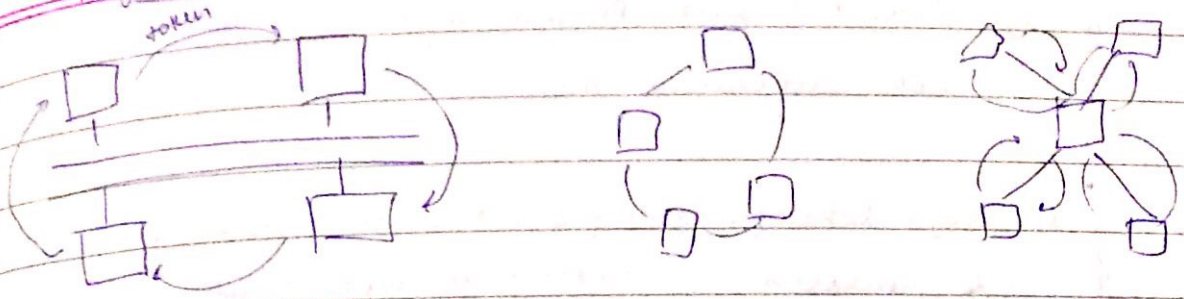
DATE : / /
PAGE NO. :

Physical Topologies



Bus Topology

⇓

Physical top is bus but logical is Ring

⇓

Ring possible only using primary

5/2/18

Ethernet & Wireless LAN → Data link layer & Physical layer involved.

(802.3) (802.11)

IEEE proposed these standards

* Diff. mfg. companies manufacture same device. They should have same protocol in order for them to communicate. IEEE standardizes the protocol.

802.11

.11a .11b .11p ...

DL
- framing — (1)
- error control — (2)
- flow control — (3)
- Medium access — (4)
  - CSMA/CD
  - Token Ring
    - polling
    - Time sharing
  - CSMA/CA

Teacher's Signature

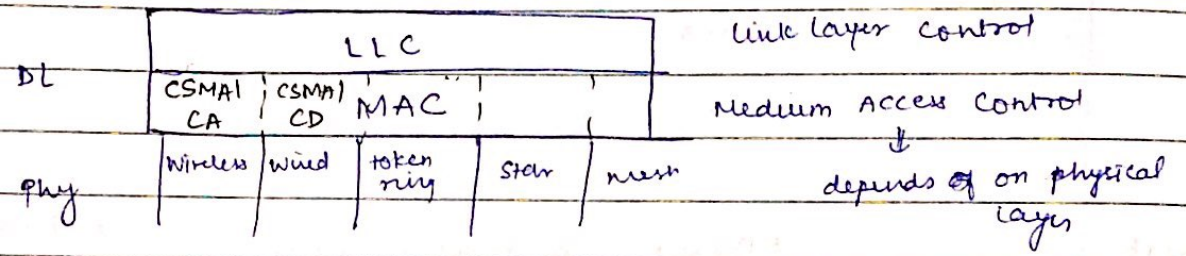
Tightly linked with Physical layer : ④

not dependent on " " : ①, ② & ③

If phy. topology changes, medium access will change

if wireless : CSMA/CA will be used.

wired : CD " "



② & ③ :→ functionality of LLC

① : general : " " "

some : " " " MAC

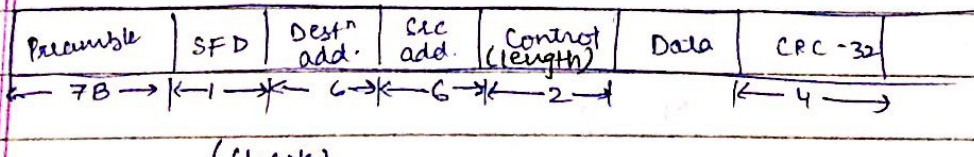
(frame length)

Ethernet

- Std. Ethernet (10 Mbps)
- Fast " (100 Mbps)
- GB " (1 GB)
- Super " (10 Gb)

Std. Ethernet :

Physical address : MAC add. (depends on NIC)



Teacher's Signature

→ Nowdays, GPS is used for synchronizing of clocks

→ And MAC add (in Hexadecimal form)

SFD Start frame delimeter

↓
flag signifying start of frame

↳ Preamble : sequence of 10101010 ---
7 bytes

↳ SFD - 1010 ---- 11

•) Preamble : utility is to synchronise the clocks.

added at Phy. layer, not at DL layer

•) SFD :

sequence of 1010 ---- 11
⇓
frame is going to start

last change for clocks to synchronise

MAC add : AB : 4A : Tc : 4A

↳ Unicast
↳ Multicast
↳ Broad cast

} add. itself tells → dest" add.

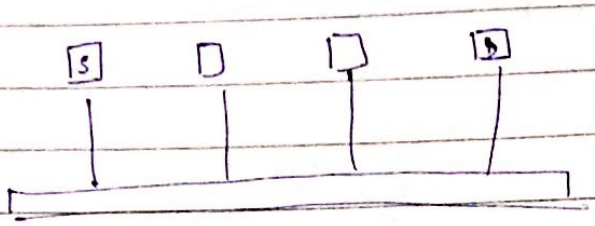
•) Source add. can only be unicast.
(Only 1 is sending data)

(LSB)
if ~~left~~ most bit is 0 ⇒ unicast
else ⇒ multicast
~~right~~ ao

4A : 0100 1010
47 : 0100 0111

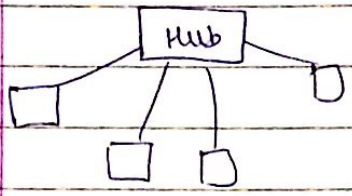
Teacher's Signatur

→ Bus :



every body receives whatever src is sending
↓
Broadcast
But only one will process it further.
(whose MAC Add. will match to dest" add.)

→ Star



Hub : broadcasting

→ Switch

Unicast / Multicast
it doesn't repeat what it is getting

•) min length of frame : 512 B b
leaving preamble, Data = (512 - 18) B b
if length of data coming is less ⇒ Padding should occur
more ⇒ fragmentation
max length : 1500 B b → needed in order to give equal opportunity to all stations

Q. Std. Ethernet = 10 Mbps
512 b

Time to transfer 512 bits on std. ethernet ?

$= \frac{512}{10 \times 10^6} = 51.2\ \mu s$: time required to send 1 frame

Teacher's Signature

Access point → talking about Wireless

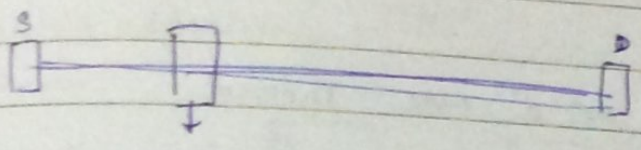
8. If propagation speed in channel is 2×10^8 m/s 51.2 μs req to transfer 1 frame. What should be length of channel

T = 2 Transmission time

$\frac{d}{2 \times 10^8} = 2 \times 51.2 \times 10^{-6}$

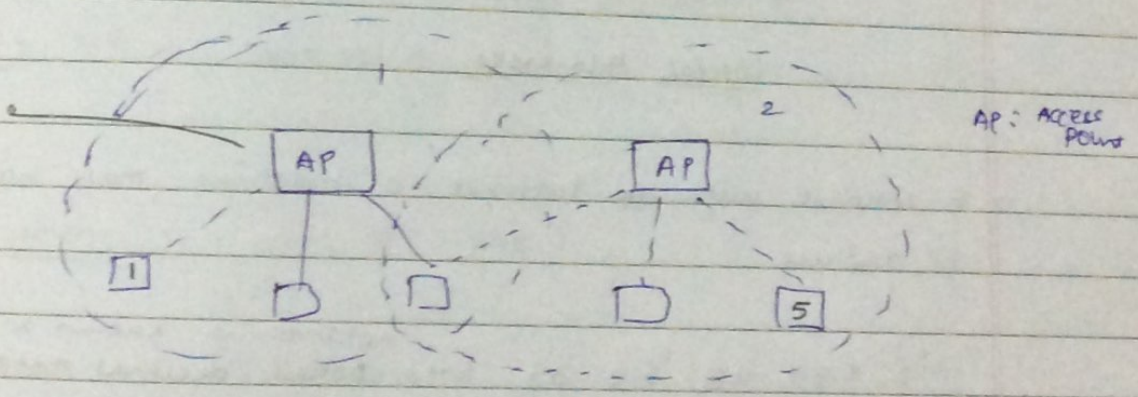
Total length (for RTT) = 10240 m.

d = 5120 m (single side)



Repeater
amplify signal
also cause delays

$\therefore (\text{length})_{\text{effective}} \cong 2500$ m



Architecture

- BSS : Basic Service Set
- ESS : Extended " "

Teacher's Signature

DATE : / /
PAGE NO. :

→ AP

→ N/w in which no external infrastructure is used - Ad Hoc N/w
(within same circle

→ use of " " : Infrastructure N/w
(AP)

BSS : available with both

ESS : if Machine wants to communicate with another which is not in range of its AP

→ 802.3 don't define how AP are connected

→ CSMA/CA is used as a MAC Protocol.

↳ WLAN has smaller frame length because collision has to be avoided.

7-3-18

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}

Teacher's Signature

Binary :→ binary ko equivalent of above octets without dot

DATE : / /
PAGE NO :

~~Theoretically~~

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

$\simeq$ 4 B

Addresses

Classfull addressing — Classless addressing

(Binary Not ⁿ)

① classfull addressing (1st byte)

5 classes :	I B	II B	III B	IV B	Decimal notation
class A	0 (started at 0)				0 - 127
class B	10				128 - 191
class C	110				192 - ...
class D	1110				
class E	1111				

110011 111 .... ⇒ 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

→ 2nd / 3rd / 4th byte

within this ~~total hosts possible~~ ⇒ 2^{24}

Only big organis"s would go for it.

Teacher's Signature

ICAN :→ give IP add

DATE : / /
PAGE NO. :

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

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

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 : notn 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 ---
8
Class B 11111111 11111111 0000 ---- 0
16
Class C 11111111 11111111 11111111 0 ---- 0
24

* Valid only for class A, B & C.

Teacher's Signature

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

DATE : / /
PAGE NO.:

→ 1 cr people ⇒ can't be connected by class B
class A capacity : 1 B
lots 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^n's
↳ Add. were under utilised.

→ More flexible method is needed

① Classless Addressing : give only as much as needed

If I need 2 add ⇒ can manipulate only 1 bit

Mask : 1111 ---- 10 / 31
31 / fixed
can be manipulated

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

↳ ID address ⇒ /28

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

Teacher's Signature

DATE : / /
PAGE NO. :

→ 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

32+16

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

ex 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 make 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

/30

2 bits manipulated

0 0 1 1 0 1 0 0

1st add. " .52 } 4 for 1st subnetwork

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

56

Q 198 · 44 · 53 · 48 / 26 (6) → 4 bits needed

16 → .59
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)

/29

.40)

.47)

.35)

.39)

Teacher's Signature

9/3/24

DATE: / /
PAGE NO.:

Special Addresses — can't be used by host 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 wants more add

160 70 14 63 → already ~~to be~~ allocated

160 70 14 127 → already ~~to be~~ allocated

160 70 14 255 → already ~~to be~~ allocated

→ ISP can't give add to that org^n 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.

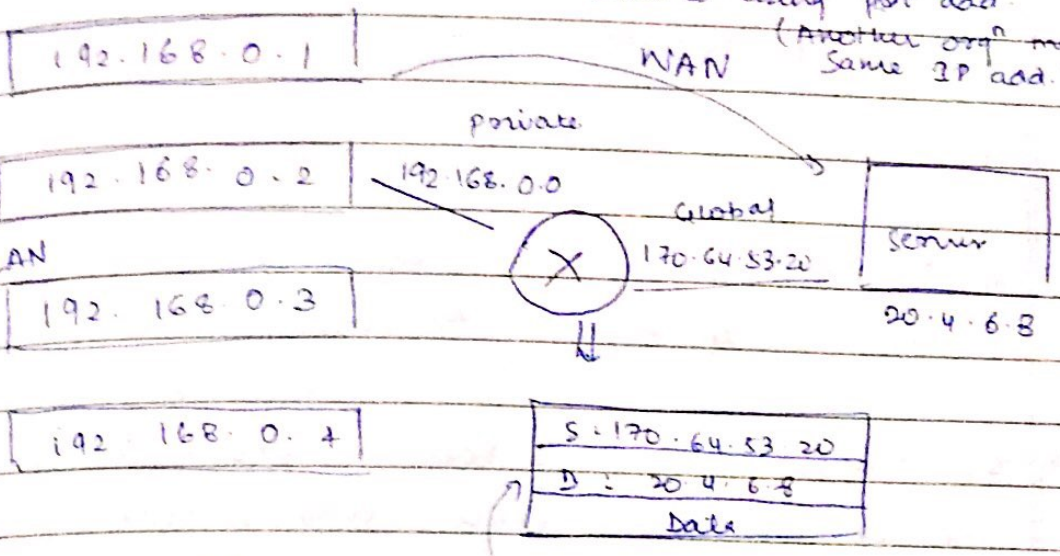
→ S : 0000 /32 will be used as source add

DHCP will give it ~~to~~ 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

when sent outside, private add. is changed to global

S : 192.168.0.2
D : 20.4.6.8
Data

→ Private add. can never go through router

For reply :

D : 170.64.—
S : 20.4.6.8
Data

How 'll it reach 1 PC ?

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

Teacher's Signature

Correct :

2) 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	120.64.53.2
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~~ have 1 connection with 1 IP Add.

→ Practically, 1 PC only has 1 IP Add.

Teacher's Signature

website are hosted on servers
↓have
IP Addresses

DATE: / /
PAGE NO.:

first IP add. are matched within it, Routers are matched

Port No :
every appn has diff port no.
↓
1 PC can have multiple conn's with same IP add.

NAT : Andar kitne bhi IP Add. use kare, baahar kam hi dikhenge

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

→ S : 192.168.0.2
D = 172.16.0.4 → want to send to all duplicate add.
⇓
Router won't pass any pkt which has D as pvt. add.

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

* In a subnet ⇒ masksize 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 H001 - H128 add.
Router hoga

Teacher's Signature

DATE: / /
PAGE NO.:

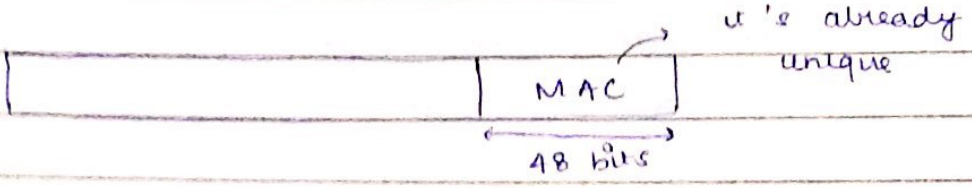
IPV6

2^{128}

AG49 : 0000 : 0000 : 0000 : 00F9 : 0654 : -

Abbrev^n : AG49 : 0 : 0 : 0 : F9 : 654 :

AG49 :: F9 : 654 :



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

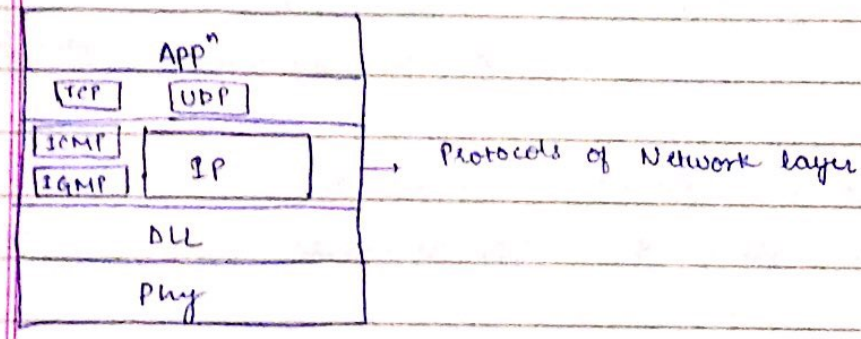
IPV6 —wants to communicate→ IPV4 ⇒ Not vice versa

⇓

add translation 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

Teacher's Signature

For N/W layer :

Payload : Packetizing

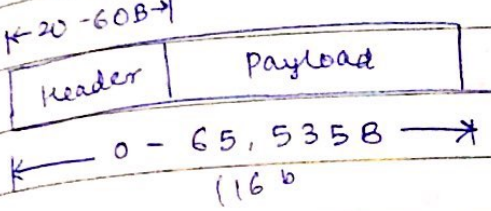
Add some header

create Packet & deliver to DLL

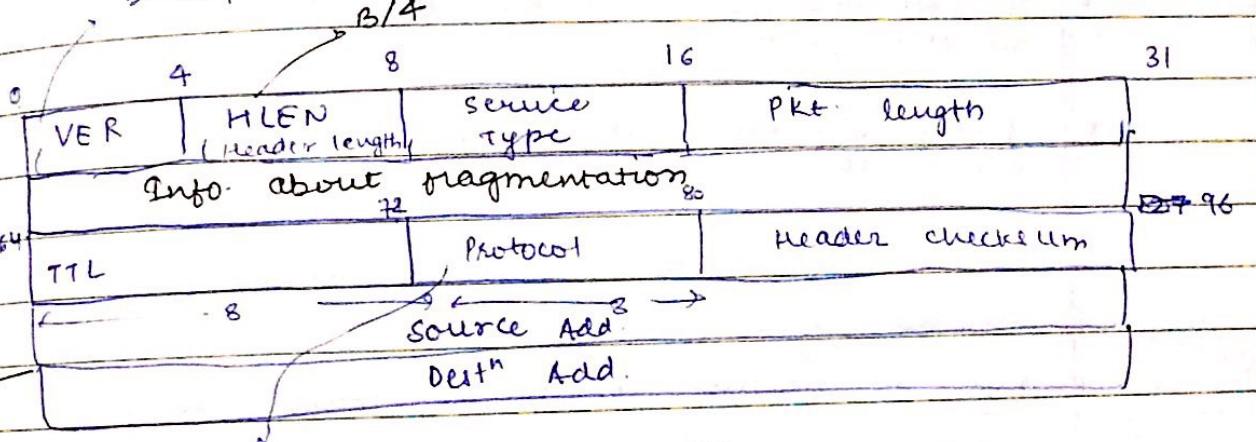
Structure of Packet : (or a Datagram)

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 'll total 60 B

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

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

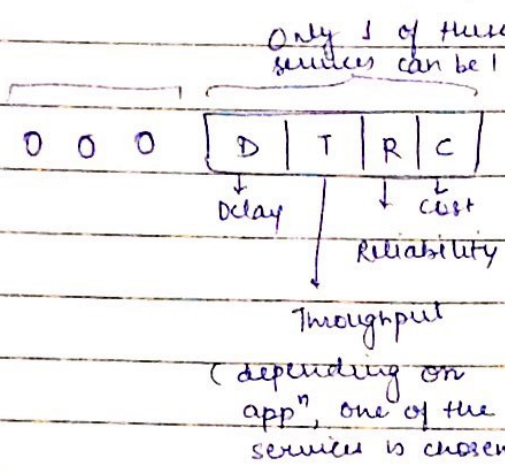
Teacher's Signature

→ 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" ⇒ Reliability ↑
(all pkts should reach)

* 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 (16 8 4 2 1 0) 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

DATE: / /
PAGE NO.:

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

100 - 5 × 4 = 80

⇒ TTL (Time to Live) → 8 bits ⇒ 255

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

* TTL says that, when pkt should be dropped

2 × (Maxm no of routers) is stored in TTL. Whenever it's value 'll decrement when it reaches a router. When value = 0, Router 'll drop pkt.

⇓

who'll check at router ⇒ Reliability

↓

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 khud 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

DATE: / /
PAGE NO.:

Logical add : 32 bits

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

96 + 64 → 160

32 × 5 = 160

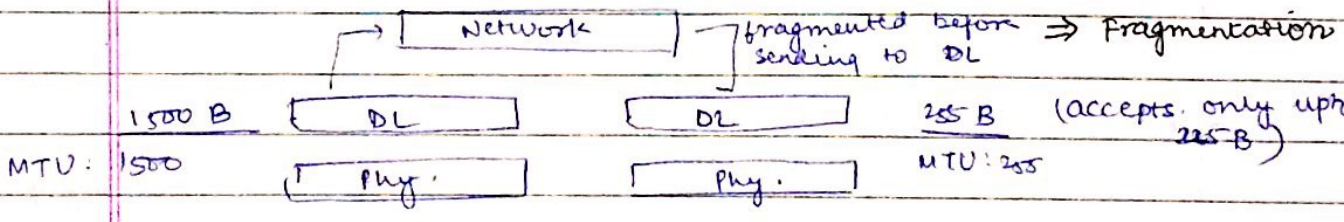
Total header length = $\frac{160}{8}$ = 20 B

→ 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

Teacher's Signature

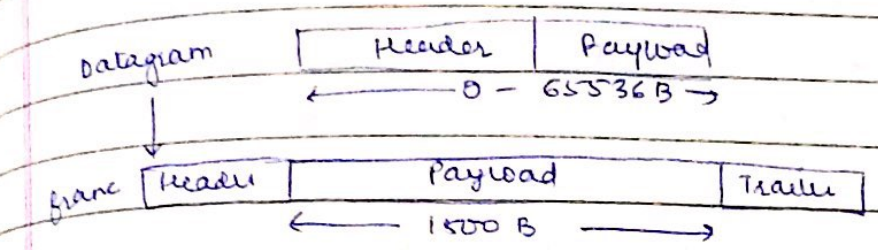
DATE: / /
PAGE NO.:

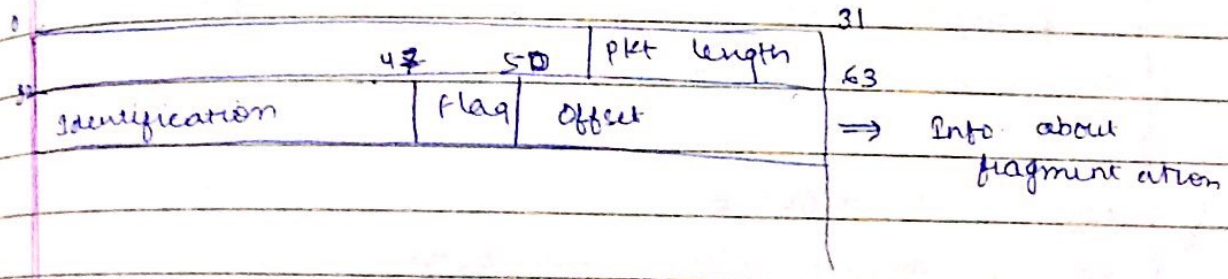
Fragmentation : Datagram

- Time stamp : included in options
  ↓
  how much each router is taking to process the datagram

- Packets come out of order ⇒ we need sequencing

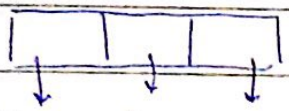
- MTU : Maximum Transmission / Unit Transfer





⇒ Info about fragmentation

Flag (3 bits) :



U (Unused) D (Don't fragment) M (More) ⇒ are there 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 envelope pkt

size of data

DATE: / /
PAGE NO.:

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) : $0 - 2^{13}-1$

$0 - 8191$

Can group Bytes to get 1 offset

Min header length : 20 B

Max. payload : 65,516 B

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

1600

* ⇒ 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 1420 → Only the payload is fragmented.

4020

4000 B : 0-1399 1420

1400 - 2799 1420

2800 - 3999 1420

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

Teacher's Signature

For 1st packet, Offset : 0

2nd : 175 $\left(\frac{1400}{8}\right)$ } Id will remain same

3rd : 350

value of M : 1

: 1

: 0

→ now Receiver will assemble it?

Using info: identification, pkt length, M, ...

• Fragment source pe bhi NO extra 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

Teacher's Signature

16/3/16

DATE: / /
PAGE NO.:

IP (V6) (Read from Book)

When

Options → Router has to take care of this

fragmentation → Source / Router

header checksum → needs processing at every Router → delay

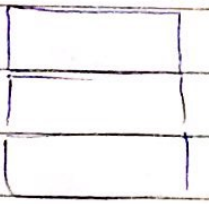
→ Due to change in characteristics of data, header of IP V6 (text → multimedia) is modified : much delay can't be tolerated

→ In IPV4 : there was no security for pkts

But security has to be built in the pkt itself.

IP (V6)
- security
- To avoid delay:
  - real time xfer of data
  - flow ~~label~~ label

→ can remove header checksum



App^n puts its own checksum

transport layer also ... ✓

can remove checksum bcz 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

Teacher's Signature

Rabi 16/3/18 00:00

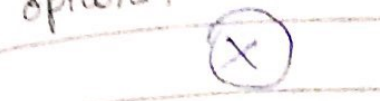
DATE: / /
PAGE NO.:

Flow label :→ 1,2:

1	... etc
2	abc

used by src machine

instead of options. it has flow label (fixed len) ⇒ header len : fixed
(IPV4) ⇒ we had op^ns



packets arriving

buffer (Q)

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.

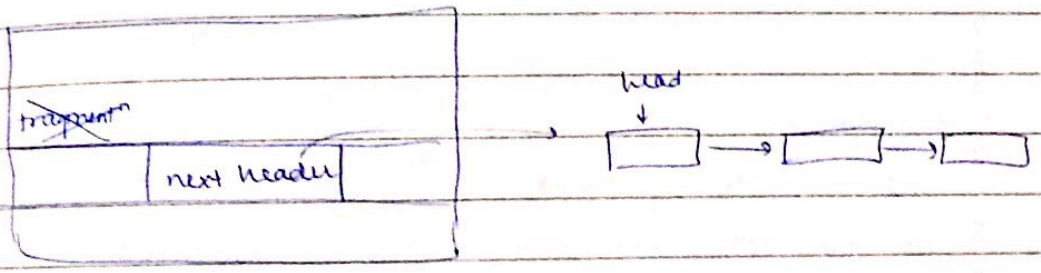
HD video is sent :- extra info (redundant info) is sent ⇒ priority ↓
→ some pkts can be dropped

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

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

↳ hop by hop : word addressable

Teacher's Signature

Wireshark

↓

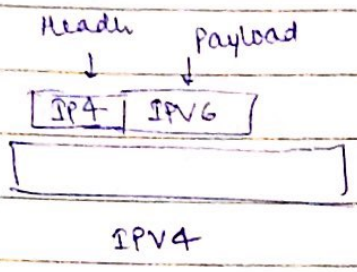
Install it

DATE: / /

PAGE NO.:

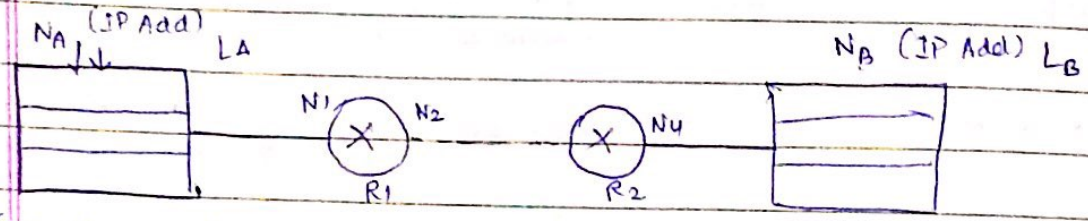
If IPV6 enabled router → it'll avoid next header part

→ Machines have both IPV4 & IPV6 version : dual stack



S D

IPV6 IPV4 IPV6



Client → 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?

DNS is used

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

Teacher's Signature

Command: arp

DATE: / /
PAGE NO.:

NA	NB	Data

NA	NB

DL

How to get Mac. Add. of next stop?

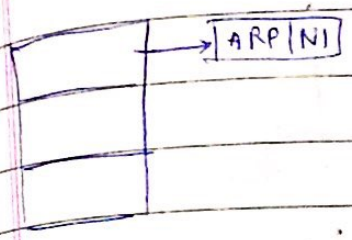
(know IP of next)

IP → ARP : maps IP add. to Mac Add.

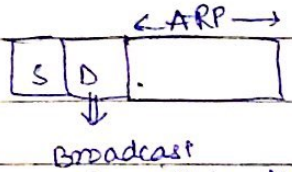
Address Resolⁿ Protocol

at N/W level

ARP : sends request with N1 ~~xxx~~ IP add. → Msg is broadcast to n/w



This msg is converted to frame



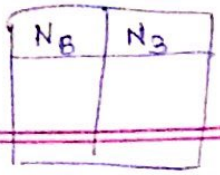
Broadcast add. will be destⁿ add.

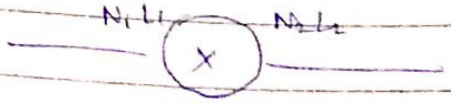
When decapsulated, N1 is recovered Only Router will reply to it (having IP add. N1)

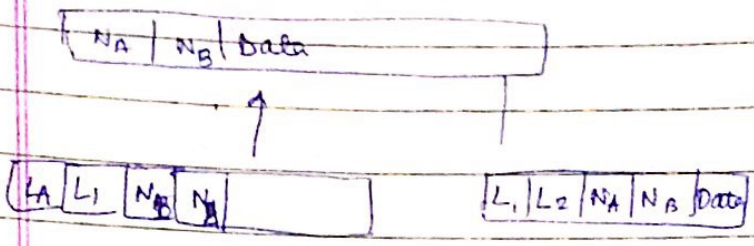
ARP reply pkt will be unicast to source

(Have both N & L of source)

Teacher's Signature

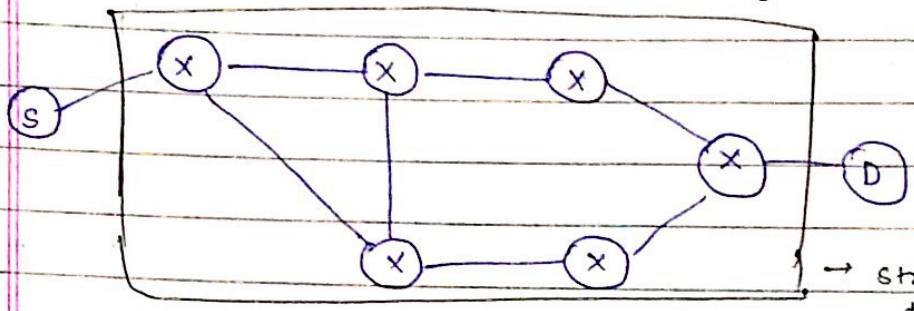






19/3/18

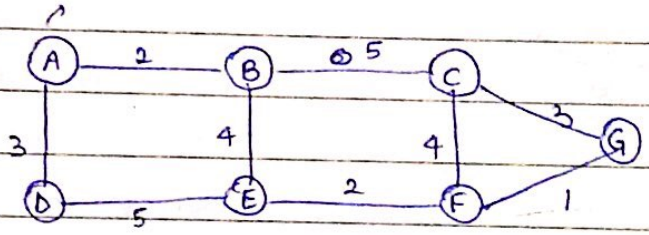
~~Distance Vector Routing~~



→ structure similar to graph

→ Path taken should be least cost path

Default Route of course (ur)



→ At every node, (n-1) least cost path will exist to reach each of other (n-1) nodes.

n routers ⇒ no of paths : $n(n-1)$

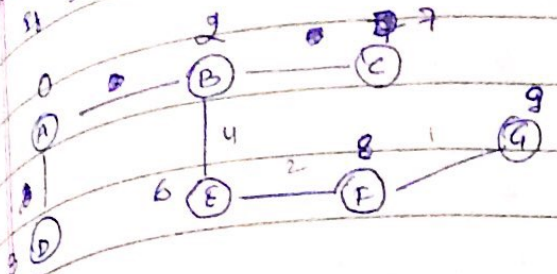
↳ Instead of this, each node maintains least cost tree.

Teacher's Signature.

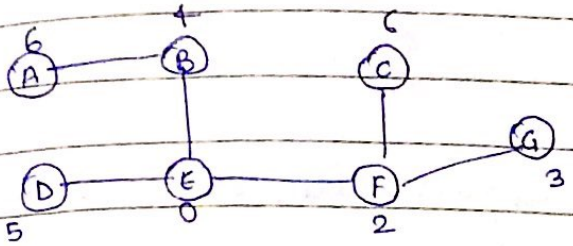
DATE: / /
PAGE NO.:

Node: Represents Router

→ If source : A



→ If source : E



• Routers aren't given complete graph. Every router knows only about its neighbours.

→ Distance vector Routing

→ At A :

A	0
B	2
C	7
D	3
E	6
F	8
G	9

→ At D

A	∞
B	∞
C	∞
D	0
E	∞
F	∞
G	∞

Initially (doesn't know about other nodes)

→

At A

0
2
∞
3
∞
∞
∞

Through messages, it knows about its neighbours

similarly, B received msgs from A, E & C

Teacher's Signature

→ Table for B :

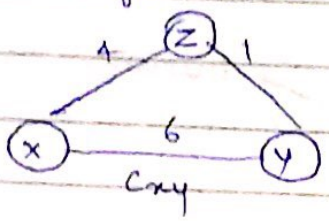
A	2
B	0
C	5
D	∞
E	4
F	∞
G	∞

A gets info from B

New A		
A	0	0, 1 (A → B, then B → A)
B	2	
C	7	∞ ; 2 + 5 (A → C ; A → B, B → C)
D	3	3 ; ∞ (A → D ; A → B, B → D)
E	6	2 + 4 , ∞
F	∞	
G	∞	

→ Whenever any node updates table, it sends its updation to other nodes (about which it knows)

→ New A 'll be formed using Bellman Ford Eqn :



$$\min\left(C_{xy}, C_{xz} + C_{zy}\right) = \min(6, 5) = 5$$

→ Overtime, every node will come to a stable state.

Link State Routing : Uses Djiketra's algo

	A	B	C	D	E	F	G
A	0	2	∞	3	∞	∞	∞
B	2	0	5	∞	4	∞	∞
C							
D							
E							
F							
G							

→ Construct this matrix at every node

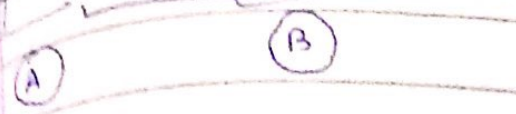
↓

should know structure of entire graph

Teacher's Signature

Table : A knows its distance from A, B & D

	A	2
	B	0
	C	5
	E	4



~~These kinds of~~ pkts are exchanged b/w all nodes,

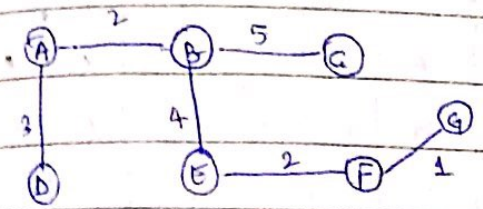
B receives pkt from C & forwards to A

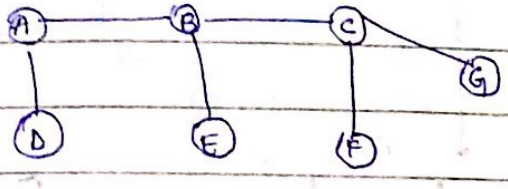
Difference b/w 2 ?

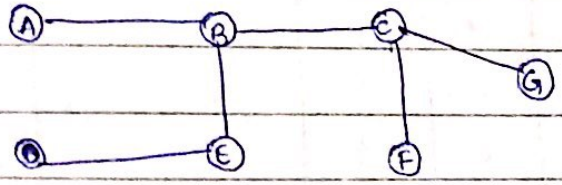
(1) exchanging info. about entire n/w with its neighbour : Distance vector Routing

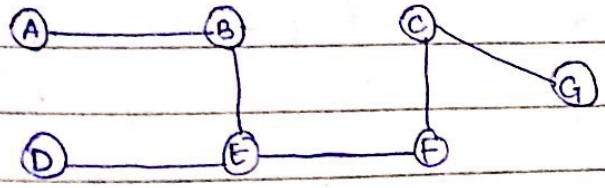
(2) " " about its neighbours with whole n/w : Link state Routing

→ spanning tree with source node as B









Teacher's Signature

21/3/18

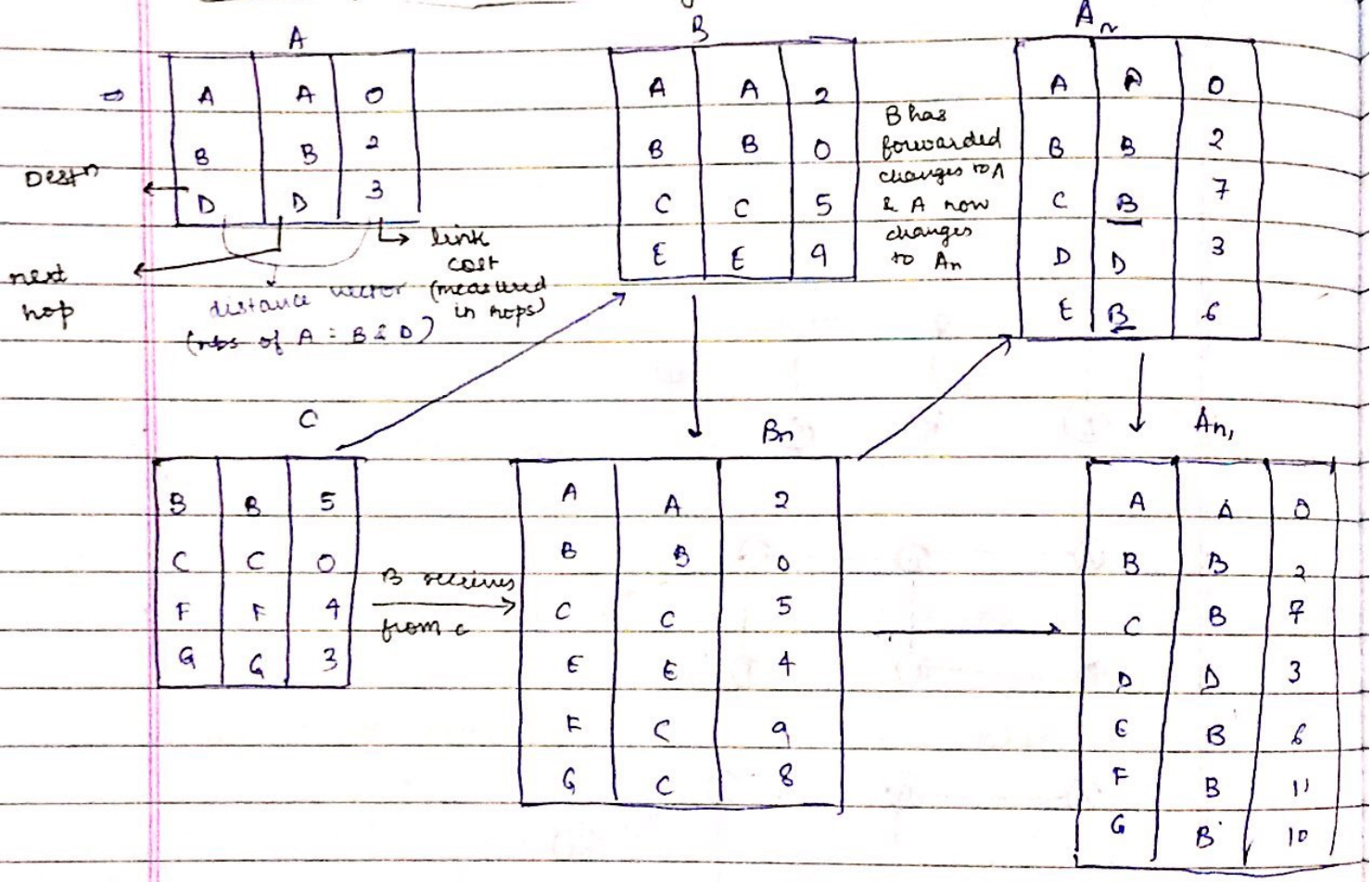
DATE: / /
PAGE NO.:

→ Distance vector at B :

A	2
B	0
C	5
D	6
E	4
F	6
G	7

⇒ knows min cost ~~to~~ to ~~go~~ go to G but doesn't know path to go to G

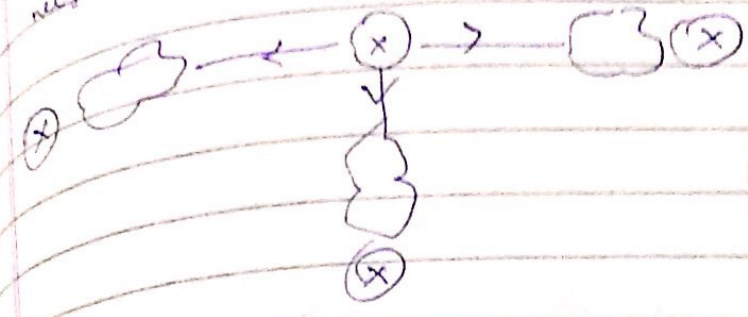
How to form a forwarding table ?



→ An was updated from info received from B ⇒ for any changes 2nd col in An will have value 'B'.

Teacher's Signature

Every node after making table will broadcast it to all other neighbours



→ After C sends info to B

A	A	2
B	B	0
C	C	5
D	E	4
F	C	9
G	C	8

Table is not yet stabilised

→ If we just see An & Bn ⇒ Pkt from A can't be send to G

→ Stabilization : when any update doesn't change the table

→ get update from E (~~has to~~ has D, B, F & G)

⇒ A node only broadcast its own table.

Dest^n	next hop	Distance	Time of last updation
E			5

B

E	C

A

When B broadcasts it, A receives it.

Only current ~~updated~~ entries need to be updated

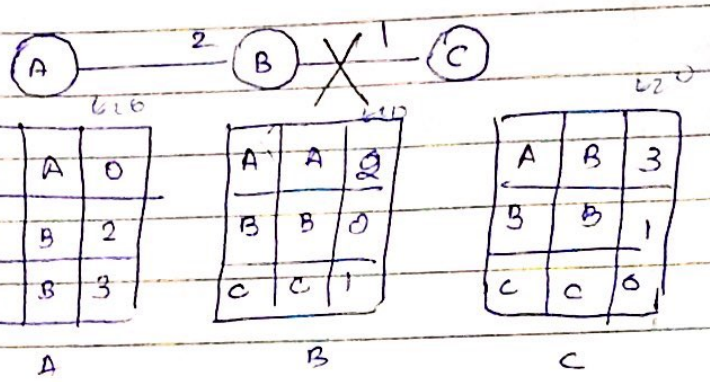
A will ~~update~~ not update E in this case

Teacher's Signature

→ RIP : Routing Internet Protocol

Algo
↓
Protocol
↓
what will be header, pkt, ...
which algo doesn't care
⇓
exact implementation of algo approved by some std. organisations.

⟹



A B C

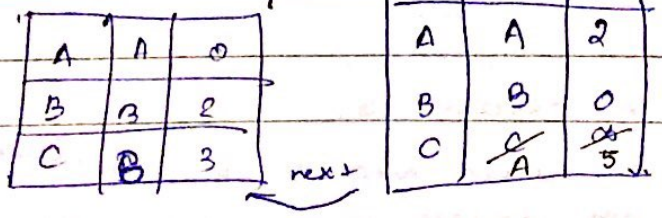
Suppose link b/w B & C breaks ⇒ There is no updates from C.

→ If B don't recieve any info. from C till a time limit
↓
defined in protocol
⇒ it will store dist. from C = ∞

Can't store ∞.

→ RIP : Max^m distance it can cover = 15
If distance > 15 ⇒ it'll treat it as ∞.

NOW,



Before A gets info from B, B gets info from A. B assumes there is a route from A to C, it will update C A 5

Then, B sends update to A, ~~update~~ → A will update

A → then B, ... upto ∞

(because of time, time has more preference than shortest distance)

Count to Infinity Problem

Every node having some dist ∞ should

Problem kyu? A se update aaya. B ne change kiya & again A ko send kr diya

Split Horizon: A se update aaya h, so A ko send back mat karo

But then, B won't send back updates to A → A will treat it as ∞ → Problem

New tech:

Split Horizon with poision reverse. Send C ∞ at A (from B), won't make change at B

E (already has F&G)

E	E	0
B	B	4
D	D	5
F	F	2
C	F	6
G	F	3

A

E		4
G		
E		
F		1

+

B

A	A	2
B	B	0
C	C	5
E	E	4

B

A	A	2
B	B	0
C	C	5
D	E	9
G	E	4
F	E	6
G	E	7

A

A	A	2
B	B	2
D	D	3

Teacher's Signature

DATE : / /
PAGE NO. :

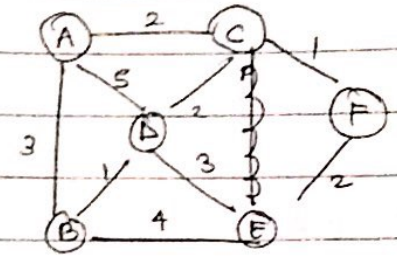
An

A	A	0
B	B	2
C	B_D	7
D	D	3
E	B	6
F	B	8
G	B	9

23/2/18

Link state Routing

→



LSP (B)

B	1
A	
D	
E	

Link State DataBase (LSDB)

	A	B	C	D	E	F
A	0	3	2	5	∞	∞
B	3	0	~~∞~~ 6	~~1~~ 4	~~4~~ 5	∞
C			0			
D				0		
E					0	
F						0

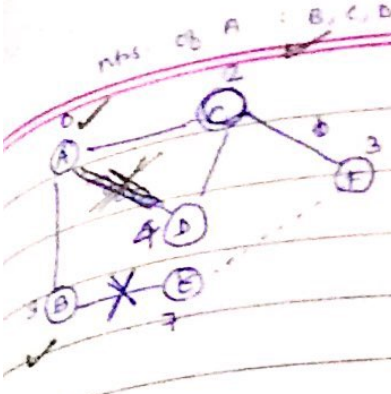
→ LSP is created at each node which helps to form LSDB at each node

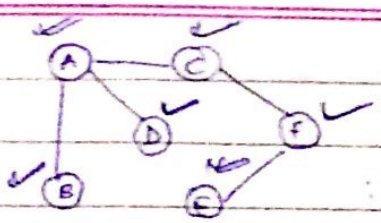
Dijkstra's Algo

Let source : (A)

Teacher's Signature

nbrs of A : B, C, D

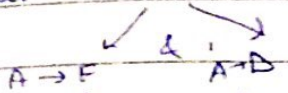




: Shortest path from A to any other

METRICS

First find min : C → try to find from c



↗ A - D

→ Original link will ~~break~~ other will form → c - D

Now, min cost : B & F

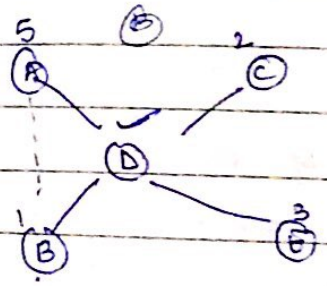
Take B

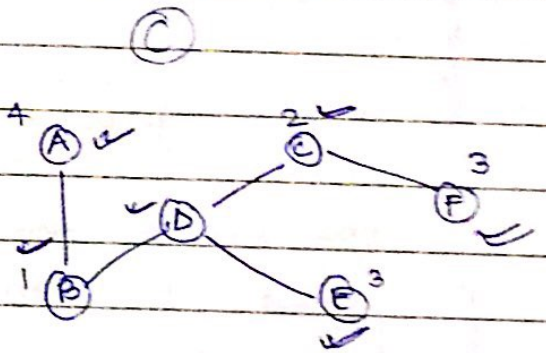
Now, we have F : Consider from F

Now, we have D :

E : ✓

With (D) as source node





How to make forwarding table ? Each node already knows how many nodes are there in total

dest	next hop	cost
A	A	5
B	B	3
C	C	2
D	C	4
E	C	5
F	C	3

Teacher's Signature

Cost depends on TOS (Type of Service)

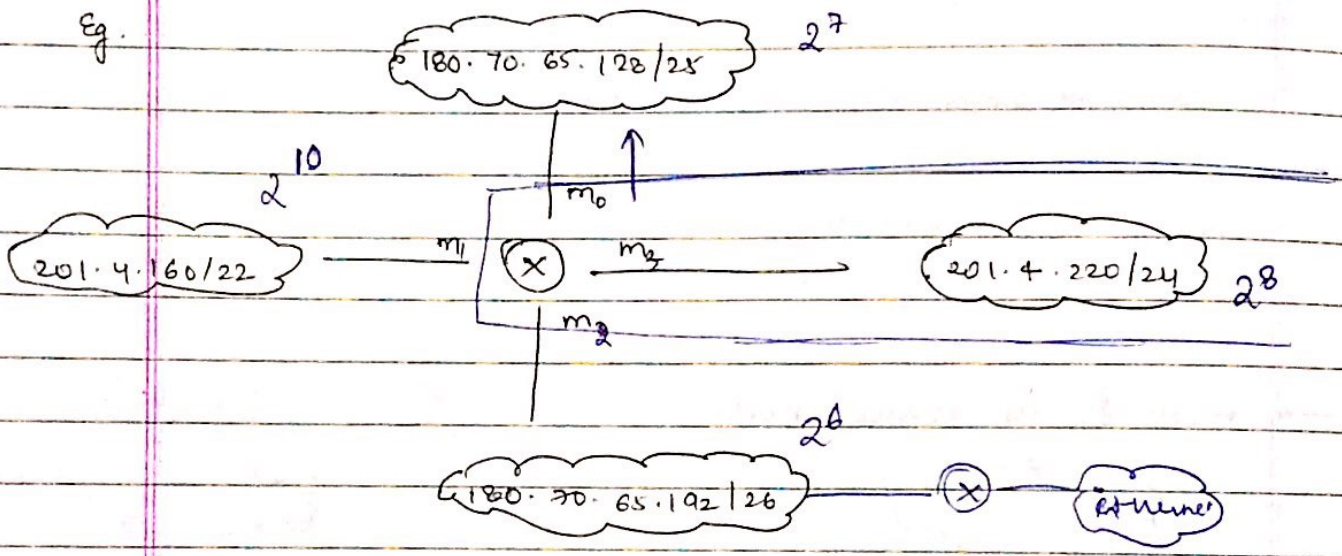
→ In case of Distance Vector :
Hard distance instead of cost.

LSDB : same for every node : Every node should have picture of whole network.

→ In Internet : we use OSPF which uses link Routing
↓
Open Shortest Path First

→ Every node wants info about every other node → lots of msgs need to be passed
→ Matrix will be very large

Eg.



What will be forwarding table of this router ?

	Mask	Dest^n (IP Add)	Nexthop	interface
Take care of Order	/26	180.70.65.192		m_2
	/25	180.70.65.128		m_0
	/24	201.4.220		m_3
	/23	201.4.160		m_1

Teacher's Signature

A pkt with dest^n add : 180.70.65.140 arrives at this router find n/w to which it belongs

Apply .140/26 → No. of add. possible $= 2^{32-26} = 2^6 = 64$

140 : 128 + ~~8 16 32 64~~ 12

180.70.65.10000110

↳ can be changed

1st add : 10000000 : add. of n/w

→ 180.70.65.128 : This doesn't belong to n/w in table (have .192)

Apply .140/25

.10000000

.128 : This belongs ← → pkt will be forwarded to m0

Routing table at Router should contain all add.

⇓

should've $2^{10} + 2^8 + 2^7 + 2^6$

But we just've n/w address (4)

↓

address aggregation

Inside 201.4.220/24 n/w :

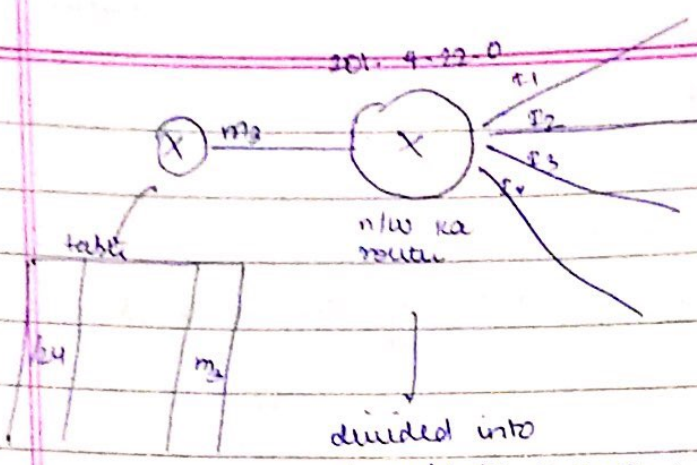
let say pkt has add. : 201.4.22.70

Can directly decide n/w using matching func^n (don't start from starting) since prefix of some n/w are same

Teacher's Signature

netstat → print routing table

DATE : / /
PAGE NO. :



divided into 4 subnets → mask : /26

/24 : — — — 0 0 0 0 0 0 → mask at each subnet : /26

for identifying subnets

64 add. in each subnet

At this router : table :

/26	.0	I1
/26	.64	I2
/26		
/26		

pkt received : 201.4.22.70 → : m3 pe bheja

↓

forward to I_2 ← ab masking lagegi /26 ki

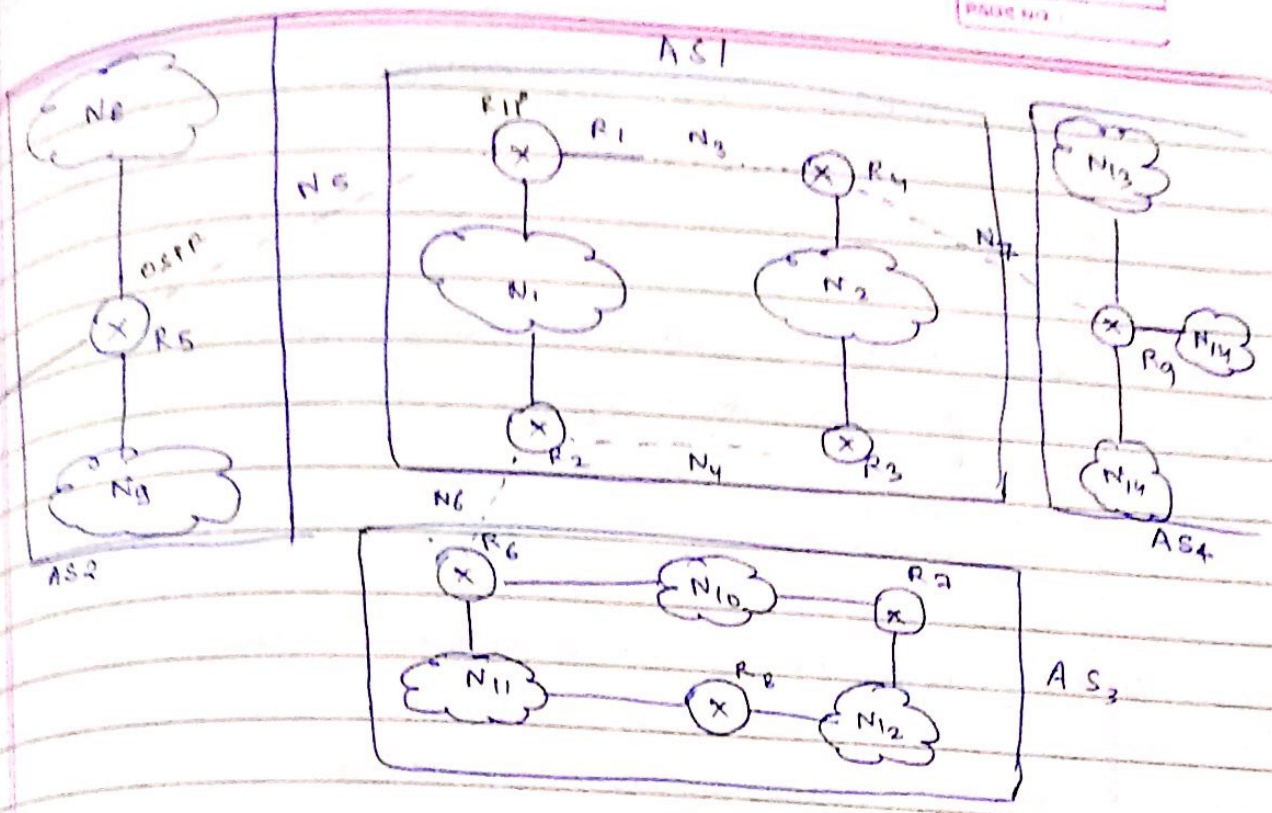
⇒ Distance vector isn't very popular (max. distance : 15 only)

→ if pkt : .0 → add. of Router also & subnet also has add.

last add : ~~forward~~ broadcast add.

Teacher's Signature

In practical n/w : link state Protocol is used



AS : Autonomous Systems

In OSPF : every router must have complete info about all routers → used matrix.

R_5 : all entries for N_8, N_9, N_1, N_2, N_3, N_4 ...

- very difficult to find path. ↱ only if hops
- Though OSPF is more scalable than RIP, matrix gets big → algo will take lots of time.
- We need another method to minimize the matrix.

→ AS : 1 kind of service provider (like Airtel, Aircel, etc.)
Within AS1 ⇒ can have OSPF / RIP.

→ For intercommunication (AS1 - AS2) : must have same protocol.

→ Autonomy : controlled by 1 ϕ service provider
have policies : include intradomain protocol
(like OSPF & RIP)

Teacher's Signature

→ BGP : Border Gateaway Protocol → don't have count to ∞, looping problem

eBGP iBGP

Border Routers which exchange info with other A.S. (diagram)

↓ Eg R_5, R_4, R_9

follow eBGP

→ Every Border Router has 2 protocols atleast

Intradomain eBGP

→ Every Internal Router has atleast 1 protocol → Intradomain

→ R_5 will exchange info with R1 : Point-to Point conn" ; also a n/w

R1 knows :

N_8, N_9	R_5	AS_2

N_8, N_9 → dest" ; R_5 → go through this router ; AS_2 → through R_5, you can reach to all networks in AS_2.

* Every AS no is unique
  ↳ could be private / public

R_5 has table

N_1, N_2, N_3, N_4	R_1	AS_1

→ Every router has this info using Intradomain Protocol

→ Whatever BGP follows, info will be sent in that format.

Teacher's Signature

→ One A.S. is either a source or sink → Stub A.S.
→ Other one is also a source or sink but it can allow traffic to pass through it → Transient A.S.

AS1 : Transient AS2 : Stub

→ By default, any AS with just 1 router is always stub A.S.
→ Stub n/w are connected through Transient n/w
Stub
Local ISP
National ISP

→ All routers within an AS must also exchange info b/w them → i BGP needed (Internal Border Gateway Protocol)
↳ Intra Domain not considered in this case because it works at micro level. Now, we need to talk about AS level.

→ Border Router — eBGP — Internal Router
IDP
IDP
iBGP
iBGP
won't be needed in case of AS having single router

→ F.T. is created just like in distance vector

→ We don't talk about shortest path in BGP. Just talk about reachability (spanning tree) some policies have to be met.

→ R0 NB → NI2

Path : R5 — R1 — R2 — R6 — R7 — NI2

Suppose R2 isn't able to forward pkt
↳ maybe, congestion
↳ forwarding table doesn't have entry) → It'll drop the pkt

then, atleast NB should be informed ⇒ ICMP (Internet Control Msg. Protocol)

hops

TTL = 2 ⇒ # pass to 2 routers only.

TTL = 0 ⇒ drop (time exceeded) & send msg back to source

DATE: / /
PAGE NO:

→ ICMP : for error

IP	ICMP
IP	

Only IP pkts are sent.

R_2 will create an ICMP msg & send back to N_8 that dest^n was unreachable

How will it know the source add.? Using src add. in IP msg.

→ Common routines using ICMP are : Ping and Traceroute

① Ping : echo request & reply : have Types

ICMP msg is sent in echo request form

② Traceroute :

For same eg.

~~If TTL = 0~~ ⇒ we will send 1st ICMP with TTL value = 1.

⇒ reaches ⇒ last router TTL = 0 ⇒ sends back msg. ⇒ get add

Next, send with TTL = 2 & so on.

→ R_7 won't send back layer ⇒ goes to ~~Port~~ N/W layer

→ If reached R_7 : don't send reply ⇒ specify wrong Port No.

↓

again, ICMP will be sent.

Msg sent :- 'Time exceeded'

Eg: IP addres : 245.248.128.0/20 → 12 bits can be manipulated

↓

CRDC scheme.

Divide blw 2 org^{n}s. 1st → half of add.

2nd → 1/4th of add.

What should be masking used ?

Teacher's Signature

27

DATE : / /
PAGE NO :

Total addresses possible : 2^{20}

245. 248. 136. 0 / 21 & 245. 248. 128. 0 / 22

245. 248. 128. 0 / 21 & 245. 248. 128. 0 / 22

1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 | 22 → 10

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

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

128 + 8 = 136

10

Teacher's Signature

2/9/18

Transport Layer

→ why use 2 diff. add. for single msg? (data comm^n)
↓ (MAC Add & IP Add)

∵ certain nodes in path of src to dest^n

Link Layer add : written of next hop

N/w " " : " " final dest^n

At Router : DLL will see if Dest^n MAC Add is same as its own —Yes→ accept it, pass to N/w layer

↳ During whole comm^n : IP add is not changing
: Mac add. is changing at each link

→ Router : Why to have IP Add. at router since IP add. of only src & dest^n are used ?

↳ Broadcasting : Router can act as src/dest^n in that case ⇒ need to have an IP add

have different N & L add. : Inside n/w, each node has to be identified uniquely
Each is able to identify router uniquely in a n/w.

Router : have 1 protocol at n/w acting on a pkt.
at DLL : may have multiple protocols
⇓
can have diff. protocols working at DLL ⇒ have 2 diff. boxes.

→ In DLL & n/w layer : all nodes in b/w will deal with msg

→ In Xport layer : only src & dest^n node will deal with msg, not other nodes.

Teacher's Signature

OSI model : theoretically exists
TCP/IP model : reality me exist karta h.

DATE : / /
PAGE NO. :

→ when pkt is received, within device, which process gets it : done by Xport layer (appl^n)

→ How appl^n takes care of pkt : Appl^n layer

Transport Protocols

1. Run on end hosts
2. multiple Xport protocol available.
   TCP & UDP ← (Primary Protocols)

3. N/W layer :
   for transmission : IP
   support protocol : ICMP : for smooth running of n/w

X Port layer Services

1) process - to - Process comm^n :
2) Addressing : Port Numbers (16 bits)
   (pid src / dest) ↳ used to identify processes
   ↓ why not use pid we get from OS which is unique
   It is unique in that device only → locally unique
   But X port layer works on both devices
   ↓
   can only use globally unique id
   ← Hence
   Xport layer uses its own id / add ↓ Port No.

→ For specific services : specific port no are reserved (well-known) world wide.

→ Daytime server : provides current time & date. : have reserved port no 13.
   ⇒ pkt me set dest^n port no 13
   src port no. ⇒ can choose port no. which suits you.

Teacher's Signature

ICANN : reserves port no.

well known

0	1024	49151	49152	65535
(mail, google)	Registered (fb)		Private (want to fetch something from google using a code)	

4/9/18

3) Encapsulation and Decapsulation

4) Multiplexing and Demultiplexing : N/W layer ⇒ No multiplexing / demultiplexing
S → receive from upper layer (only src)
R → receives " lower " (only dest)

→ All protocols of layers are supported by OS → using func" call.
(like UDP)

Sender: Xport layer gets UDP's from 2 diff. processes (in App" layer)

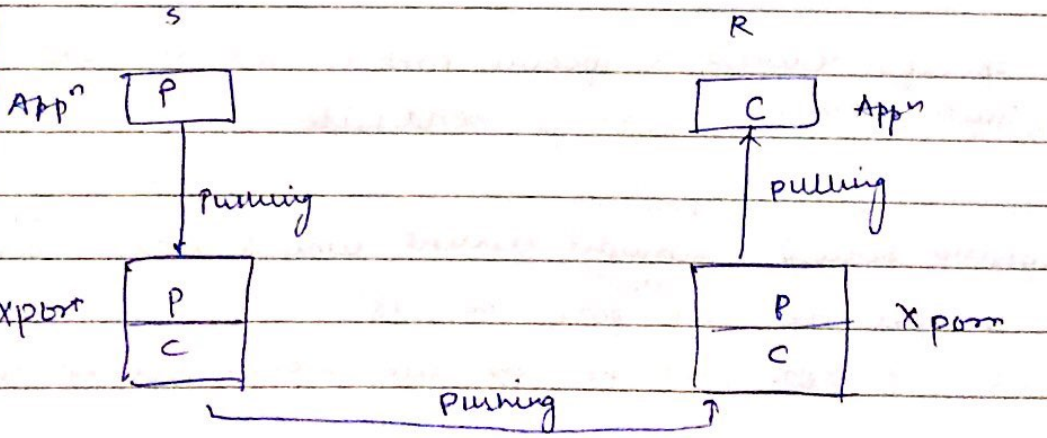
Receiver " receives from n/w layer only, deassemble it & send to suitable process

5) Flow Control : (Also in DLL) → only concerned b/w S & R.

Prod" ↑, Consumption ↓ → mismatch : need control mechanism

Pushing
Producer pushes whatever it is generating

Pulling : don't need flow control
Consumer requests for pkt & then only producer sends.



Teacher's Signature

* Same Consumer - Producer problem as OS.

6) Error Control :

- detect & discard corrupted pkt, duplicate
- keeping track of lost & discarded pkts. & resending them
- Recognizing duplicate pkts & discarding them
- Buffering out of order pkts until missing pkts arrive.

if msg is of small length & can fit within 1 pkt
⇒ pkt can't arrive out-of-order. (DON't have to look to 3 & 4 point)

if msg is divided in multiple pkts : 3 & 4 point apply

→ UDP : don't support error control

TCP : supports error control

→ why flow & Error ctrl at Xport layer?
(XYZ) (X, Y, Z)

DLL will divide pkt to frame & send at receiver side, DLL will assemble X, Y, Z correctly. But to reassemble Xport layer will take care of XYZ

↳ Sequence No.) → for error control
↳ Acknowledgement No.

7) Congestion Control

- Congestion : takes into care of whole path unlike Flow Ctrl. whatever path can handle, send only that much.
- TCP : ensure Congestion control
  UDP : don't " " "

Teacher's Signature

→ Connectionless & Connection oriented

↓ UDP — ↓ TCP/IP

N/w layer defn: Message divided into M_1, M_2, M_3 at this layer

- if all M_1, M_2, M_3 take same path
- free to take any path

Xport layer defn:
- if all datagrams are independent of each other
- have dependency on each other ↓ pkts are sent in-order to Appⁿ layer

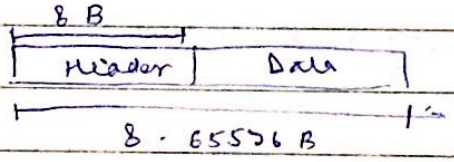
At R: pkts are sent directly to Appⁿ layer without bothering about order

User Datagram Protocol (UDP) :

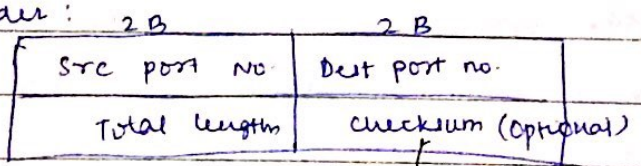
→ Connectionless, unreliable TL Protocol
↳ don't support error control
(if using checksum, may find corrupted pkt)

Although unreliable, UDP fits best in some cases

→ User Datagram packet format :

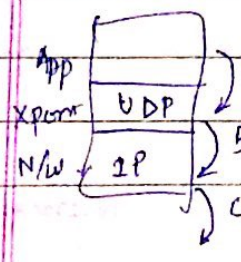


Header :



↳ to check if pkt is corrupted or not

Q.



A ⇒ what is max. size of A ?

Teacher's Signature

→ Header of IP is used in checksum

DATE: / /
PAGE NO.:

→ Don't support flow control, Error & Congestion control

IPV6 → checksum

Slide 24 : Header is given

1) Src Port No : C B B 4 (16 bits) (Private)

2) Dest Port No : 0 0 0 D → 13 : well known

3) Total len = 6 0 1 C → 16 + 12 = 28

4) len of data = 28 - 8 = 20 B

5) checksum = 0 0 1 C

⇓

Both Header & Data are taken into account & 1 more thing

+

Pseudo header → multiple of 4 B

⇓

have components which are subset of IP header

Pseudo header : has src IP, dest IP at R

↳ If N/w layer checks using checksum & finds it correct, it'll send to X port layer.

↳ why to include src IP / Destⁿ IP / Protocol for checksum calcⁿ in UDP?

for addⁿal security :-

a malicious user in b/w can change src IP & dest IP

⇓

N/w layer will find it correct but X port layer will find fault.

Some protocols are capable of working on all UDP, TCP, SCTP at X port layer.

→ Port No. is valid for 3 diff categories.

→ X port layer has to verify the category. (send that only which is asked by upper layer)

Eg. →

UDP

H	D
8	3

Src. IP: 32 bit put everything in 16 bit format (checksum): 16 bits)

do for every thing (Pseudo header)

→ Padding at end

0 0 0 0 0 1

1st 2 bytes as it is | 3rd byte | 4th byte

← 16 → ← 16 →

Teacher's Signature

→ If value of checksum : 00 ... 0

⇓

sender hasn't included.

→ If sum comes out to be all 0's : take complement & get all 1's.

∴ For checksum : sum all + Overflow then find compliment

↳ Why to use UDP ?

↳ Burden of n/w ↓

↳ Time ~~to send for~~ duration of communication ↓.

→ for reliability → may send several times.

(∵ send what is IP add. of google)

{ can send everything is one datagram }

↳ If connection oriented : have to send several datagrams asking for availability & termination.

Real time : reliability not imp. → Prefer UDP

(watching due match at current time is imp. If I miss something in b/w : not imp. Just don't want to lag behind)

↳ If downloading a book : I can't get entire book if ~~do~~ some datagrams are lost / out of order.

⇓

reliability is imp : Prefer TCP

TCP

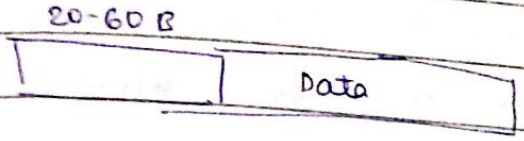
Transmission Control Protocol

- connection oriented
- assumes delivery as stream of bytes
- reliable, in order delivery
  - ↳ supports error control
- flow control
- Congestion control

TCP Services

- Full Duplex Commn : Both can send data at same time with same conn

- TCP segment :



- Header :

  HLEN : 4 bits

  → Divide header length/4 to get 4.

  min. value HLEN : ≥ 5 [5 × 4 = 20]

- Control bits : 6 flags
- Checksum : mandatory

Checksum : Header + Data + Pseudo header

Teacher's Signature

→ At R: usually have well known / reserved port
↓
Server will pull data from Xport layer & if it gets

Passive open : It's listening
Active : when it'll establish connⁿ

S : Active open : Open for connⁿ
For connecⁿ establishment : send pkt with
sync flag (S) = 1 , and a pkt no.
↓ ready to establish connⁿ
↓ It is not numbering the pkt but numbering the Bytes
Initial seq no.
(Random value b/w : $0 - 2^{32}-1$)

Q. Why this flexibility to randomly select?
Why not always start from 0?

Connection Establishment : "Three way hand-shaking"
When received at other side, if it is ready to establish connⁿ, it will send back ack with ack no. = 8000+1. Seq no. for this will again be a random no. Flag : S + A
↓ opening connⁿ from your side (want to establish connⁿ)
↓ acknowledging

S → received sync → send ack. to R.
for this pkt : ack = 15001
seq no = 8001 { Offset / starting add. = 8000 }
↓
Because even though 1st packet was empty, it sent S → needs atleast 1 B
→ 8000 + 1
(S = 0 , A = 1)

This pkt need not be acknowledged back. Conn is open now at both side

Teacher's Signature

SYN bit = 1 ⇒ can't have any data

Window size

R: rwnd : → 500 → 500 size of buffer
→ atmost 500 B of data can be sent to R

S: rwnd : 10000 ⇒ can send atmost 10000 B of data to S.

→ Both S & R have buffers (2 pairs at each side) → buffer size : ltd.
↳ for flow control.

→ Buffer at R will get vacant only when app^n layer puts

	A	B
(1)	S	R
(2)	R	S

— 2 pairs of buffers

↳ The connection Establishment phase can be violated : SYN flooding attack

→ SYN flooding attack:
malicious user can send SYN pkt to server. Server will reserve a buffer (mem.) to that user. But if it doesn't come :→ I'm expecting ACK back from (3rd datagram)
Collec^n of malicious user send SYN coordinately & never send ACK back. Also, they'll write src IP as somebody else. So, 2nd datagram will go to some other node. And whoever receives that will discard it.
→ All resources of server will be blocked & it won't be able to take care of legitimate user. (gets SYN request in small duration of time) → denial of service attack.

Data Transfer :
→ After 3rd datagram transfer : client —data→ server ⇒ can send 1 pkt only with both Ack. & Data.
can use cumulative (Go back N) + selective ack (selective repeat)

Teacher's Signature

* Jab bhi ack ki zarorat h → ack. no. increments.

* Ack. no. = pkt no. of next pkt.

* In TCP : everything that needs an ack. consumes a byte.

Slide 33 : If no 3rd : don't have data part

4th : ack = 15001 (remains same because ack doesn't req uires ack no 3. so, won't increment)

Connection Termination : (3 way hand-shaking)

→ Client : put F = 1 (Fisrd bit)

closing conn' from your side

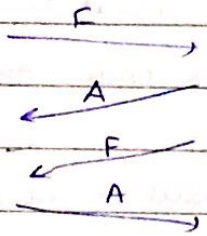
→ Other party will acknowledge it (F + A)

just like s bit

* There may be 4 way hand-shaking also- :- Half close :

↳ Closing means : can't send data, but can receive it

If Server has something to send still, it won't send with F=1, it'll just send ACK, then send all data & then send with F = 1.



Windows in TCP

S → Right wall shrinks / opens : depends on rwnd of R.

R → Window size ≤ Buffer size

some bytes exist here which aren't pulled by Appn Layer.

Teacher's Signature