NETWORK LAYER

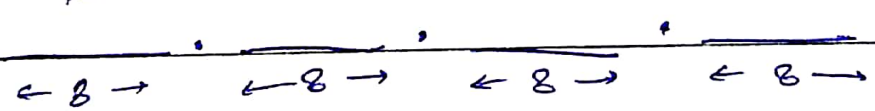
Logical Addresses (IP Add.)

→ To connect with Internet, a machine must have an IP address.

192.48.63.39 → notation is known as Dotted decimal notation

IP add. → 32 bit long

0-255



Address space : no. of add. possible = 2^{32}

Binary :- binary ko equivalent of above octet without dot

Theoretically,

→ At a time, 2^{32} can connect to Internet ~~at a time~~

$\downarrow$

$\simeq$ 4B

Addresses

Classfull addressing — Classless addressing

(Binary Notation)

(1) Classfull addressing

5 Classes :

	I B (1st Byte)	II B	III B	IV B	decimal notation
class A	0 (start at 0)				0 - 127
Class B	10				128 - 191
class C	110				192 - ...
Class D	1110				
Class E	1111				

11001111.... ⇒ Belongs to class C

192.48.... ⇒ "

Allocation :- (of IP add.)

→ 2 parts :- netid, hostid (2^7) [1st bit is fixed]

If 1st byte represents netid ⇒ 128 n/w are possible in Class A

↓

within this ~~total hosts possible~~ ⇒ 2^{24} → 2nd/3rd/4th byte

Only big organis^n's would go for it.

ICAN :→ give IP add.

Class B :→ 2 bytes : netid → Mid size org^n's
" : hostid

no. of n/w possible : 2^{14} (1st two bits are fixed)
no. of hosts possible : $2^{16} \simeq 65536$
(size of each block)

Class C → Startup (needs less IP add.)
3 bytes : netid (no. of blocks)
1 byte : hostid

no. of blocks : 2^{21} (3 bytes fixed)
block size : 2^8

Class D : 1 byte : netid : Used for Multicast app's
There's no hostid

Class E : 1 byte : netid : Reserved
There's no hostid

CIDR : not^n to represent A, B, C
Class A /8 : 1st byte is fixed
Class B /16 : 1st 2 bytes are fixed
Class C /24 : " 3 " "

Mask : 32-bits long
Class A 11111111 0000 ... 0
8
Class B 11111111 11111111 0000 ... 0
16
Class C 11111111 11111111 11111111 0...0
24

* Valid only for Class A, B & C.

Signature

IPV4 add. : 32 bits
(in both Classfull & Classless)

DATE: / /
PAGE NO.:

→ 1 lac people ⇒ can't be connected by class B
class A capacity : 1 B
↓
10 lakhs of connections are wasted
These many add. were not utilised.
Same is for B & C.

→ All add. couldn't be utilized

Disad. :-
↳ Add. got depleted fast as high chunk of add. were given to org^ns.
↳ Add. were under-utilised.

→ More flexible method is needed.

(II) Classless Addressing : give only as much as needed.

If I need 2 add. ⇒ can manipulate only 1 bit

Mask : 1 1 1 1 ... ... 1 0 / 31
(31 bits) / (fixed)
can be manipulated

* CIDR : Classless Inter Domain Routing
(How many 1's followed by how many 0's)

↳ 10 addresses ⇒ /28

→ ICAN gives these ip. add.
We go to local ISP's to get add.

→ If add. is :

192.46.64.52/20

⇒ Total add. allocated = 2^{12}

Restrictions

↳ add. allocated are in power of 2 : 2^n

↳ 1 orga^n has contiguous alloc^n of add

↳ 1st add. should be ÷ by total no. of addr.

→ .52/28

⇓

we can manipulate last 4 bits of last byte

32+16+4

bits are fixed

52 : 0 0 1 1 0 1 0 0

1st add. : 0 0 1 1 0 0 0 0 → 192.46.64.48/28

last : 0 0 1 1 1 1 1 1 → 192.46.64.63/28

↳ How many add. are possible : $N = 2^{32-n}$

∩∞ 192.46.64.48/28

not^n : x y z t /n

* 1st add. in list of all add. is used by Router (.48 here)

↳ Grouping addresses ⇒ can create hierarchy

n/w within n/w : sub net

192.46.64.52/28 → 4 ⇒ 2 bits need to be manipulated

4

4

1st 4 : netid

last 4 : hostid

Teacher's Signature

* always continous add. 'll be allocated to ~~subnet~~ subnet

DATE: / /
PAGE NO.:

/30

2 bits manipulated

0 0 1 1 0 1 ~~0 0~~ 0 0

1st add. : 0 0 1 1 0 1 0 0 .52 } For 1st subnetwork

Last add. : 0 0 1 1 0 1 1 1 .55 }

.56

.59

Q. ~~Write~~ 198 . 44 . 53 . 48 / 26 (6) → 4 bits needed

16
8
5
} 3 subnetworks

Find 1st & last & mask of subnet.

(start with highest size)

00 ( )

/28

0 0 1 1 0 0 0 0

1st add. : .48)

last add : 0 0 1 1 1 1 1 1 .63)

~~0 0 0 0 0 0 0 0~~ .~~40~~)

/29 .47)

.35)

.39)

9/9/22

Special Addresses : can't be used by host at global n/w

0 0 0 0 /32

255.255.255.255/32 (works only within same n/w) (Broadcast)

127.0.0.0/8 → loopback (server & client both on same machine)

160.70.14.0/24

$2^8 = 256$

160.70.14.0/24 → 1st add. (divisible by 256)

160.70.14.255/24

① → 160.70.14.0 } → this wastes more add.
160.70.14.63 } → already ~~is~~ allocated
160.70.14.127 }
160.70.14.255 }

→ ISP can't give add. to that orgⁿ now.

→ DHCP : Dynamic Host Control Protocol (server)
↓
allocates IP add. to systems (host)

host requests (give req) to DHCP

| S | D |

↓
add of DHCP

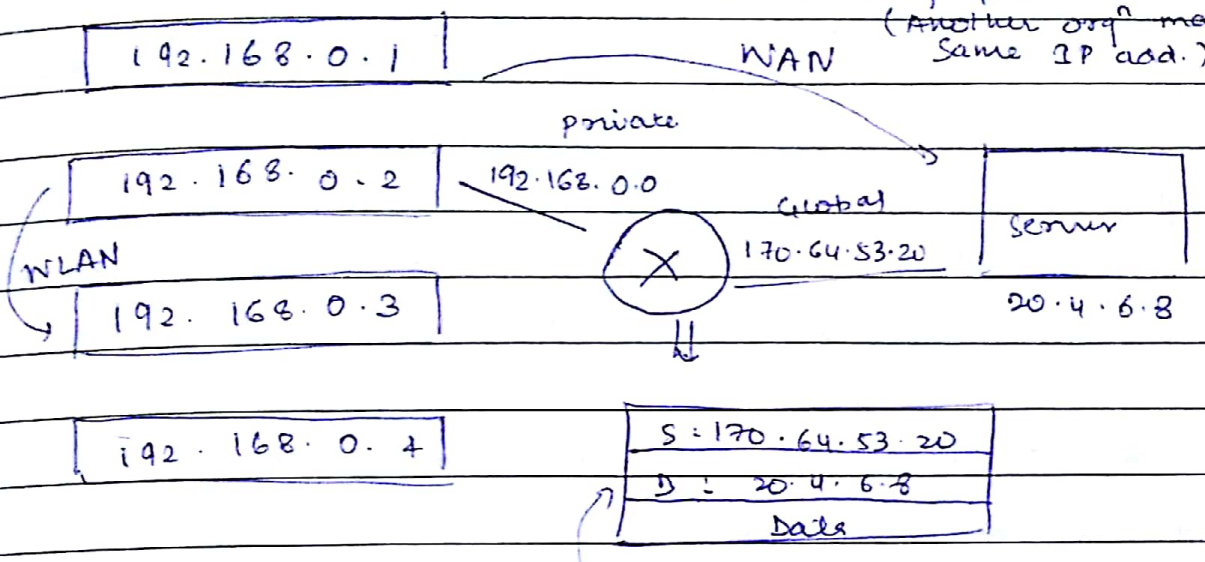
Source doesn't have IP add. now

→ S : 0000/32 will be used as source add
DHCP will give it IP add

Teacher's Signature

10.0.0.0 /8
172.16.0.0 /12
192.168.0.0/16
169.254.0.0/16
} Private add. (free add)

For ①, org's use free add. within the org's
Each is using pvt add.
(Another org may use same IP add.)



→ Upto now : IP add. were unique globally.
→ When IP add. got depleted → start using duplicate Add.

Pkt sent to server : through router

S : 192.168.0.2
D : 20.4.6.8
Data

when sent outside, private add. is changed to global

→ Private add. can never go through router

For reply :

D : 170.64.—
S : 20.4.6.8
Data

How 'll it reach 1 PC ?

1) Broadcast :- But other 'll also have to process it to

Teacher's Signature

Contd :

→ Table is stored at Router : Translation Table

Mapping of Private Add. to Global IP Add. :

Network Address Translation (NAT)

Translation Table

192.168.0.2	20.4.68	170.64.53.2
192.168.0.3	20.4.68	" "

- If only 1 global add, only 1 conn^n will occur at a time (if 2 want to connect to same server)

→ If server sends reply back → who'll get it ? Not clear

→ If we use above table, only 1 conn^n is possible

Add External IP add. also

⇓

Still a problem.

→ Can have multiple IP Add.

1 con^n — 1 Global Add.

→ Can have server inside org^n.

- Routers are NAT-enabled.

→ Now also, 1 PC can connection ~~with~~ have 1 with 1 IP Add.

→ Practically, 1 PC only has 1 IP Add.

Website are hosted on servers
↓ have
IP Addresses

DATE: / /
PAGE NO.:

* First IP add. are matched. Within it, Routers are matched

Port No.:

every app^n has diff. port no.
↓
1 PC can have multiple conn^n's with same IP add.

* NAT: Andar kitne bhi IP Add. use kare, baahar kamchi dikhenge

* 2 people want to use ~~same~~ gmail within same org^n:
Dono ka port no. same hoga
But IP add. alag hoga

→ S: 192.168.0.2
D: 192.16.0.4 → want to send to all duplicate add^n
⇓
Router won't pass any pkt which has D as pvt. add.

→ S: " "
D: 192.168.0.4 } But Router isn't allowing
⇓
Can use a switch (don't look for IP Add.)

* In a subnet ⇒ mask size always ↑
1st house is getting 4. If it needs 10 ⇒ use NAT enabled Router

→ Check for 1st byte → 2nd byte
⇓
address aggregation

IBP: aggregation of all H 001 - H 128 add.
Router hoss

Teacher's Signature

IPV6

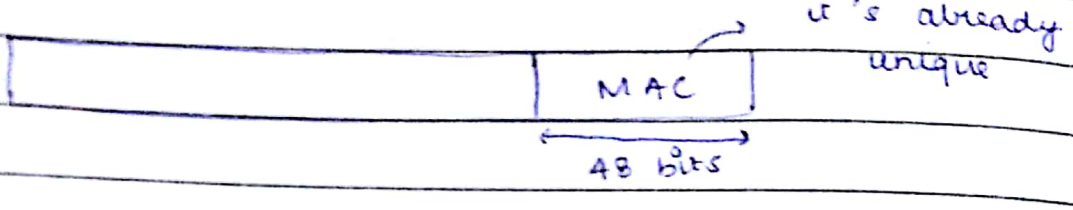
2^{128}

A649 : 0000 : 0000 : 0000 : 00F9 : 0654 : -

Abbrev^n :

A649 : 0 : 0 : 0 : F9 : 654 :

A649 :: F9 : 654 : -



Here, auto config^n is possible (unlike earlier)

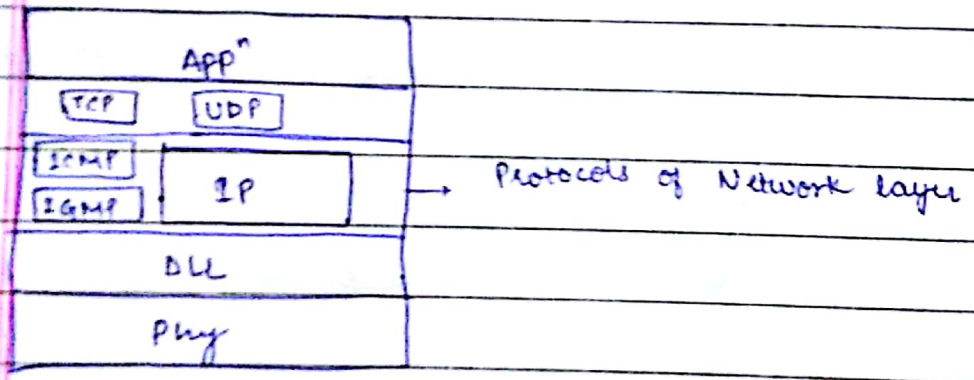
IPV6 —want to communicate→ IPV4 ⇒ Not nice usually

⇓

add trans^n is possible here

12/3/18

Network layer - IP



→ ICMP : used for error (App^n :- Ping)

Internet Control Msg. Protocol

→ IGMP : used for Multicasting

Internet Group Msg. Protocol

For N/W layer :

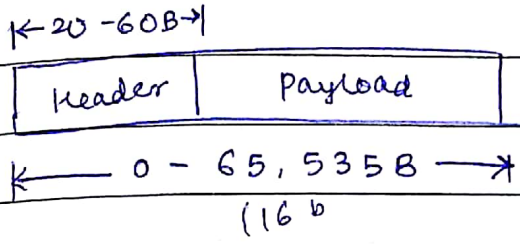
Add some header | Payload | : Packetizing

create Packet & deliver to DLL

Structure of Packet : (or a Datagram)

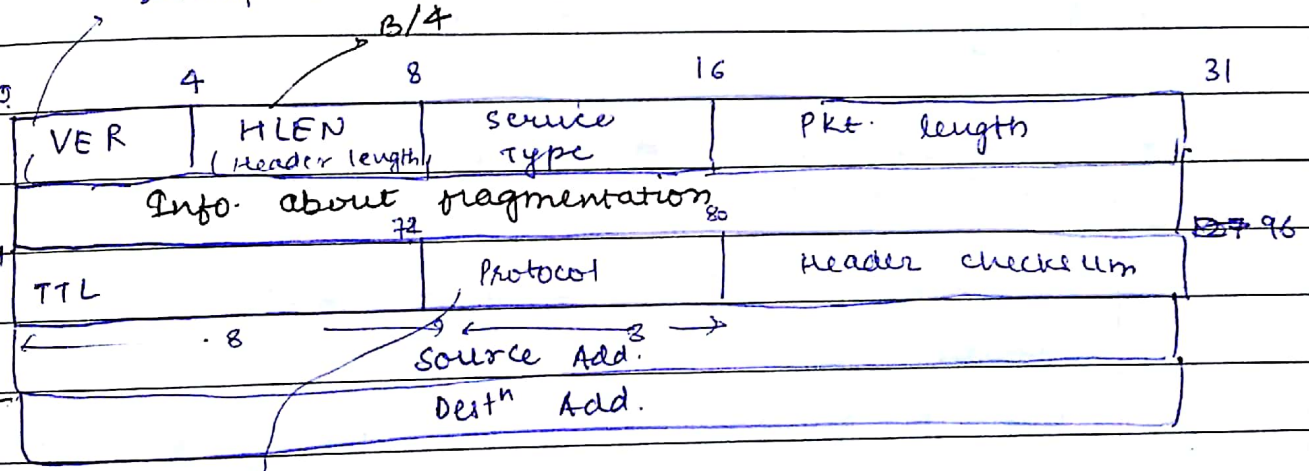
IP is available in form of IPV4 and IPV6

→ IPV4 :



Ethernet expects only 1500 bytes, how to accomodate in frame)

binary eqvt. of 4/6 (IPV4/ IPV6)



→ HLEN : what protocol uses services of IP

4 bits ⇒ Only 16 header lengths possible.

↳ variable header length. To know from where payload starts.

But we've total 60 B.

So, for every 4 bytes, there is 1 length.

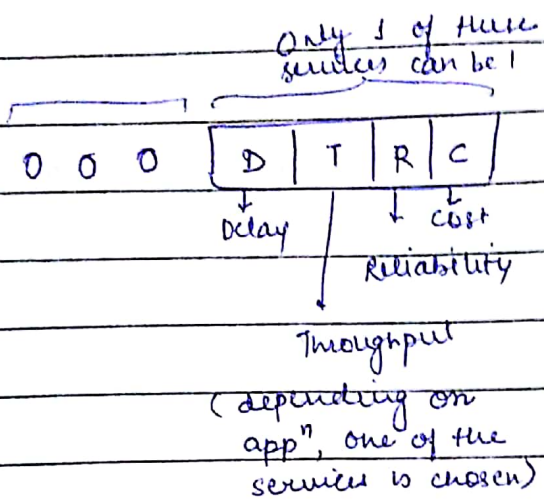
Max. header length = $\frac{60}{4}$ = 15 (4 bits)

→ Service Type : 8 bits : for Quality of Service

1st 3 bits : for priority

⇓

due to congestion, routers are forced to drop pkts with less priority



* If Interactive (video chat) ⇒ Delay ↓ ⇒ need delay

* In Real time app" (all pkts. should reach) ⇒ Reliability ↑

* Background job ⇒ Cost ↓
N/w Management "

→ Pkt length : Header length + Payload

Required :

Info. to receiver kitna pkt. milna chahiye ki NO single pkt. pahuch gaya h

Payload = Pkt length − 4 × Header length

→ 0 1 0 0 0 0 1 0 (bits 8 7 6 5 4 3 2 1)

Receiver KO 1st 8 bits ye mile

Discarded. Why?

Version no. = 4

Header : 2 × 4 = 8

⇒ Header length should be 20 − 60

⇒ Discarded.

Teacher's Signature

→ Value of total length field is 100 B. & value of HLEN = 5. Size of Payload ?

$100 - 5 \times 4 = 80$

⇒ TTL (Time to Live) → 8 bits ⇒ 255

Because of an error in Routing algo., pkt circulate in same routers. This goes on. It is utilizing resources when

* TTL says that pkt should be dropped

2 × (Maxm no. of routers) is stored in TTL.

It's value 'll decrement when it reaches a router. Wherever value = 0, Router 'll drop pkt.

⇓

who'll check at ⇒ Reliability
router

↓

jisne bheja h (upper layer) no hi dekhega.

* N/W layer is only providing best effort service.
(aage agar kuch gadbad hoya toh upper layer ko kaun hi dekhna padega)

→ Even protocols ICMP & IGMP use IP.

⇒ Header checksum :

N/W is adding header. whatever it has add, it can check.

*⇒ Upper layer are responsible for payload
N/W layer is only header, not payload

Teacher's Signature

not same everytime you connect

logical add. : 32 bits

⇒ length of source add. : 32
dest^n add. : 32

96 + 64 = 160

32 × 5 = 160

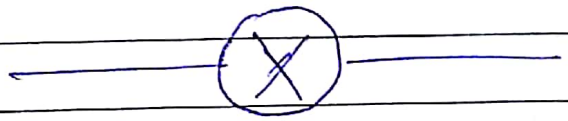
Total header length = $\frac{160}{8}$ = 20 B

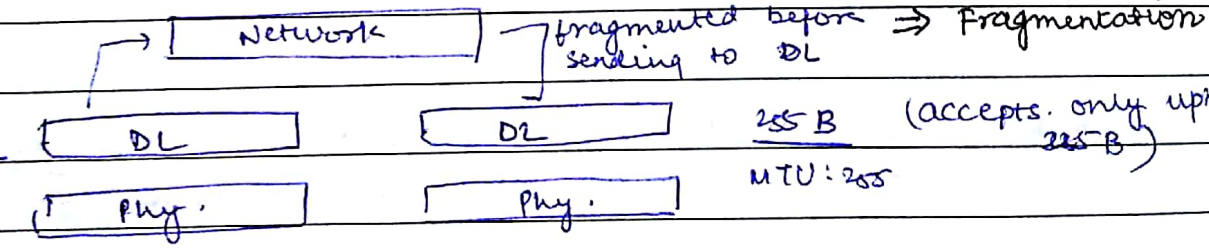
40 Bytes reserved for this

⇒ In 'options' : routers add. should be mentioned.
If pkt goes to any other router, it'll be discarded. ⇒ Strict Routing

Loose Routing : can be sent to other routers also (have to pass through all routers mentioned in option)

→





⇒ The 32 - 64 bit have info. about these. (fragmentation)
since pkts arrive in sequence, we know start & ending. But it follows pkt switching. Every pkt may follow diff. path (route), may come out of order. So, to sort them, this info. is needed

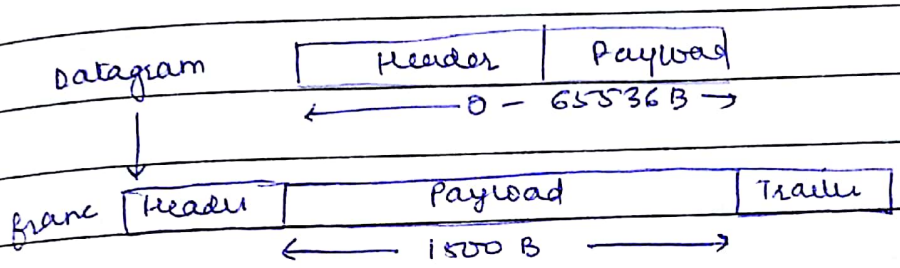
Fragmentation : Datagram

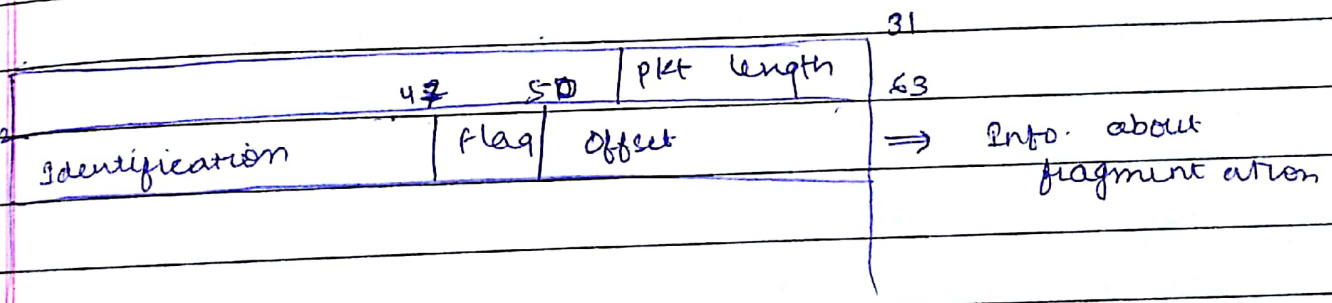
→ Time Stamp : included in options

how much each router is taking to process the datagram

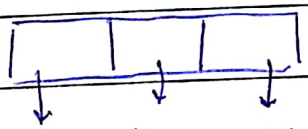
→ Packets come out of order ⇒ we need sequencing

→ MTU : Maxm Transmission/ Unit Transfer





Flag (3 bits) :



U (Unused) — D (Don't fragment) — M (More) ⇔ are their more pkts arriving for this id

D = 1 ⇒ No fragmentation ⇒ M = 0 (It's 1st & last pkt

→ Pkt length will tell how many pages are in each size of data envelope pkt

Teacher's Signature

identification : : upto 2^{16} values

ek type ke nodes ke pkt ki id → same

For 1st packet : Offset = 0

→ Max. length of a datagram : 65,536 B

Offset (13 bits): $0 - 2^{13}-1$

$0 - 8191$

Min. header length : 20 B

Max. payload : 65,516 B

can group Bytes to get 1 offset

$10 \times 8 = 8192$

$\frac{65516}{8192} = 8$ ⇒ can group 8 bytes to get 1 offset

1400

* ⇒ packet length should be divisible by 8.

if length = 1500 ✗ can't be sent in 1 packet (not in units of 8)

= 1400 ✓

Eg. Payload = 4000 B

Ethernet can only take = 1500 Bytes

So, fragmented :

Pkt. length (header = 20B)

Pkt. length			
4020	4000B	0-1399	1420
		1400 - 2799	1420
		2800 - 3999	1220

→ Only the payload is fragmented.

Header is attached to every datagram (Hence, taken 1400 instead of 1500)

For 1st packet, Offset : 0
2nd. : 175 $\left(\frac{1400}{8}\right)$
3rd : 350
} id will remain same

value of M : 1
: 1
: 0

→ now Receiver will assemble it?
Using info: identification, pkt. length, M, ...

* Fragment source pe bhi ho sakta h or Router pe bhi
there could be many fragments reaching the receiver.
Offset / pkt. length may change at every router.
so header ~~also~~ may be diff. at R & S side.
Header checksum can't work properly in this case.

Sol^n :
→ every router when modifies something, it'll recalculate & modify router
↓ problem
Time delay
Processing at each router should be avoided ⇒ Use IPV6.

→ 2000 B ka packet with D=1. ⇒ Ethernet can't pass it further
↓
Router will drop that pkt.
ICMP msg. is

16/3/18

IP (V6) (Read from Book)

Hlen

Options → Router has to take care of this

fragmentation → Source / Router

header checksum → head processing at every Router ⇓ delay

→ Due to change in characteristics of data, header of IP V6 is modified : (text → multimedia) much delay can't be tolerated

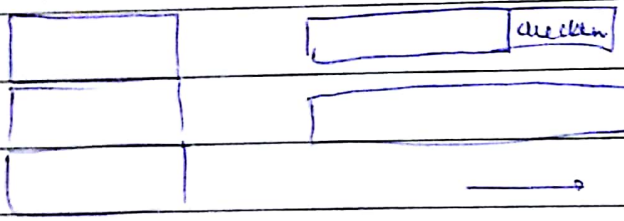
→ In IPV4 : there was no security for pkts

But security has to be built in the pkt itself.

IP (V6)
- security
- real time xfer of data
- flow label

To avoid delay:

→ can remove header checksum



App^n puts its own checksum

transport layer also...

can remove checksum here as

↳ medium are error free now

↳ upper layers already have checksum & it has to send to upper layer only.

→ can remove fragmentation at Router

A special pkt is sent 1st, it brings min. MTU.

So no fragmentation at Router is required

→ Options :

data can be given special services

Valid 16/3/18 00:00

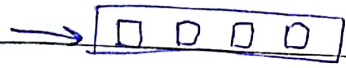
Options ~~in~~ pages

flow label :→ 1,2:

1	... em
2	abc

↓
set by src machine

→ Instead of options, it has flow label ⇒ header len : fixed (fixed len)
earlier → we had op's (IPV4)



if congestion ⇒ drop pkts with least priority

→ distinguishes

flow label 'll say that these pkts. should have spl. space.

↑ buffer should always be reserved for them.

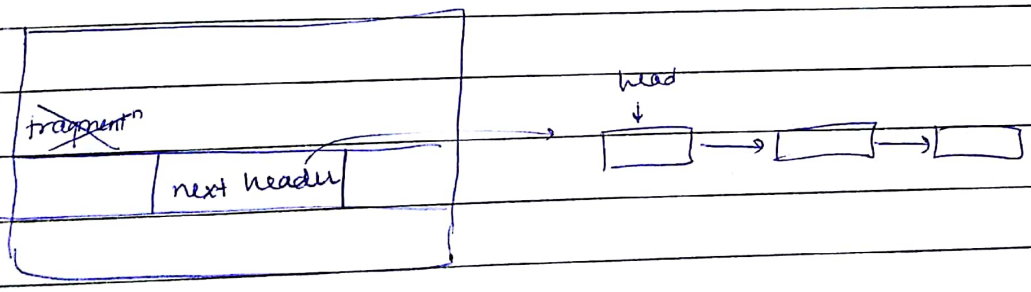
HD video is sent :- extra info is sent ⇒ Priority ↓
(redundant info) → some pkts can be dropped

Non-HD → no redundant info ⇒ Priority ↑
→ pkts can't be dropped

IPV6 :

→ Header : fixed length ⇒ NO HLEN

→



In b/w some may be IPV4
↓
fragmentation bits 've to check

dynamic header → if fragment" → IPV4 is occuring : then only next header 'll be added
at any router

↳ hop by hop : word addressable

Teacher's Signature

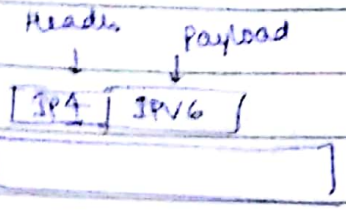
Wireshark

↓ Install it

DATE : / /
PAGE NO. :

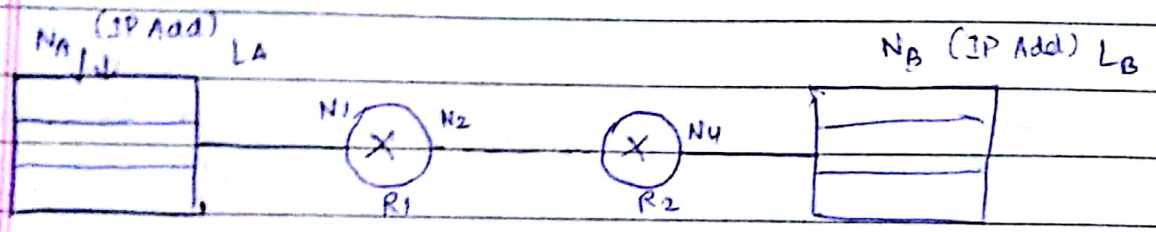
If IPV6 enabled router → it'll avoid next header part

→ Machines have both IPV4 & IPV6 version : dual stack



S — D

IPV6 — IPV4 — IPV6



Note: DHCP client program must be installed in your system

N_B	N_1	forwarding table

→ How to see

CAT / VAR / log / syslog

→ S knows its own add

How it gets IP add of N_B?

is DNS used

S gives DNS add → DNS converts to IP add of src.

Command: / Arp

N_A	N_B	Data

↓

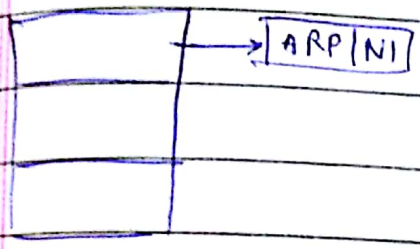
LA				
LB	N_A	N_B		DL

How to get Mac. Add. of next Hop?
(know D N of next)

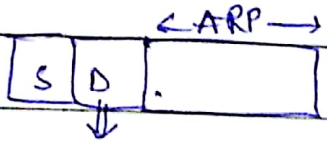
IP ⟶ ARP : maps IP add. to Mac Add

Address Resol^n Protocol at N/W layer

ARP : sends request with N_1 as IP add. → Msg. is broadcast to n/w



This msg is converted to frame



Broadcast add. will be dest^n add.

When decapsulated, N_1 is recovered. Only ROUTER 'A' reply to it (having IP add. N_1)

ARP reply pkt 'll be unicast to source
(Have both N & L of source)