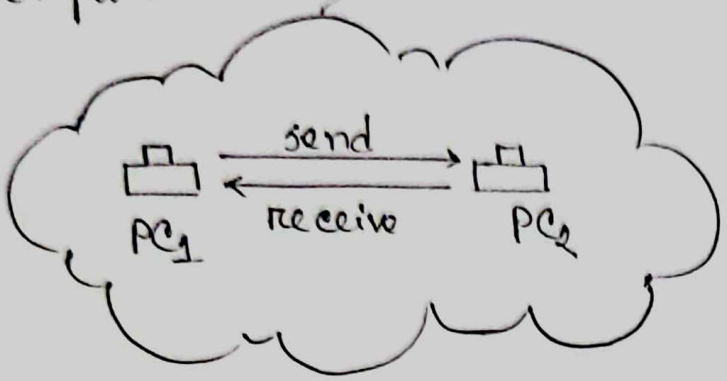
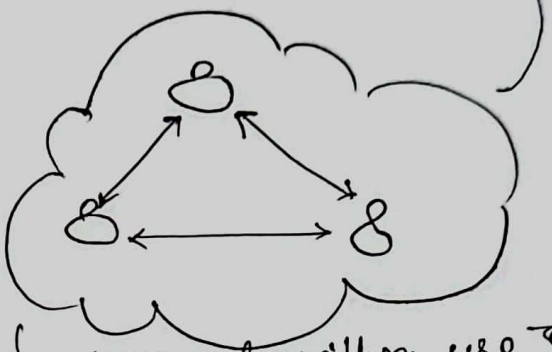
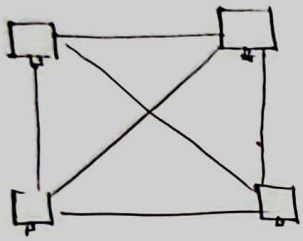
Computer Network



ARPANET
—agen advanced research
. . . . .





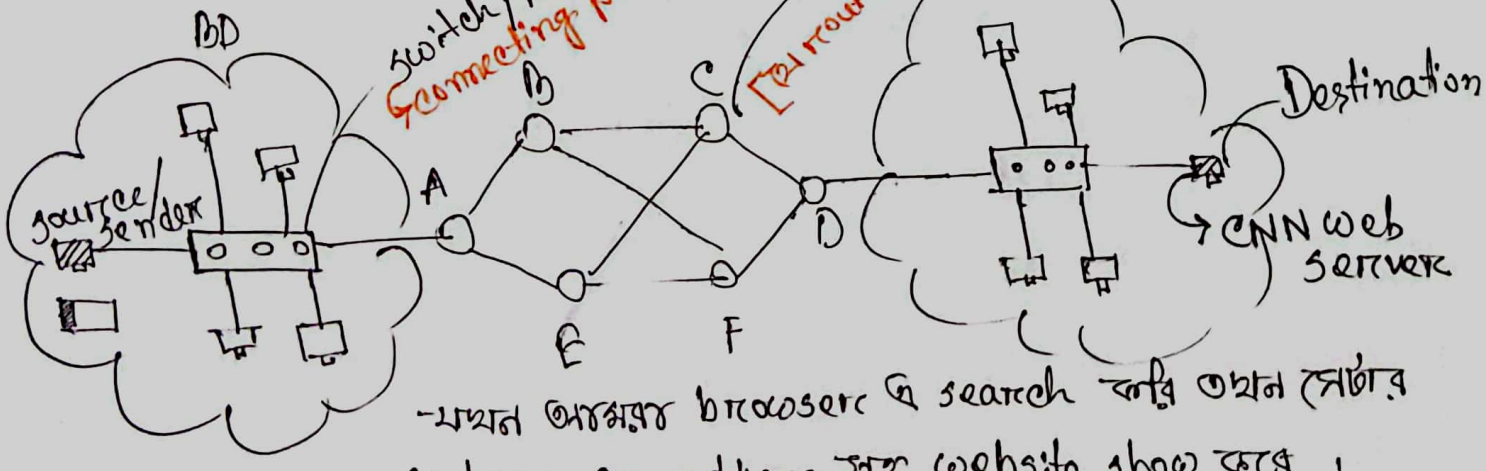
যদি সবাই same algorithm use করত তাহলে communicate করতে পারত। এভাবে তৈরী হয় internet

standard → OSI Ref. Model

আরেকটা system এর সাথে communicate করতে পারবে (open system interconnection reference model)

—The system is opened to others

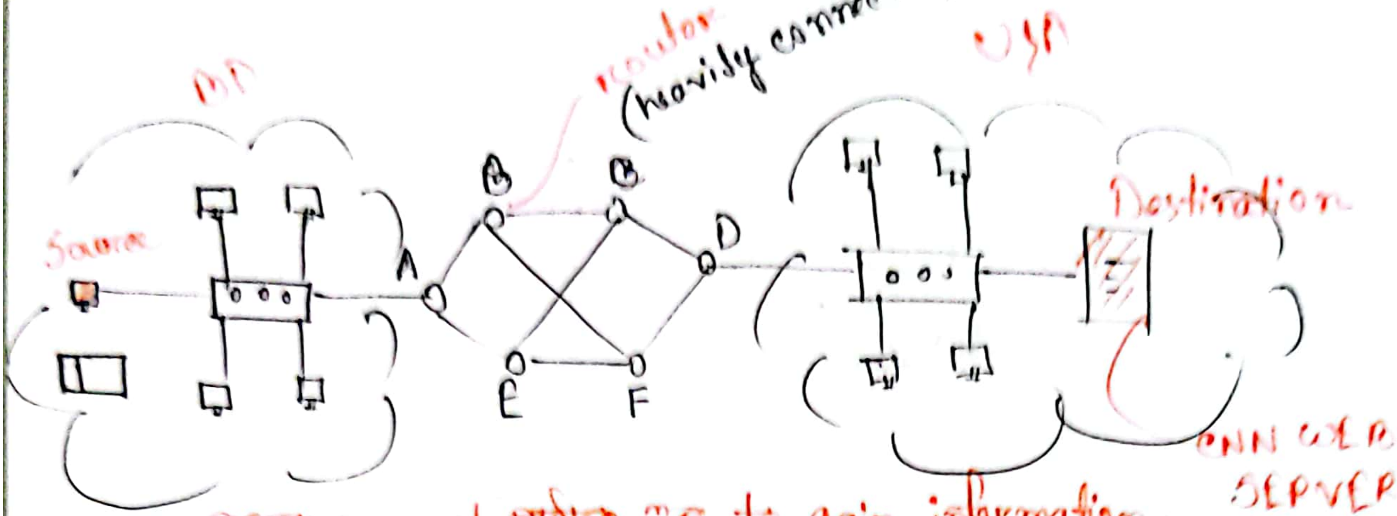
router গুলো heavy.... connection create করে



—যখন আমরা browser এ search করি তখন সেটার webpage এর address সহ website show করে।

আনেকগুলো connection কে এক point এ connect করে যেটা switch; It works like a multiplug

5-02-24



- প্রথমে request পাঠাতে হবে to gain information; তারপর request কে generate করে তারপর send করতে হবে।

- router carries my information

Standard — ISO এর

OSI Ref. model

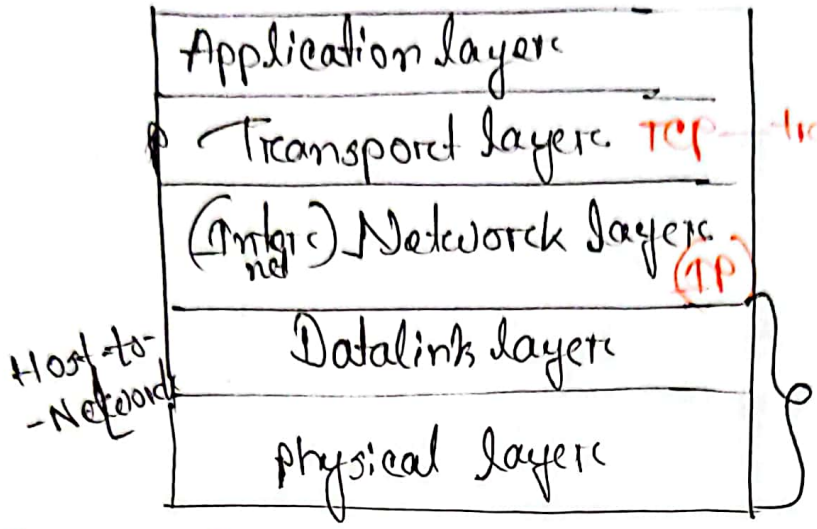
(layers & layers)

- Application layer ← software; opposite of physical layer → abstract layers
- Presentation layer → interpret, present করে
- Session layer → management
- (Mother layer) Transport layer → source থেকে generate হওয়া থেকে destination পর্যন্ত supervise করে
- Network layer → just to show the path (Path finding device)
- Datalink layer ← এটা adjacent device এর মধ্যে error free data পাঠায়
- Physical layer → all the hardwares are defined

*datalink ensures error free network (communication) to adjacent devices

* একটা device multiple layer এর কাজ করতে পারে

TCP/IP Ref. model — smaller/derivative form of OSI



Down Encapsulation

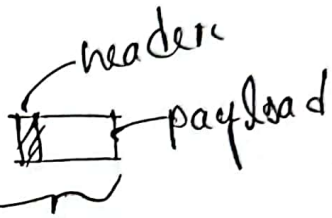
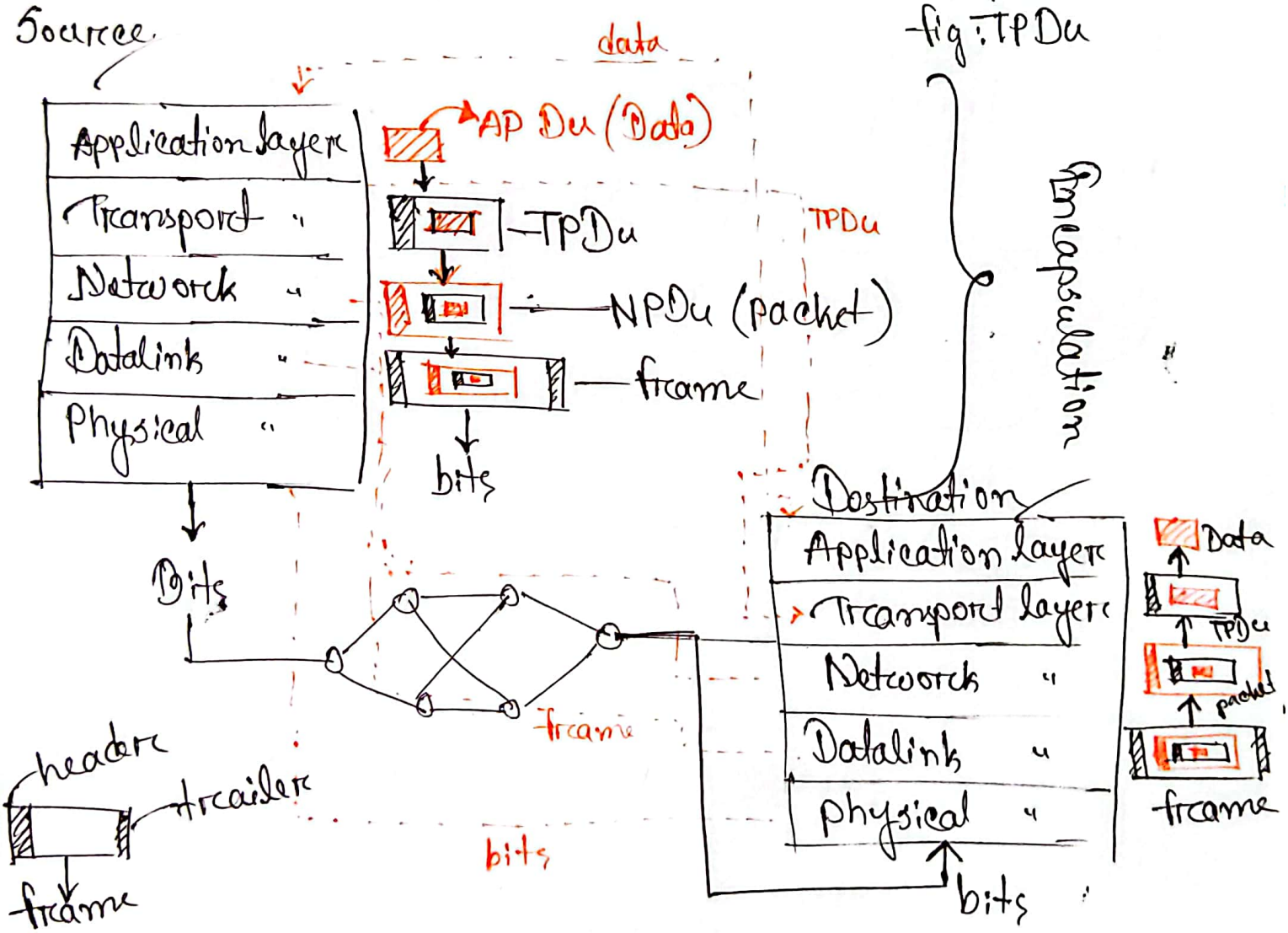
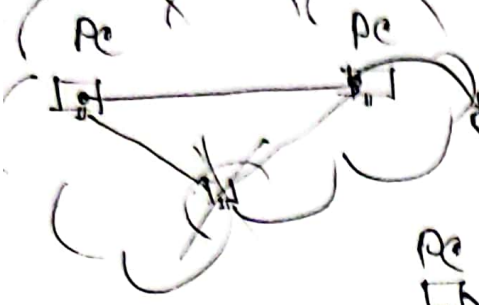


fig: TPDu

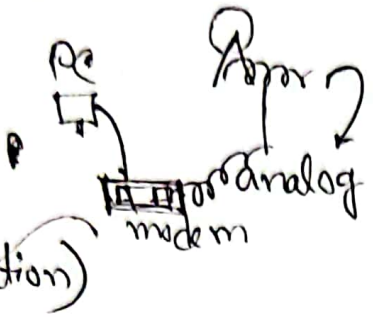


Lab-1

Computer network



ethernet cord



(demodulation)

→ Computing Devices (PC)

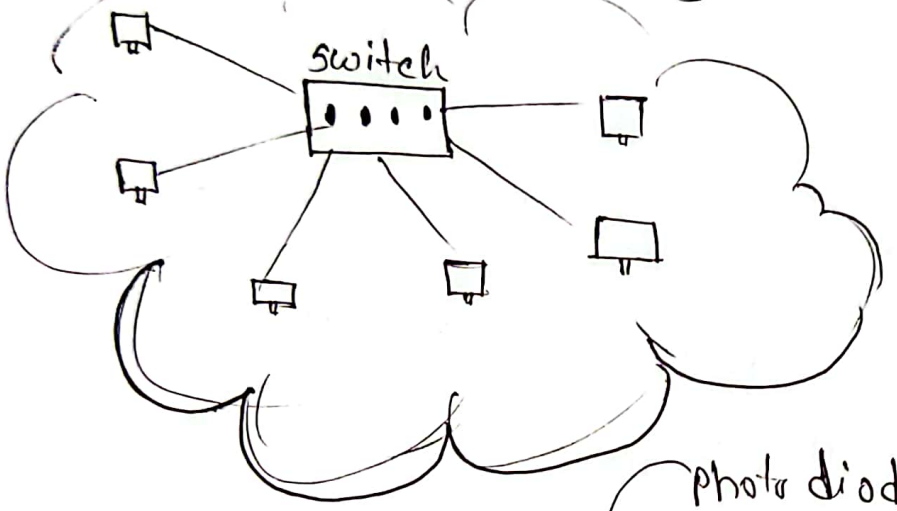
→ (NIC network interface card)

→ Transmission medium

a) unguided (wireless) ✗

b) Guided
- Fiber Optic
- Coaxial (Dish TV signal)
- Twisted pair cable

multiple PC connection এর জন্য switch use করতে হয়,



a) CAT5 (সাধারণ)
b) CAT6 (professional)
c)

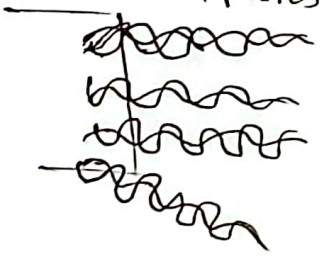
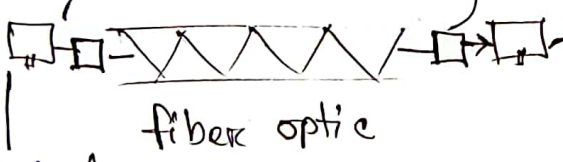
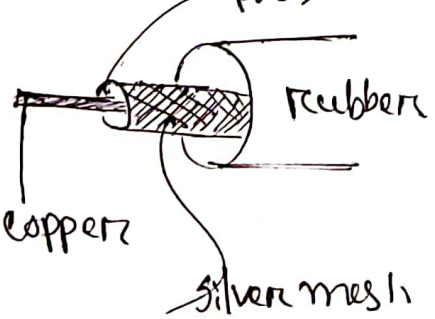


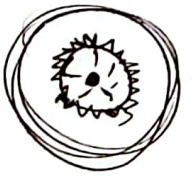
photo diode

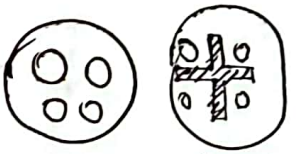


electrical signal

electrical signal



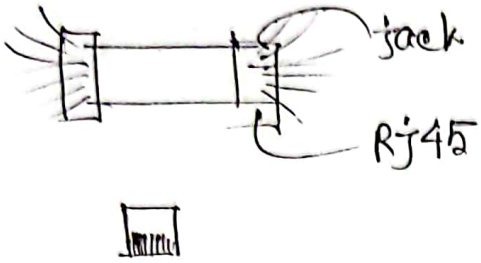




per inch-এ twist
বেশি CAT6 এ

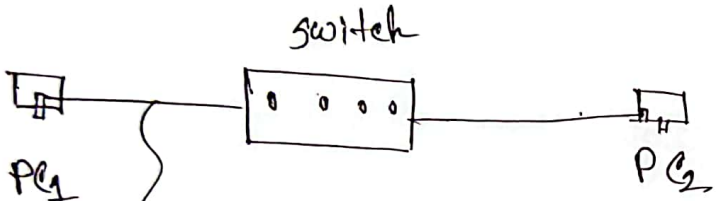
Twisted

- 4 solid colors [Green, blue, orange, brown]
- 4 Mixed with white [white - Brown
  white - Green
  white - Blue
  white - Orange]



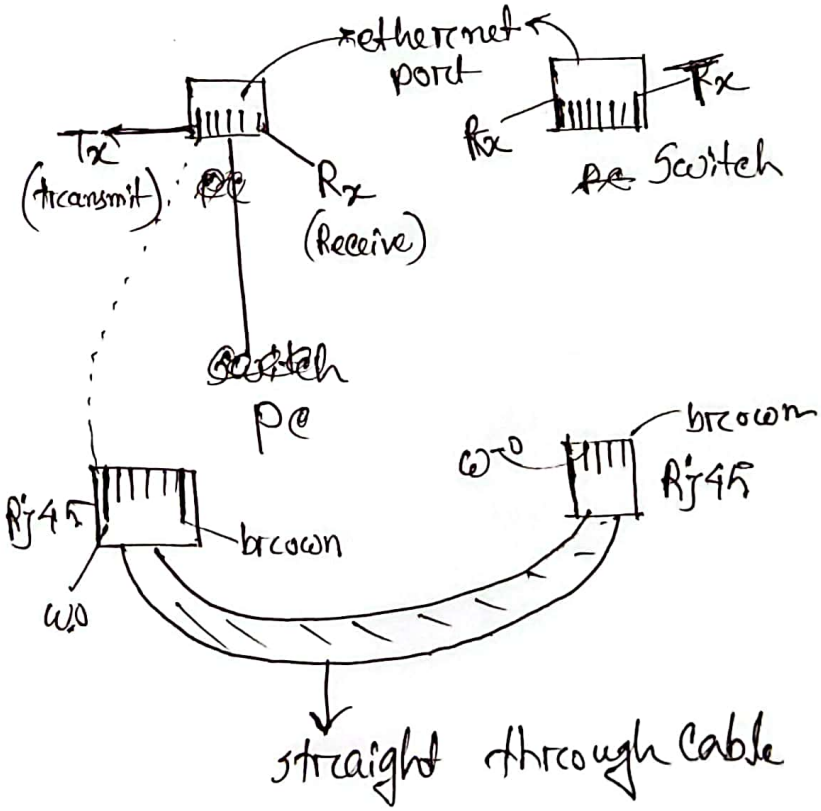
Next lab
CAT5 — 2m
Rj45 — 5/6 pcs

Crimping tool



straight Through Cable

fig: 568 'B' standard



Lijenta

CamScanner

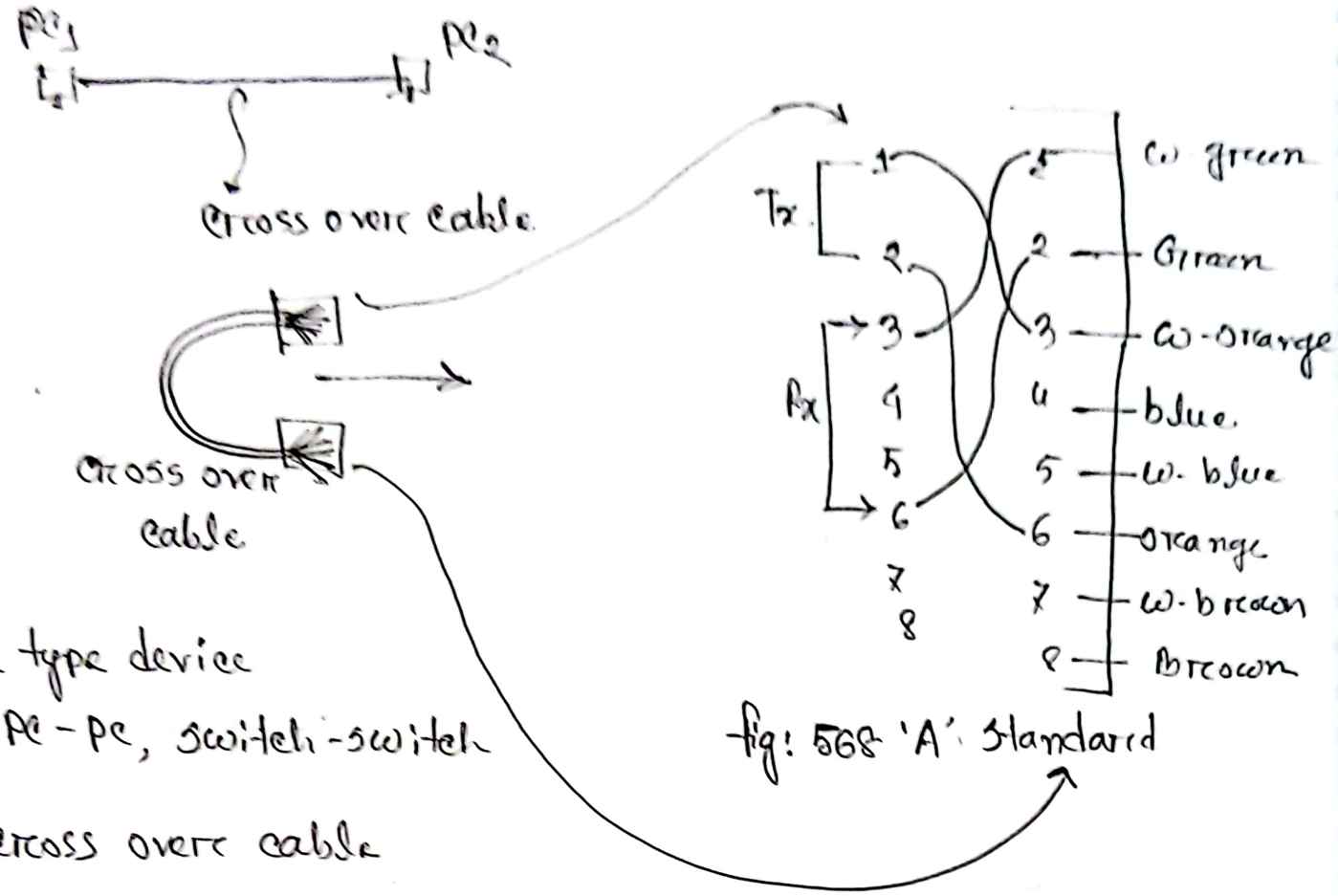
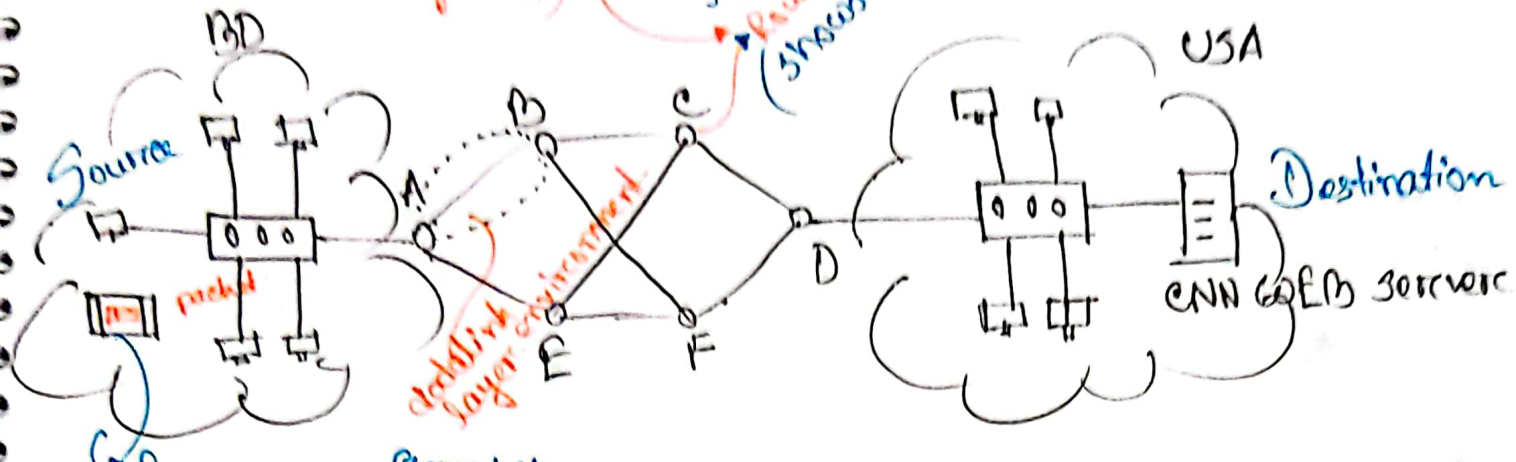


fig: 568 'A' standard

Same type device
PC - PC, switch - switch
→ cross over cable

Diff type of device — straight through cable



এই frame এর মধ্যে bit আকারে data থাকবে প্রথমে nex device (switch → router ... → destination)

Router define করার হয়েছে → physical layer এ
কাজ করছে → Network layer এর মত
algorithm যেখানে থাকে → application layer এ

Packet - এই packet এর header টার মধ্যে থাকে destination এর address

Destination address (IP address)

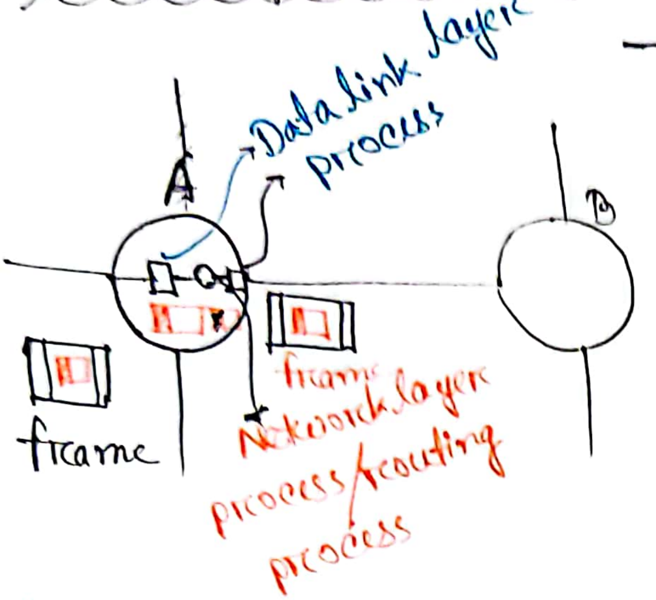
frame যখন A এর কাছে যাবে, A কে জানতে হবে frame এর address-টি কি? address জানলেই A কোথায় তার algorithm কোন route দেখাবে। Packet এর header read করতে পারে Network layer. এ packet device to device travel করার পর destination-এ পৌঁছাবে। এখানে কোন route/ link-এ দিক দিয়ে যাবে or depend করবে packet এর এই destination এর address এর উপর A থেকে B তে যে environment create হবে সেটির data link layer environment.

Datalink layer environment :—

→ frame টা পাঠিয়েছে datalink layer

→ frame কে আগে accept করবে A; অতএব কাজেতে হবে packet এর ভিতর যে process আছে সেটা datalink layer process accept করে। এখন header and tailer সব ফিল্টার করে তারপর routing process করবে।

→ Network layer process ই পারবে packet এর header কে read করতে। header - এর ভিতর আছে destination address.



[Process is an executing programme]

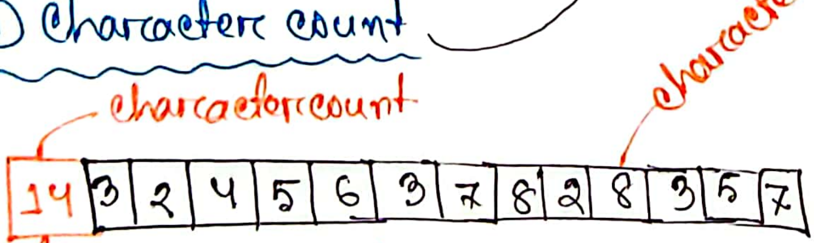
* frame A থেকে B তে যাওয়ার সময় address প্রত্যেক বার change হচ্ছে।

Framing techniques

(i) character count
(ii) Byte stuffing
(iii) Bit stuffing
(iv) Physical layer coding violation

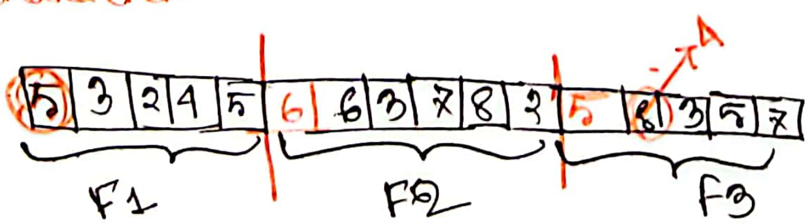
(i) Character count

→ এই technique follow করলে এই frame এর total 14 টা character read করে next router এ যাবে।



total count + 1 character
= 13+1
= 14

error কিভাবে solve হয়?



Ethernet → creates LAN

→ frame: 1460 Byte (max.)

ATM Network (asynchronous ...

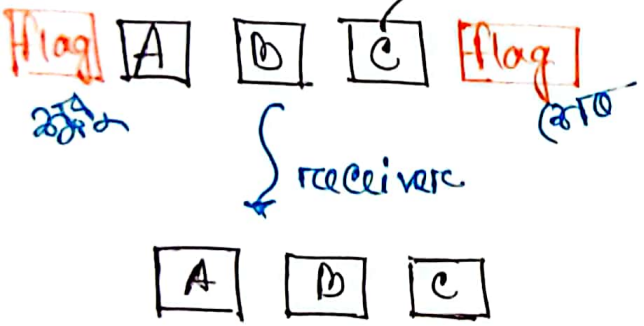
backbone এ use করা

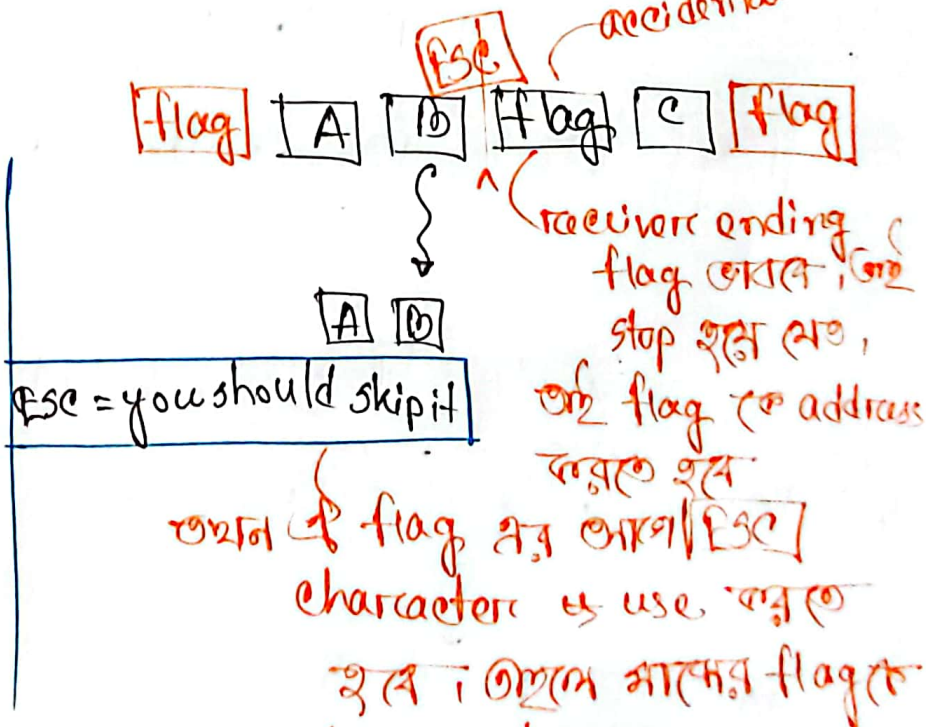
frame (cell): 53 Byte (max.)

sender যা data পাঠাবে receiver read করবে, যখন receiver 6 read করবে (কিন্তু data আসছে) [error] তখন সে 6 count করবে। আবার 6 read করবে। আবার 4 read করল না, কিন্তু error না হয়ে character read করল কিন্তু কি বুঝতে পারছে না।

① Byte stuffing — আগে পিছে flag insert

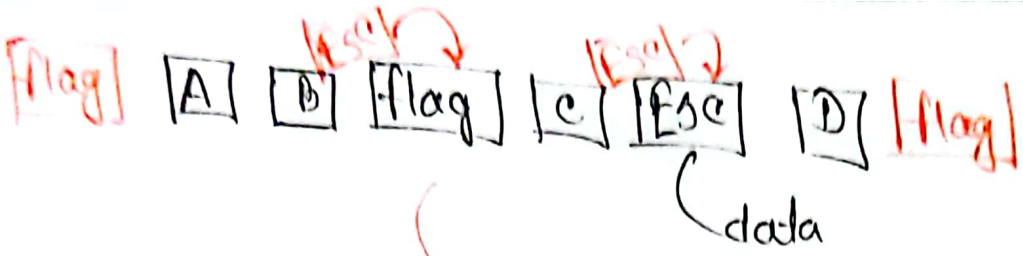
Flag: pattern is not fixed



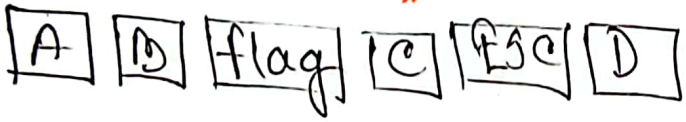


flag: 10101010

Esc: 0000111



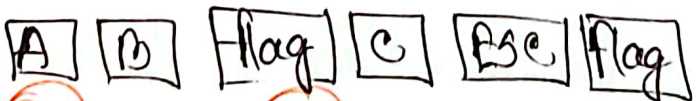
receiver কে বলতে হবে মাঝের Esc আর্থার data. এই প্রত্যেকটার আগে Esc (skip character) দিতে হবে.



not fixed
flag : 10101010
Esc : 00001111

প্রত্যেকবার Esc দিতে হবে.

receiver



12-02-24

9-4/ Bit Stuffings

- flag fixed
- pattern fixed

flag (fixed): 01111110

[flag] 10101110100 11 [flag]
(data)

01111110 10101110100 11 01111110

(receiver)

10101110100 11 | 01111110 flag pattern
(data)

01111110 10101101111111010 01111110
flag
receiver
10101101111111010
instruction for sender

flag পাঠানোর আগে অবশ্যই address করতে হবে

after every 5 consecutive 1's, insert a '0'

আগে পিছে flag insert করতে হবে

Receiver instruction

- flag read করবে
- 5 consecutive 1 এর পর যদি 0 থাকে সেটা omit করতে হবে

01111110 10101101111110101 01111110
receiver
10101101011111010 | 01111110

কখনোই Data র মধ্যে 111111 থাকবে না।

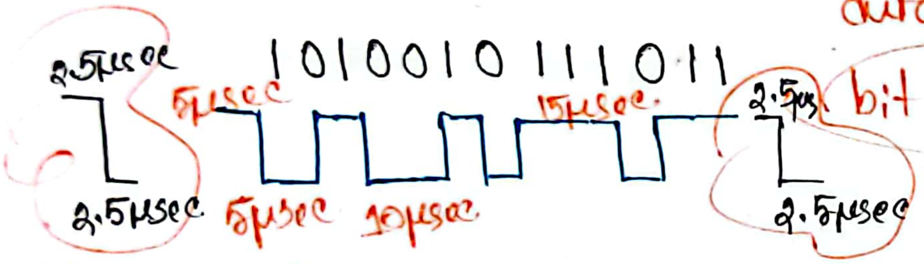
01111110   1010110101111110001010   01111110
(flag)   (flag)

↓

1010110101111100 1010 | 01111110

Physical layer Coding violation

durationwise change হয়।



bit-time বোঝানোর জন্য পাঠাবে

bit time : 5μsec

— violation হচ্ছে flag*

— প্রথমে ও শেষে যে violation ওটি flag হিসেবে consider হবে

→ ঐ hardware এর violation মেকানিজম মত প্রস্তুত থাকতে হবে

→ বেশি কাছাকাছি (bit time) সময় হলে violation ধরতে সমস্যা হবে

আবার 0-এর কাছাকাছি হলে receiver বুঝবেই না violation হচ্ছে কিনা

— violation bit-time এর বেশি হবে না ; তখন flag detect করতে না পেরে data consider করবে

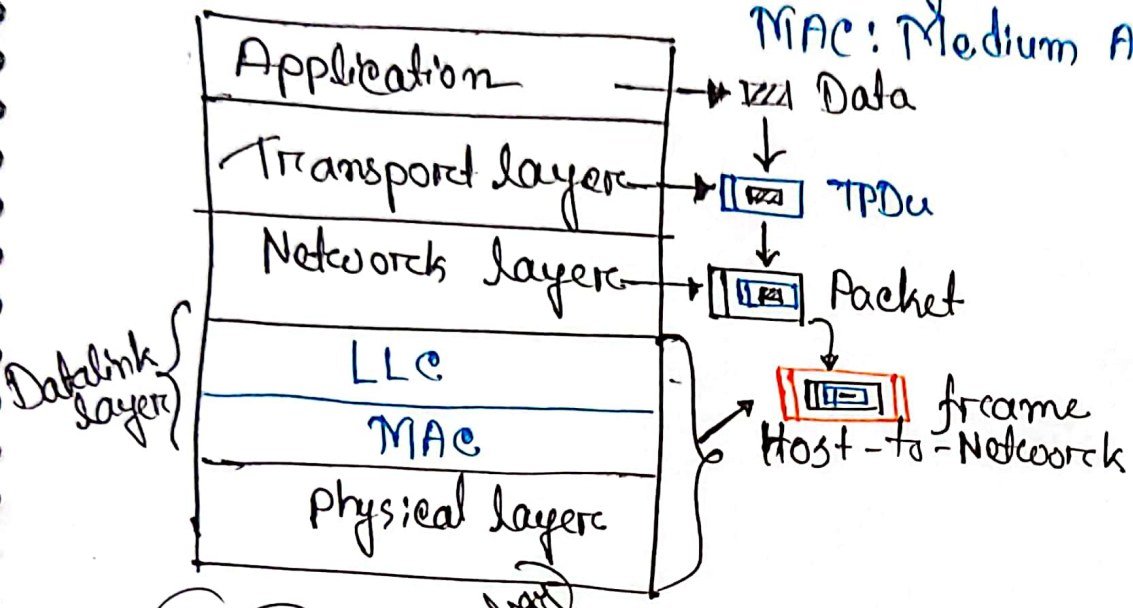
— middle দিলে best হবে

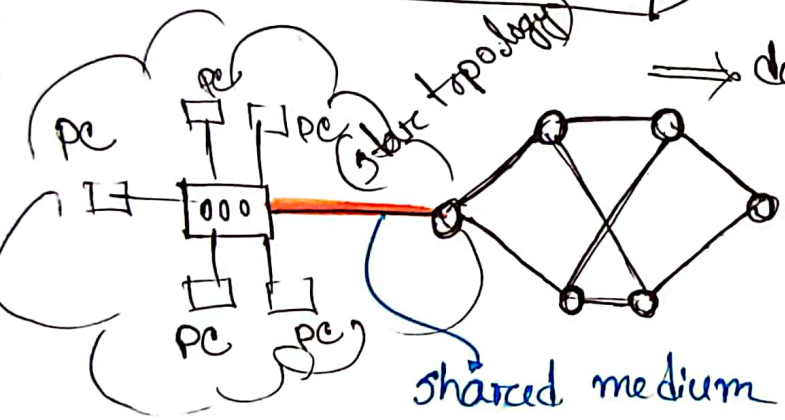
19-02-24

lec-5

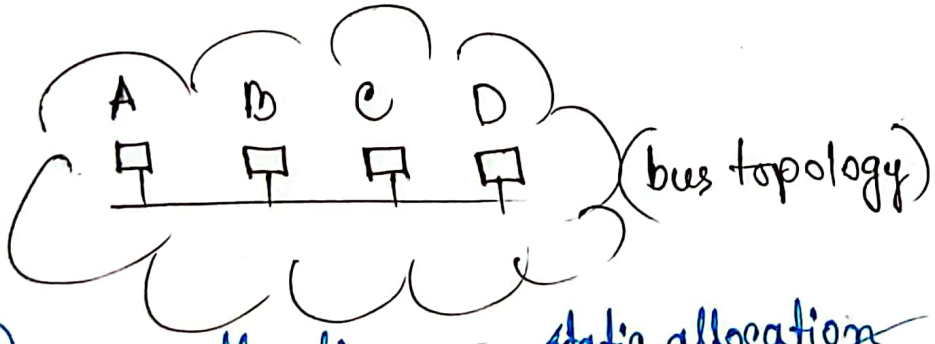
TCP/IP Ref. Model

LLC : logical link control sublayer
MAC : Medium Access Control sublayer

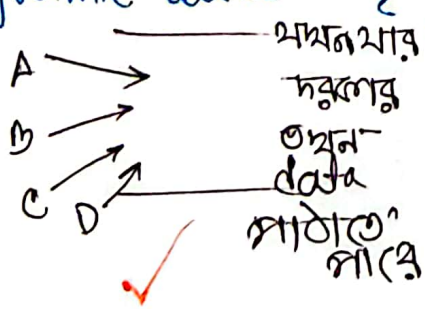




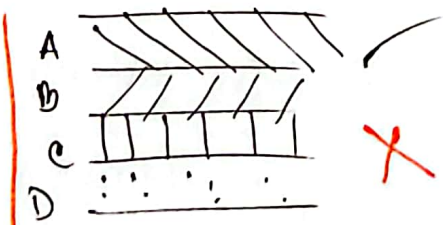
⟹ datalink layer এর কিছু host device থেকে অন্য device-এ data transfer share করবে বা কোনো instruction এর control access করে সেটা হচ্ছে MAC.



Dynamic allocation

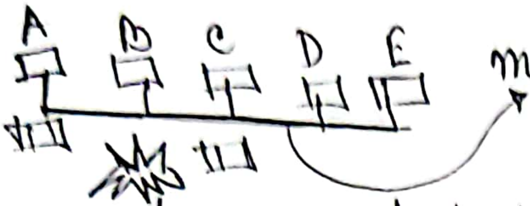


static allocation



একবারে অনেক data পাঠানো যাচ্ছে, specific কাজের জন্য; তাই এটির static allocation (আগে করে দেয়া) It's not very useful

Dynamic Channel Allocation Protocols → কখন কোনটা transmit করবে



medium এর মধ্যে channel share করে

Assumptions: protocol design করার আগে assumption থাকতে হবে

i) Station Model [There are N num of independent stations/nodes (integer)]

ii) Single channel

iv) Collision — Time
- continuous [যখনই data থাকবে you can transmit]
- Slotted

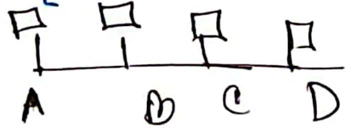
iii) Collision [নিজের turn এর জন্য wait করবে]

v) Sense
- Yes
- No

(Carrier sensing capability)

[একই সাথে A আর C frame পাঠালে তখন collision হবে, তখন they have to retransmit again. collision প্রথমেও হতে পারে; শেষেও হতে পারে, কারণ পুরো frame নষ্ট হয়ে যায়।]

- ALOHA → যখন যার data থাকবে; তখন সে data পাঠাবে। যদি data বেড়ে যায় তখন কারও না কারও সাথে collision শুরু হবে। একবার collision হলে বারবার collision হওয়ার possibility বেড়ে যায়। তখন free slot পাওয়া সম্ভব হয় না।

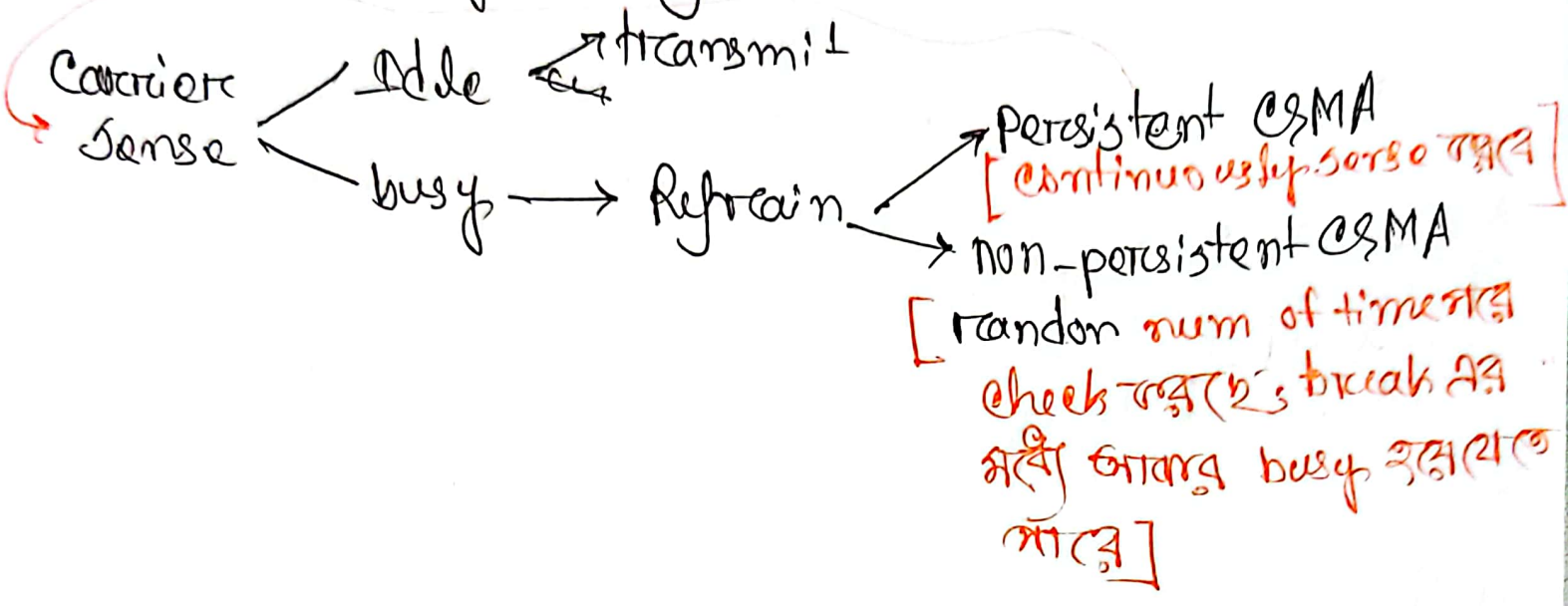


কোনোটাই use করি না

- Slotted ALOHA — number of nodes বেশি থাকলে এটা preferable
- CSMA (carrier sense multiple access) [Sensing capability থাকতে হবে]

Sense (carrier)
- → Idle → Transmit
- → Busy → Refrain from transmission [বিরত থাকতে বলা হচ্ছে]

Idle হলে collision হওয়ার chance কম তাই তখন transmit করতে পারবে। busy হলে refrain করতে হবে।



20-02-24

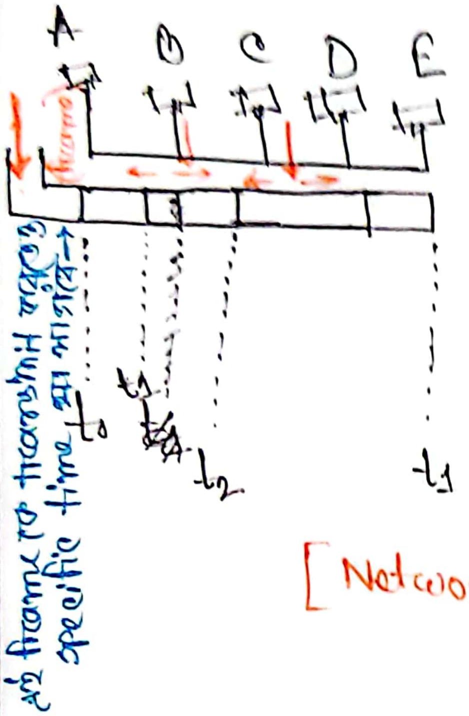
Lec-6

CSMA (Carrier sense multiple access) Protocol

[collision হবার chance আছে]

Sense → Idle → transmit
Sense → Busy → Refrain from transmission.

→ [t₀ শুরু হওয়ার পর data যেতে থাকলে কিন্তু frame যেতে হয়ে যাবে না]



[collision হলো সব frame গুলো frame থাকলে অনেকগুলো involve থাকবে সবাই নষ্ট হবে]

→ at t_1 time, 1 bit B তে পৌঁছায়নি তাই B তে idle পাচ্ছে পারে

→ sense করতে হবে কোনো signal আছে কিনা, if there is going any current is going on.

[Network layer কাজ করছে]

Sense? → by detecting current or voltage in the carrier.

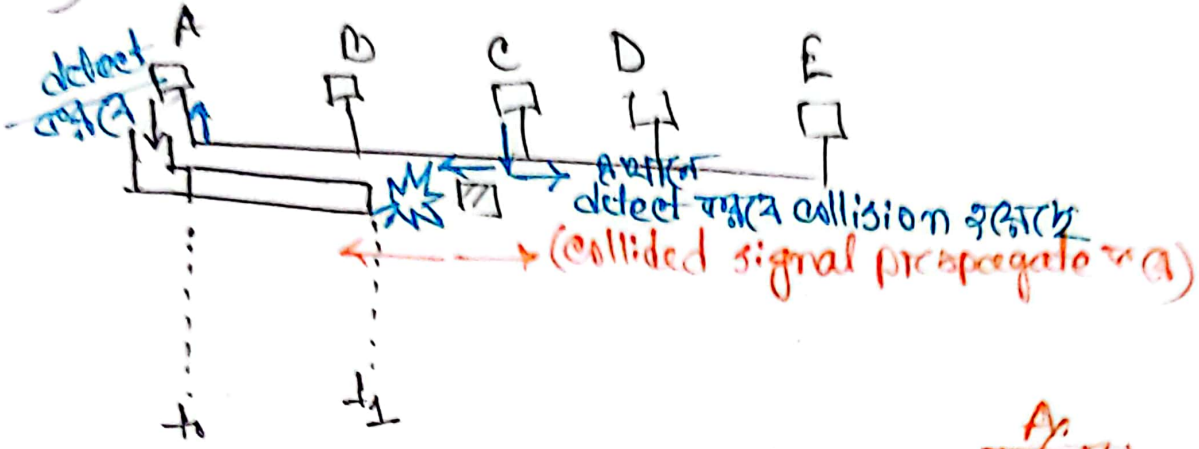
at time = t_2 → C, D, E idle; transmit হবে কিন্তু B তে busy পাবে that's why B can't send any data.

→ CSMA তে collision হতে পারে, (tiny window আছে)

→ যদি t_1 E তে গিয়ে পৌঁছায় তাহলে আর collision হবে না; কারণ সবাই জানে data পাঠানো হয়ে গেছে।

CSMA/CD (Carrier sense multiple access with collision detection) ?

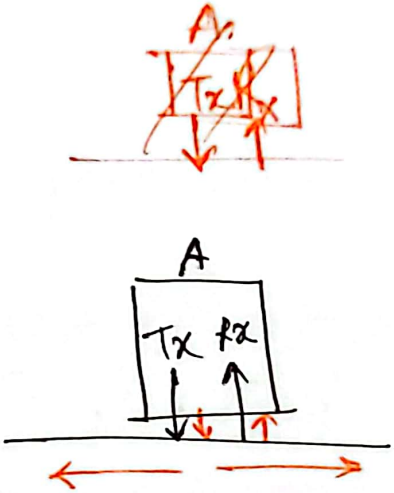
(better of than CSMA)



Transmitted signal and received signal compare করা হয়।

- same হলে → no collision
- different হলে → collision

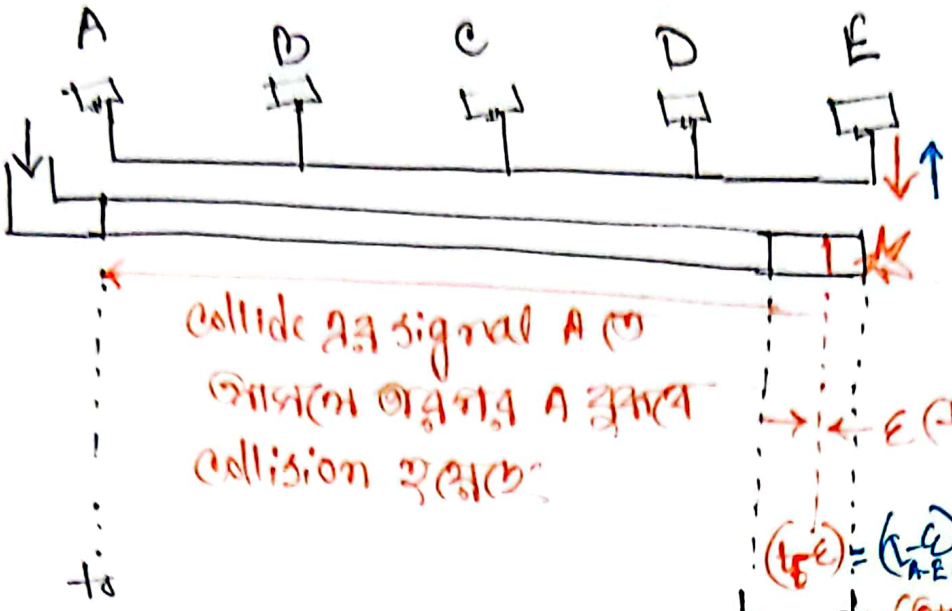
(signal) যে station থেকে সে আবার receive করছে



যদি transmitted signal = received signal

∴ collision হয়নি

যদি equal না হয় তাহলে collision হবে



- $t_1 - \varepsilon$ is the last possible time of the collision

যেতে সময় + আসতে সময় ⇒ যদি collision হয়েছে সেটা A তে এ আসতে সময়

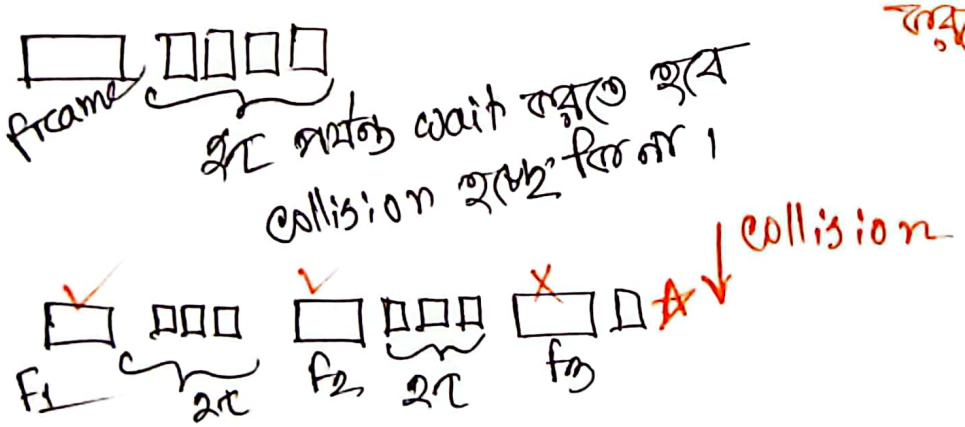
$(t_1 - \varepsilon) + (t_1 - \varepsilon)$

$= 2t_1 - \varepsilon \approx 2t_1 = 2\tau_{A-E}$ = Contention period

A-E = farthest node

Propagation delay = τ_{A-E} (দূরের time)

Contention period = $2\tau_{A-E}$ [এই সময়ের মধ্যে collision হলেও না হলে detect-3 করবে না।]



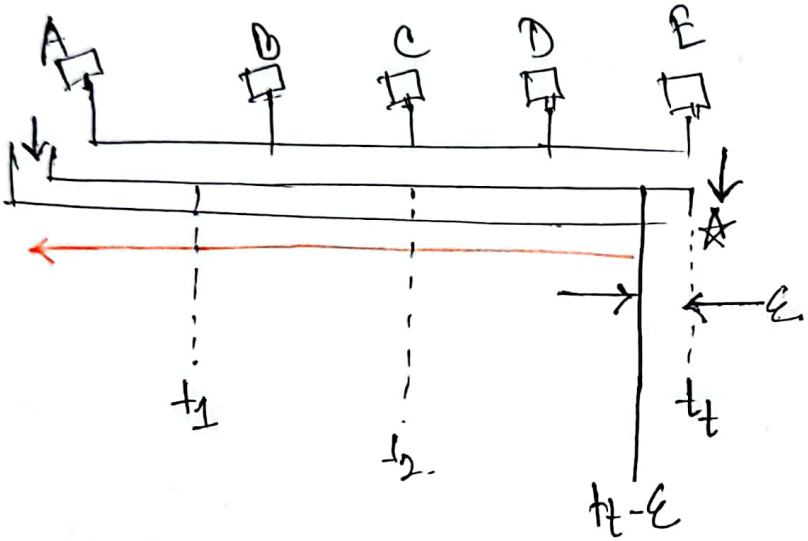
4-03-24

Lec-7 / CSMA/CD — is always designed as half duplex

τ : propagation Delay

How it detects collision?

Contention period = 2τ

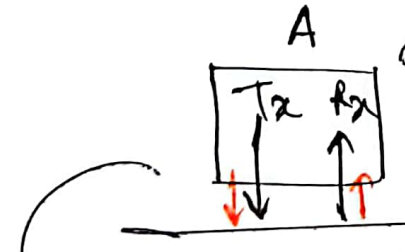


Destination C হওয়ার সত্ত্বেও signal A থেকে E পর্যন্ত যাবে, (here every elements are covered) তার মানেই (contention period) বলতে পুরোর propagation delay বোঝাবে

∴ contention period = $2\tau_{A-E}$

= ~~$2\tau_{A-C}$~~

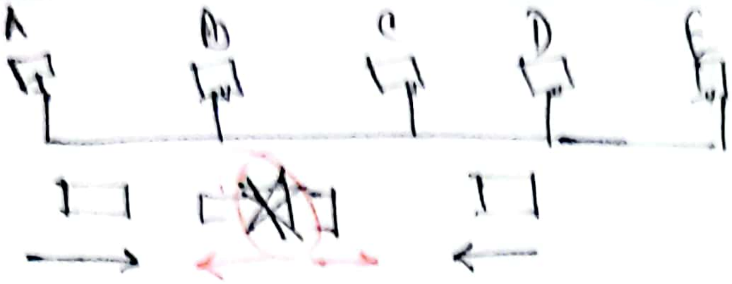
between 2 farthest nodes যেহেতু আমরা exact position জানিনা পুরোর medium এর propagation নিব।



এটা always busy; কেননা signal receive করতে পারবে না।

এটা always engaged for detecting collision

যদি full duplex হয় তাহলে আলাদা signal receive করতে পারবে এবং collision detect করতে পারবে না।



collided signal পুরো medium-এ propagate করবে। collision থেকে আবার নতুন করে signal পাঠাতে হবে কারণ আগের সব frame নষ্ট হয়ে গেছে।

same সময় same signal পাঠালে আবারও collision হতে পারে। তখন BEB apply করতে বলা হয়েছে। (আবার collision ঠেকাতে)

BEB (Binary Exponential Backoff) Algorithm :

collision হয়েছে মানে frame নষ্ট কিন্তু ওর দেখার দরকার নেই কার সাথে collision হয়েছে।

এখন এই algo run করবে তখন ওর set হবে

$$Set = \{0, \ldots, 2^n - 1\} \quad n: \# \text{ of collision (counter)}$$

(pick an element from the set randomly) × (512 bit time) = waiting time

For A,

collision # 1

$$Set = \{0, \ldots, 2^1 - 1\} = \{0, \ldots, 1\} = \{0, 1\}$$

0 × 512 Bit time = 0 time (waiting time – doesn't need to wait সাথে সাথে transmit করবে)

1 × 512 Bit time = 512 Bit time (waiting time)

For D, collision #1

Set = {0, 1}

0 × 512 Bit time = 0 time (waiting time)

1 × 512 " " = 512 Bit time (")

50% chance for next collision — worst case.

For A

Collision #2 (n = 2)

Set = {0, ... $2^n - 1$}

= {0, 1, 2, 3}

0 × 512 Bit time = 0 time (waiting time)

1 × 512 " =

2 × 512 " =

3 × 512 " =

for D

collision #2 (n = 2)

Set = {0, 1, 2, 3}

0/1/2/3 × 512 Bit time = ? (waiting time)

একই same num pick করার chance 25%.

for A

collision #3 (n = 3)

Set = {0, . . . , $2^3 - 1$}

= {0, 1, 2, 3, 4, 5, 6, 7}

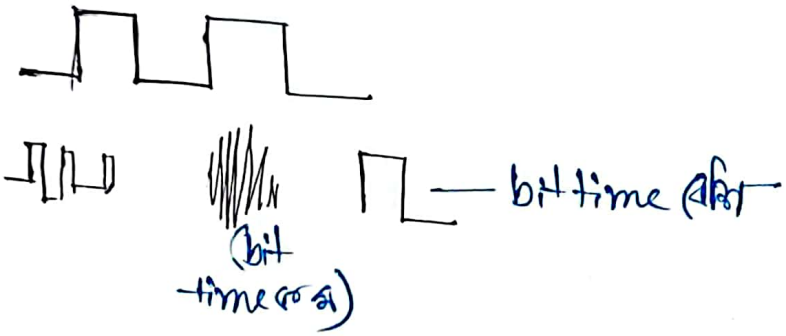
0/1/2/3/4/5/6/7 × 512 Bit time = ? (waiting time)

আবার same num pick করার chance 12.5%.

↑ 10th collision * consecutive 10 বার collision হলে বুঝতে হবে trouble আছে কোথাও হচ্ছে।

* 2 জনের set different হতে পারে

* collision হওয়ার time = $t_2 - \epsilon$



lab-03

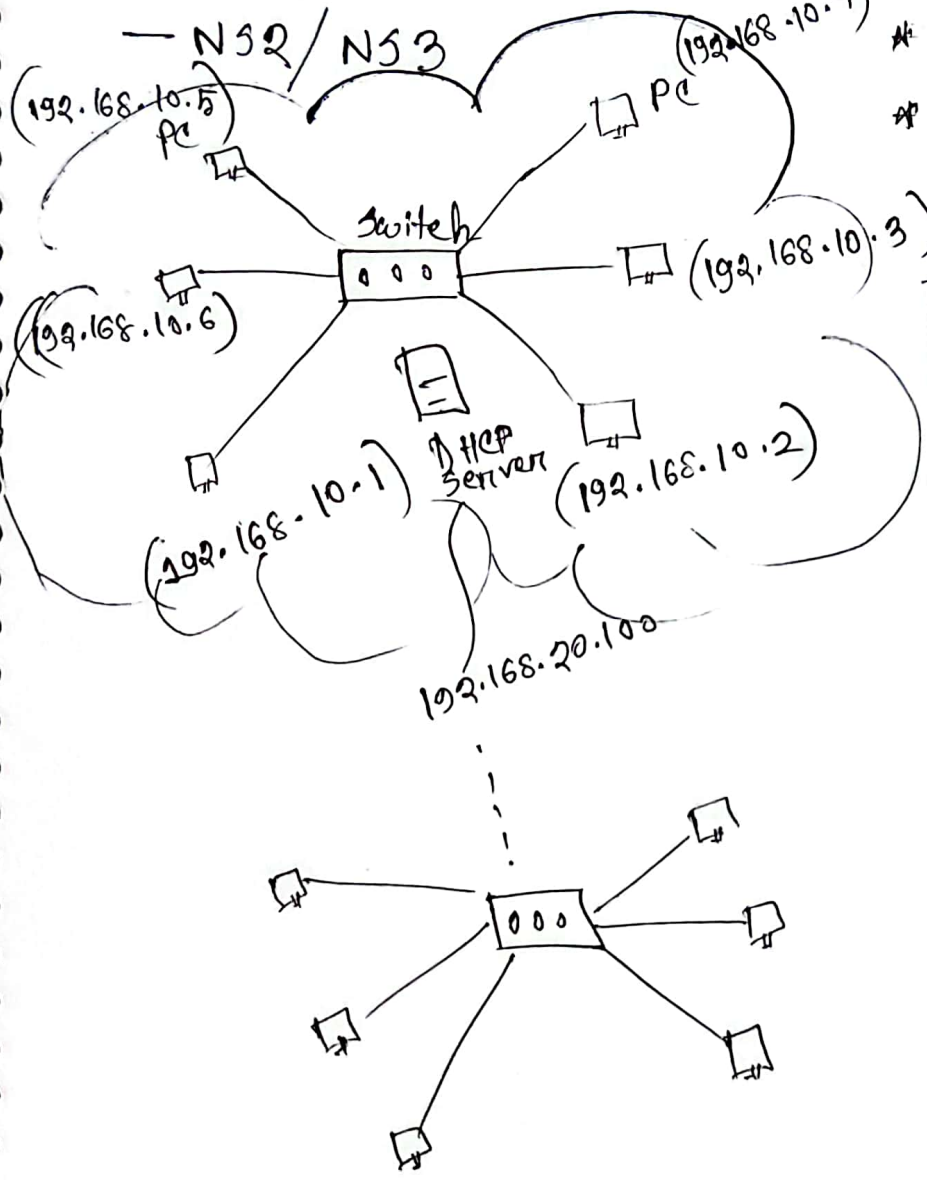
Simulation

✓ CISCO packet Tracer (pre. defined আছে)

- OPNET
- OMNET
- Simulink
- NS2 / NS3

Logical design

* PC – end device = 6
* 2960 switch



TTL : Time to live
(সময়) যতক্ষণ device থাকতে পারবে
(Hop count)

Date: / /

IPv4 Address

Internet Protocol (IP)

specific address for the device

32 bit long

total # 2^{32}

00000000 (0)

⋮

11111111 (255)

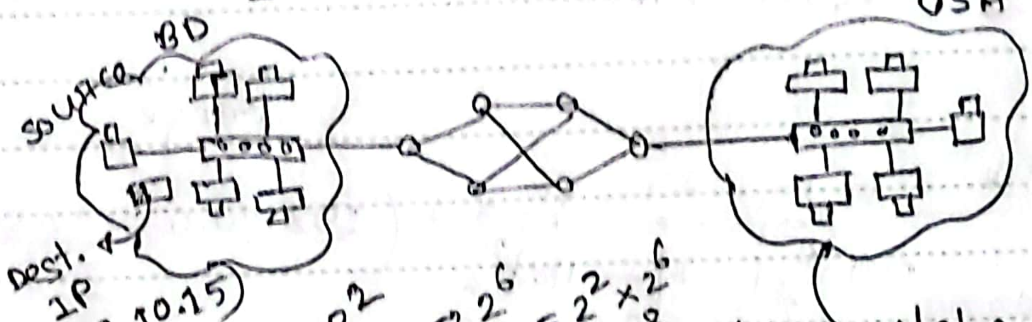
11000000 00001010 ....

192.10.15.5 ← (decimal Notation)

IPv6 (128 Bits long)

2^{128}

(Quiz #1 13 March '24)



00000000 → 2^2, 2^6 $= 2^2 \times 2^6 = 2^8 = 256$

⋮

1111 1111

$2^8 = 256$

Network part, Host part

11 | 000000

00	000000	} 64
00	000001	
	⋮	
00	111111	

01	000000	} 64
01	000001	
	⋮	
01	111111	

10	000000	} 64
10	000001	
	⋮	
10	111111	

class A: 0 — 8 — 16 — 24 — 32

Net → | ← Host →

$= 2^8$, 2^7 ; 2^{24}

(0) 0000000 | 0000........0000

⋮

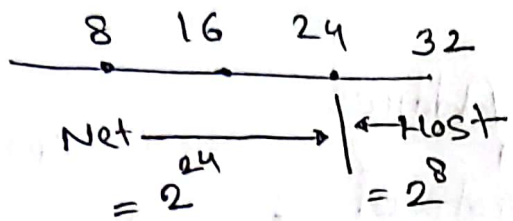
(127) 1111111 | 1111 1111

class B: 8 — 16 — 24 — 32

Net → | ← Host →

2^{16} ; 2^{16}

Class C

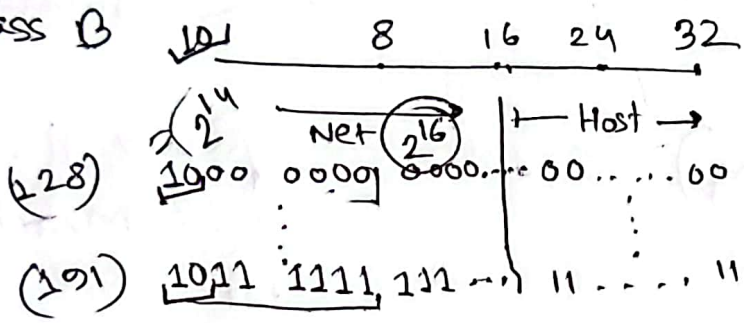


$2^{15} - 2^{0}$

32 16 8 4 2 1

Class A Range (0.0.0.0 — 127.255.255.255)

Class B

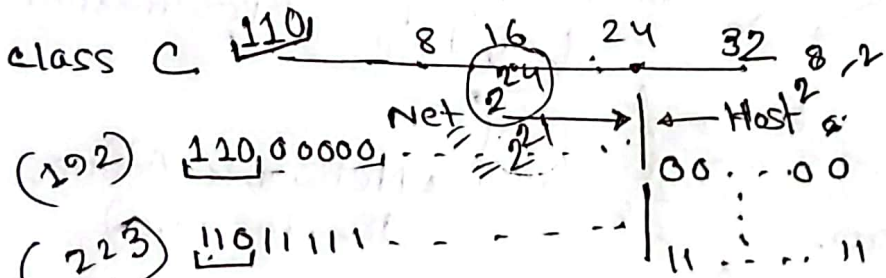


Range

128.0.0.0 — 191.255.255.255)

Range

Class C



(192.00.0 — 223.255.255.255)

Multicasting

Class D (224.0.0.0 — 239.255.255.255)

Class E (240.0.0.0 — 255.255.255.255)

Future use

(128.120.255.255)

00...00

Date: 00000000 (0)

Destination IP: 192.168.10.15
Net / Host → 1111 1111 (255)

class C

Network IP Address: 192.168.10.0

Broadcast IP Address: 192.168.10.255

Subnet Mask IP Address:

11...11 11...11 11...11 00...00

255.255.255.0

Network Address
✗ (192.168.10.0
192.168.10.1
⋮
192.168.10.254
✗ 192.168.10.255

Total 254 (Host)

Dest IP: 000.00 00...00

156.140.35.10
Net Host 11...11 11...11

class: B

Network IP Address: 156.140.00

Broadcast IP Address: 156.140.255.255

Subnet Mask IP: 255.255.0.0

Rupa

Lijenta

11-08-24

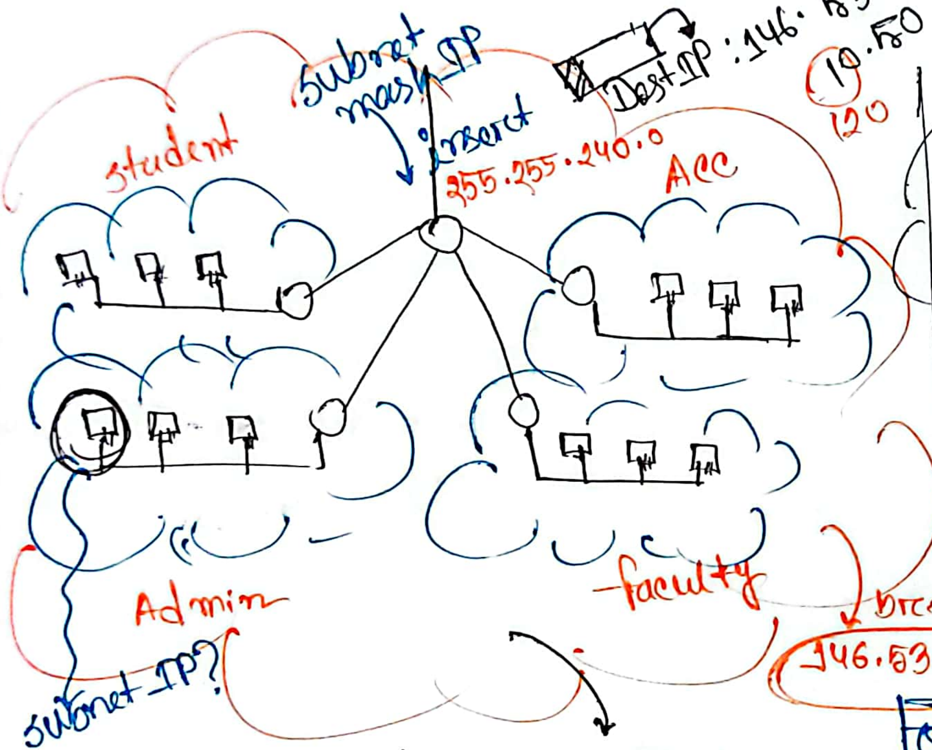
lec-9/

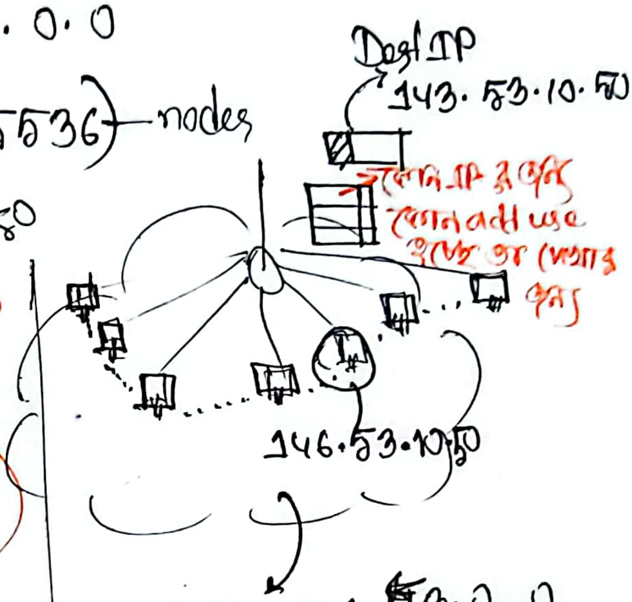
* address থাকে datalink layer এর মাধ্যমে। যা MAC address দিবে

Destination IP : 146.53.10.50 (Net: 146.53 ; Host: 10.50)
(Class: B)

Network IP address : 146.53.0.0

Total # of host : 2^{16} (65536) nodes





Net IP : 146.53.0.0

এখানে search করতে গেলে পুরোটা desrupted

Net IP : 146.53.0.0

এখানে এটা এটা select করে প্রয়োজন অনুযায়ী down করে কাজ করা যায়। এটা use করা বেশি useful.

কোন network এর মধ্যে আছে জানতে subnetwork জানতে হবে। (network এর ভিতর network)

Subnet × Host: $2^4 \times 2^{12} = 2^{16}$

to denote subnet — subnet mask

146	53	0000	0000	0000 0000

Network part unchange রাখতে হবে (Host)

1st sub : 0000 0000 00000000

2nd sub : 0001 0000 00000000

⋮

last sub : 1111

subnet mask = separation

~~1st subnet IP : 146.53.0.0~~

2nd subnet IP : 146.53.16.0

→ 1st

subnet হলে আমাদের কোনো change করা যাবে না।

এখানে 1st subnet আর Net IP same; তাই use করা যাবে না।

— o —

146	53	0000 | 0000	00000000
← Net →			
1st sub	:	0001 | 0000	00000000 → 1st sub
2nd sub	:	0010 | 0000	00000000 → 2nd sub

[প্রথম num network এর সাথে শেষ num broadcast এর সাথে match করে]

1st subnet IP : 146.53.16.0

2nd subnet IP : 146.53.32.0

3rd subnet IP : 146.53.48.0

— o —

146	53	|	
1st subnet :	0001	0000	00000000 → 1st sub
	0001	0000	~~00000000 → 1st host~~
		0000	00000001 → 1st host
		0000	00000010 → 2nd host
	0001	1111	11111111
	0001	1111	11111110 → last host

1st subnet 1st host IP : ~~146.53.16.0~~ 146.53.16.1

1st subnet 2nd host IP : 146.53.16.2

⋮

1st subnet ~~last host IP~~ Broadcast : 146.53.31.255

CamScanner

1st subnet last host : 146.53.31.254 [যেহেতু last number টা broadcast তাই সেটাকে last host না নিয়ে তার আগের number টা হোস্ট নিতে হবে]

subnet এর প্রথম এবং last বাদে যাবে।

146   53   0000 | 0000   00000000

3rd sub : 0011 | 0000   00000001 → 1st host

0011 | 1111   11111110 → last host

3rd subnet 1st host IP : 146.53.48.1

3rd subnet last host IP : 146.53.63.254

146   53   0000 | 0000   00000000

last sub : 1111 | 0000   00000000

~~1111 | 1111~~   11111111

last sub : 1110 | 0000   00000000 → last sub

~~last subnet IP : 146.53.240.0~~

last subnet Broadcast IP : 146.53.255.255

because this broadcast IP is same as the mother broadcast IP

∴ last subnet IP : 146.53.~~50~~.224.0

Subnet = $2^4 - 2 = 16 - 2 = 14$ Subnet

→ 1 bit subnet হতে পারবে না ; atleast ২টা নিতে হবে । $2^n - 2 = 2$

last subnet 1st IP host : 146.53.224.1

1110 | 0000 → 1110 0000 → 1111

0000 0001 → 1st host

11111111 → last Broadcast

last subnet Broadcast IP : 146.53.239.255

Dest IP : 146.53.10.50 (10 = 00)

subnet | subnet mask

146 | 53 | 0000 | 0000 | 00000000

11....11 | 11....11 | 1111 | 0000 | 00000000 → subnet mask IP

Subnet mask IP (decimal notation) : network + subnet portion = 1
+ host part = 0

→ 146.255.255.240.0

Subnet mask IP (CIDR) : 146.53.10.50/20 — কত বিট separation আছে

(class less inter Domain)

11111111 11111111 11110000 00000000

subnet mask IP : 255.255.240.0 ↑ AND

Dest : 146.53.10.50 ←

AND → 10010010 00110101 00001010 00110010

10010010 00110101 0000 (1st subnet)

0111 0000

146 | 53 | 112 | 0

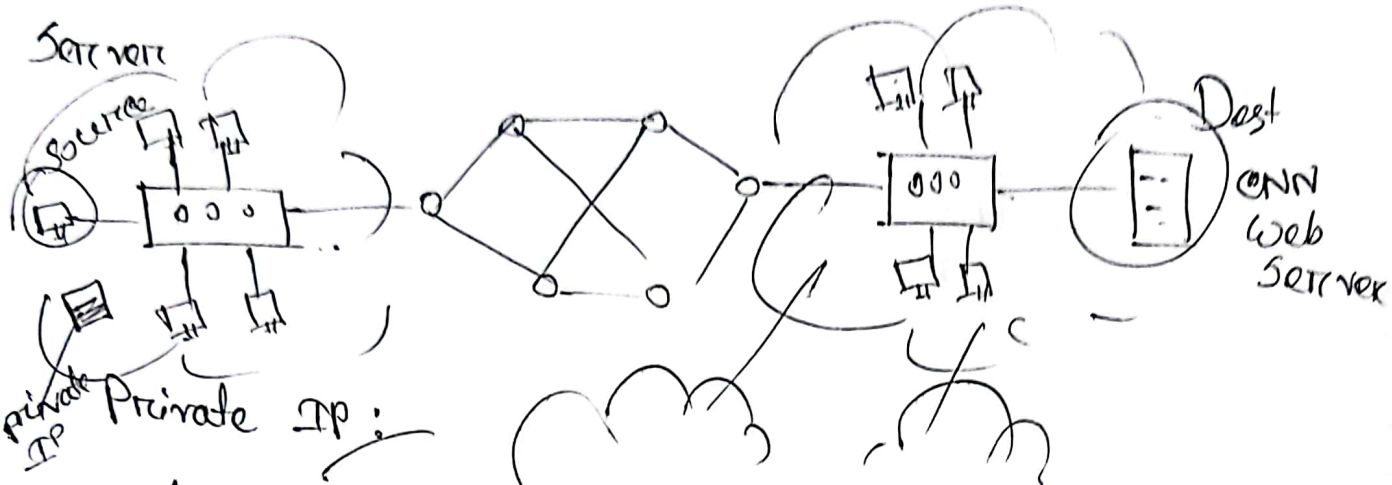
Subnet IP : 146.53.112.0

11111111	11111111	11110000	00000000
10010010	00110101	01111000	00110010

lab-9
lec-10

IPv4 (32 Bits) (Quick fix?) → IPv6 (128 Bits)

total = 2^{128}



Private IP :

class A : 10.0.0.0 — 10.255.255.255 (>16x10^6)

class B : 172.16.0.0 — 172.31.255.255 (>1x10^6)

class C : 192.168.0.0 — 192.168.255.255 (>65x10^3)

Real (Public)
Rest of them

NAT (Network Address Translation)
(feature এটা help করবে)

Real (Public)

Rest of them

10.2.4.1

A

10.2.4.2

10.2.4.3

10.2.4.4

Packet 1

S: 10.2.4.1

D: 76.20.10.100 (CNN)

Packet 2

S: 10.2.4.4

D: 76.20.10.100

NAT Box

Packet 1

S: 120.5.6.10

D: 76.20.10.100

S: 120.5.6.10

D: 76.20.10.100

B

(120.5.6.10)

ISP's router

USA

CNN Web Server

76.20.10.100 (public)

Private IP দুলের Network এর বাইরে যেতে পারবে না; কারণ অন্য কেউ same IP পাঠাতে পারে।

Source এর IP এর সাথে নিজের ISP's router এর IP replace করে

NAT যাবে replace করল তার জন্য Index maintain করবে

10.2.4.1	A
10.2.4.4	B

Reply1

S: 76.20.10.100

D: 120.5.6.10

A

insert করে

এভাবে private IP-র মাধ্যমে world এর সবার সাথে communicate করতে পারে

Reply2

S: 76.20.10.100

D: 120.5.6.10

B

Lijenta

CamScanner

Port address (16 bits long)
(header)

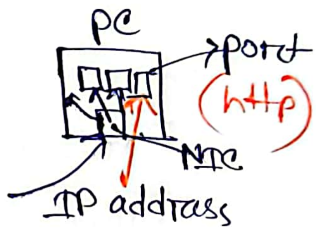
- TPDU এর ভিতরে থাকে port address
- frame এর মধ্যে থাকে MAC address

memory location যেমনে specific .. locate করে

Source port (16)	Dest Port
⋮	⋮

2^{16} ports = 65536 ports আছে প্রতিটা device এর জন্য, প্রত্যেকটা port এ আলাদা algo দেয়া আছে

- IP add দিয়ে খুঁজে পাবে কিন্তু service দিবেনা তাই নির্দিষ্ট port খুঁজতে হবে



DNS server — specific IP address আছে
Machine. Host name
port

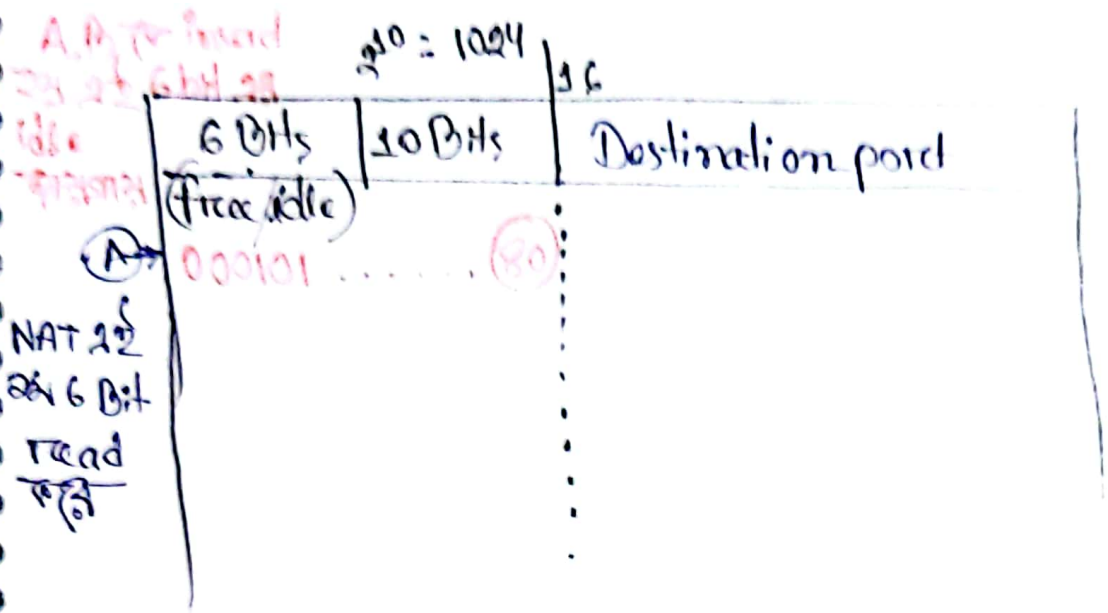
http://www.CNN.com/filename

Protocol — http
requested file — filename

CNN domain এর under এ host-name এর সাথে contact করবে

DNS — Domain name system server
- ঐ নামের corresponding IP address provide করে
- IP দিয়ে address খুঁজে বের করে

- Reserved port (0 – 1023) → Well known services এর জন্য
  ex: http: (port 80)
  Telnet: 23
  ftp: 21
  smtp: 110
  ⋮
- Rest → free (not for specific use) (idle)



NAT में 6 Bit read करते