→ CN used for communication b/w computers

→ Needed for communication :

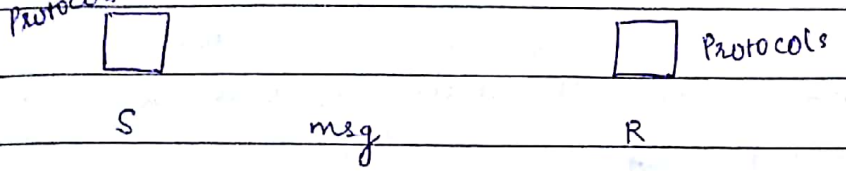
Sender

Receiver

Message

Transmission medium

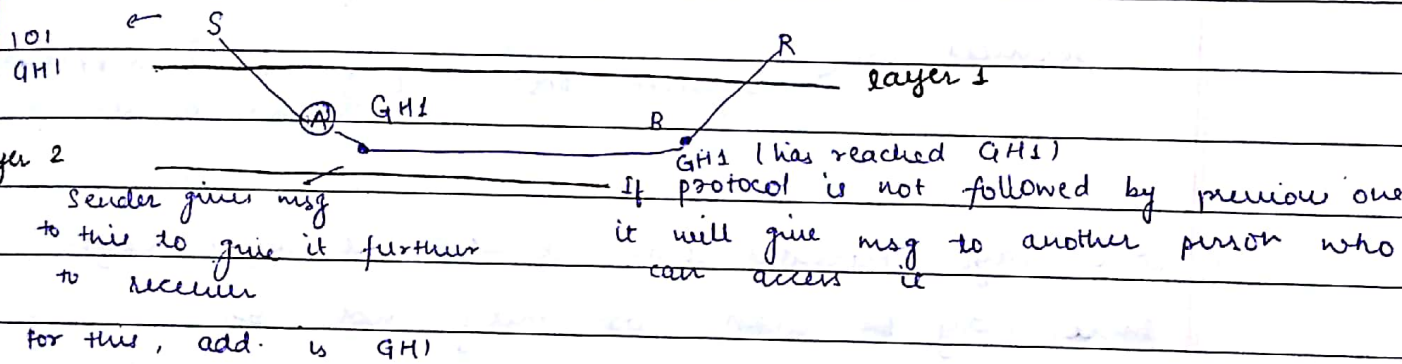
Common rules : Protocol → some set of rules on which two parties agree to communicate



→ Face-to-Face : medium : wireless (air)

→ Communication not possible if they don't agree to same rules (protocol)

→ → This forms network



Comp 1 Comp 2

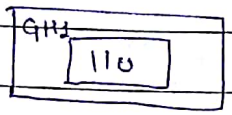
→ Network is the backbone over which communication can occur.

TCP / IP → 5 layers

OSI Reference – 7 layers

DATE : / /
PAGE NO. :

→ Since S can't directly communicate to R, it is sent to entity in another layer. Now, we don't know how msg is going to be delivered.
So, Ⓐ is providing some services to S ∴ delivering msg



A puts it in another envelope GH1. B knows GH1. So, A has to communicate with B. Thus, they follow same protocol.

B will remove GH1. It will provide services to R → delivering msg

Communication is not yet completed
↓
Comm. without acknowledgement.

→ In p. case of posts :

Services
- Normal post : if you don't want ack.
- Registered post : if you want acknowledgement (receive a receipt)

→ Lower layer provides some services to upper layer.
There may be ackn. or may not be

Services
- UDP : don't get acknowledgement
- TCP : get acknowledgement

→ diff. layers :

Applin
Transport
Network
Data link
Physical

Dr. Subrat Das

Teacher's Signature

Online gaming
Chat Rooms
Forums
Twitter
Facebook

Emails
Texting

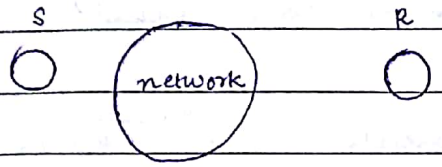
Interactive films / television

DATE : / /
PAGE NO. :

→ It is the same network which is supporting all these app's. (all modes of communication) [Eg. emails, fb, ...]

→ All communication are working on the same network.

10/1/19



→ When you make your phone a hotspot, you configure its name & password. The other person must also have to know its name for exchange of files.

The 2 devices should be in range to send files.

→ Other way is to use Bluetooth. → create Personal network (radio waves)

⇒ A network can only made up by devices

⇒ CN : Network is created using h/w.

→ Connecting comp. without internet :

Video

- WLAN wire use
- Some config" needs to be done before 2 PCs can connect.

create new network

- OS of PC's also plays an important role.
- Adhock n/w : without WLAN wire.

→ Cable should also have a definite length. Wifi → definite range

This is PAN (Personal Area Networks)

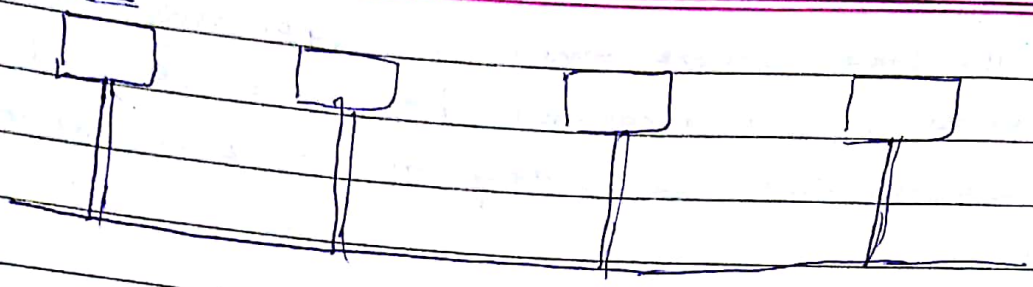
No. of devices : 2 - 8 3

→ In COA, comm" occurs using bus → comm" channel

→ Bus arbitar controls who'll use the bus ⇒ again network

Teacher's Signature

→ PCs



Connected through a common cable (shared)

→ Network makes comm^n possible (provides resources sharing)

Components :

Channel b/w devices

↳ Computer, Router, Printer, Phones, Laptops, Network switch

devices

Switches Router

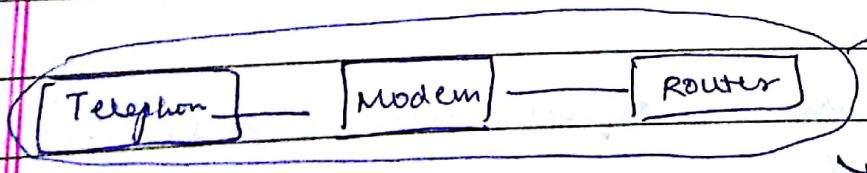
It is interconnection of these devices which make comm^n possible (over a comm^n channel). This is called CN.

(CN consist of H/W only : we think for now)

→ Components of Home Wi-Fi :

- Router
- Telephone Cable
- Modem : converts telephone signals to digital signal

* If only a Router is there, you can interact with other devices



→ Again, a network

(7 - 8 devices)

(beyond 10m range)

↳ This is special category of CN called Home Network

Teacher's Signature

→ To connect to outer world, then only you need telephonic cable.

→ Telephonic cable & modem are aiding network & hence, part of CN.

- PAN, LAN, WAN : All are networks diff. in range
- Router can be accessed

PAN
LAN - Local Area Network (All devices in an institute)
WAN - Wide Area Network (Whole city)

→ diff. is only b/w size & scale & range b/w devices

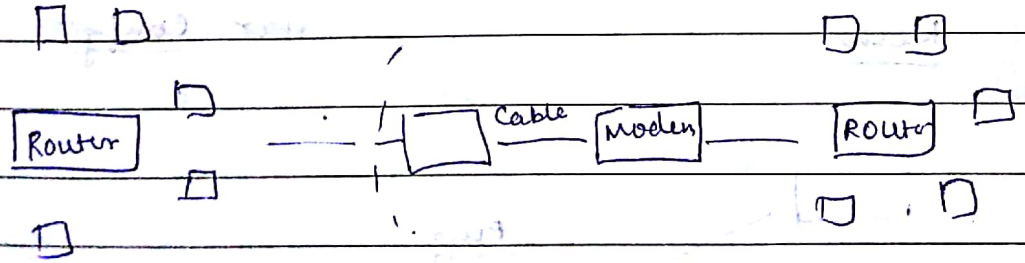
PAN → (2 - 3)

PAN -
LAN -
WAN -

→ We can configure Router for Home Network : N/W Operator
User
N/W Operator → CCNA course
App^n developer

→ We 'll learn how to design a n/w — h/w / s/w

→ Wi-Fi v/s Internet



Wi-Fi

Internet
(Connected by cable to another Router)

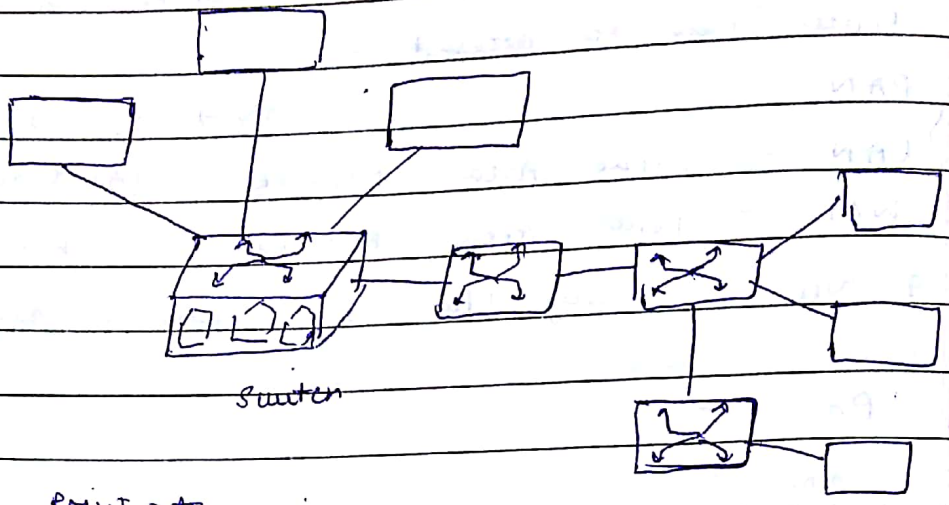
Router

1)

(It is n/w)

Point to Point n/w

2)

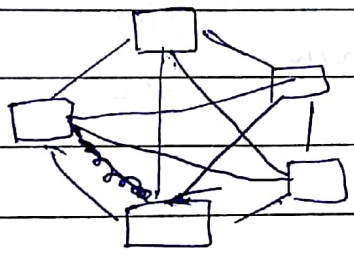


Devices : Point-to-point

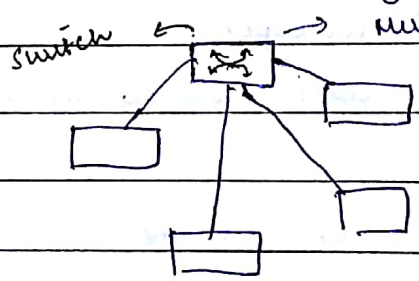
Switch : Multi point

If comm^n has to happen within these 6 PCs :

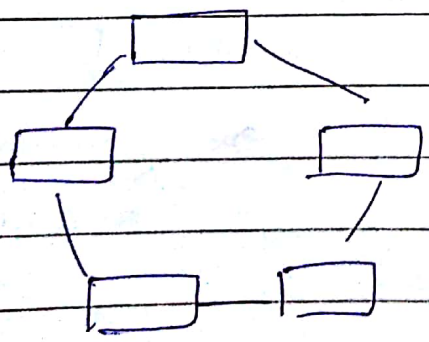
Eg. 5 PC's : Can also communicate using intermediate PCs



Mesh



Star Config^n



Ring

Wi-Fi : multi point which
(no specific port on Comm^n is happening)

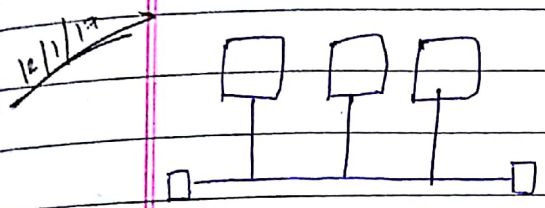
DATE : / /
PAGE NO. :

: Multi-point

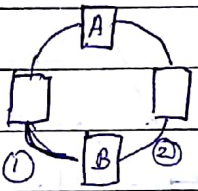
Mesh : Very fasto

→ Switch : less cables are used to connect more PCs.
↳ has to be some s/w to find route to PC where info has to be sent.

So, Network Topologies



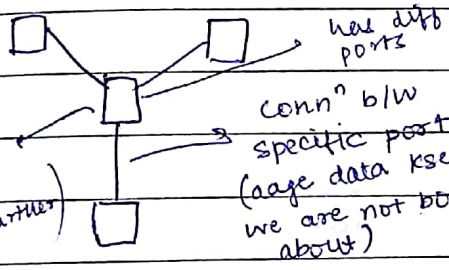
Bus
Connection : (multi-point)



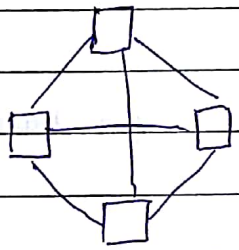
A — B
going to use ①, or ②
↓
shared by various devices
↓
multi-point

also forwarding other's traffic

ring
multi point



star



mesh

Connection : Point-to-Point

Topologies : ~~physical~~ representation in which devices ~~network~~ are linked

A Link : connects 2 devices → how much devices can access it

↳ point-to-point

↳ multi-point : can use same conn^n for different ~~comm^n~~ communication.

→ Shared bus in COA : multi-point bus
diff. purposes, diff. devices

→ Point-to-Point :→ Connect 3 PCs with your PC. No. of ports required : 3
↳ info. going b/w port of 1 PC to port of another PC
↳ shared port : multi point

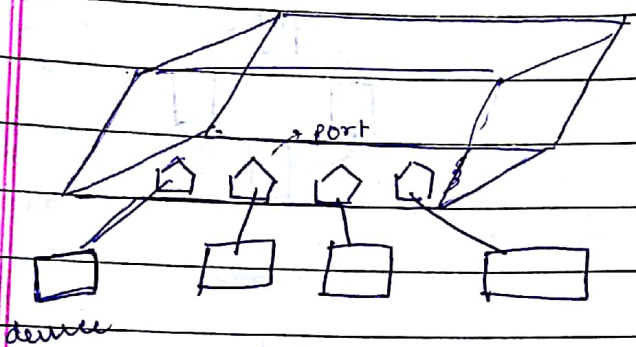
Teacher's Signature

→ Generally, switches are connected through a mesh devices are connected to a switch through a star.

→ Generally, star topology is used in dynamic structure

→ Switch : has multiple port.

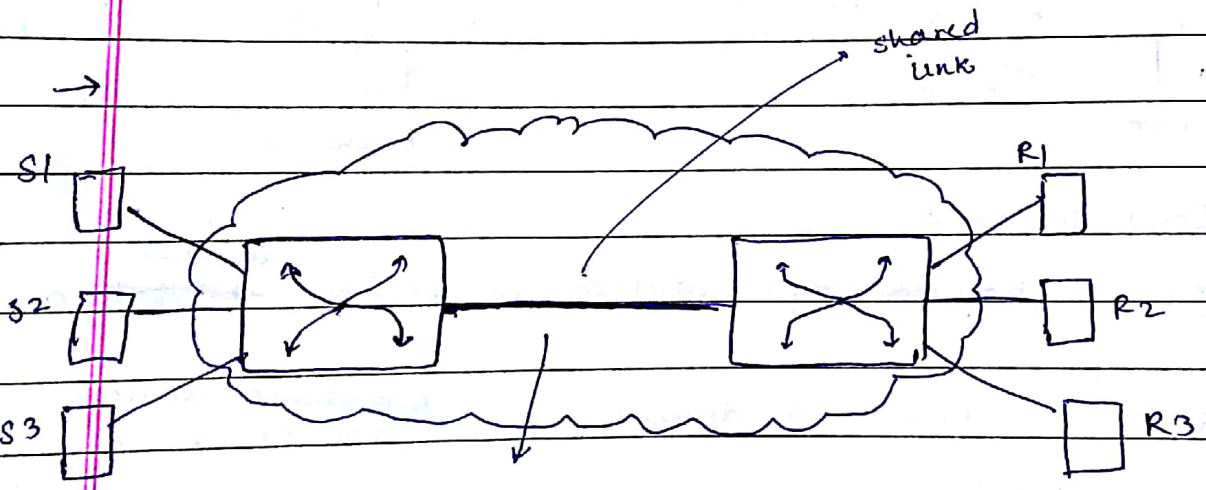
Eg. : Extension board



1 channel for 1 device ⇒ Point-to-Point

→ mesh : for backbone network

→



Think of only 2 devices : are is this link shared by any other device?

No ⇒ Point-to-Point

P1

→ devices outside cloud : user ⇒ hosts

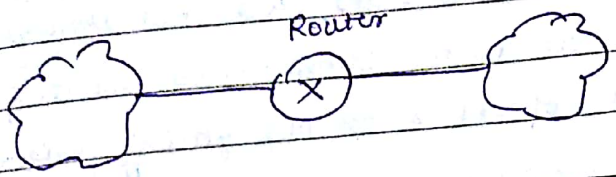
inside : implement ⇒

Cloud : Rep^n of a network.

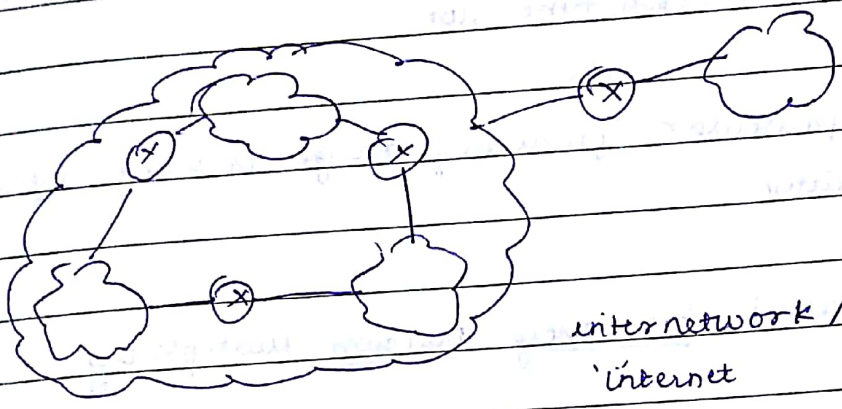
Switch : only providing comm^n b/w devices

Teacher's Signature

Capitalization of I.



Router :- provides comm^{n} b/w networks



internetwork / 'internet

This can be WAN also.

→ 2 protocols communicate through translator

Router ||| internet : TCP-IP ||| Internet

15/1/18

→ In DS, middle link has to be shared.

→ Time Division Multiplexing :

Every device is given a time slot to communicate. Capacity of channel is not divided, time is divided.

Disadvantage :- Time to send the complete data will ↑ 3rd device has to wait for 1st 2 devices. There is a delay.

Advantage :- All are getting equal time & chance. There is no partiality.

Teacher's Signature

If S3 has to transmit & S1 & S2 are not using link, it has to wait for its turn. There is a wastage of time (time slot of S1 & S2 are getting wasted)

Other option →

We can divide the capacity of channel and everyone can transmit at the same time. Each device doesn't have to wait for its own time slot.

Eg. Radio

↳ There is particular frequency range in which signals can be transmitted.

This is called Frequency Division Multiplexing.

⇒ It is kind of static allocation. ↑ng no. of devices is not easy

⇒ If S_1 & S_2 not using ⇒ again wastage (frequency is alloted to them)

↳ Statistical Division Multiplexing :

Time slot ✓

Time Slot are on Demand :- could be 1st come 1st serve.

disadvantage :- If 1 is taking longer time, other has to wait for that much time for data transmission.

This is partial.

So, let S_1 send 1 ~~data~~ unit, then if S_2 & S_3 also have to send, then S_2 will send 1 unit then S_3

~~If S_2 1~~ 1 unit ⇒ Packet

data, which can't be transferred as whole, it transmitted as chunks of data

17) Packet Switching Network :- (Internet)

Teacher's Signature

Switch : kind of Multiplexer

DATE : / /
PAGE NO. :

All packets are sent to the switch (by S_1, S_2, S_3).
Switch has to send all packets on link & then on next switch ⟹ de-multiplexer.

What should identify sender & reciever ? ⟹ address.

↳ network should have facility to broadcast → for everyone (everyone processes it)
multicasting → for some people
unique casting → for single person (add. by your name) (though everybody listens, only you process it)

have to specify addresses.

* Every host on the network has an address.
→ switch will transfer packets to proper receiver

⟹ Speed at which senders are sending packets may not be same. But if switch can manage & forward packets : it just acts as a forwarding unit.

But, it also has capacity. If it can't manage, it will store packet in DS (Queue) : → act as storing & forwarding unit.

→ Switch is not just h/w, it is s/w also.
→ We can transmit packets from diff. route also. Packets are disassembled & assembled back at reciever by a s/w. So, sequence no. should be also there. They can arrive out of order also. So, there can be a time delay

Another option :

2) Circuit Switching Network :- (telephone)

Before transmitting, firstly, a route is decided & all packets are sent through that route only.

→ A set up phase is there already. S/w is transferring msgs Only when all links agree for entire transmission, data transfer occurs

Teacher's Signatur

3 phases :

1) Set up phase
2) Data transfer phase
3) Tear down phase → when data transfer is finished, another transmission can occur using the same circuit.

Switch : ∞ If 1 path don't agree, packets could be sent through another route.

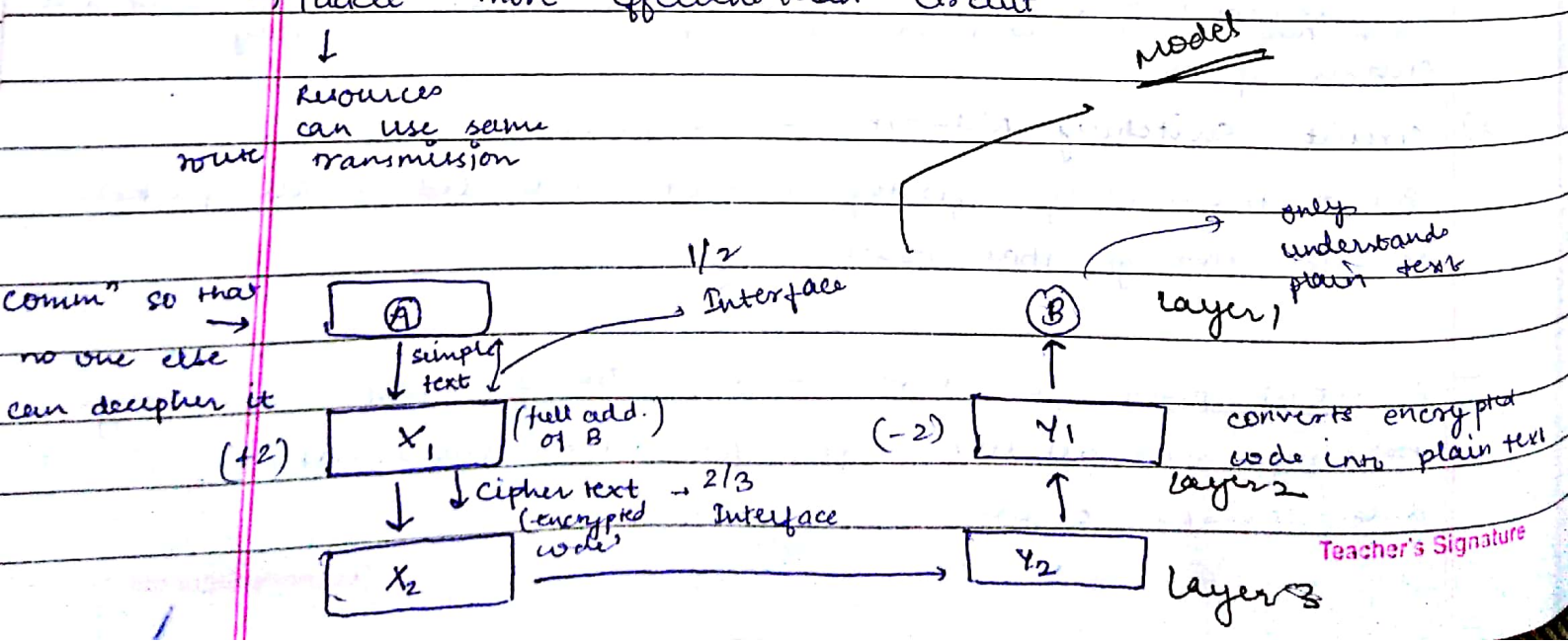
→ Pure Circuit Switching : Only for some dedicated people.
→ Virtual : Not always, only established on demand

→ Interface : prototype of a func°. ⇒ In programming language
eg. : float sqrt (float)
→ Service
for abstraction : hide the implementation's details

→ A network system is also distributed in layers - provides abstraction

17-1-18

→ Packet more efficient than Circuit
↓
Resources can use same transmission route



comm° so that no one else can decipher it →

Scanned by CamScanner

Network models — OSI / TCP/IP

DATE : / /
PAGE NO.:

→ If A writes in Eng., B recieves in French : comm^n not complete

→ at 1st level, both should have English

cipher text

	A	B	C
+2	↓	↓	↓
	D	E	F

→ X_1 & Y_1 should agree on a formula on how to convert plain text to cipher text & vice versa

→ Comm^n b/w A & B is happening in layers : Layered Architecture

→ A has sent the msg, it's not bothered how msg will be sent to B

When layer 1 forwards msg to layer 2, it's work is completed

→ layer 2 is providing service to layer 1.

(lower) (upper)

need not be bothered about how services are provided

→ 1/2 Interface :

Some op^n need two things op1 (pt, addrs) — pt: plaintext

Advantage :

→ Division of tasks

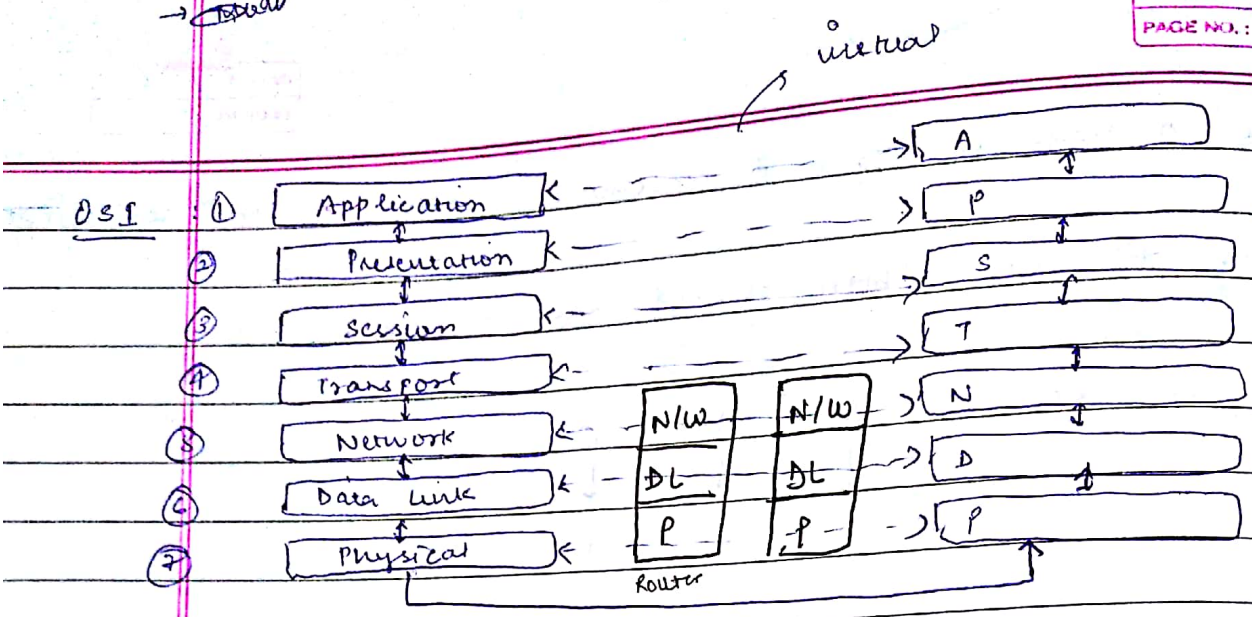
→ Modularity has happened

→ Provides abstrac^n, independence

→ Net. model v/s Net. Architecture

No. of layers, which layer follows what protocol

Teacher's Signature



Open System Interconnection (OSI) Reference Model

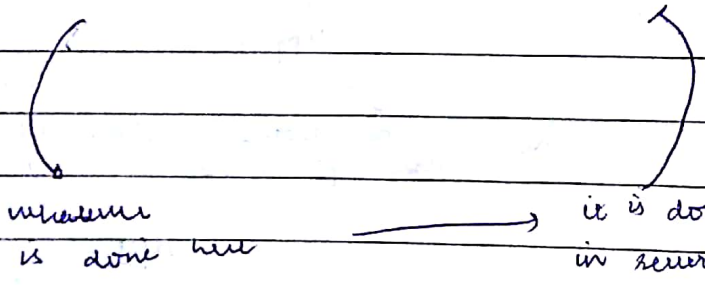
→ Both have to agree on same set of rules → protocol (on same layer)

⇓

called Peers

→ formula used by x_2 for conversion, if it is not used in reverse order, comm^n won't occur

same formula



(7) Physical layer : concerned of Physical medium (Modulation / Demodulation, etc.)

(6) Data link : ~~Responsible for~~ how to go to that route → decided by data link

Kaunsa route lena

h : Network layer decides

Teacher's Signature

→ which port should be used to transfer ~~dat~~ packet to dest^n ⇒ Network Layer

How to transfer :→ Data link

data ⇓
also, error should not arise

Many packets could arrive at switch which can cause congestion. Data link also manages flow & Control

⑤ Congestion : managed by Network layer.

⇒ Some commn^s are

1) Simplex : data can transfer ~~at~~ in 1 dir^n at a time
2) Duplex : transmission can occur in both dir^n but only in 1 dir^n at a time
3) Full Duplex : commn^ in both dir^ns.

④ Transport layer :

Router can also have layers
↓

3 layers are connected to corresponding neighbour

only Transport layer is end-to-end
↳ takes care of " ~~error~~, flow of control

•) data Router tak pahucha ya nhi ⇒ Data link takes care
Dest^ " " ⇒ Transport " "
↳ Reacknowledgement

→ Links could be diff. in diff. devices (wired / wireless)
⇓
DL take care of this.

Hubs : Physical layer
Switch : Data link "
Router : Network layer

2-3

DATE : / /
PAGE NO. :

Transport layer :-

⇒ service : some functionality

protocol :
↳ deals with implementation

interface : concerned of service

⇒ Comm^n is happening in layers (not peers). Peers only have to agree on same set of rules (→ protocol) in order to communicate.

→ If upper layer needs to request lower layer to provide network services, it does so using an interface.

⇒ Transport : fragments the packets

→ Transport provides 2 types of services

↓ Connection oriented

Reliable (Once conn^n is established, only then packet are transmitted)
↳ used when all reliable transmission needs to be done

Non-reliable

↓ Connectionless (direct send msg conn^n)

Reliable

Non-reliable (don't check)

→ When it fragments the packet, it gives sequence no. to packets. If certain packet hasn't reached its dest^n, it sends msg to resend that fragmented packet
↳ Reliable transmission

→ We don't want respond to every msg we send

⇒ Multimedia : Connectionless

Earliest network : ARPANET — 2 objectives → IP

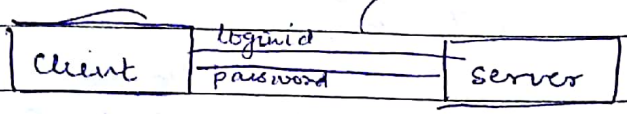
initially, used for only point-to-point ~~can't be used for broadcast~~ multicasting

N/W should always be on (even when Routers were down)

DATE : / /
PAGE NO. :

(3) Session Layer :-

It is an intention to connect to an app^n.



Conn^n established by Transport

When you login, you create a session within that conn^n (gmail)

→ Can have multiple sessions within same connection

(2) Presentation :-

format of information.

presentation of byte stream

(1) App^n :- App^n are created, people use it.

→ It is just a model. Don't have protocol defined for each layer.

IP : Protocol works on packet switching

TCP :

~~Internet~~ :-

→ These didn't have model. They directly made protocols (simultaneously while OSI was being developed).

→ TCP/IP became standardized.

→ Whenever you create an app^n, you've to give a port no.

DATE: / /
PAGE NO.:

TCP/IP

App^n layer
Transport layer (end-end) ← TCP
Internet layer (network layer in OSI corresponds to this) ← IP
Data link
Physical
host-to-Network

→ 3 layered model (made 5 layered for theoretical purposes)

→ In an adhoc way, protocols were just designed & grouped together

OSI : Most of work was done in lower layers. So, present^n & session were grouped together in TCP/IP (App^n layer)

Address :

App^n layer ⟹ Sender Address
↓ when sending msg
TCP — kaunsi app^n ne msg send kiya h, transport layer is concerned with this only.
App^n address is called : port no.
↓
IP → translates add. into IP address (network layer address)
↓
Data link → not concerned with IP add., it is concerned with machine to which it has to send msg
machine add.

⟹ Segments : On Transport layer
frame : Data link
bits / bytes : Physical
Packets : Internet / network

Teacher's Signature

→ standardize for Internet.

bodies that standardize protocols

IETF : Internet Engineering Task Force

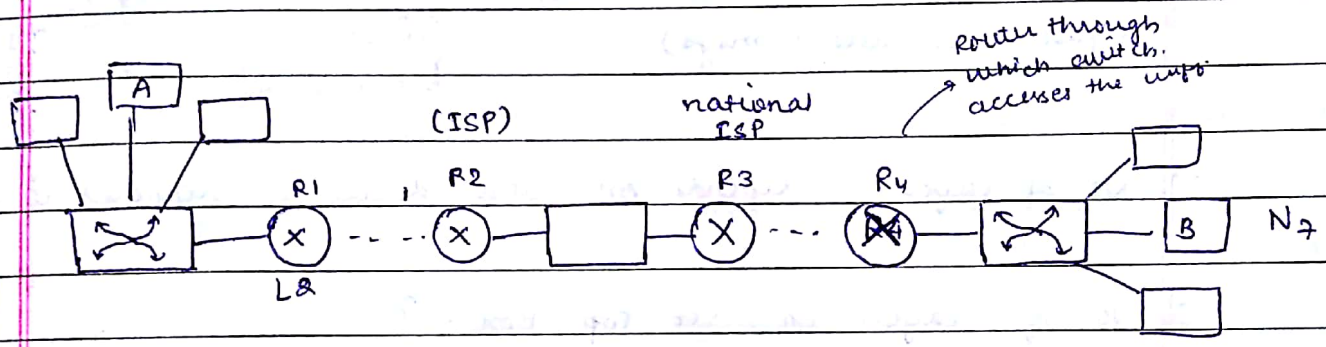
→ RFC : Request for Comment : specification on how protocol should be implemented

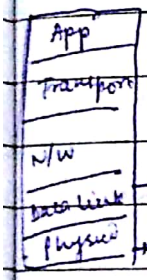
IEEE → deals with lower level (physical level stdardiz^{n}s

→ Implementation can change but services should not change.

22-1-18

Data Link Layer



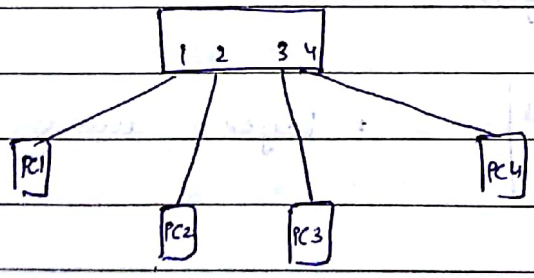


→ Router : basically an internetworking device

→ There is a hierarchy of routers

→ Only source & dest^n machine have all 5 TCP/IP layers

→ Layers in a switch :



There are many no. of ports in a switch.

PC1 communicate → PC4 . 2 options

dev1 : 1) PC1 sends to device, it sends to all PC's, only PC4 processes it. (Kind of broadcasting (don't bother about dest^n))

dev 2 : 2) " , device only forwards it to PC4 ↔ bothered about dest^n.

Teacher's Signature

Network → IP address

→ Only when you need to connect to Internet, you need IP add.

DATE: / /
PAGE NO.:

→ Physical Layer : Just forwards it.
Data link layer : sending info. to appropriate destⁿ.

Hub
Dev. 1
Physical

Switch
Dev 2
Physical
Data link

→ Wherever there is n/w, there will be layers.

→ Mobile Phone (Not smart Phones) : No. of Layers : 1 (Physical Layer)
(just voice calls & msgs)
(to convert calls to signals & send to base station) → (Only transmit & receive signals)
data link

→ No. of layers : depends on what device is supposed to do.

→ No. of layers on Set Top Box : ?
Physical Layer
Data link layer : time to which channel I've subscribed for

→ wherever you have broadcasting, you need data link.
(Earlier phones didn't need it because they were point to point)

→ Router : has 3 layers

Switch : | Data link | Physical | : layer 2 devices

Router : | Network | Data link | Physical | : layer 3 devices

Teacher's Signature

② Determine IP address & Mac Add. of your machine → Add. of machine & hardcored.

* Every device has an address (Mac. Addresses) ↓ Add. of Network Card (Adapter)

→ Now days, we've Routers which have inbuilt switches & vice-versa (talk about pure form only in CN)

How does msg go from source to dest^n.

N7 : IP add. of B

segment

Msg → App^n → sends msg to Transp

Encapsulation → H1 | Msg → Transport → Recieve msg & add its own header (says which app^n it has to be forwarded) { deals with kaunsi app^n se aaya h & kisko app^n ko jaayega

N1 | N7 | H1 | Msg → Network → Protocol which find IP add. of server IP address of source is known by that machine

Payload

datagram / packet

L1

L2 | N1 | N7 | H1 | Msg | Data → Link → needs to know address where of device (next hub) where msg has to be forwarded

ARP : Add Resolution Protocol

↳ Protocol to determine add. of next Hum

Decapsulation

frames → Physical layer → No protocol, sends bits of info. to next hub.

decapsulation → gets L_2, sends data to L_2.

→ Switch : It knows which device is connected to which port no.

How ?

1) Earlier, N/W administrator made table statically.

2) Now, as soon as device is connected, it sends msg, & switch fills table (table is empty initially)

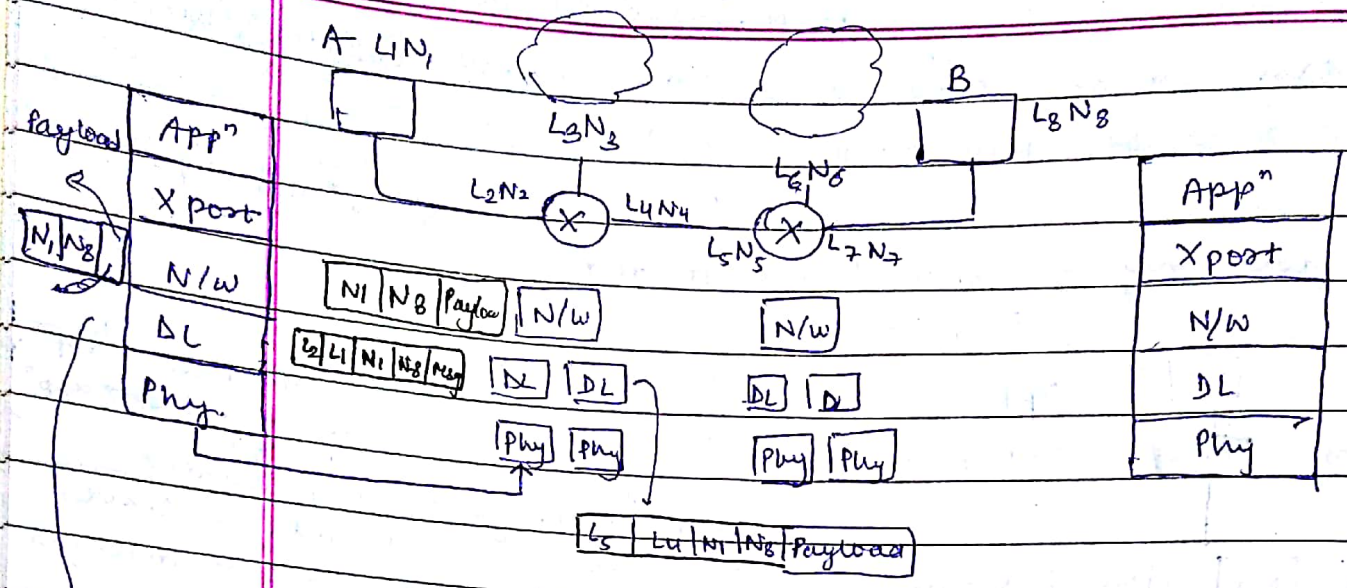
→ Initially, PC 1 sends msg : write it in table, send msg to all device When PC 4 processes it & sends an acknowledgement signal, it will see it & write PC4 in table.

→ L_2 : recieved.

Router connected to many n/ws.

3 line networks se connected → 3 IP add.

Has Adapter (each having Mac Add.) → with each n/w.



N → add. of N/w layer (IP Add.)

L → Link (Mac. Add.)

at DL layer : L_2 | L_1 | N_1 | N_8 | ~~Msg~~ → Payload

→ a Protocol ~~at~~ ARP at N/W Layer gets N/w add. of dest^n.

→ A table is already in n/w ~~whose to f~~ which shows on which hub it has to forward (N_2) to send msg to dest^n (N_8)

Routing table

ARP → L_2

at Router : N/W layer decides where to send it to. (next hub)

So, L_2 | L_1 is decapsulated by Data link layer & data forwarded to N/W layer is :

N_1 | N_8 | Payload → Packet

N/W layer will again check dest^n add. It ~~is~~ also has table that (shows Hub) it has to send msg ~~not~~ to ~~na~~ which hub (N_5) to reach dest^n (N_8)

Now, it has to go to Data link layer. Data Link Layer knows which port (L_4) is used to connect which n/w

Teacher's Signature

ARP → will give MAC Add. L5, packet will now be:

L_5	L_4	N_1	N_8	Payload

→ If medium b/w L_4-L_2 & L_4-L_5 are diff., data link will be accordingly.

→ Protocols in L_1-L_2 & L_4-L_5 can be diff. ⇒ 2 diff. Data links are needed.

⇒ No. of Connection in Router = No. of DL layer = No. of Physical layer

⇒ At 2nd Router → Router of ISP

[Router connected to dest^n ⇓ Router of ISP]

dest^n source source dest^n

DL (1) :

L_5	L_4	N_1	N_8	Payload

DL (2) :

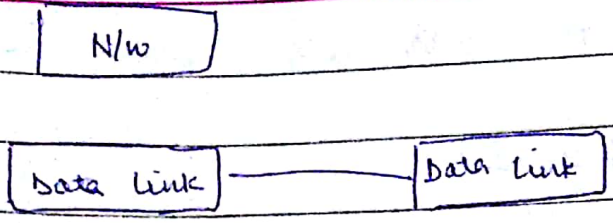
L_8	L_7	N_1	N_8	Payload

* Payload = Msg + Header written by Transport layer

→ At B : app^n ko msg chala jaayega

→ Routing table :→ at N/W layer

→ Domain Name Service : Convert www.... into IP add.
↓
at app^n add.



•) When N/w layer sends packet to DL, it doesn't check on how DL is performing its task → reliable data link

DL → responsible for Error detection & Correction

→ Very large packet can't be send by DL. So, it divide packets into frame.

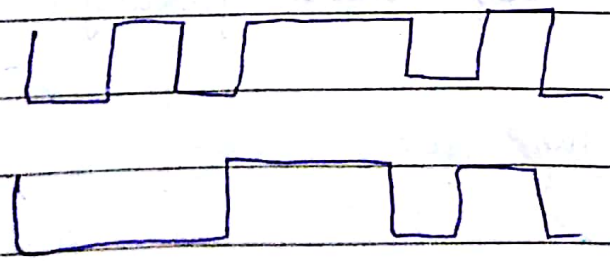
dividing packets into frames + how frames are transferred ≡ Framing

(msg is divided, other part is same)

→ Any info. is sent in analog signal
Only bits will go in msg

→ How bits can go ?

1.) 0 : as low signal
1 : as high signal



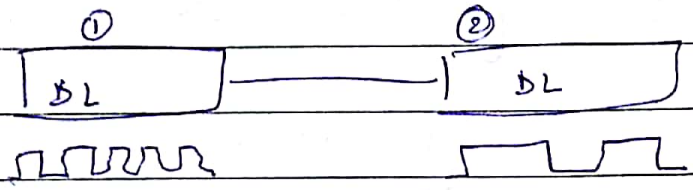
: Error here
⇓
Error Detection

Packet actually has → correspond to Hash

L2	L1	N1	N3	Payload	Trailer

↳ DL layer : also takes care how to find where 1 frame has ended & next has started.

↳

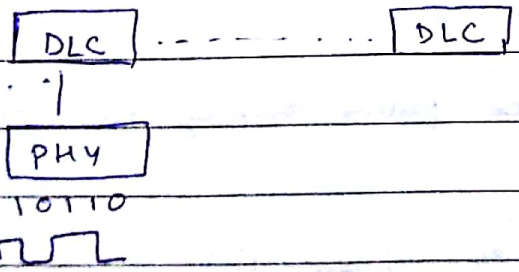


Clock should be synchronised
(frequency should be same)

→ There are queues associated
→ If ② isn't able to accomodate packets, it sends some feedback to ① that slow down the process : Flow Control

25-01-18

Data Link Control - Framing — Error Detection
— Flow Control



Data Link : 2 layers

Data Link Control Mac

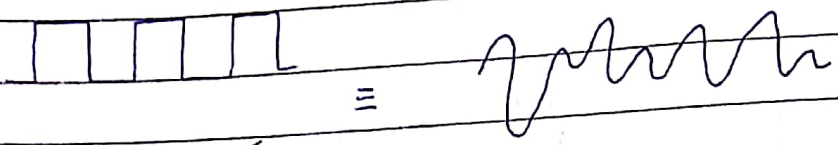
Teacher's Signature

Frame

Fixed size (ATM) — Variable size (WLAN)

3 services provided by DL layer to N/W layer

→ Digital signal (bits are converted to ~~Hoot~~) needs to be transmitted through communication channel.

Digital  ≡

analog signal → modulated

superimposed (amplitude is changed)

→ Next DLC will just receive 1010110

NIC cards have an adapter which does this thing

DLC → Bit stream → [~~Converted to~~ modulated, then transferred] → Bit stream → DLC

Byte stream : 8 bits transferred parallelly.

⇒ If we transfer frames in intervals : bandwidth will be wasted

So, we can send them regularly one by one.

⇒ Framing : identifies which frame belongs to which bit stream.

(or (length : 45 ~~bytes~~ bits))

⇒ Supposing all frames are of same size, we don't need diff s/w to differentiate ~~the~~ frames.

After 45 bits, we know that 2nd frame has started

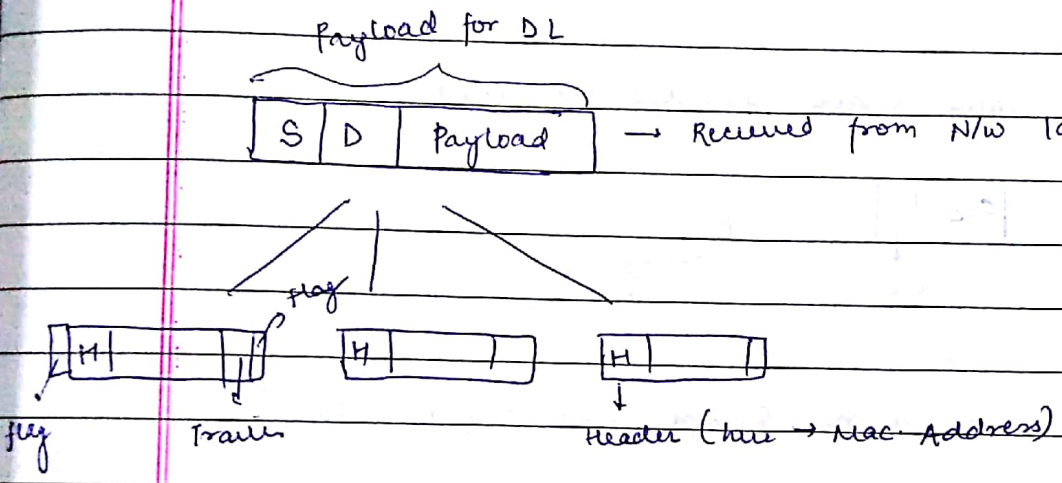
→ Variable : less Overhead, utilises bandwidth

Variable sized frame

Character oriented

Bit oriented

1) Char. oriented :



In character oriented : Frame size is in multiples of 8 bits (each char. is 8 bits) (ASCII value) Even add. are in 8 bits

→ some char. like □ ↓ [,] are never used (ASCII values are known)

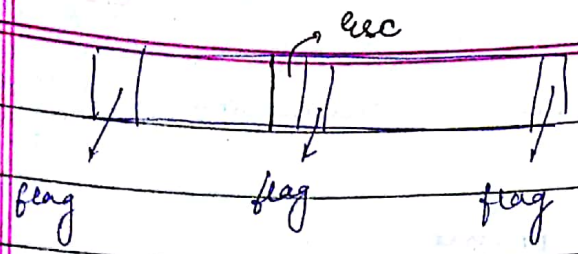
→ How to separate ~~frames~~ stream of bits ? : Using these special characters to form frame

We can insert ASCII value of □ → At receiver's end, as soon as □ ~~is detected~~, its ASCII value is obtained, it'll know frame has finished.

→ DL at sending side will add these special flags

→ It may arise a problem :

• ASCII value of □ could be present in stream of bits.

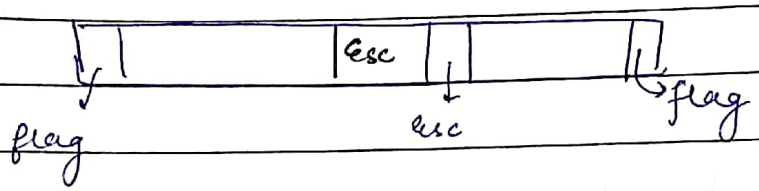


If flag is in ~~text~~ b/w somewhere

TO avoid this, we can add escape before flag

this means iske bad jo flag aayega usko flag nhi lena h, it has meant to be literally.

2) If Escape is some where inserted in text



we can add one more Escape just before "Esc".

This tech. in which extra special ~~data~~ bytes are inserted by source : Byte Stuffing Technique

→ These protocols only work with 8 bits

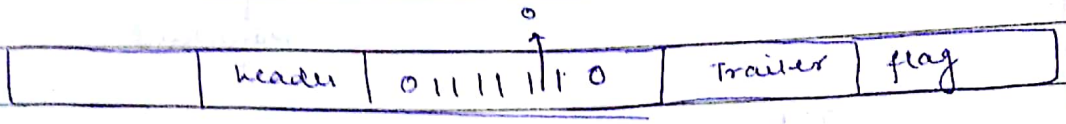
may / may not be multiple of 8

2) Bit Oriented :

Flags should be there to differentiate frames

Problems:

1) get flag in b/w . It does Bit stuffing after 5 1's, it'll stuff a 0.
↳ sender



(after 5th 1)

at receiver's end, it'll remove that 0 & retain the sequence. So, flag will always be unique.

31/01/18

Data Link : Error Detection and Correction

Sender : 1 1 0 0 1 1 0 0 1 1 0 0 0 1 1 0

Receiver : 1 1 0 0 1 1 0 1 1 1 0 0 0 1 1 0 — Single error (1 bit flipped)

OR 1 1 0 0 0 0 1 ✱ ✱ 1 0 0 0 1 1 0 — Burst error (multiple bits flipped)

Why changed ?

↳ Because medium is never free of noise. This presence of noise make error

↳ Wired : lots of error

Relatively error free medium : Optical

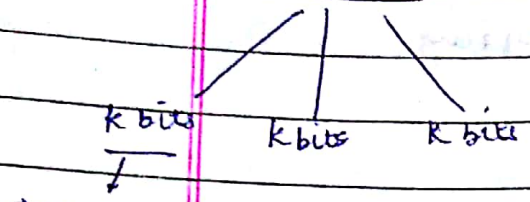
→ Detect ?

With original bits, transfer some redundant bits (depends on data) so that a bit stream can be coded with redundant bits to give unique bits
(similar to hashing)

Teacher's Signature

→ How to add redundant bits ?

Block Codes :



Msg is divided into fixed-size bits (k-bits)

To these k-bits, r-bits are added ↓ redundant bits

This forms n-bits code word

K-bits + r-bits ≡ n-bits

→ k-bits

Max^m data words possible = 2^K —— ①

→ n-bits

Max^m code words possible = 2^n —— ②

Legal codes : ① which have ② as presequence

→ illegal codes : $2^n - 2^K$

Eg. k = 2 , r = 1 n = 3

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

⇓

8

illegal : 8 − 4

$(2)^3 - (2)^2 = 4$

suppose : 001 ⟶ 011

Sender sent this

In b/w it changed

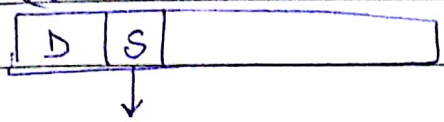
→ if this is not legal (valid code word) ⇕ we can find it is some error.

→ We need to know exactly where error has occured.

→ In Optical fibre / Coaxial cable : only error detection is done
(less noise)

Wired cable : error detection + corr^n

(MAC Add)



have this because if there is some error, receiver will send msg to ask src to resend it.

$$k + r = n$$

→ $k = 3$ $r = 1$ ← Eg. 1

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

1 1 0 0

1 0 0

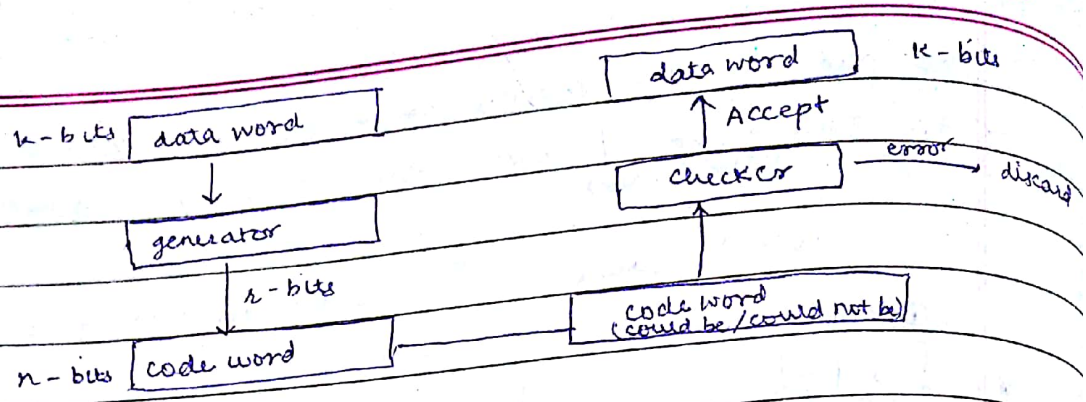
→ Linear Code

⇓

If XOR any 2 codes ≡ diff code word (valid)

→ Sender always sends a valid code word

Receiver has list of valid " "



↳ 0011 → 0011 — (1)

→ Appending : left shifting

→ Most of DL is h/w.

prev. (eg.) ↳ 0011 → 0010 — (2) [Here, redundant bit has corrupted]
sent received

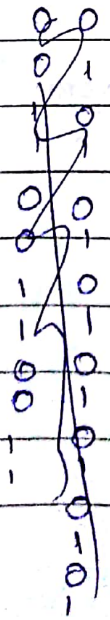
discard (because this isn't a valid code word)

↳ 0011 → 0101 — (3) (2 bits changed)
↳ valid code word

→ All error-det^n mechanism don't work in all cases.

This (above) code is valid only for single bit error.

Q. Let K = 4 bits. Possible Data words :



Teacher's Signature

Even (even no. of 1's) Parity

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

Hamming Distance : ⟶ no. of bits corrupted during transmission

① ⟶ 0 ⇒ XOR of bits sent & recieved

② ⟶ 1

③ ⟶ 2

Minimum Hamming Distance :

If $d_{min} = s + 1$

⇒ no. of error detected = s by coding scheme

In Eg 1 : Since we could find 1 error

⇒ d_{min} in that case = 2

Teacher's Signature

→ ~~Draw the table~~

					Hamming distance
0	0	0	0	) 2	4
0	0	1	1	) 2	2
0	1	0	1	) 2	2
0	1	1	0	) 2	2
1	0	0	0		2
1	0	1	1		:
1	1	0	1		4
1	1	1	0		

d_{min} ⇓ 4

Calculate min^n Hamming distance for any 2 comb^n's. min^n of them will be d_{min}

$d_{min} = 2$

Q. Find d_{min} for :

2 (	0	0	0	) 2
	0	1	1	) 2
	1	0	1	) 2
	1	1	0	) 2

$d_{min} = 2$

⇒ errors that can be caught = 1

* Pick up each code & find min^m no. of 1's.

d_{min} = min^m no. of 1's.

* This is only Parity checker [linear code]

↓

where $r = 1$ ⇒ mostly, even parity

5-02-18

Data Link Layer : Error Detection

Cyclic Redundancy Check (CRC)

→ App^n Layer : actually generates the msg.

1. How the sequence of bits can be grouped into frames
2. How can errors be detected.

Subtraction : just perform XOR

→ CRC : very common in Internet

Ex. k (data-word) = 8 bits r = 4 bits

n = 12

Let k = 10100 111

Left shift by 4 : 01110000

```
                  1
       4 3 2 1 0    11 10 9 8 7 6 5 4 3 2 1 0
       10111     )  101001110000
                    10111
                       11

                      0001
```

2048
512
64
32
16
2672

```
           1287
   23 ) 2672
        23
        372
        23
        142
        147
          5
```

Remainder = 1

Code word : 101001110000 + 1 (Rem)

= 101001110001

→ This is recieved at other end. If this code word is divided by same divisor,

rem = 0 ⇒ no error

rem ≠ 0 ⇒ error detected

→ Divisor : how to calculate ?

CRC : basically works on how to generate divisors so that this scheme be implemented.

Generator :

7 6 5 4 3 2 1 0
1 0 1 0 0 1 1 1 : $x^7 + x^5 + x^2 + x + 1$

Left shift by 4 ÷ (* x^4) : $x^{11} + x^9 + x^6 + x^5 + x^4$

$$x^4 + x^3 + x^2 + 1 \;\overline{)\; x^{11} + x^9 + x^6 + x^5 + x^4 \;}\; x^7 - x^6 + x^5 - 2x^3 + 4x^2 - 2x$$

$-x^{11} \pm x^9 \qquad \pm x^{10} \pm x^7$

$-x^{10} - x^7 + x^6 + x^5 + x^4$

$+x^{10} \qquad \mp x^6 \qquad \mp x^9 \mp x^8$

$x^9 + x^8 - x^7 + 2x^6 + x^5 + x^4$

$-x^9 \pm x^8 \pm x^7 \qquad \pm x^5$

$-2x^7 + 2x^6 + x^4$

$\mp 2x^7 \quad \mp 2x^6 \qquad \mp 2x^5 \quad \mp 2x^3$

$4x^6 + 2x^5 + 2x^4 + 2x^3$

$-4x^6 \pm 4x^5 \pm 4x^4 \qquad \pm 4x^2$

$-2x^5 - 3x^4 + 2x^3 - 4x^2$

$+2x^5 \mp 2x^4 \mp 2x^3 \qquad \mp 2x$

$-x^4 + 4x^3 - 4x^2 + 2x$

→ $x^{11} + x^9 + x^6 + x^5 + x^4 + 1$ → $c(x)$ (If Rem = 1)

$\left(\frac{c(x)}{g(x)}\right) = 0$: No error

$c(x) \longrightarrow C(x) + e(x)$

⇓

error bits

$\frac{c(x) + e(x)}{g(x)} = \frac{C(x)}{g(x)} + \left(\frac{e(x)}{g(x)}\right) \rightarrow R$

To detect error, $\text{Rem.}\left(\frac{c(x)}{g(x)}\right) = 0$ but $\text{Rem}\left(\frac{e(x)}{g(x)}\right) \neq 0$

①

↳ If R is single bit error, it should be of form x^i

⇒ $\text{Rem}\left(\frac{x^i}{g(x)}\right) \neq 0$

Eg. $\frac{x^2}{x}$ ✗ $\frac{x^2}{x^2}$ ✗ $\frac{x^2}{x^3}$

→ It will work well if power in denom. > power in num

→ It 'll always work if 0^{th} bit is 1, ie, $x^0 = 1$

Eg. : $\frac{x^4}{x^3+1}$, $\frac{x^2}{x^7+1}$

→ There should always be 2 terms.

Eg. Error : x^i, $x^3 + x^2$: good divisor or not

It could be because it always leaves a remainder

Eg. $g(x) = x + 1$ ✓

$g(x) = 1$ ✗

(2) If 2 bits are in error :

(at positions i & j)

$\Rightarrow e(x) = x^j + x^i$

g(x) should be such that $\left(\frac{x^j + x^i}{g(x)}\right)_{rem} \neq 0$

$$\frac{e(x)}{g(x)} = \frac{x^i(x^{j-i}+1)}{g(x)}$$

↓

leaves remainder when 0^{th} bit in $g(x) = 1$

→ Let $j - i = 4$

$\frac{x^4+1}{x^2+1}$: leaves rem

→ $\frac{x^4+x^2}{x^{②}+1}$: Rem = 0 → can't check error 2 bits apart

→ $\frac{x^6+x^2}{x^{④}+1}$ " " 4 bits apart

So, we should have atleast 3 terms to check this with 0^{th} bits being 1

Eg $\frac{x^8+x^4}{x^5+x^4+1}$

③ Odd no. of errors

if $g(x)$ has a factor $(x+1)$: It ~~has~~ can detect error

Eg. $x^4 + x^2 + x + 1$

④ Any no. of errors can be possible : (Burst Error)

$$e(x) = x^j + \dots + x^i$$

$$\frac{e(x)}{g(x)} = \frac{x^i (x^{j-i} + \dots + 1)}{g(x)}$$

0^{th} bit of $g(x) = 1$

$\frac{x^{j-i} + \dots + 1}{g(x)}$: should leave remainder

Earlier Conclusions :

→ g: $\frac{x^{j-i} + \dots + 1}{x^n + \dots + 1}$ (more than 2 terms, 0^{th} bit = 1)

Cases :

1.) $j-i < n$ ⇒ can detect all the errors

2.) $j-i = n$

3.) $j-i > n$

Eg. $\frac{x^5 + \dots + 1}{x^6 + \dots + 1}$ ⇒ $(j-i) < n$: always leave remainder

$\frac{x^2 + \ldots}{x^5 + \ldots + 1}$

It is not able to show errors in $\left(\frac{1}{2}\right)^{n-1}$ cases

↑ probability

eg. $\frac{x^5 + \ldots + 1}{x^5 + \ldots + 1}$

P (error not detected) = $\left(\frac{1}{2}\right)^n$

eg. $\frac{x^{13} + \ldots + 1}{x^{17} + \ldots + 1}$: can detect all the errors

eg. $\frac{x^{17} + \ldots + 1}{x^{17} + \ldots + 1}$:

how many errors of less than 17 bits can be detected ? : all $(j-i < n)$

— — — of value 17 — can't

— — ∴ $\left(\frac{1}{2}\right)^{16}$

— — — greater — —

— : $\left(\frac{1}{2}\right)^{17}$ [won't be detected for very few times]

→ For LAN : We use CRC-32

→ Point to Point : CRC-16

CRC

PAGE NO.:

→ Cyclic : In previous class, we stopped at linear codes
↳ When rotated, then also it forms another code
(Additional property)

Error Detection at Xport Layer : Check Sum

14, 16, 7, 5

add sum & append

14 + 16 + 17 + 5 | + 42 ⟶ Receiver

Problem :

16 ⟶ 15

17 ⟶ 18

Subtract all (n-1) terms from n^{th} term
if 0 → no error
not 0 → error detected

Sum remains same ⇒ Error not detected.

That's why it is used at 2nd level of check.

Data link control

1. Framing
2. Error Detection & Correction
3. Flow Control

- Parity check codes (checks for 1 bit error)
- CRC - 32 (checks for any no. of bits error)
- Checksum

detect s errors : hamming : $s+1$

to correct errors, $d_{min} = 2t + 1$

→ 10 10001 —(single bit Error)→ 101010

- 101101
- 100011
- 101100

Compare with all valid. Check which has only 1 bit diff.

* Not used in LAN's (usually resend)

Teacher's Signature

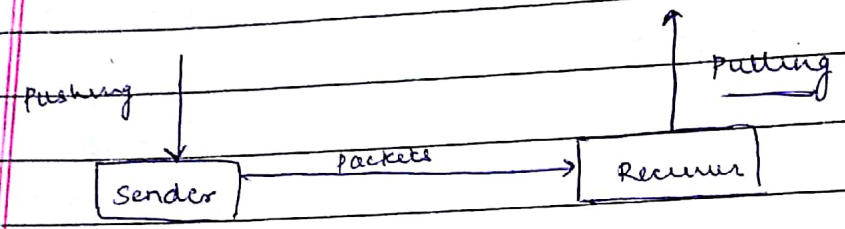
Even Parity

→

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

(1 error or 2 errors ↓ if in diff rows & col)

Flow Control



2 techniques :

→ **Pushing** : Whatever comes to producer, push to consumer as soon as it gets it

→ **Polling** : consumer will send msg when it is ready Only then packets will be sent

* Reciever will send to upper layer only when it asks for packets

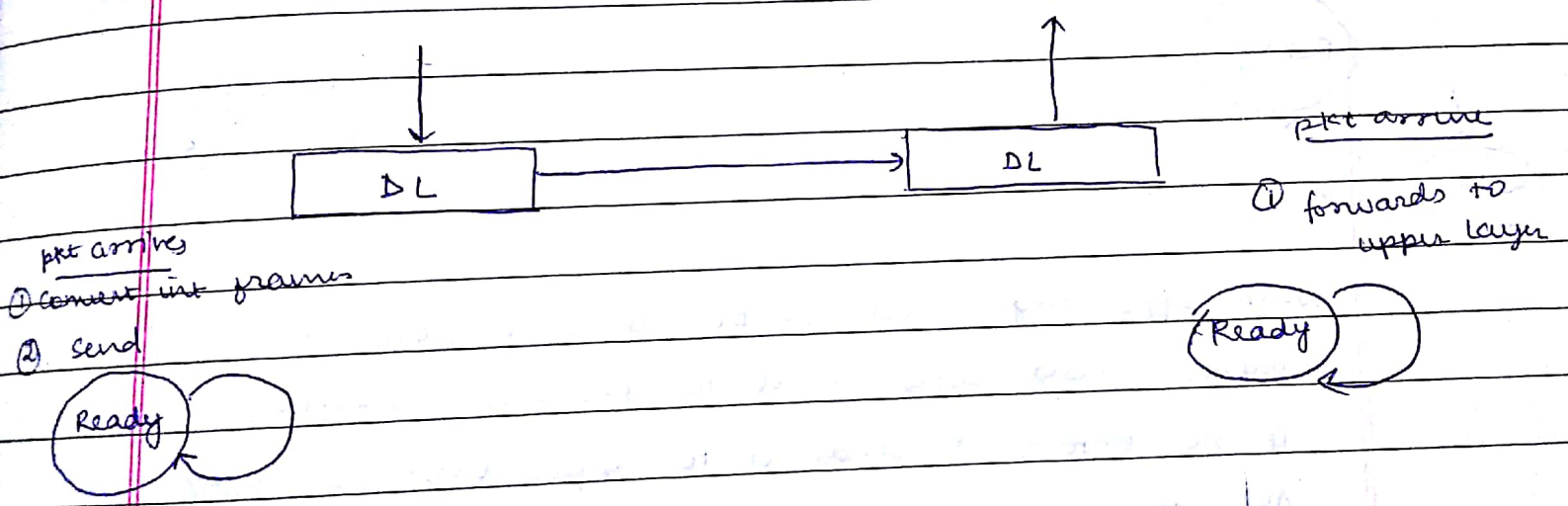
* Flow of Control is required when : Pushing occurs (Rate of producing packets ≠ Rate of consuming packets)

* Generally, both error detection & Flow Control go hand in hand.

→ Receiver will communicate with sender using packets 'ACK'.
If sender doesn't receive : could be due to dropped packeets or error. It can do :
waiting before sending next pkt.
resending lost packet

Protocol for Flow Control

1) **Simple Protocol** : where there can never be errors due to transmission.
DL at Receiver's level is able to manage whatever rate at which sender is sending packets.
'ACK' is also not needed (because of above 2 reasons)



→ Sender has Ready state.
When pkt comes to DL, it
① converts into frames
② sends to o/p gate
& then, again it is in Ready state

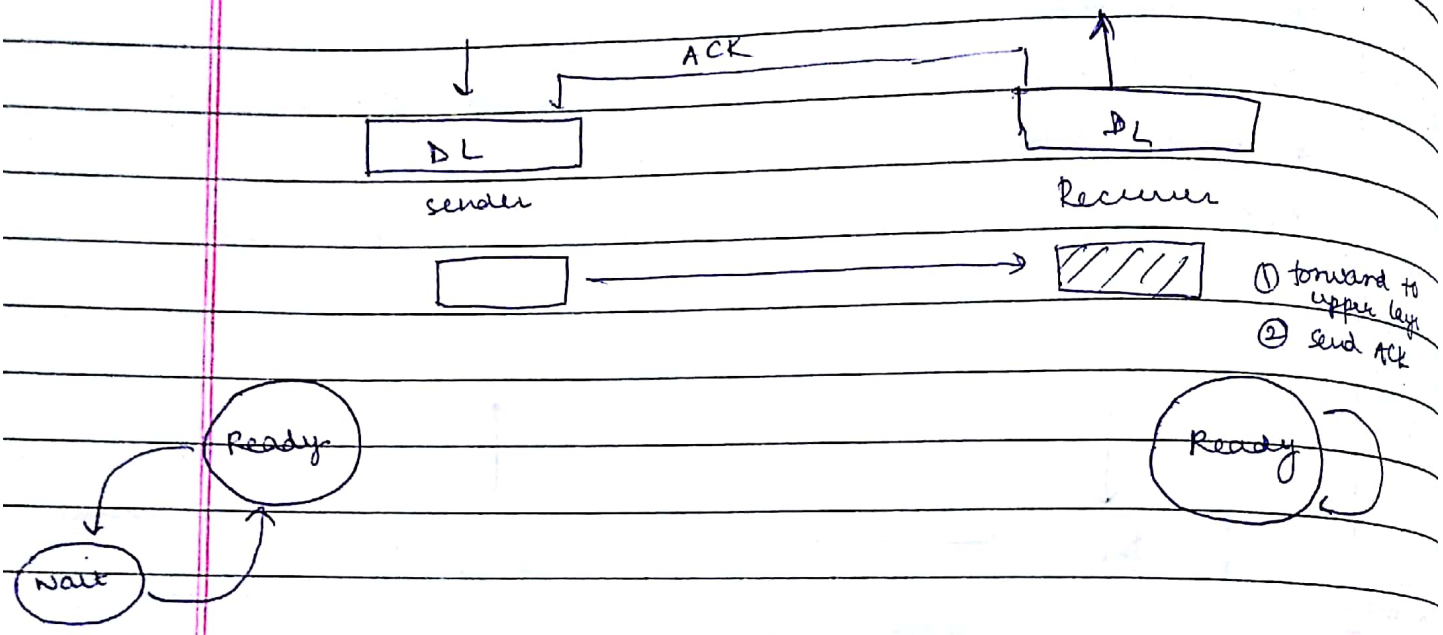
always in ready state.
(pkt arriving occurs). It sends pkt to upper layer

Teacher's Signature

→ Flow Control can be managed with help of buffer

2.) Stop and Wait Protocol :

→ If a buffer to hold 1 packet at each sender & receiver's side



→ When upper layer sends to DL, it will store pkt in buffer, send copy of it to receiver's buffer. If no error, it sends it to upper layer & sends 'ACK' to sender. NOW, buffers are free.

→ If buffer is full, sender will send msg that buffer is full, don't send any more msg.

Sender → NOW, it's not always in Ready State. It has to wait for ACK

Ready state : As soon as pkt arrives, it sends to receiver again

Receiver → It only needs to communicate to the sender ('ACK')

Teacher's Signature

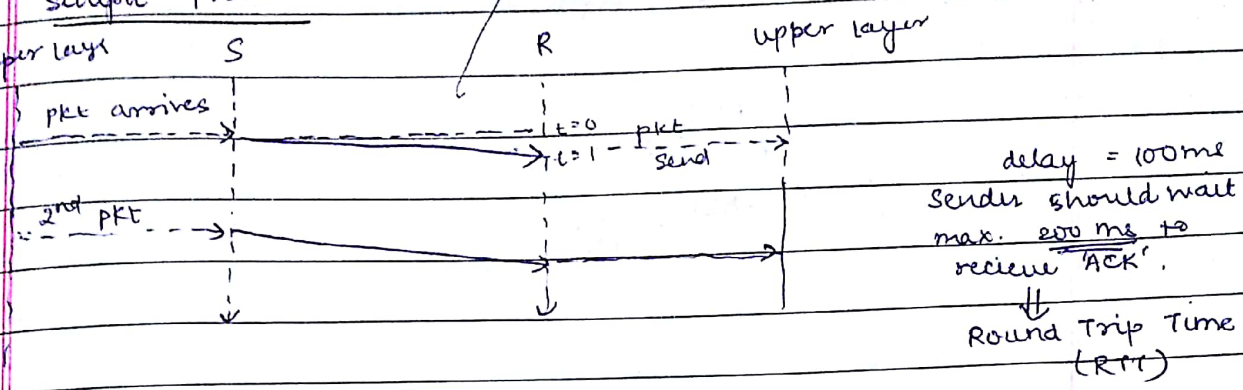
ARQ : Automatic Repeat Request

DATE : / /
PAGE NO. :

Another Representation : Timing Diagram

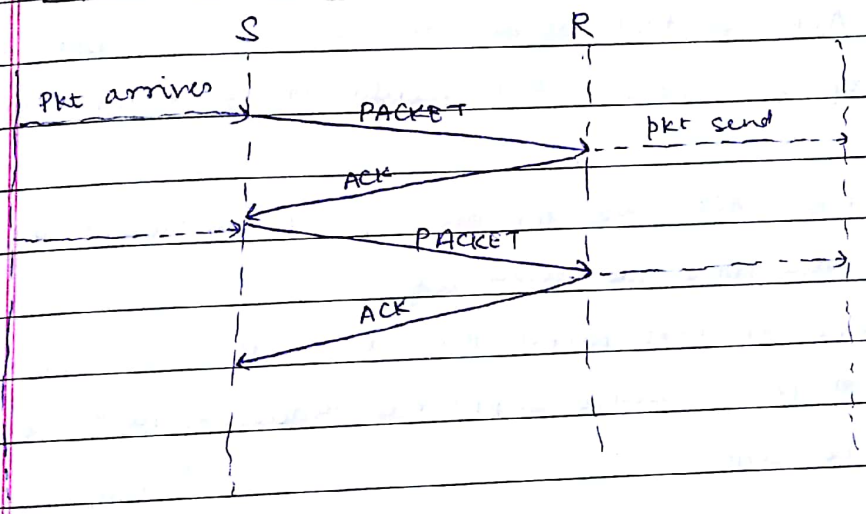
transmission delay

1) Simple Protocol



delay = 100ms
Sender should wait max. 200 ms to recieve 'ACK'.
⇓
Round Trip Time (RTT)

2) Stop and Wait



Problem : a) Packet can get lost

1) either corrupted
2) due to transmission problem, not recieved
3) duplicate packets are recieved & need to be discarded

b) ACK can get lost

Sender has to wait for how much time ? It is given by timer

Sender : 3rd action by Ready state : Set Timer

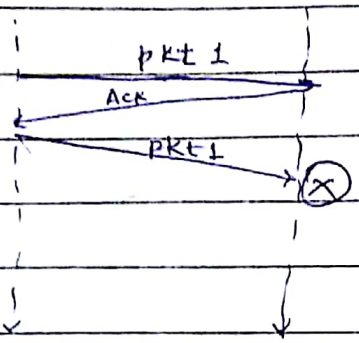
It starts to wait after timer starts. After it stops, it 'll again send same pkt. But now, it 'll be duplicate pkt & they should be discarded. So,

Teacher's Signature

→ How would receiver know it is original or duplicate?

We need 'sequence no.' attached to each packet

Sender : 4th action : set sequence no.



→ Receiver can check this using buffer

Duplicate ko discard karne ke saath receiver should send an 'ACK' so that Sender will send a new pkt with a new sequence no. ['ACK' should arrive within Time-Out]

Sender → It doesn't know 'ACK' was for prev. pkt or discarded pkt

'ACK' can have some additional info :

Sequence no of next packet that it wants

so that sender knows earlier pkt has reached & next one has to be sent.

a) Packet gets lost :

→ Receiver can't send ACK with same sequence no. of pkt which it received. (PKT 1 → ACK 1 ✗)

→ So, it won't send any msg & after time out, sender itself will send pkt again

Sequence No :

We try to use least no. of bits

→ In above case, only 1 bit is needed (modulo 2)

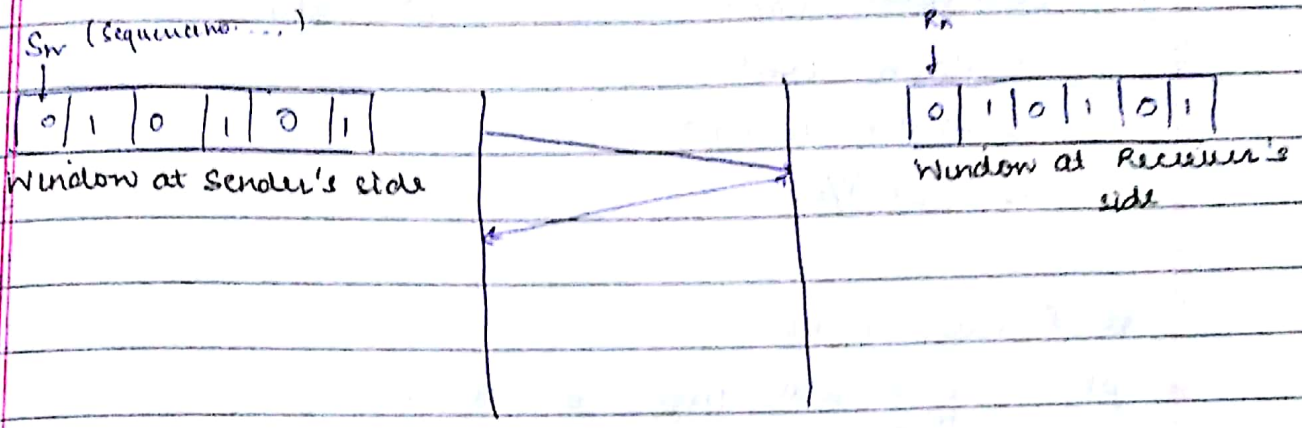
0, 1, 0, 1, 0, ' . . .

Teacher's Signature

Flow & Error Control

↳ Simple Protocol

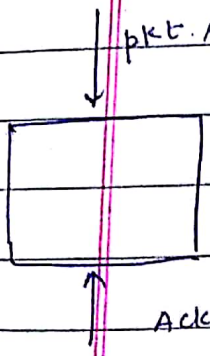
↳ Stop and Wait ARQ



S_n : sequence no. of next pkt to be sent

R_n : -- of pkt for which it is waiting

Initially $S_n = 0$, $R_n = 0$: Expecting packet with seq.no. = 0



getpacket ();
createframe (Sn);
storeframe (Sn)
sendframe (Sn)
InitializeTimer ();
$S_n = S_n + 1$;

At Sender DL :

1) Packet is received from upper layer
2) Ack in -- - lower layer
3) Time Out

} these 3 triggers an event

1) & 2) : handled by handleMessage()

Teacher's Signature

From this pkt, you create a frame

get packet ( ) ; a-pdu

create frame (S_n) d-pdu

store frame (S_n) → Buffer

send frame (S_n)

Initialize Timer ( )

$S_n = (S_n + 1) \% 2$

→ At Receiver's DL

↳ pkt received will trigger an event

f-pdu If $R_n = S_n$ (pkt 2 / S_n) ⇒ it will ~~send ACK~~ & increment R_n (1st) & send ACK

→ When ACK is received, 2nd packet will be sent if it's received from upper layer.

S_n : window of size 1 which is sliding throughout the sequence

Drawbacks :

Transmission med : cylinder

able to send data = vol. of cylinder at a time

But, we aren't using channel efficiently

(Only ~~using~~ transmitting single frame at time)

→ Utilisation of channel is very less.

How many bits can a channel carry : bandwidth delay product

Eg. Bandwidth of channel = 1 MBps = 10^6 Bps

Delay of channel = 20 ms = 20×10^{-3} s

bandwidth * product

$\frac{10^6 \text{ Bps}}{20 \times 10^{-3}} = 10^9 \times 20 = 20000$ bits

Teacher's Signature

→ We are sending only 1000-bit frame (For eg.)

$$\text{Utilisation} = \frac{1000}{20000} \times 100 = 5\%$$

If we send 15 frames at a time = 5 × 15 = 75%

Next protocol will do this :

3) Go Back N (GBN) :

→ Many pkts can be there in transition

→ Have to specify how many bits in sequence no.

n bits → 2^n sequence no. $[0 - 2^n - 1]$

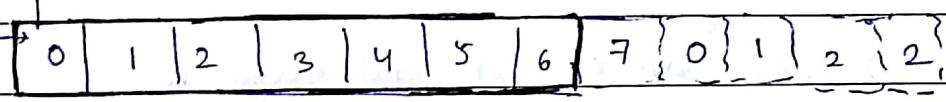
Sender : $S_n = 0$, $S_f = 0$, $S_{size} = 2^{m-1}$

Reciever : $R_n = 0$

Seq is (% m) form

$m = 3$

$\Rightarrow S_{size} = 2^{m-1}$



Stmt : At t = 0, sender will send frame with seq. no. 0

⇒ False. If it hasn't recieve pkt from upper layer, it can't send.

→ Packet has been recieved from upper layer

$S_n = 0$ 1st it has to send & then only it will

→ 3

↳ Multiple packets recieved from upper layer

we assume packets are sent as soon as they are recieved

So, $S_n = 3$.

→ 4 more pkts recieved

↗ $S_n = 7$

If more pkts are sent, reciever doesn't have window to recieve ~~next~~ this pkt.

$S_n - S_f > S_{size}$

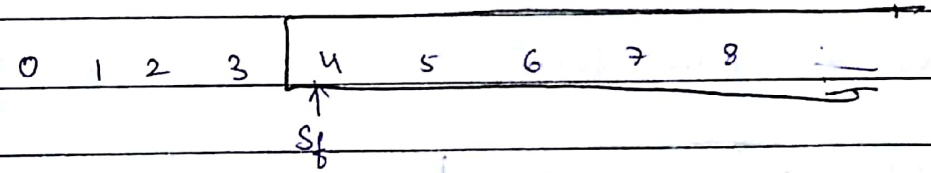
A window is acting as a flow control

→ 6 pkts have been sent

When sender recieves Ack 1 ⇒ $S_f = S_f + 1$

So, window is sliding forward

→ If Ack 1 is lost, Ack 2 is lost, Ack 3 recieved



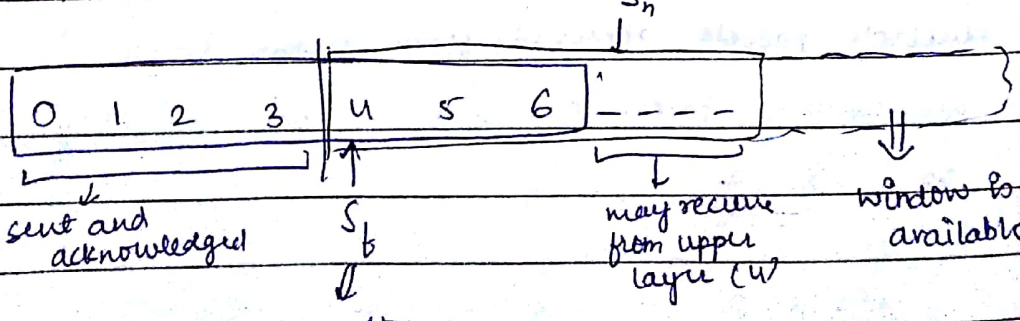
It assumes 0, 1, 2, sab ka ack. mil gaya h.

⇒ At reciever's end :

~~Recize = 2^{3} - 1~~ $R_{size} = 1$

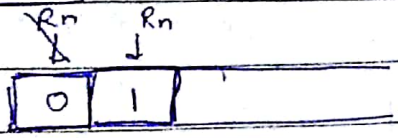
→ If it got Ack for 0, 1, 2, 3 $S_f = 4$ $S_n = 7$ &

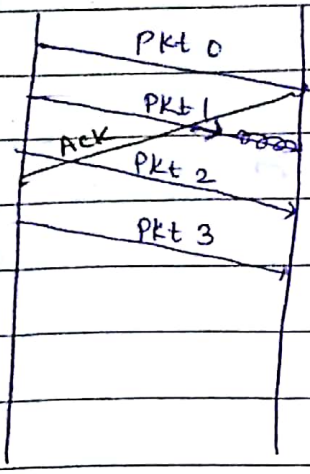
Ack for 4, 5, 6 aren't recieved



Teacher's Signature

Reciever :





Pkt 1 lost

Pkt 2 arrives → Rn ~~doesn't expect~~ ~~can't~~ it, discard it.

* Problem : Lots of pkts discarded

* Time Out : It 'll send S_f to (S_{n-1}) packets again

⇓

Go Back N Protocol

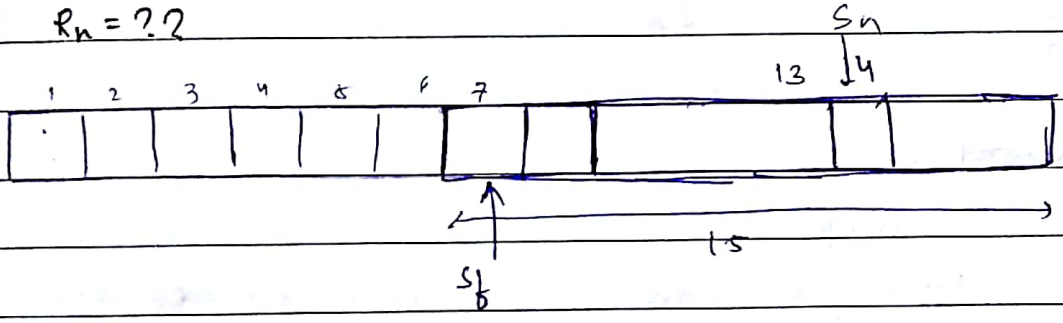
$S_f = 0$

$S_n = 4$.

Eg. $m = 4$. $S_{size} = 15$

5 packets have been sent by sender. $S_f = 7$ $S_n = 14$

$R_n = ??$



$R_n = 7$ ⇒ 7th ka ack. nhi mila

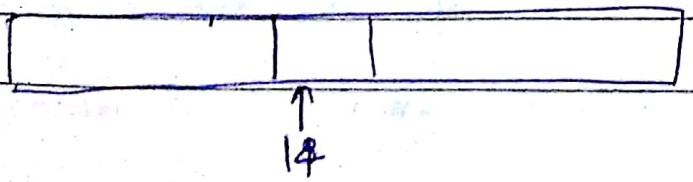
⇒ R_n is expecting value of 7

$S_n - S_f$ =

→ sender receives ack for 13. and gets 6 more pkts from upper layer.

$S_f = 14$

14



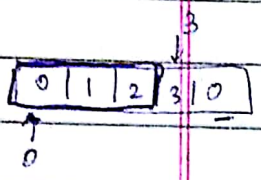
$S_n = 4$

Teacher's Signature

S_n : 14 + 6 = 20 % 15 = 4

$m = 2$, $\rightarrow S_{size} = 3$

All ack. lost

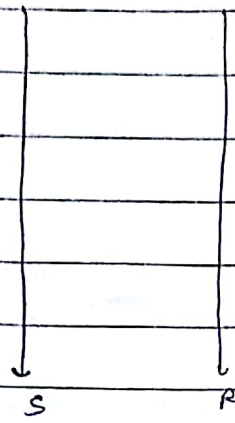


$R_n = 2$

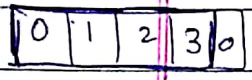
can receiver distinguish b/w 1st 0 & 2nd 0

$R_n = 3$

3 भेजेगा



$\rightarrow$ $2^m (2^m - 1)$ se kam hi hona chahiy



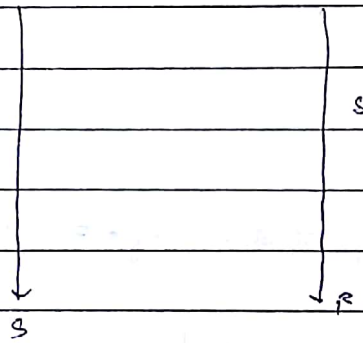
$R_n = ?$

$R_n = 3$

S → Waapis 0 bhejega kyuki No Ack

$R_n = 0$

उसको जाएगा नया Packet h



12/02/18 **FLOW Control :**

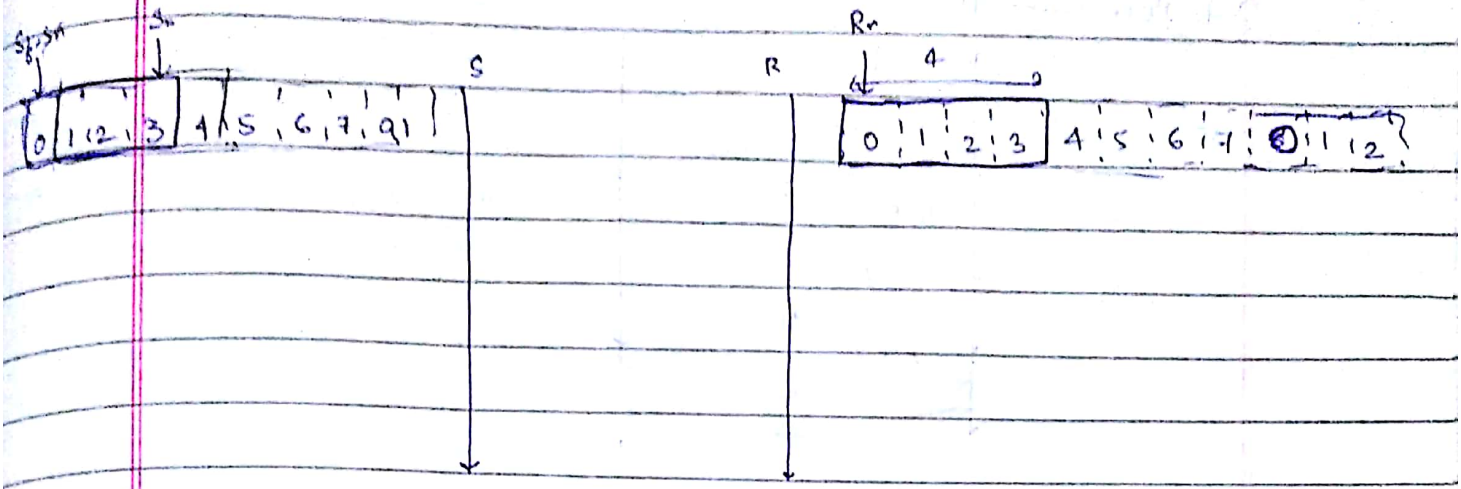
- ↳ simple
- ↳ Stop & Wait (waits for ack before sending next)
- ↳ Go Back N (whenever ack not received, lots of duplicate pkts are sent, later discarded)
- ↳ Selective Repeat

	Go Back N	Selective Repeat
	m - bits	m - bits
Range	$0 - 2^m - 1$	$0 - 2^m - 1$
Sender window size	$2^m - 1$	2^{m-1}
Receiver window size	1	2^{m-1}

Teacher's Signature

Selective Repeat :-

→ m = 3 ⇒ Range → upto 7



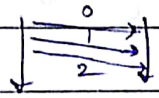
→ Window size : $2^{m-1} = 2^2 = 4$ (at both sides)

→ Using same variables

→ S_n : incremented when it recieves new pkt.

→ If 3 pkts recieved from upper layer

⇒ $S_n = 3$

→ In go back N : 

0 को ACK मिल गया लेकिन नहीं पता मिले
⇓
have to recieve all pkts 0, 1, 2

~~⇒ If it recieved pkt 0 Ack but not pkt 1 & 2~~

→ Here, only those pkts will be sent again whose Ack isn't recieved.

→ When '0' is recieved, it will send Ack with same sequence no. as pkt recieved. (Ack 0 for pkt 0)

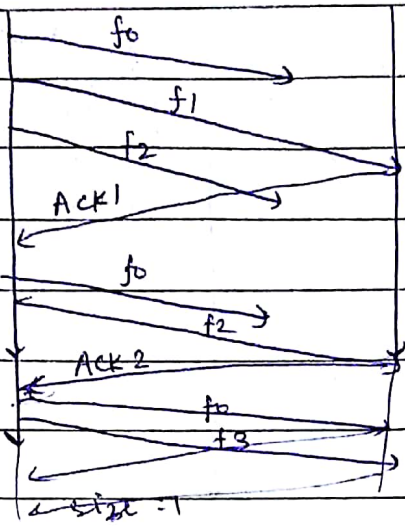
→ At Reciever's side :

If it sends Ack for 0, it moves forward (window)

Teacher's Signature

→ Lossy channels :

1) Packets are lost



2) Go back N : If received f1 : discard it

3) Selective : It receives f1 ⇒ store in buffer for 1.

Since It has not sent ack for f0 ⇒ window won't move

Sends, ACK for f1

Sender marks f1

Once a timer is out ⇒ Sender will see which pkts are lost & send them again.

Now f0 is again lost & f2 received ↓ Window same

Receiver 'll send ack for pkt 0.

Sender 'll again send pkt 0. ⇒ Window still same

If meanwhile, pkt 3 is received ⇒ it 'll send that pkt also as it is in window.

When it receives ACK for both 0 & 3., Sender will check 1, 2, 3. ⇒ ACK for all received ⇒ $S_f = 4$

Window slides forward by 4 locations

⇒ First few frames should be in sequence order. Then only it is sent to upper layer

So, only when window gets full, & receiver sends it for upper layer

Teacher's Signature

2) Ack are lost :

Pkt 0, 1, 2 are sent again after time out.

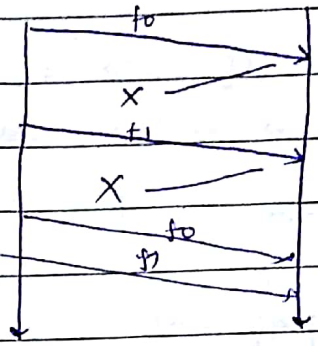
⇒ Why Window size = 2^{m-1} and not $> 2^{m-1}$??

1) $m = 2$

S/R side { Window size = $2^{2-1} = 2$

Range : 0, 1, 2, 3

0	1	2	3	0	1	2	3



R side

0	1	2	3

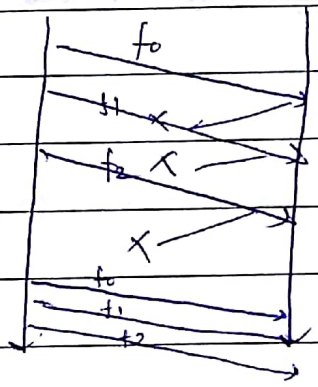
Sends f0 & f1 ⇒ ack of both lost → at R side : Window slides forward

→ after time out ⇒ both f0 & f1 will be sent again : → correctly discarded by R ⇒ it'll send ack for 0 & 1 so that S knows they have been received.

2) $m = 2$

let window size = 3

0	1	2	3	0	1	2	3



0	1	2	3	0	1

window slides forward

→ all 0, 1, 2 would be sent

at R : it expects f0 & f1. Though we are sending old f0 & f1 ⇒ it'll accept it :

⇒ It won't be able to differentiate b/w duplicate pkts and new pkts.

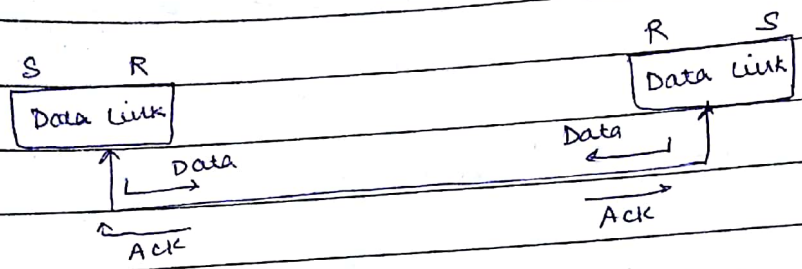
Hence, Window Size : 2^{m-1}

Teacher's Signature

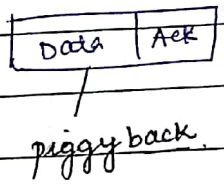
* It is actually window size that is responsible for flow control

DATE : / /
PAGE NO. :

→

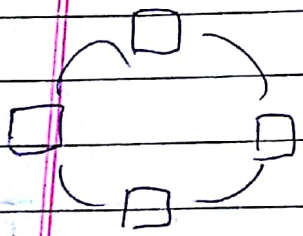


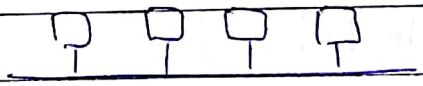
Instead of sending separate Ack msg, Ack of prev. msg. is sent in data ℓ msg only



→ In shared network, without someone coordinating, there may be collisions !z of data

Medium Access Control





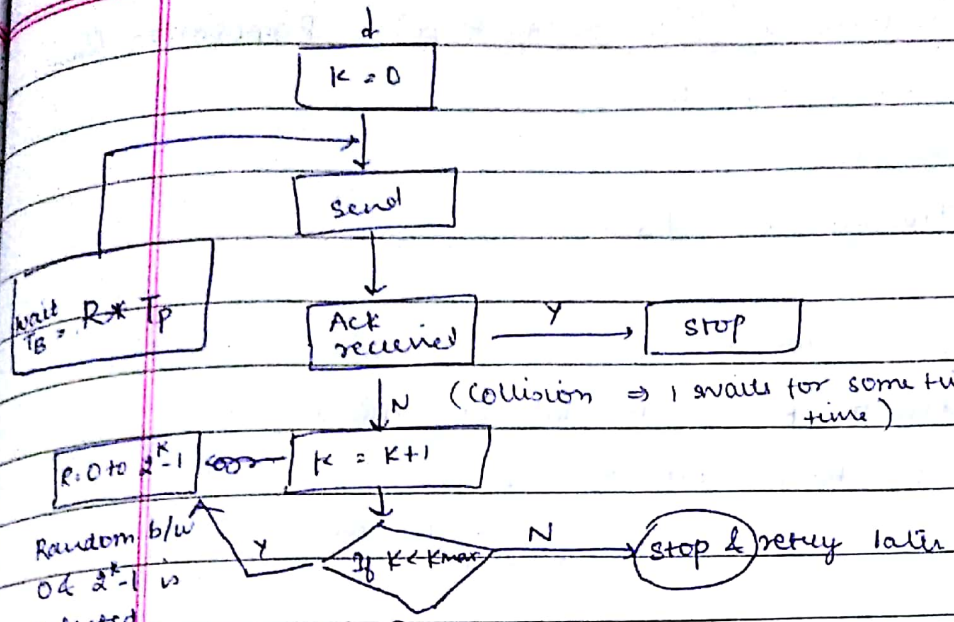
DL : also provides medium access control in shared n/w.

→ In p-ℓ-p : there won't be a problem.

→ In case of shared n/w, ~~they~~ ~~have~~ there may be collision of data ⟶ ALOHA (used to minimize collision)
(signals interfere with each other)

ALOHA : Protocol

Anybody can send at any time. But this may cause collision



So, wait T_B (back of time)
till that random no. even if you've data
(1 ms, 2 ms, ...)

$T_B = R * T_p$

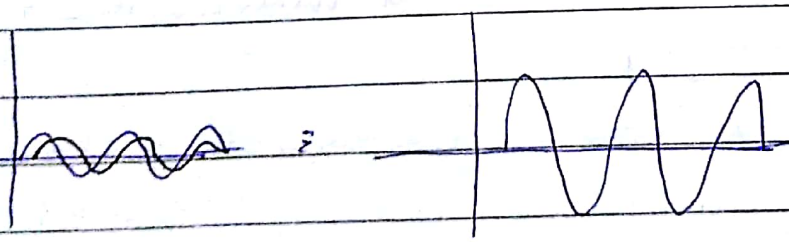
↓
propogation time

→ Slotted ALOHA : Have time slots and everyone will send data at the starting of the ~~st~~ slot.

* $\frac{135}{1000}$: Only 135 are recieved in ALOHA out of 1000

$\frac{350}{1000}$: ... 350

14/02/18 MAC : Carrier Sense Multiple Access (CSMA)



→ There may be problem even in non-shared n/w (if both are full duplex)

Teacher's Signature

↳ how much time 1 bit takes from src to dest^n : Propogation Time
(delay is also included in this)

T_p :→ Propogation Time

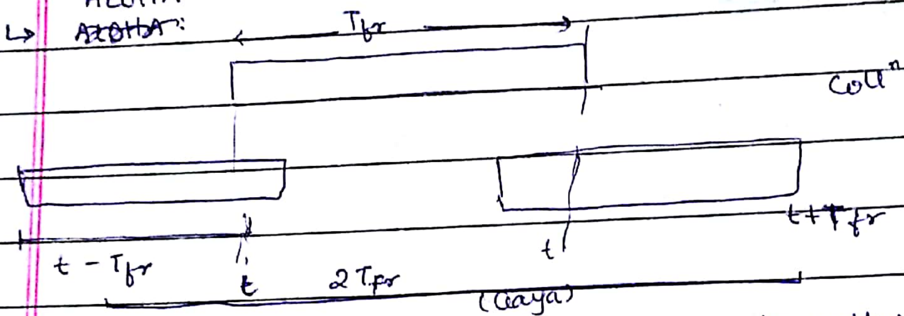
T_{fr} :→ Frame Transmission Time

Eg. length of frame = 100 bits

T_{fr} :→ Time taken to put all the bits on channel

depends on bandwidth of channel
(volume) ↳ No. of bits on channel per sec
(bits / sec)

↳ ALOHA ~~ALOHA~~ :



Coll^n : Even if 1 bit is collided in both ~~fr~~ frame

→ $t - T_{fr}$ ke baad kuch nikla toh woh collide hoga

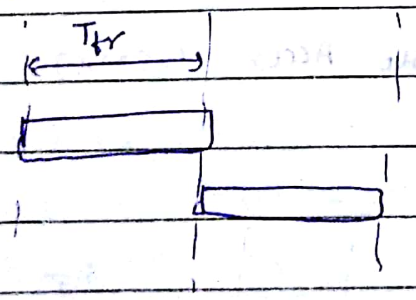
→ $t + T_{fr}$ - - - - - - -

this time diff b/w '$t + T_{fr}$' & '$t - T_{fr}$' = $2T_{fr}$ is called

vulnerable time → chances when collision would happen

↳ In slotted ALOHA :

slots :



they can start only at end of slot

→ vulnerable time = T_{fr}

→ since both collided will transmit again, randomization happen & coll^n decreases

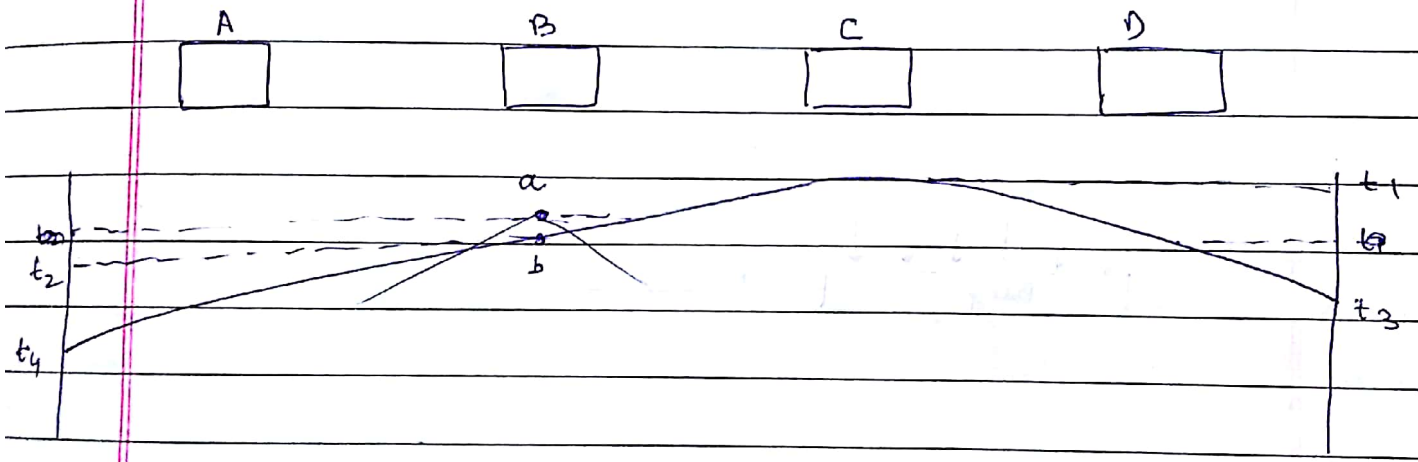
Carrier Sense Multiple Access (CSMA)

→ Carrier signal : super imposition of original signal to transmit in channel

→ Carrier Sense : (Sense) Listen if there is transmission goin on. If not, start transmission

↳ freq. which is noiseless is diff. from with noise.
↳ if altered → becomes noise.

↳ Carrier Sense : knows change in frequency whether 1 station is transmitting or 2 or more (Energy ↑ as no. of stations ↑)



t_1 → C has started transmitting
t_2 → reach B t_4 → reach A (1st bit)
t_3 → reach D

C want to send. It will 1st sense freq. If no other, it 'll start transmitting

→ C can always sense and thus, there 'll be no coll^{n}s → False

at time < t_2 : B won't sense any change in freq. (a)
so it 'll consider the channel is idle. It 'll start transmit.
This may cause coll^n

Teacher's Signature

* Vulnerable time = Propogation Time

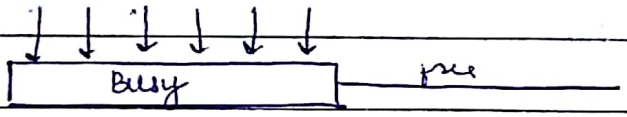
→ CSMA is not collision free

Vulnerable time :		Prob. that pkts reach
Aloha	→ 2 T_{fr}	13.5%
Slotted Aloha	→ T_{fr}	33.6%
CSMA	→ T_p	80% (P & Non-persistent)

How to avoid collisions ? → 3 Methods

↳ 1 - persistent
↳ non - persistent
↳ p - persistent

1) 1 - persistent



Station want to use channel. It'll keep on checking for channel, will be in loop (Battery life ↓) As soon as it finds channel free ⇒ it'll start using the channel.

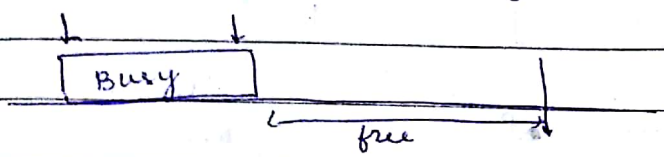
Problem : Chances of Coll^n ↑

2) Non-persistent :

Doesn't check in loop. (Ideally, > T_p)

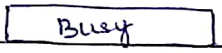
1st Check → if b' busy : waits for some time

Next check → if still busy : waits for some more (more than prev) time



Problem : Channel is free but not being used for some time
(Chances of Collision ↓)

3) p - persistent : (b/w 1) & 2))



p is set → 0.6 (let) ⟹ 60% of time channel is transmitting

[p → some prob. which protocol has fixed]

→ keeps monitoring channel unless it finds the channel busy.

→ If ~~both~~ 2 stations finds channel idle → may be collided.

↳ randomly select no. b/w 0 & 1.

If that no. < p ⇒ that station 'll get channel.

Other channel will wait for time T_p & again select a no.

Problem : If both stations to select same no. → Coll" will occur

It avoids coll" more than 1- Persistent.

CSMA : just an approach which senses a carrier & when to transmit., won't detect the problem

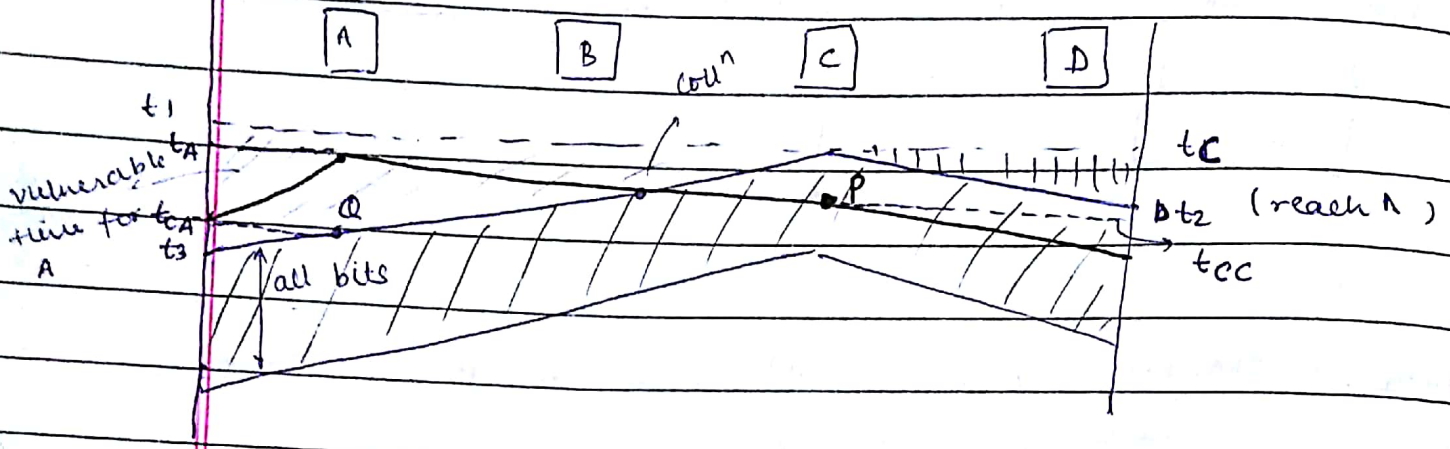
→ It didn't take any action if coll" occured. just, sent again

Teacher's Signature

→ WLAN Ethernet uses this

Collision Detection (CD)

→ What happens when coll^n is detected



C → uses some persistent methode & transmits

D → if senses channel within $(t_2 - t_1)$ ⇒ it'll find it idle

→ vulnerable time ⇒ $t_2 - t_1$ for D

↳ If A finds channel idle b/w t_1 & t_3 ⇒ it transmits

↳ Only at P ⇒ C / u come to know about coll^n

Q ⇒ A " "

→ A C has sent more bits than A.

→ MAC don't keep copies of data.

→ If frame length < Propogation ⇒ saare bits agar transmit ho gaye toh they'll be lost (don't keep copies)

So, frame length > Propogation

round trip → send data & get ack. back

There may be ~~Back~~ trip also.

⇒ frame length > $2 T_p$ → worst case [T_p : A to D]

Eg. Bandwidth of channel = 10 Mbps → Bit rate

Round Trip Time = 51.2 μs

frame length = ?

$$\frac{f}{\text{Bit rate}} = 2\,T_p$$

$$f = 10 \times 10^6 \times 51.2 \times 10^{-6}$$

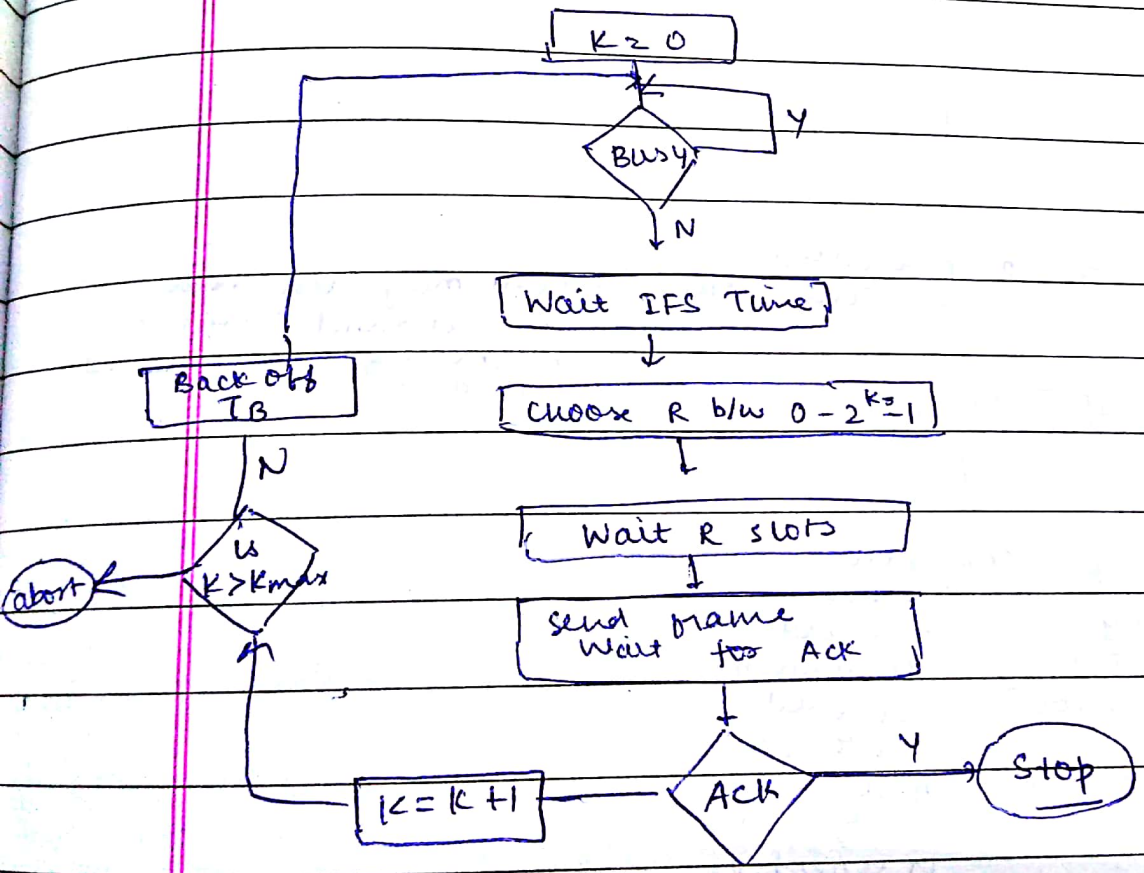
$$\Rightarrow \boxed{f = 512\ b}$$

Once coll^n has occured, other stations (B & D) should also know it has occured.

Jamming Signal : sent by (A & C) which come to know about coll^n. to all other stations so that they know that coll^n has occured & they don't start any transmission

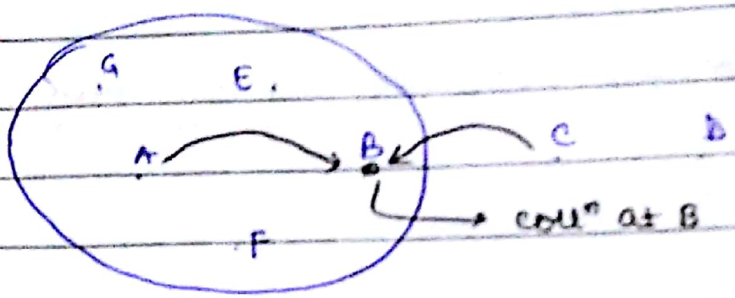
16/02/18

CSMA/CA → coll" avoidance (used in wireless N/w)



→ Have to be in range of Access Point to access the services

→ every station has some transmission range



B is in transmission range of A whatever A sends is received by B

→ Coll" :

B, E, F, G know channel is busy & won't transmit

But C don't know this & can transmit to B.

- Coll" at B

A, C → Transmitters sending to same receiver

⇓

hidden terminal problem.

→ How to avoid this ?

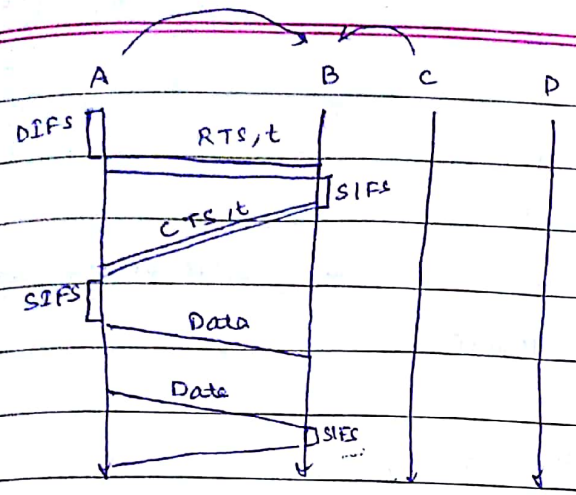
Special frames → Control frame are used

A & B → range of A

C, D → range of B

Ack : smaller in than data pkts
create diff. frame for Ack & n

RTS : request to send



if not received any other RTS
B recieves RTS, sends CTS (clear to send)
RTS ↔ CTS exchange

When ~~A is sending RTS~~ : B, C, D are listening

B CTS : " "

→ It refrains C, D to send any msg.

→ This solves hidden terminal process (para)

They'll (C & D) wait for some time but send again.

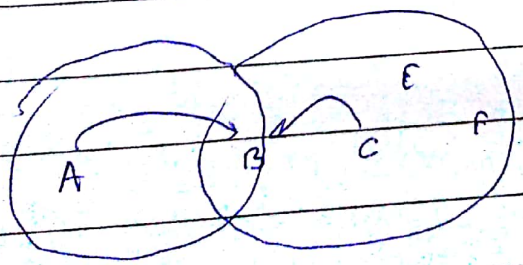
C & D 're to wait until A finishes.

So, with RTS, t will be sent by A app. time to complete transmission B will sent CTC with same t so that C & D also know the time.

NAV : N/w allocation vector

↓
variable in which this time is stored

→ Why vector : because there can be more stations communicating in other transmission ranges other than these 4 ⇒ so it actually makes table



~~But E & F can transmit~~
Agar B recieve kar rha h, toh C, E, D kabhi transmit nhi kr skte kyuki wo A ko bhi milega

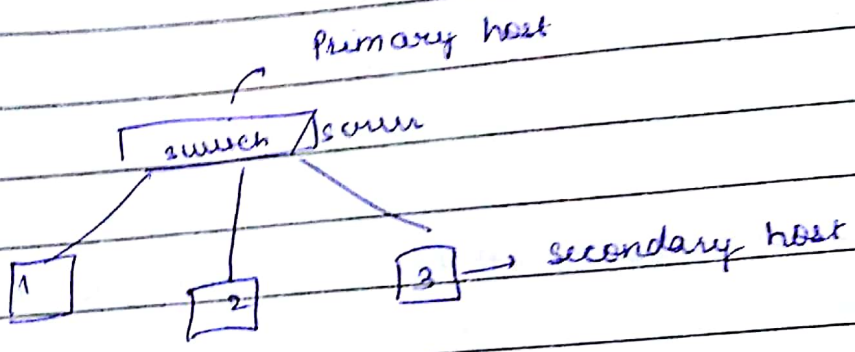
Controlled Access

① Centralised control (polling)

→ Till now → contention Access

→ polling : controls who is accessing & who is providing

→ generally possible in star topology

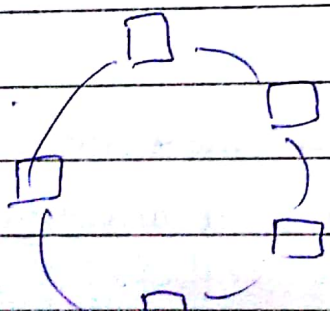


→ switch gets pkt from 1 to be sent to 3. If switch asks 3 to send Ack. After then, pkt will be sent

→ switch 'll ask stations to send.

② Time slots :

③ Token Passing :



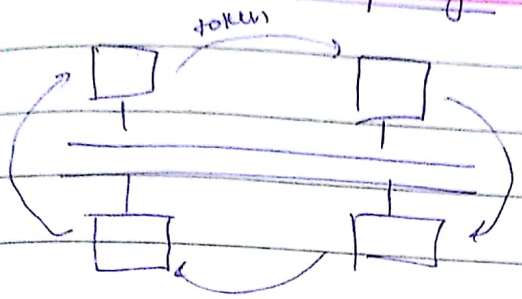
same dir^n

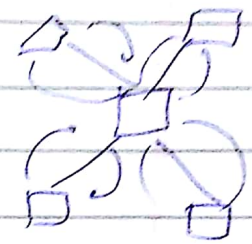
Ring topology : token topology

as it gets token, it 'll pass further if doesn't want.

Physical Topologies

Token:





Bus Topology

⇓

Physical top. is bus but logical is Ring

⇓

Ring possible only using primary