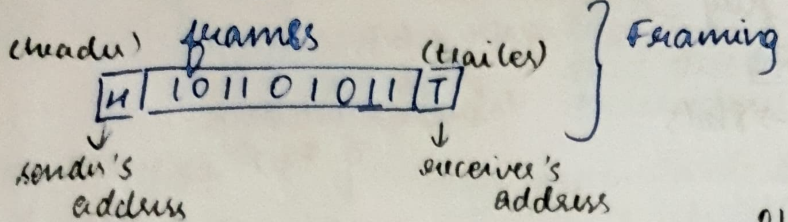
Data link control services

→ finds best route for data transmission
→ links the mac addresses
→ encapsulates datagram & transfers it in the form of frames to physical layer



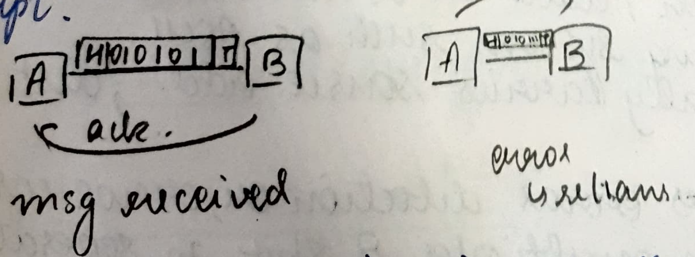
01: B2: _ : _

MAC → multiple access control → for determining which exact device to send it to

DLC services → datalink layer which helps in performing following operation
i) framing
ii) flow control
iii) error control

MAC
i) Encapsulation
ii) Access to the medium

i) Framing: Framing in the datalink layer seperates a message from one source to a destination by adding a sender's address and a destination address.
The destination address defines where the packet is to go & the sender's address helps the recepient acknowledge the receipt.



When a msg is carried in one very large frame, even a single bit error would require the re-transmission of the whole msg.
When a msg is divided into smaller frames, a single bit error affects only that small frame

1byte = 8bits

size of frame

H | 10101010 | T

flag flag flag → divides it into 4 bits each
↳ avoids wastage of BW

4bits 4bits

frame → 8bits

Two kinds of framing:
→ fixed
→ variable size

In fixed size framing, there is no need for defining boundaries on the frame. The size itself can be used as a delimiter.
Eg: The ATM wide area network (WAN) which uses the frames of fixed size called cells

remaining bits

H | Viveka000 | T

cells data

(no delimiter)
↳ differentiates btw sender's add, data & receiver's add.

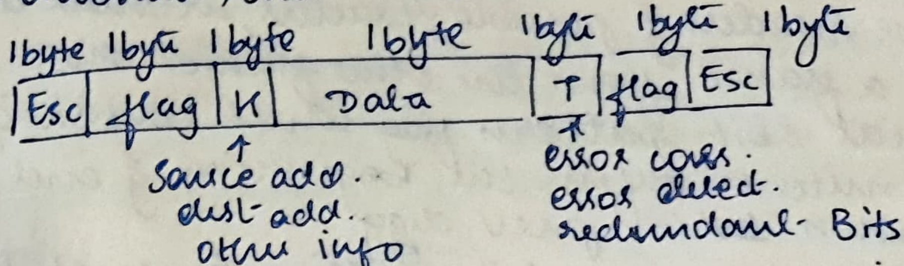
In Variable size framing, we need a way to define the end of the frame & the beginning of the next frame

Two approaches used:
→ character oriented approach
→ bit oriented approach

In char. oriented approach, data to be carried are 8bit characters from a coding system such as ASCII.
The header which normally carries source add., dest. add. & other control info. ~~&~~
The trailer which carries error detection ~~or~~, error correction redundant bits & are also multiples of 8bits to seperate one frame from the next and 8bit flag or 1byte of flag is added at the beginning & end of the frame

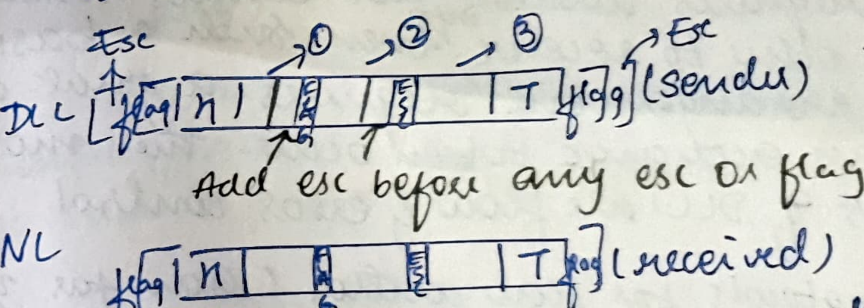
to avoid combination of multiple bytes → flag added
to avoid confusion btw flags → esc added

In char oriented → char oriented



The flag composed of protocol dependant special char, symbol start & end of the frame which is usually of length 1byte or 8bits

Byte Stuffing & unstuffing



Byte stuffing

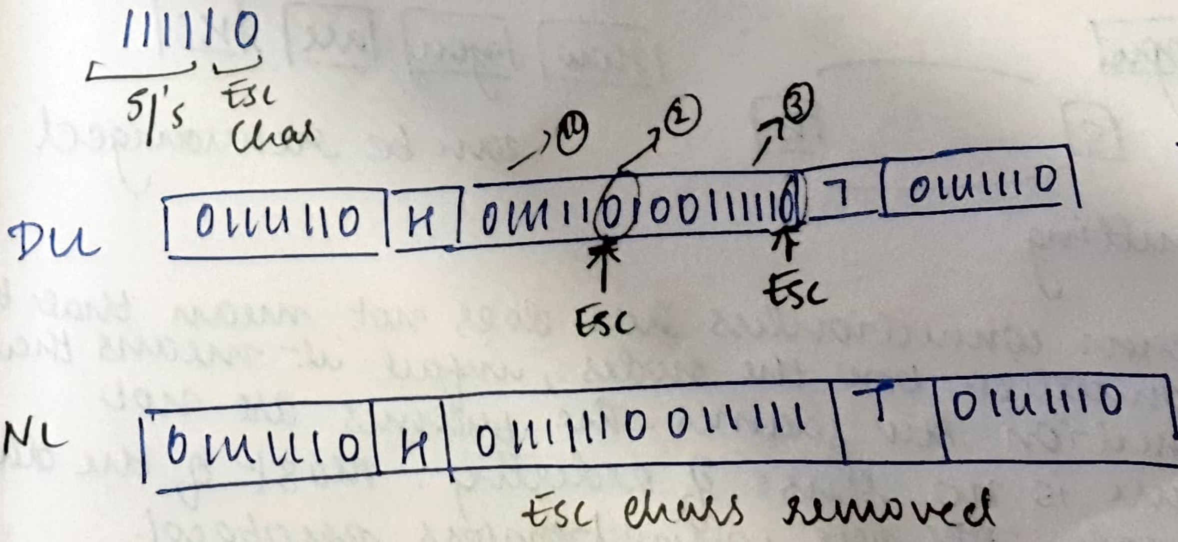
Byte unstuffing
removing the esc which got added at DL

→ Bit oriented frame

By default flag is 0111110

problem: data can contain same sequence

Soln. whenever data with five 1's are seen, you add 0 (consecutive)



Byte Stu

Byte un

In bit oriented framing, ... is a sequence of bits to be interpreted by the upper layer of text, graphic, audio, video, & so on... However in addition to the headers & possible trailers, we need a delimiter to seperate a frame from the other frame. Most protocol uses a special 8bit pattern flag which consists of 01111110 as the ~~dl~~ delimiter to define the beginning & end of the frame as shown ~~to~~ in prev digg.

Here we are performing bit stuffing & unstuffing by adding a zero after every ~~5 one~~ five 1's in a dataframe. By unstuffing, the receiver will check for five consecutive 1's & remove that zero.

Data communication requires atleast two devices working together, one to send & other to receive info. Even such a basic ~~argument requires~~ arrangement requires a great deal of coordination for an exchange to ~~be~~ occur. The most imp. responsibilities of DLL are flow & error control

There are 2 types of protocols for flow control (how far you receive data).

→ connectionless protocol → connection oriented protocol

connectionless: frames are sent from one ~~s~~ node to the next node without any relationship btw the frames. Here frames are independant.

| Hi | how | are | you |

| How | you | are | Hi |

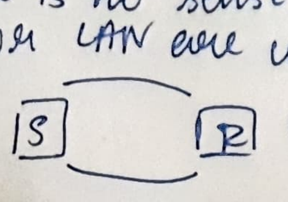
no sequence while transmitting

can be rearranged

Eg. LAN

Note that the term connectionless here does not mean that there is no physical connection btw the nodes, infact it means that there is no connection btw frames. The frames are not numbered & there is no sense of ordering. Most of the data link protocols for LAN are connectionless protocol.

1, 2, 3, 4



1, 2, —, 4

In the first method if the frame is corrupted it is silently discarded, if it is not corrupted the packets is delivered to the network layer. This method is used mostly in wired LAN & ethernet.

In the 2nd method, if the frame is corrupted, it is silently discarded & acknowledgement will be sent. If the frame is not corrupted, even then an acknowledgement is sent for the purpose for both flow & error control to the sender.

<u>Error control</u>

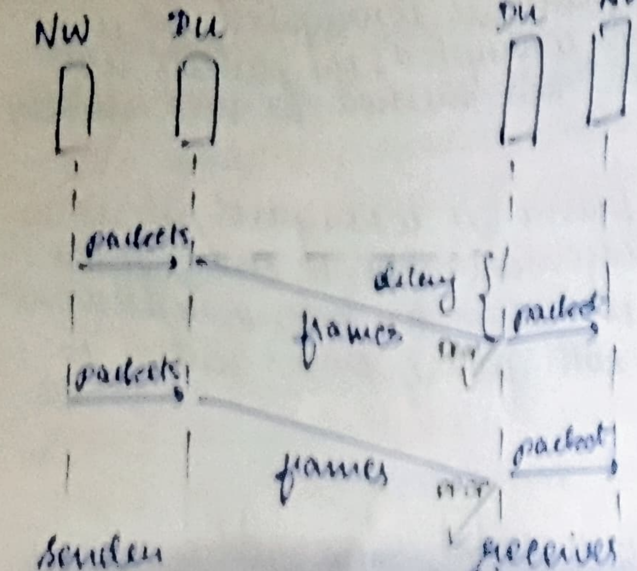
ARQ - automatic repeat request

$\boxed{S} \xleftarrow{ARQ} \boxed{R}$

tells which bit is corrupted.

connectionless oriented approach has 3 phase
↳ set-up phase (ping), transfer phase, tear-down phase

In DLL, we have 4 protocols for error detection & control
i) Simple protocol ii) Stop & wait protocol
iii) Go back N protocol iv) Selective repeat protocol.

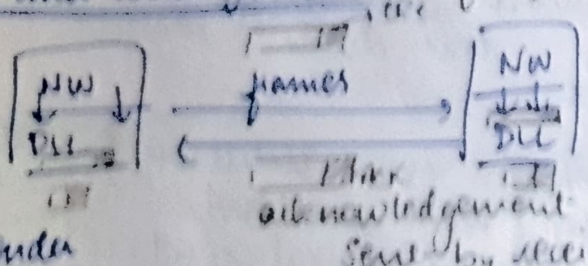


Drop when corrupted
Bcz of delay, line is not straight.

→ wireless connection

Stop and wait protocol (receiver receives valid frames)

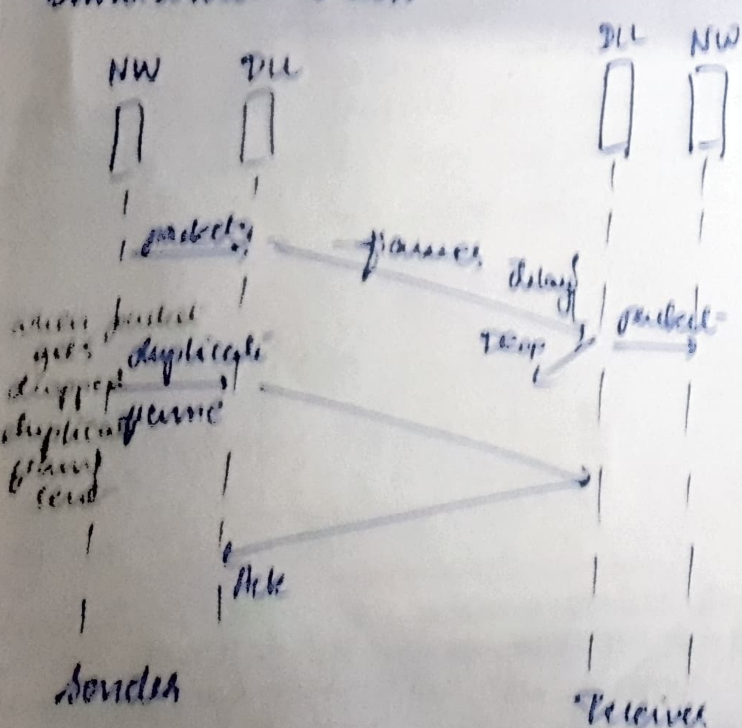
one frame transmitted at a time

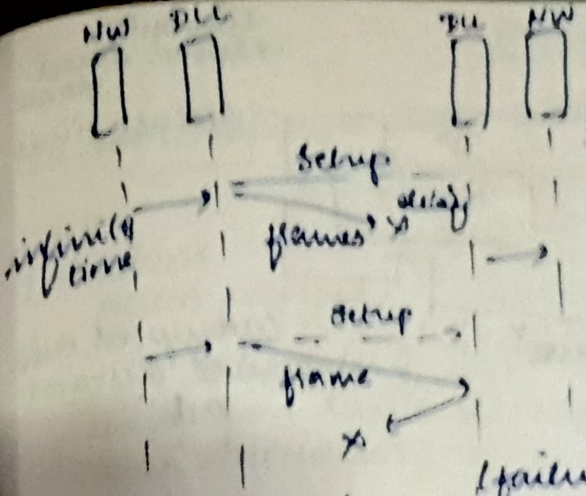


(sender keeps copy of frame sent until acknowledgement received)

instead of dropping whole msg (simple protocol), here we send one frame at a time which gets checked & retransmitted (failure reduces)

Bandwidth is less

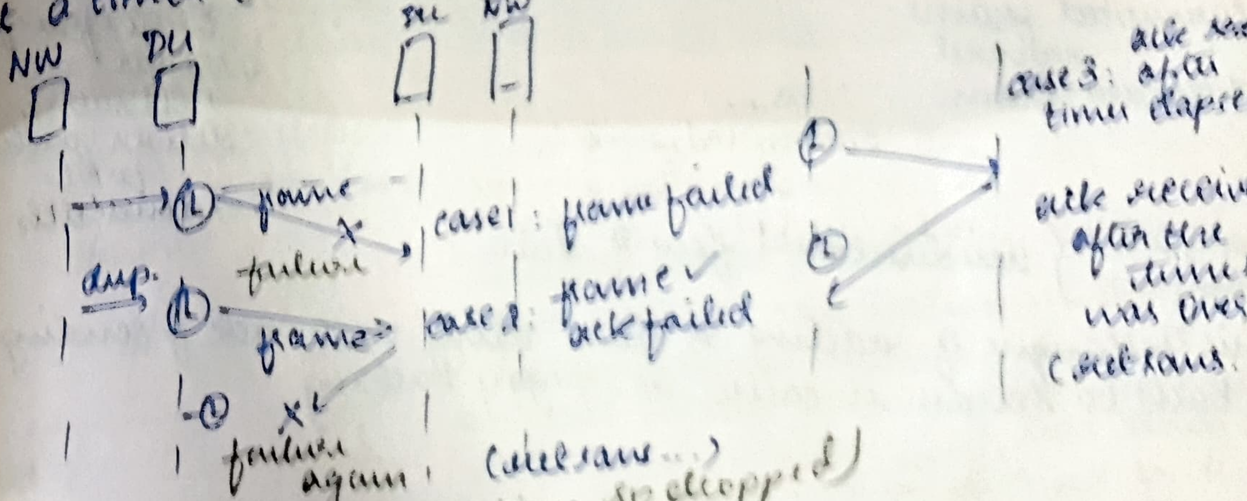




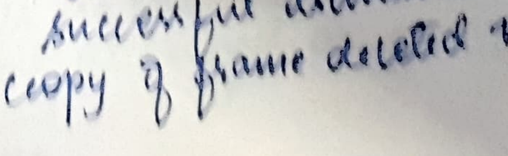
int time
failure in transmitting frame
both sender & receiver wait for infinite time

(failure in ack... sent)
sender waits for infinite time for acknowledgement

Stop and wait ARQ (diag same as prev one)
→ set a timer (acknowledgement received within time)



(didn't reach on time so dropped)



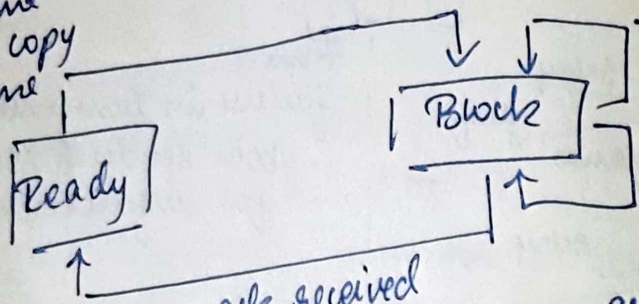
on time
successful transmission
copy of frame deleted now from DLL layer)

packet come from NL
→ make a frame
→ saved a copy
→ & send the frame

[Ready]

packet came from NL
→ make a frame
→ ~~copy~~ save a copy
→ send the frame
→ start timer

→ Time-out
→ resend saved frame
→ restart timer

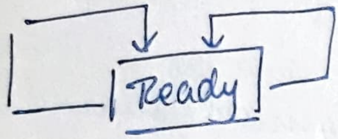


Sender

Error free ack received
→ stop timer
→ discard saved frame

Corrupted ack arrived, discard ack.

Receiver

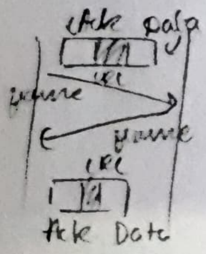


Corrupted frame arrived
→ discard frame

Error free frame arrived
→ extract & deliver packet to NL
→ send ack

Stop & wait ARQ } unidirectional flow of data

This technique of adding a data along with ack & sending back to sender is called as piggy backing



Unit IV

In MAC (multiple Access control)

RAP (random access protocol)

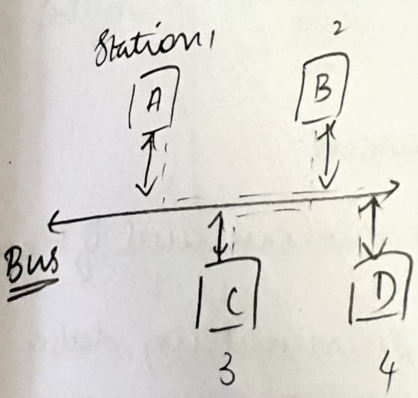
1) ALOHA (Advocates of Linux open source Hawaii association)
2) CSMA (carrier sense multiple access)
3) CSMA/CD — n — / collision detection
4) CSMA/CA — n — / collision avoidance

CAP (control access protocol)

1) Reservation based
2) Polling based
3) Token passing

CP (chanalized protocol)

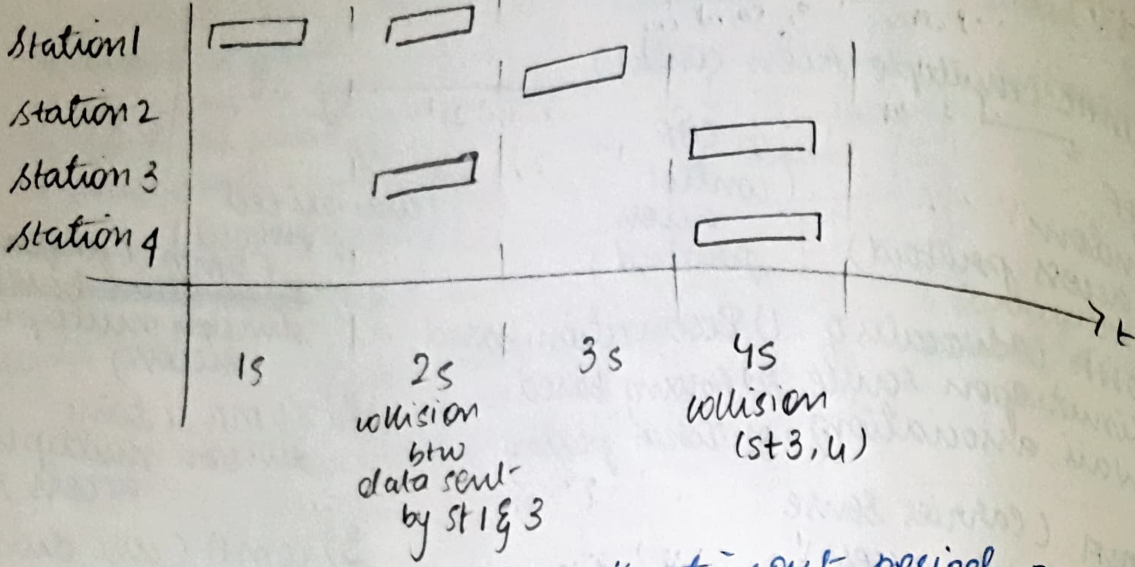
1) FDMA (frequency ~~Reservation based~~ division multiple access)
2) TDMA (time division multiple access)
3) CDMA (code division multiple access)



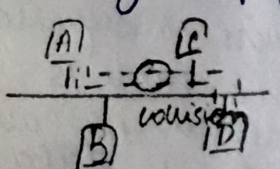
Random access / contention method.
In this method, no station is superior to ~~other~~ another station & none is assigned control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send data.

In a medium access method, each station has right to the medium without being controlled by any other station, however if more than one station tries to send the data, there is an access conflict - collision and the frames will either destroyed or modified.

ALOHA:
The original ALOHA protocol is called as pure ALOHA. This is a simple but elegant protocol, the idea is that each station sends a frame whenever it has a frame to send however since there's only one channel to share, there is a possibility of collision btw frames from different station.



- Pure ALOHA dictates that when the time out period passes each station waits a random amt. of time before resending its frame. This randomness will help avoid more collisions. This random time is called a backoff time TB.
- Pure ALOHA has a 2nd method to prevent collision in the channel with retransmitted frames after max. no. of re-transmission attempts that is called as Kmax. A station must give up & try later

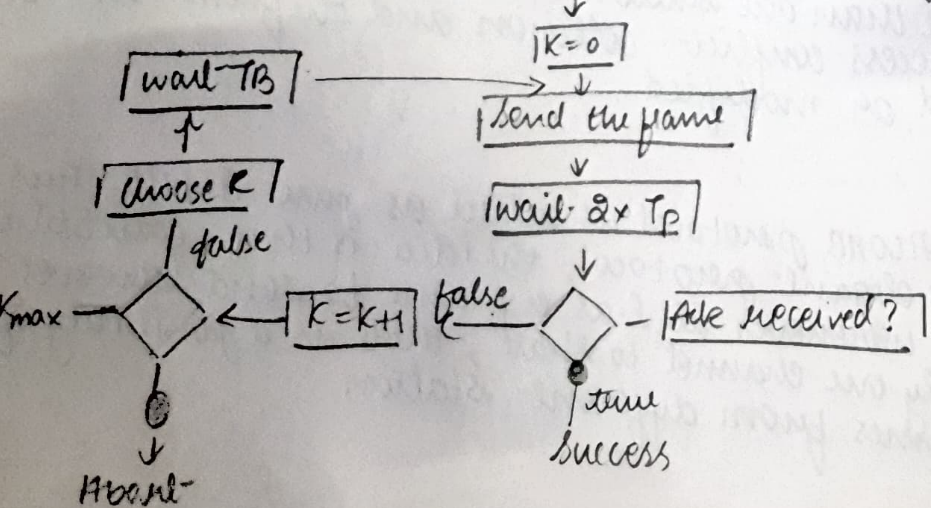


→ collision
→ both wait for 1ms then resend
→ again collision
→ so make them wait for random amt of time

time for all st. is same

→ same RT
→ collision

→ Kmax reached, both stop transmitting data
↳ both reached max attempts to transmit data



K = no of attempts

T_p = max propogation time

T_{fr} = avg. transition time

T_B = Back off time
$R \times T_p$ or
$R \times T_{fr}$

R = random no: 0 to 2^{K-1}

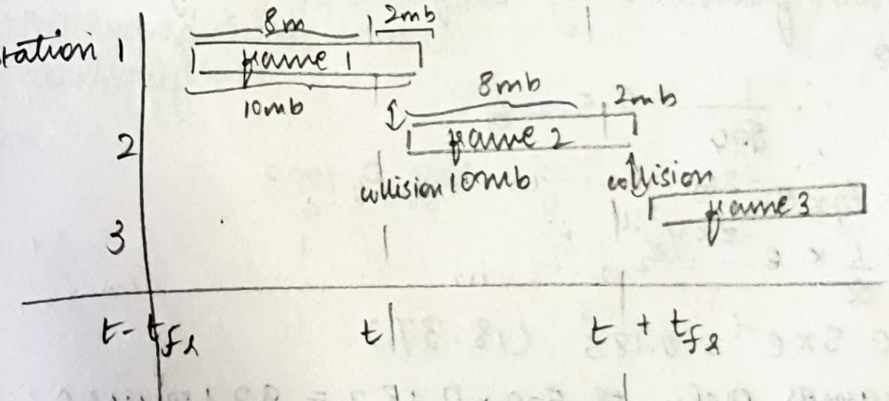
Station A sends
$K=0 \rightarrow$ attempt 1 of sending
A sends fram
sending & receiving ack ($2 \times T_p$) fixed

(CAW) Ack received
success

no ack (false)
$K = K+1$ (attempt inc.) $= 0+1 = 1$
$K > K_{max}$ (check cond.)
$1 > max$
(false)
$R \rightarrow 2^{1-1} = 2^0 = 1ms$ wait time
Loop until $K > K_{max}$

if $K > K_{max}$ (true)
then **abort** it

Vulnerable time: $2 \times t_{fr}$ t_{fr} = frame trans. time



The length of time in which there is a possibility of collision is called as vulnerable time. We assume that that the station sends fixed length of frames with each frame taking t_{fr} seconds to send where t_{fr} = frame transmission time. In ALOHA, the vulnerable time is fixed at $2 \times t_{fr}$

Throughput: The throughput for ALOHA is given as. (G) and is defined as avg. no. of frames generated by the system during one frame transmission time so the max throughput is S_{max}.

$$S = G \times e^{-2G}$$

$$S_{max} = \frac{1}{2e} \text{ at } G = 1/2.$$

8) A pure ALOHA network transmits 200 bit frames on a shared channel of 200 Kbps. What is the throughput if the system (all systems together) produce

i) 1000 frames per sec.

ii) 500 frames/s

iii) 250 frames/s

Soln. $t_{fr} = \frac{200}{200kbps} = 1ms$ $\frac{\text{no. of frames}}{\text{speed}}$ = Avg transmission time

i) 1000 frames → 1sec

1 frame → ?

$\therefore \frac{1}{1000} = 1 \text{ frame} = G$

↳ 1ms for 1 frame

$S = G \times e^{-2G}$

$= 1 \times e^{-2} = 0.135$

$= 13.5\%$ → 100 × 0.135

∴ out of 1000 frames only 135 frames will get transmitted

↳ max throughput ALOHA can give here

ii) 500 → 1

1 → ?

$\therefore \frac{1}{500} = 0.5 = G$

500 is half of 1000

↓ ↓

½ 1

$S = G \times e^{-2G}$

$= \frac{1}{2} \times e^{-\cancel{2} \times 1/\cancel{2}}$

$= 0.5 \times e^{-1} = 0.183$ (18.3%)

∴ out of 500 frames only ~~f~~ 500 × 0.183 = 93 frames can be transmitted

iii) 250 → 1

1 → ?

$\therefore \frac{1}{250} = \frac{1}{4} = G$

$S = G \times e^{-2G}$

$= \frac{1}{4} \times e^{-2 \times \frac{1}{4}}$

$= 0.1516 = 15.16\%$

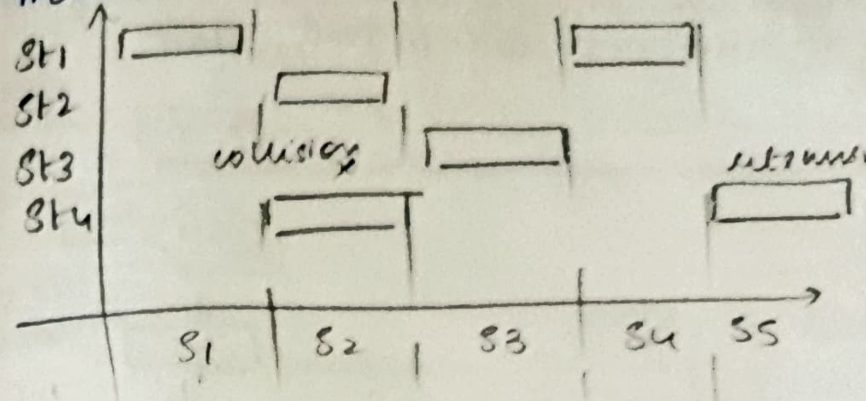
∴ out of 250 frames only 250 × 0.1516 = 38 frames can be transmitted

If 4 station → all st tog is giving 38 frames trans

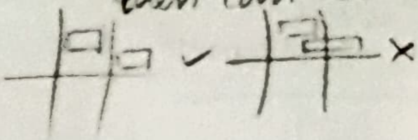
for 1st. → $\frac{38}{4}$ = ___ frames transmitted

} A case when st are given

Slotted ALOHA



if frame is ready & is at beginning of time slot then can send



The vulnerable time for slotted ALOHA is t_{fr}. So throughput is given as,

$$S = G \times e^{-G}$$

$S_{max} = 0.386$ at $G = 1$

) For given G, solve using slotted ALOHA method

i) $S = 1 \times e^{-1} = 0.3678 = 36.78\%$

ii) $S = \frac{1}{2} \times e^{-1/2} = 0.3032 = 30.32\%$

iii) $S = \frac{1}{4} \times e^{-1/4} = 0.1947 = 19.47\%$

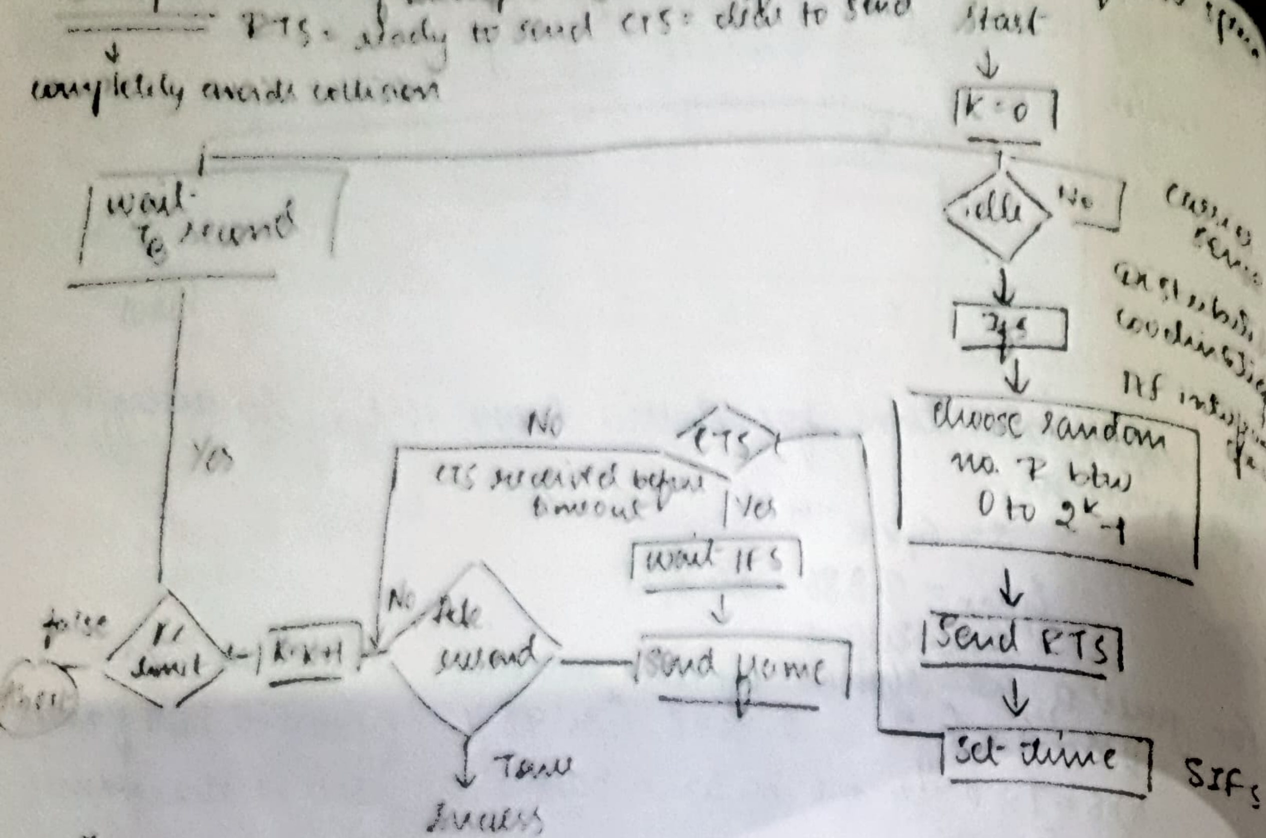
1000 → 367 frames
500 → 152 frames
250 → 49 frames
→ successful transmission

CSMA

CSMA/CA K = No of attempts T_B = back off time I_{FS} = interframe space

RTS = ready to send CTS = clear to send

completely avoids collision



Q) 10M

Consider a CSMA/CA network that transmits data at a rate of 100 Mbps over 1 km cable with no repeaters and if the min. frame size required for this network is 1250 bytes. What is the signal speed in the cable and efficiency.

Soln. $V \geq \frac{2 \times 1 \times 10^8}{8 \times 1250}$

$BW = 100 \times 10^6$ bps

$D = 1$ km

$L = 8 \times 1250$ bits

$TT \geq 2PD$

$\frac{L}{BW} \geq 2 \times \frac{D}{V}$

$V \geq \frac{2 \times D \times BW}{L}$

$V \geq 0.2 \times 10^5$

$V \geq 20000$ km/s

$\eta = \frac{1}{1 + 6.44a}$

$a = \frac{1}{20000} \times \frac{10^8}{1250 \times 8}$

$= 0.5$

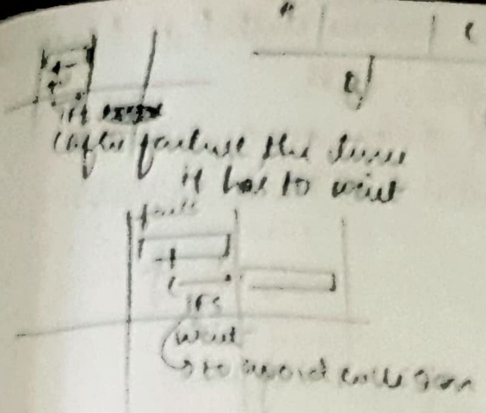
$= \frac{1}{1 + 3.2} = 0.236$

$\therefore$ efficiency $= 23.6\%$

$A = \frac{PD}{TT} = \frac{D/V}{L/BW} = \frac{D}{V} \times \frac{BW}{L}$

$\eta = \frac{1}{1 + 6.44a}$

[D = distance - cable dist
L = length of msg
V = velocity
BW = bandwidth

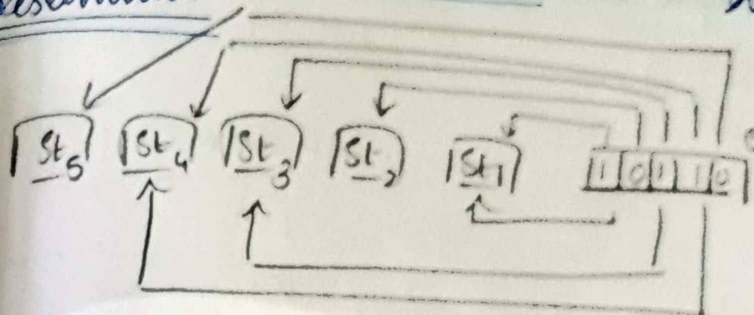


all want to
send tog.

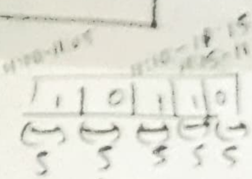
NOTE: In CAP, there is no chance
of collision & in all the
protocols of CAP, we have
round-robin system

CAP (control access protocol)

1) Reservation

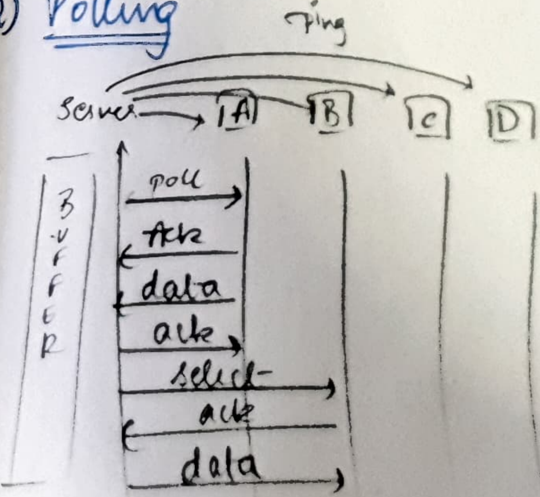


server
checks if any station wants
to transmit data
1 = Yes 0 = no
time slot reserved
for 1, 3, 4
each will run for
a fixed amt of time
↳ Eg. 5min
(round-robin)



↳ nothing to
transmit but
system still waits to avoid
collision

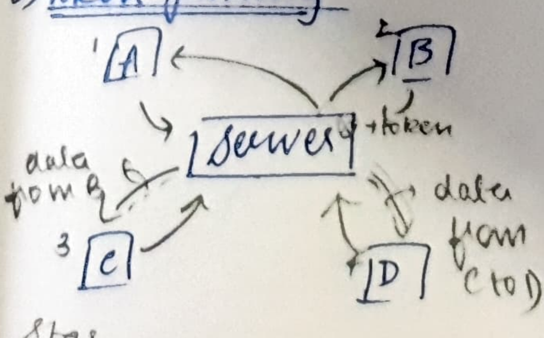
2) Polling



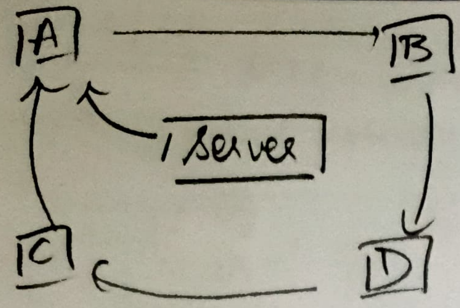
↳ (A wants to send data to B)
System A sent ack to server
then sends data,
server sends ack to A
select to check if B is free to receive
msgs.
if B sends ack to server then
data can be transmitted

FCFS if multiple systems want to
send data.

3) Token passing



token passed to A first, if A has
nothing to transfer then token
sent back to server
B has data, keeps the token as long
as it has to send data, once done, ...
sent back to server

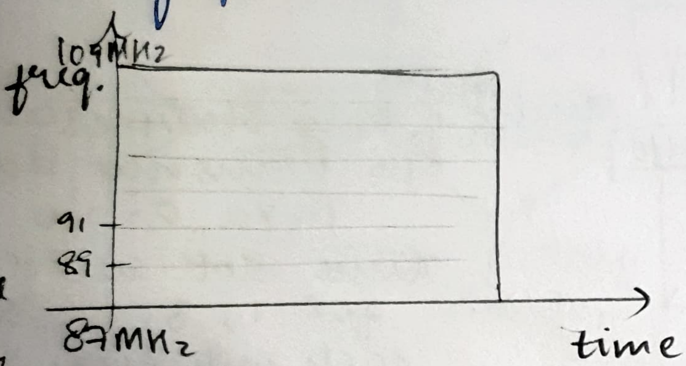


Ring

token to A, keeps token until it [illegible] data, then token given to B.
A wants to send data to C, server holds data until C's turn comes.

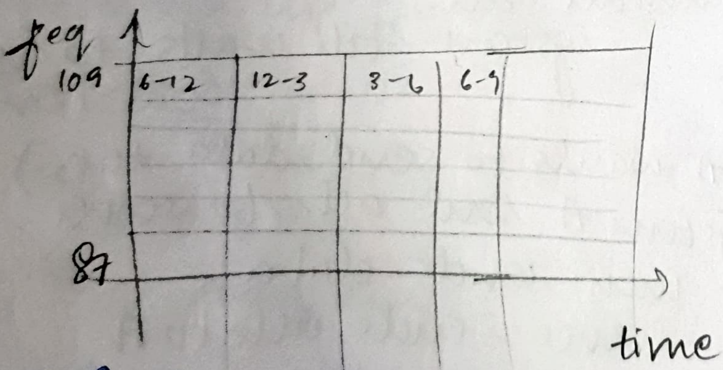
channelized protocol (collision can happen)

→ FDMA (freq. division multiple access)

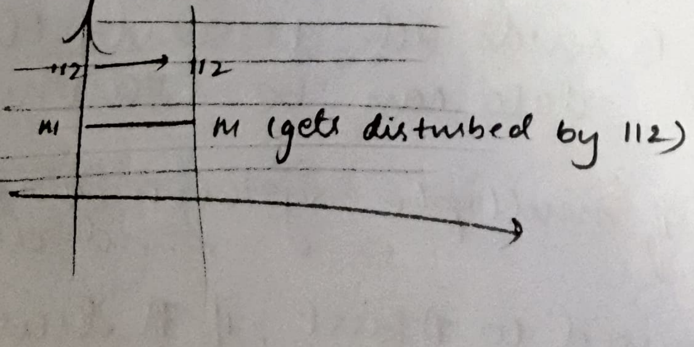


sound ↓ as you go near 89, new station starts

→ TDMA (time division MA)



→ CDMA (code division MA)



IEEE 802.11

deals with MAC layer (combination of DLL & NL)

LLC	IEEE 802.1
DLL / MAC	point coordination fn. (PCF)
	Distributed coordination fn. DCF

contention free service → PCF → no competition btw devices (10-11 → Device 1, 12-1 → D2)

contention service → DCF → competition, at wt time which device should be given priority, depends on time

DCF → CSMA/CA (wireless)

2.4 GHz
5GHz

physical layer	802.11 B	802.11 G	802.11 N	802.11 ac	802.11 ax

2 bytes	2 bytes	6 bytes	6 bytes	6 bytes	2 bytes	6 bytes	0-2312 bytes	4 bytes
frame control	Duration	Add-1	Add-2	Add-3	Seq	Add-4	Data	Check sum

Duration → how long data transmitted

Seq → when data missing 1 2 ? 4, seq. imp

Data → longest byte

entire thing is 2 bytes

2 bits	2 bits	4 bits	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit	1 bit
protocol version	Type	sub-type	To DS	from DS	more flag	retry	Power manage-ment	more data	WEP	order

protocol version → newer version taken into inside.

DS = dist. station → AP, comes → 1, dev → 0

from/to or to/from

retry → resending frame, set to = 1

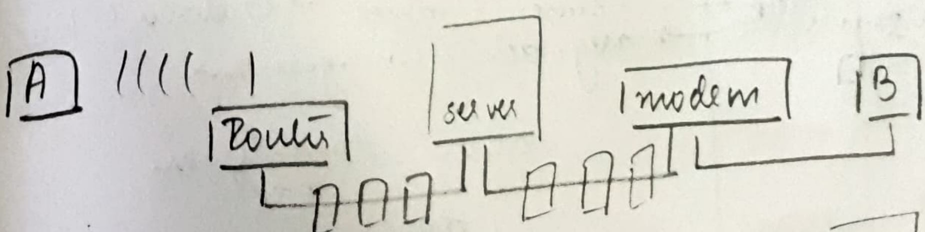
WEP → wired encryption

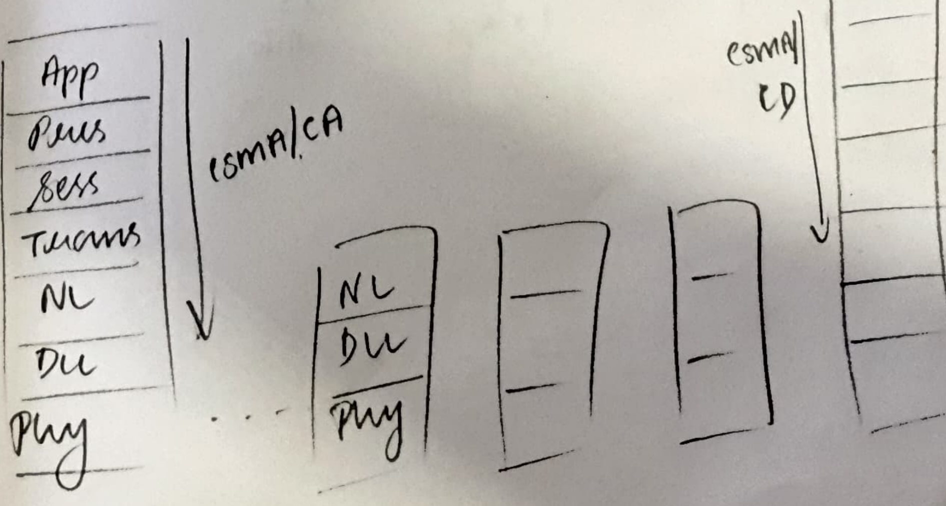
order → to send in orderly manner

Bits	Type
00	manage (frames communicate with each other cont.)
01	control frame
10	data frame

Bits	subtype
1011	RTS (ready to send)
1100	CTS (clear to send)
1101	Ack
0000	Data frame

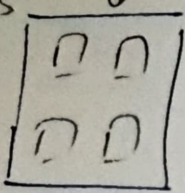
Standard	Operating freq.	Channel width	Speed
802.11	2.4GHz Can support 3 devices channel at a time while 5GHz can support 20 channel at a time	20MHz 2.4 - 2.42 2.42 - 2.44 2.44 - 2.46 3 channels diff 5 - 5.02 5.04 - 5.08	
802.11 B	2.4GHz	20MHz	10Mbps
802.11A	5GHz	20 MHz	54Mbps
802.11G	2.4 GHz	20MHz	54Mbps
802.11 N	2.4GHz & 5GHz supported both	20MHz & 40 MHz	450Mbps
	in total = 23 devices supported		
802.11 ac	5GHz	40MHz	866 Mbps
802.11 ac wave 2	5GHz	20MHz & 40MHz & 80MHz 60 devices	1.73Gbps
802.11 ax	2.4GHz & 5GHz	20MHz & 40MHz & 80MHz & 160MHz	2.4Gbps
802.11 az	2.4GHz & 5GHz	20MHz & 40MHz & 80MHz & 160 MHz & 320MHz 500 devices at a time	10Gbps





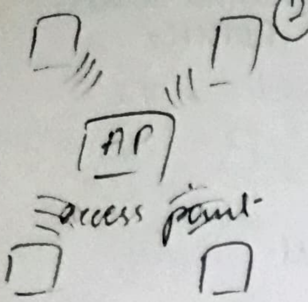
only dest. a[...]
keeps
changin[...]
source add.
rema[...]
sa[...]

BSS
ESS ①

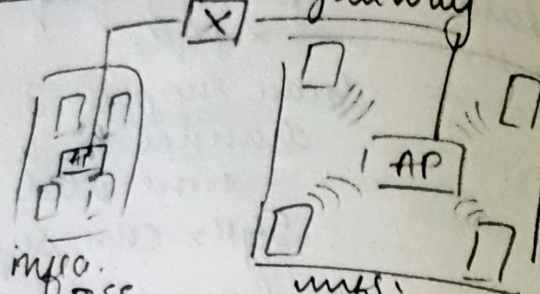


Adhoc
BSS

②



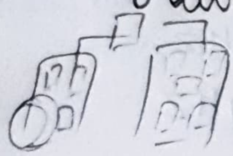
infrastructure
BSS
(Basic standd. set)
all devices connected
to access pt. (AP)



infra.
BSS

infra
extension service set-

- when there is no movement of any device from 1 BSS to another BSS
  ↳ no info. – state
- when movement is there BSS transition state
- device moves outside of ESS then
  ↳ ESS transition state

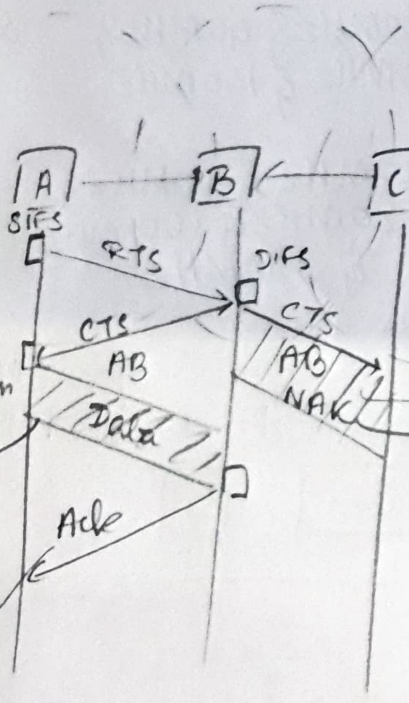


↳ moved

A's range, doesn't know C exists

wireless → CSMA/CA

Both want to communicate with B, collision
Hidden node prblm
↳ A doesn't know C
↳ C doesn't know A

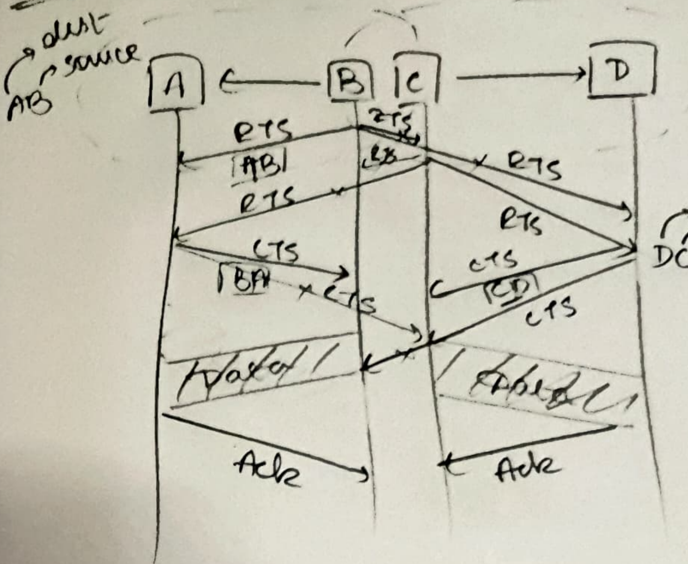


...ure AB dist
B is free CTS in all direction
...ds data amt of time
...received

discards it
→ time it takes
→ neg. ack

0000	Data frame
1011	RTS
1100	CTS
1101	Ack

Exposed node prblm



comm. B → A then C waits until they're done to comm with D

when B → A RTS sent to all
C & D have no prblm, ignore RTS

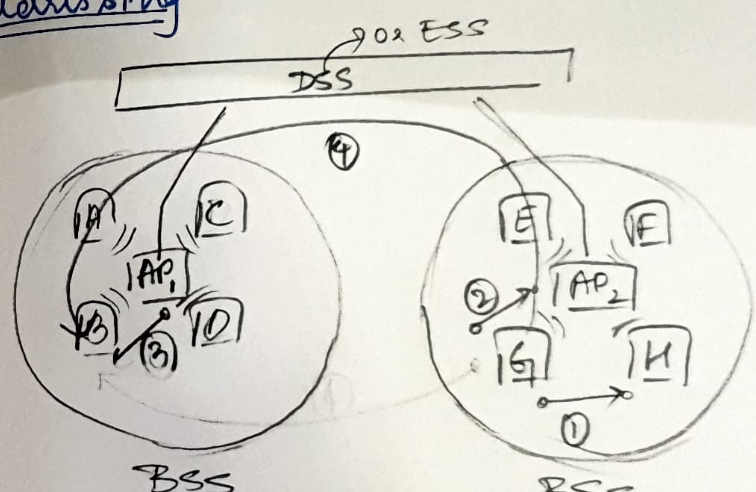
C → D

→×→ ignored

After which
A sends CTS to B, C
$A \xrightarrow{CTS} B$ $A \xrightarrow{CTS} C$
now B sends data to A
& A sends back ack

D sends CTS to C, B
now C sends to D
D sends Ack

Addressing



AP1 has no do... some other s...

some other pl... can be dest.

st → st com... in which region

no intermediates

	To DS	From DS	Add1	Add2	Add3	Add4	seq
two stations / mobile points (G → H)	0	0	Destn.	Source	AP2	NA	
G → AP2 (st & routers)	1	0	AP2	Source	Destn.	NA	
AP1 → B	0	1	Destn.	AP1	Source	NA	
all stationary G → B	1	1	Destn.	Source	AP1 ↓ Source	AP2 ↓ Destn.	