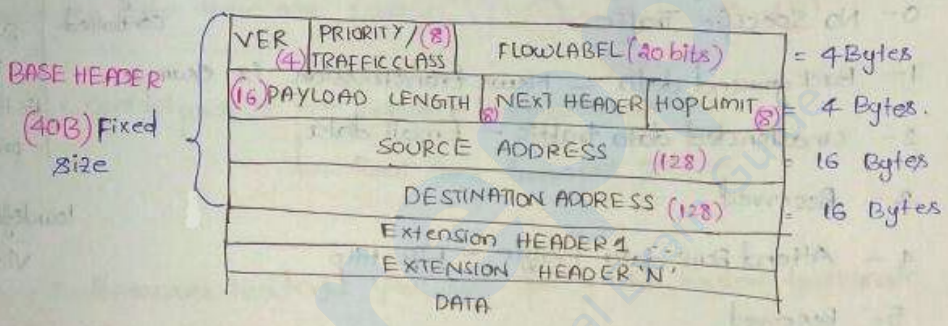
⇒ Not used for Encryption and Decryption used only for Key exchange

20. IPV6 AND WIFI

1. Need for IPV6:

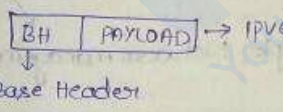
- IPv4 Address depletion (Not Available no of Ip Address)
- Realtime Audio/video transmissions
- Encryption (provided at N/w level).
- Authentication
- past processing (Avoid the process that makes the routers weak/Busy)
- Additional functionalities. (Extend Headers concept is used)
- Ipv6 is called Ipng (ng = Next Generation)

2. IPV6 HEADER



VER → what type of version we are holding. (If VERSION = 0110 = IPV6
0100 = IPV4)

PAYLOAD LNGTH:



NEXT HEADER: It says what type of Extension Header is just next to the destination Address (Extension Header 1) (same as options)

Hop LIMIT: Is same as TTL.

Questions that might be possible are

1) what are the fields that are present in both IPV4 and IPV6? (VERSION)
2) what is the size of so and so field.
3) TRUE/FALSE QUESTIONS!
4) which of the following has static header IPV6/IPV4? (Ans: IPV6)

3. TRAFFIC CLASS

The traffic can be classified into 2 types

Traffic
- Congestion controlled Traffic: The protocols that are sending the data observes there will be congestion and they will adjust their sizes (window sizes) appropriately.
- uncontrolled Traffic: No such above facility.
  Ex: Audio video Traffic. (Live streaming)

→ Now the first 4 bits of Traffic field will be given to priority [4 bits (0-15)]
(0-7) (8-15)
↓ congestion controlled
↓ uncontrolled
8 - least priority
15 - high priority
low definition Videos.

Pri	Meaning
0	No Specific Trafic
1	Background data — News transmission is example
2	unattended data traffic — Email data
3	Reserved
4	Attend Bulk data traffic — FTP, Http
5	Reserved
6	Interactive traffic — Remote login, putty, ssh
7	control traffic. — Highest priority among all (SNMP, Routing data)

SNMP: Simple Network management protocol

→ The priorities are set by the upper layers. (above N|w layer) and IPv6 should provide an interface to the above layers by which priorities are set.

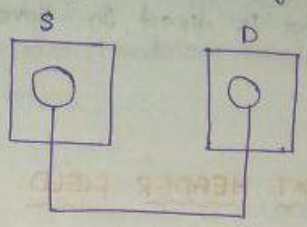
→ Source can set priority, Routers can change priority.

→ Destination should not assume the packet received is of same priority as set by source.

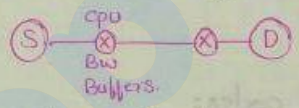
4. FLOW LABEL FEILD

Now, Assume, there are two process within 2 hosts. The stream of packets that go from source-destination is called "flow". And all the packets that belong to same flow have the same requirements, They are

1. Source Address
2. Destination Address
3. priority of all packets should be same
4. Extension headers should be same



→ This says that whenever you are sending the flow, before sending Inform all the intermediate routers to allocate appropriate resources for a flow, Generally for informing the routers that you are going to send the data, there are 2 ways.



1. Control packets
   - Resource Reservation protocol
   - Real time transport protocol.
2. Extension Headers: you can specify the Routers to allocate Resources.

→ For all the packets in the flow we use a label called flow label. (Randomly chosen)

→ The advantage of flow label is whenever a packet with label no. 10 arrives at the router, now router will form the flow table and the contents are

Source Address	Flow label	Information
A	10	Bw, Buffers, CPU

→ one entry is enough for all the packets

→ Hashing is used on flow table

Que that might be asked:

→ Between a source and destination there could be more than one flow.

→ All the packets need not have flow label, a packet can have flow label = 0.

→ Router can ignore this facility.

5. PAYLOAD LENGTH

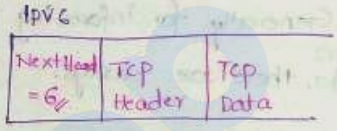
The entire thing that is present after Base header is called payload

The payload contains: Extension Header, upper layer packets. **

→ Header is fixed in Ipv6.

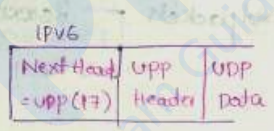
6. NEXT HEADER FIELD

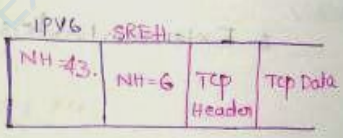
Now, consider an Ipv6 packet that contains a TCP packet (TCP Header, TCP Data) then Next Header field should contain '6' which indicates the packet next is TCP packet

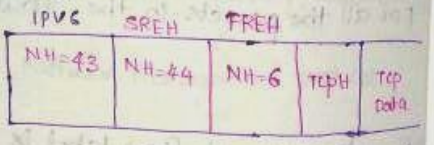


Next Header Codes:

Code	Next header
0	Hop by Hop option
2	Icmp
6	Tcp
17	UDP
43	Source routing
44	Fragmentation
50	Encrypted Security payload
51	Authentication
59	NULL (No Next header)
60	Destination option.







7. ORDERING OF EXTENSION HEADER

The order in which the Extension Headers should be placed after Base Header are defined by RFC, and the order is.

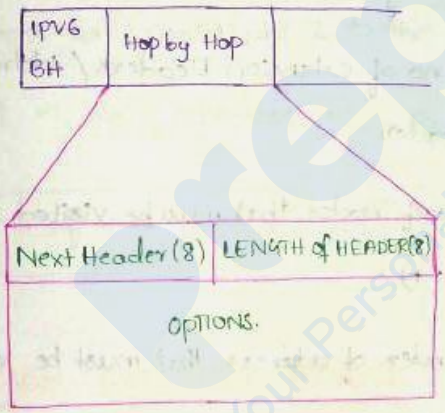
1. Hop by Hop options Header
2. Destination option Header
3. Routing Header
4. Fragment Header
5. Authentication Header
6. Encapsulating Security payload Header
7. Destination Header
8. Upper layer Header.

* This order is not mandatory but it is a general convention.
* Each Extension Header should not appear more than once, except "Destination options Header" *** (Atmost 2 times)
* If Hop-by-Hop options is even present in a packet then it should always be present immediately after Base Header (Mandatory Rule)

1. HOP BY HOP OPTIONS HEADER

⇒ The importance of Hop by Hop options header is whenever a source is sending the IPV6 packet via some intermediate routers then the packet should be Examined by all the intermediate routers and the destination, and check if any functions are specified that must be carried out.

The Hop-by-Hop options look like.



payload = 16 bits

⇒ max size of payload: $2^{16}-1$

$= 65535$

∴ If payload/Datasize is > 65535 we use "Jumbo packet"

1. Jumbo packet: payload size > 65,535 then we use "JUMBO PACKET"
2. pad i: we have to see all the options of 8 bytes in length if we are having option <8B then we pad some bits.

2. ROUTING EXTENSION HEADERS--1

⇒ Source Routing option Header: source takes the control over the path in which its packet should be transmitted.

* Assume that source (A) wants to send a packet to destination (B) via two 3 Intermediate Routers ($R_1 R_2 R_3$). $(A - R_1 - R_2 - R_3 - B)$

⇒ Now, say the Ipv6 contains the Baseheader and Routing Header (RH) and we know that the Base Header contains (Source Address and Destination Address)

∴ BH
- Source Address
- Destination Address

RH
- The order of address that the packet should go
- SL field

DA = R_1
A[1] = R_2
A[2] = R_3
A[3] = B

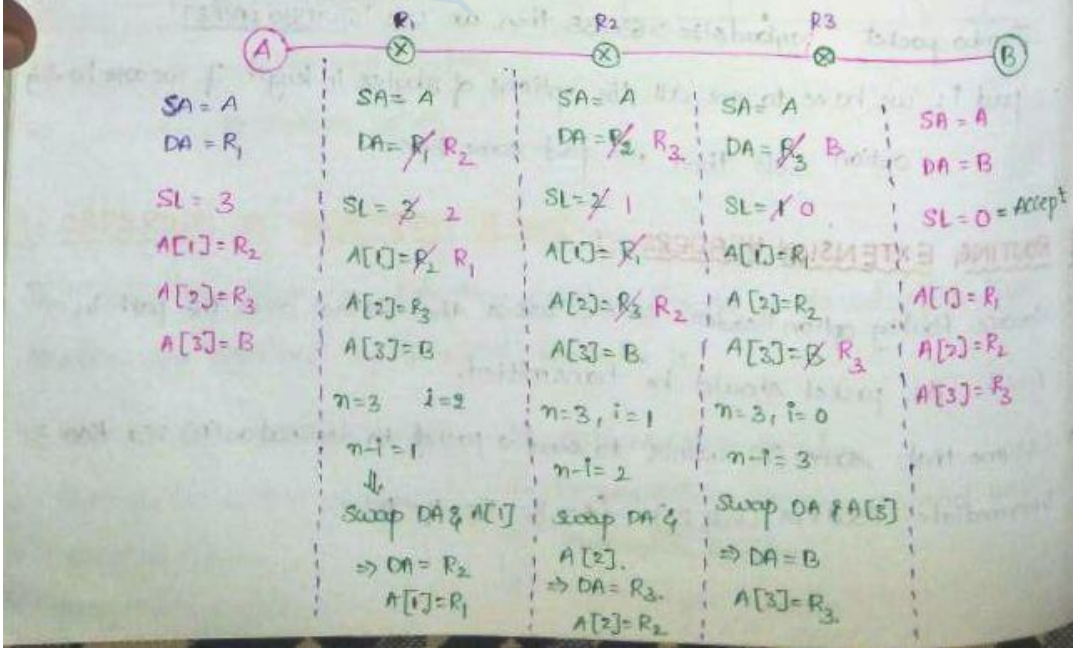
⇒ Along with BH, RH we will have a field called "segments left" which represent the no of of intermediate nodes which are left to be visited before we reach the final destination. In this case, when the packets starts, its SL value = 3 (No of Intermediate routers that have to be visited before reaching 'B'.

⇒ Let 'n' represent the no of extension Headers / Address that are present in Extension Header

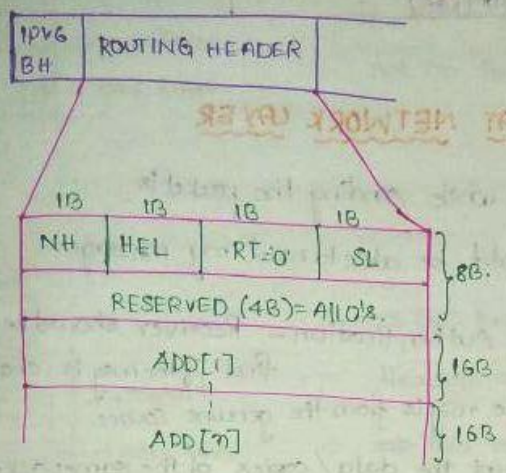
⇒ Let 'i' represent the no of nodes that must be visited before we reach final destination. (i = SL-1)

⇒ Now, (n-i) gives the Index of address that must be visited next

⇒ Now, swap DA and the A[i]



10. ROUTING EXTENSION HEADER-2



NH = Next Header
HEL = Header Extension length
RT = Routing Type
SL = Segments left

⇒ HEL represent the size of header in terms/units/multiples of '8'.

⇒ Now, say HEL = 8 then what are no. of Addresses that are present?

Each Address will contribute 2 to the HEL ⇒ $16/8 = 2$ (→ Address size in IPV6)

∴ HEL = 8 ⇒ No. of Addresses = $\frac{HEL}{2} = 8/2 = 4$ Addresses are present.

∴ No. of ipv6 Address = $\frac{HEL}{2}$

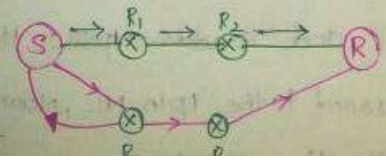
11. FRAGMENTATION EXTENSION HEADER

⇒ Not required for all packets

⇒ In Ipv4 the fragmentation is done both at the "routers" and at "source."

⇒ In IpV6 the fragmentation is done only at the source. ***
The Routers are not supposed to the fragmentation.

⇒ The sender finds the path MTU before sending the packet and fragment it appropriately and starts the transmission. But there is no guarentee that the packet follows the path for which we calculated the MTU. In that case, an icmp packet will be generated and sent to the sender.



(8)	(8)	(13)	(2)	(1)
Next Header	Reserved=0	Fragment offset	Reserved	MF
IDENTIFICATION (32)				

12. NEED FOR SECURITY AT NETWORK LAYER

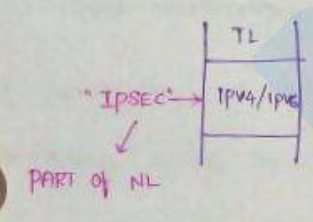
The main things that we need while sending the packet is

1) confidentiality – No one should be able to read 'my messages'
2) Message Integrity / Message Authentication – Receiver should be able to find if the msg is changed on the way
3) Source Authentication – If the msg is from the genuine source
4) Reply attack problem – corrupt the data / copies of the same packet

→ The protocol that is used to provide the above feautres is called "Ip security" protocol (or) "IPSEC"

IPSEC – collection of protocols.

IpSEC
- Authentication Headers protocols
- ESP protocols (Encapsulating Security protocols)



13. MODES OF IP SECURITY

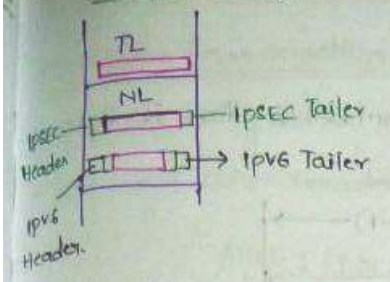
There are two modes in which Ip security operates, they are

1. Transport mode
2. Tunnel mode

Transport mode :

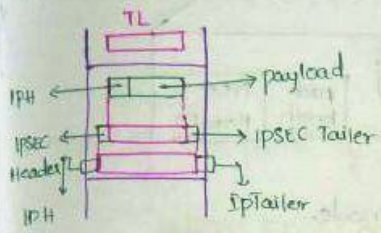
Consider a Transport layer packet is passed to the NL then in the NL, first the packet is received by IPSEC and it adds IPSEC Header and IPSEC Tailer and again it is passed to the Ipin NL, Now, IP adds Ip Header and Ip Trailer and transmits the packet.

Transport mode



⇒ The main point to note is the modification is done only on the payload part but not on the Transport, Network layer Headers.

Tunnel mode



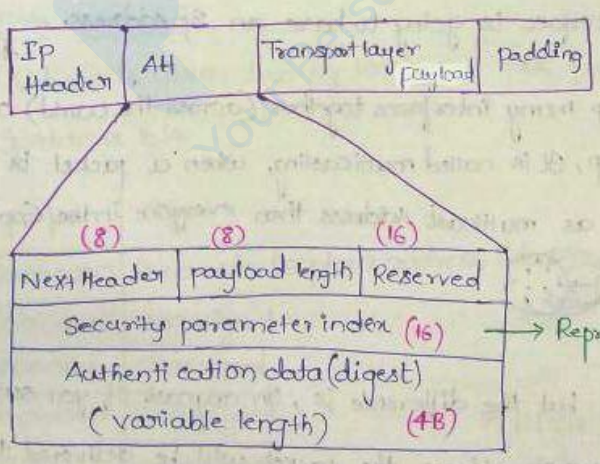
⇒ The transformation is done on both the Headers and payload.

⇒ This is called tunneling because an IP packet is kept inside another IP packet and the two IP Headers need not be same.

⇒ In IPV6 the Tunneling is used in between routers.

14. AUTHENTICATION HEADER

→ Authentication Header mainly provides us with message Integrity, Source Authentication.



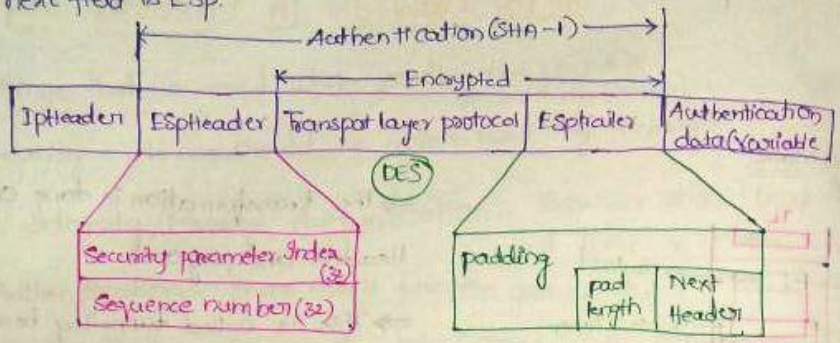
⇒ NH : (8-bit) = Tells what is present in the next field if Tcp data is present then NH = 6.

⇒ payload length : (8-bit) : The length of the "Authentication Header." The Authentication Header size is measured in size of 4 Bytes. (scale factor 4)

15. ESP

Encapsulation Security payload protocol.

→ If the protocol field of IPtheader contains the number 50 then it represents that the next field is ESP.



⇒ The AH and ESP are operated under Transport mode.

16. IPV6 ADDRESSES

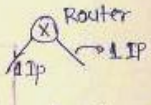
Size: 128 Bits

→ Unicast

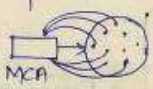
→ Multicast

→ Anycast

Unicast: Every interface is going to have an Ip Address.

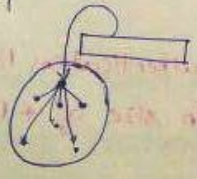


Multicast: we group many interfaces together (across the world) and assign an Ip Address to the group, It is called multicasting, when a packet is sent at having the destination Ip as multicast Address then everyone in the Group is going to receive it.



Anycast:

Same as multicast but the difference is, In anycast if you send a packet having destination Ip as Anycast address the packet will be delivered to the closest Host and it is the responsibility of that host to distribute the packet to every host in the group.



⇒ In Ipv6 Broadcasting is not defined

17. IPV6 ADDRESS REPRESENTATION

Colon Hexa Representation

⇒ IPv4 uses using dotted decimal Representation.

⇒ IPv6 uses colon-hexadecimal Representation.

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

4bits 4bits 4bits 4bits.

= 4+4+4+4 = 16bits

= 2B.

= 2B+2B+2B+2B+2B+2B+2B+2B

= 16B

= 16x8bits = 128bits.

⇒ Most of the parts of IPv6 are Zeros.

ABCD : EF01 : 0000 : 6789 : 0000 : EF01 : 2345 : 6789.

⇒ They have used short hand representation by deleting the leading 0's and if all 4 0's are present they used a single '0'.

Actual Ip: ABCD : EF01 : 0000 : 0789 : 0000 : EF01 : 0345 : 0789.

Short Hand: ABCD : EF01 : 0 : 789 : 0 : EF01 : 345 : 789.

18. TYPES OF IPV6 ADDRESS

When an IPV6 Address is given then by looking at first few bits we can decide what type of Ip Address it is.

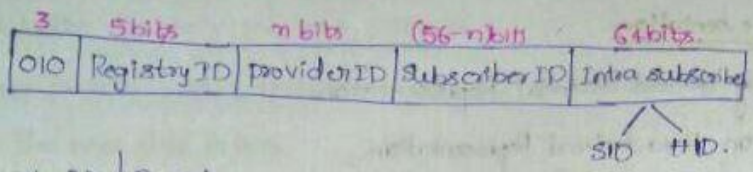
	prefix	Type
First 8 bits =	00000000	Reserved - Loopback address, Idont have Ip.
	00000001	unassigned
	0000001	unassigned
	000001	unassigned
	00001	unassigned
	0001	Reserved
	001	Reserved
*	010	provider based unicast Address
	011	unassigned
*	001	Geographic based unicast
	100	

* 1111111 010 = link local Adress
7

* 1111111 01 = site local Address
7

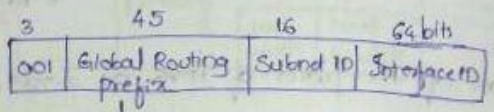
* 11111111 = multicast address
8

19. PROVIDER BASED UNICAST ADDRESS



Registry ID	Registry
10000	Multiregional
01000	RIPE NCC
11000	INTERNIC
00100	ApNIC

Geography Based unicast Address



↓

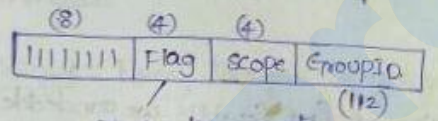
contains Latitude and Longitude locations.

Multicast Address: First 8bits are ones.

Flag: 0000 = permanent group

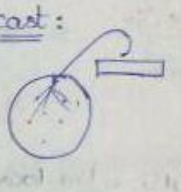
0001 = Transient group

Scope = 4bits = 0000 = Reserved
0001 = Node local
0010 = Link local
0101 = Site Local
1000 = Organizational
1110 = Global
1111 = Reserved



0000 = permanent
0001 = Transient

Anycast:



Assign the same Ipaddress to one of the interface of the Each node and send the packet with that Address.

23
20. SOME SPECIAL ADDRESS

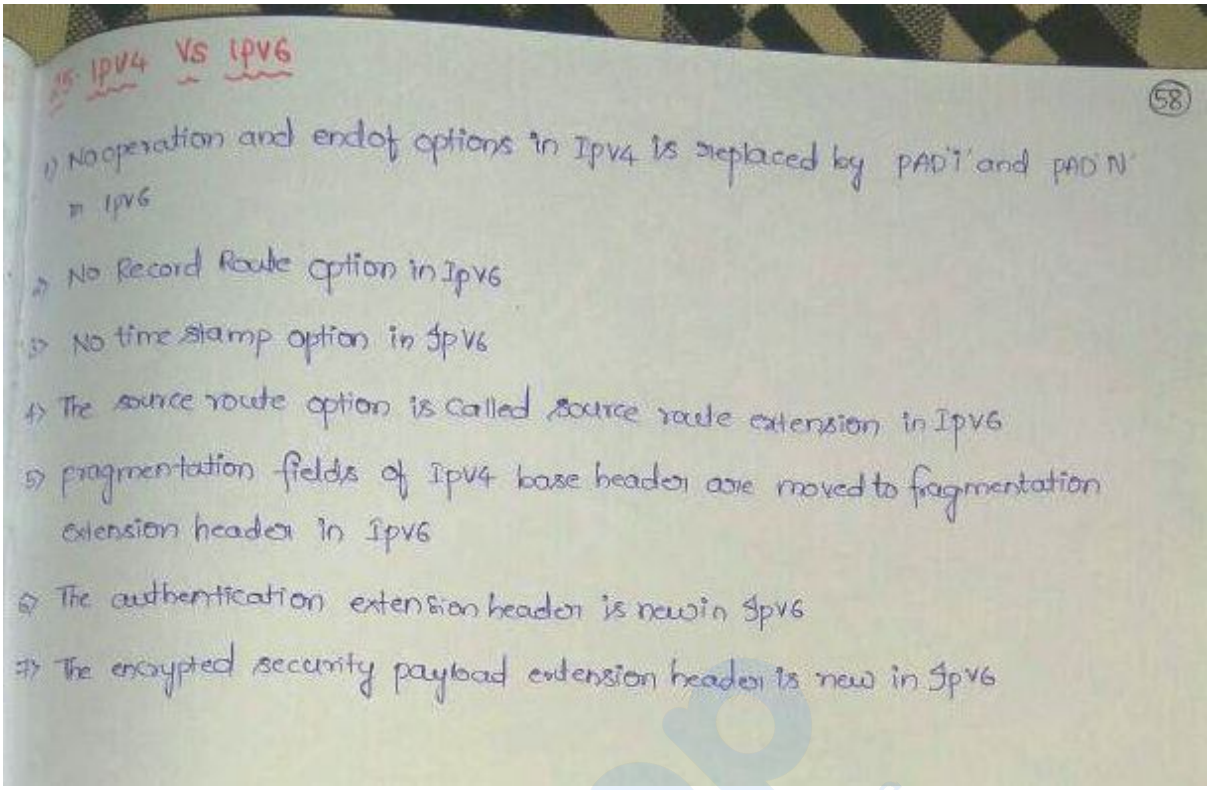
Unspecified: All 0's = 128bits are 0's.

Loopback Address: Test self connectivity = All 0's except last bit (=1)

= 127 (0's) and last '1'.

Ipv4 compatable: First 96bits 0's and last 32bits represent IPv4 Address.

Ipv4 mapped: First 80bits = 0, next 16bits 1's remaining 32bits = IPv4 Address



IPV4 VS IPV6

1) No operation and end of options in IPV4 is replaced by PAD1 and PADN in IPV6
2) No Record Route option in IPV6
3) No time stamp option in IPV6
4) The source route option is called source route extension in IPV6
5) pragmentation fields of IPV4 base header are moved to fragmentation extension header in IPV6
6) The authentication extension header is new in IPV6
7) The encrypted security payload extension header is new in IPV6