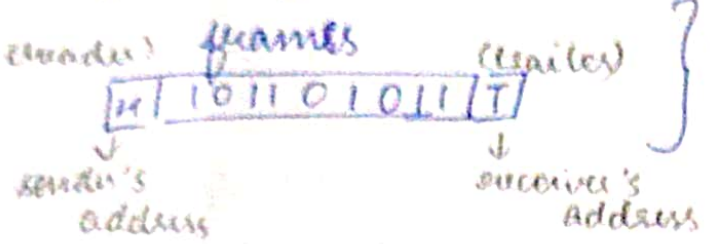
Data link control services

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



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

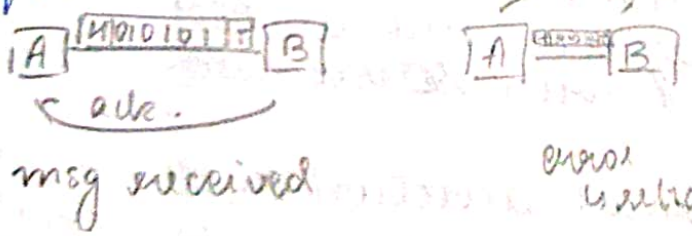
DLC services → datalink layer which helps in performing following operation

i) framing
ii) flow control
iii) errors control

MAC

i) Encapsulation
ii) Access to the medium

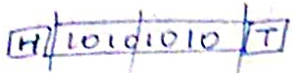
1) Framing: Framing in the datalink layer seperates a message from one source to a distination by adding a sender's address and a distination address.
The distination address defines where the packet is to go & the sender's address helps the recepient acknowledge the receipt.



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

1byte = 8bits

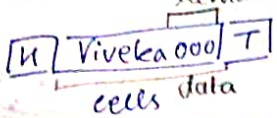
size of frame



flag flag flag → divides it into 4 bits each
4bits 4bits
frame → 8bits
↳ avoids wastage of BW

Two kinds of framing:
→ fixed
→ variable size

In fixed size framing, there is no need for defining boundaries of 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



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

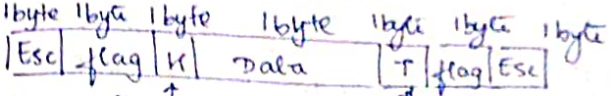
In Variable size framing; we need a way to define the end of the frame & the beginning of the next frame

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 errors detection or, errors correcting 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

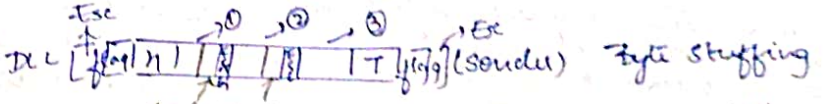


Source add.
dest add.
other info

Errors codes
errors detect.
redundant Bits

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

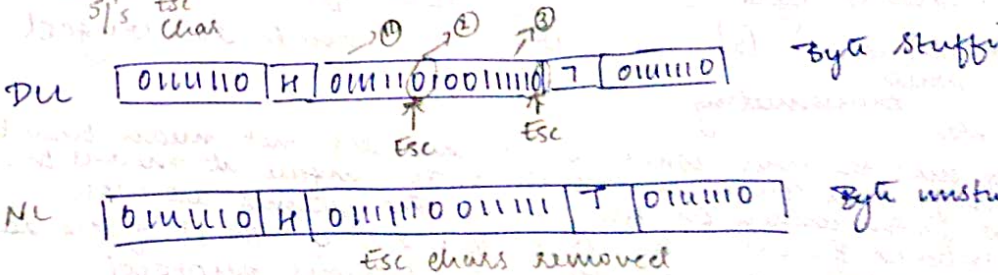
Byte Stuffing & unstuffing



add esc before any esc or flag (esc→1byte)

removing the esc which got added at DLL

→ Bit oriented frame
By default flag is 01111110
prblm: data can contain same sequence
Soln. whenever data with five 1's are seen, you add 0 (consecutive) → Esc char



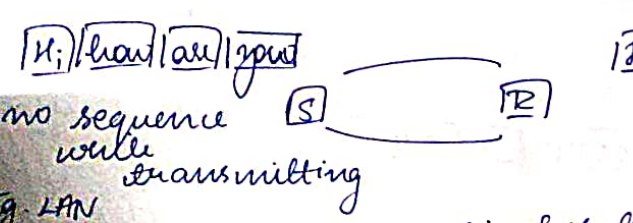
In bit oriented framing, the data section of a frame is a sequence of bits to be interpreted by the upper layer as text, graphic, audio, video, & so on... However in addition to the headers & possible trailers, we need a delimiter to separate a frame from the other frame. Most protocol uses a special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

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

Data communication requires atleast two devices working tog., one to send & other to receive info. Even such a basic ~~arrangement requires~~ arrangement requires a great deal of coordination for an exchange to 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 node to the next node without any relationship btw the frames. Here frames are independant.



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.

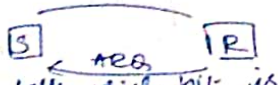
duplicate message

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.

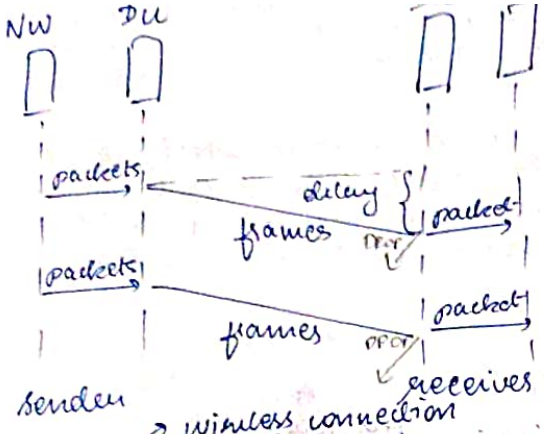
Error control

ARQ - automatic repeat request



connectionless oriented approach has 3 phase
is 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.

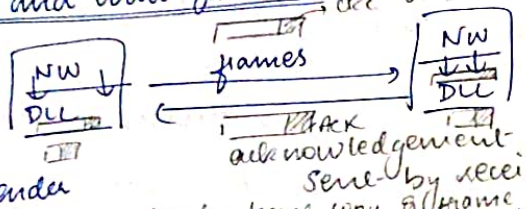


Drop when corrupted
Bcz of delay, line is not straight.

→ wireless connection

Stop and wait protocol (verifies if receiver receives valid frame)

one frame transmitted at a time

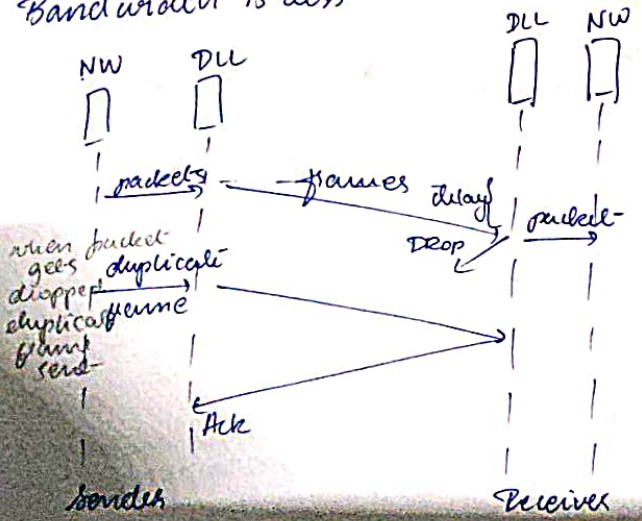


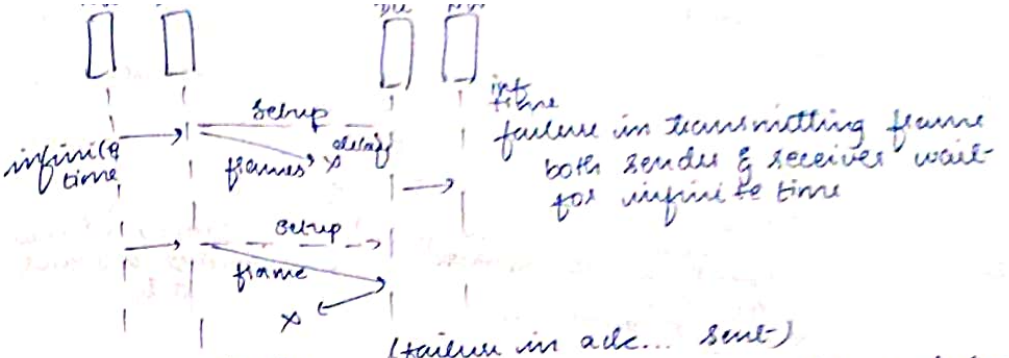
Sent by receiver

(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)

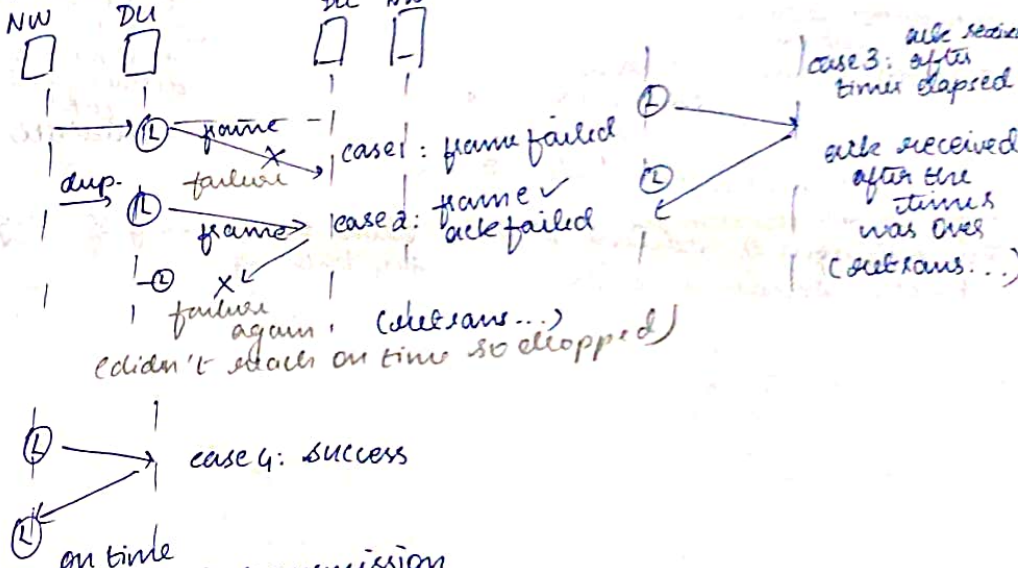
Bandwidth is less





Stop and wait ARQ (diag. same as prev. one)

→ set a timer (acknowledgement received within time)



on time successful transmission

copy of frame deleted now from DLL layer)

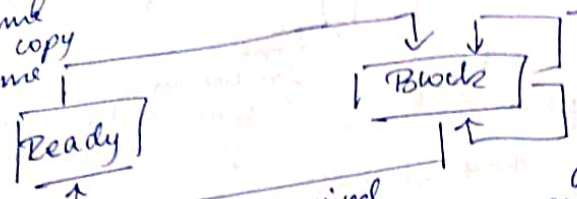
packet come from NL

→ make a frame → save a copy → & send the frame

Ready

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

timeout
→ resend saved frame
→ restart timer



Sender

Errors free ack received
→ stop timer
→ discard saved frame

Corrupted ack arrived, discard ack.

Receiver

Corrupted frame arrived
→ discard frame

Errors 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

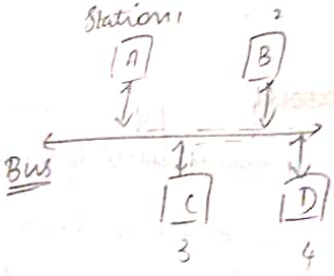
In MAC (multiple access control)

- RAP (random access protocol)
- CAP (control access protocol)
- CP (channelized protocol)

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

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

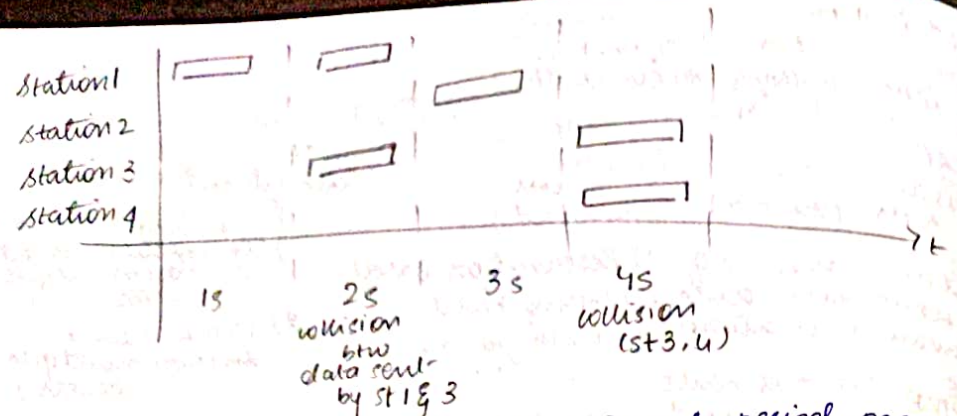
CP:
1) FDMA (frequency division multiple access)
2) TDMA (time division multiple access)
3) CDMA (code division multiple access)



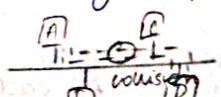
Random access / contention method.
In this method, no station is superior to another station & none is assigned control over another. At each instance, a station that has data to send uses a procedure defined by the protocol to make a decision on whether or not to send data.

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 be 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.



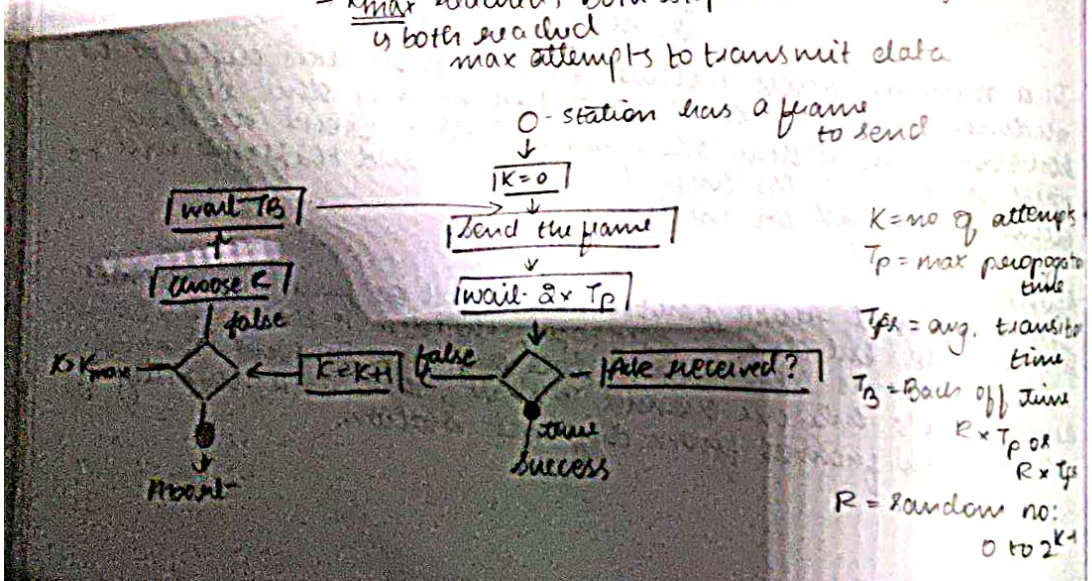
- Pure ALOHA dictates that when the time out period passes each station waits a random amt. of time before resending its frame. This randomness will help avoid more collisions. This random time is called a backoff time TB.
- Pure ALOHA has a limit method to prevent collision in the channel with retransmitted frames after max. no. of retransmission 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
→ same RT
→ collision
→ Kmax reached, both stop transmitting data

time for all st. is same

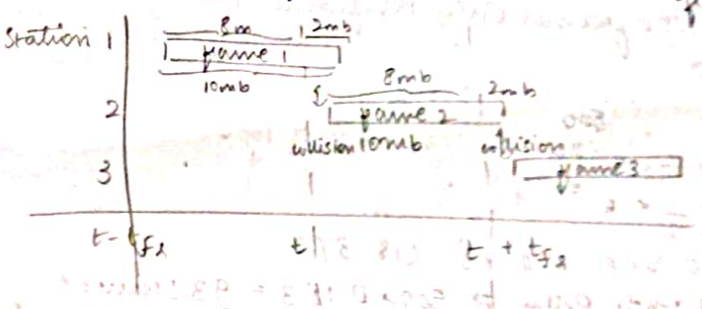
Kmax reached, both stop transmitting data
is both reached max attempts to transmit data



Station A sends
K=0 → attempt 1 of sending
A sends, frame
sending & receiving side ($2 \times T_p$) ← fixed
(time) Ack received
success
no ack (false)
K = K+1 (attempt inc.) = 0+1 = 1
K > Kmax (check cond.)
1 > max
(false)
TR → $2^{1+1} = 2^0 = 1$ ms wait time
loop until K > Kmax
if K > Kmax (true)
then abort it

Vulnerable time: $2 \times t_{fs}$

t_{fs} = frame transmission time



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

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

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

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

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

i) 1000 frames → 1 sec
1 frame → ?
$\therefore \frac{1}{1000} = 1\text{ms} = G$
↳ 1ms for 1 frame

$S = G \times e^{-2G}$
$= 1 \times e^{-2} = 0.135$
$= 13.5\%$ → 100×0.135 → 135 frames will get transmitted
∴ 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 ↓ 1
↓ ½

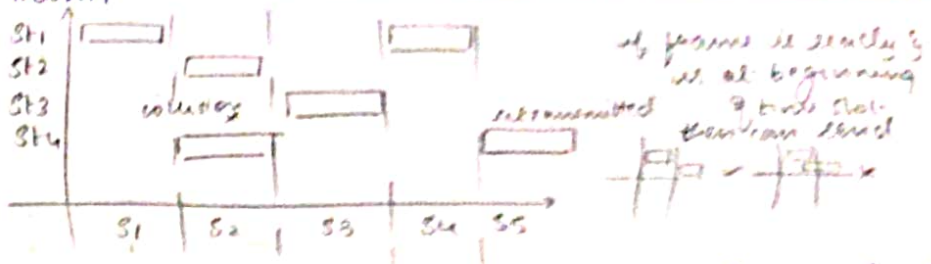
$S = G \times e^{-2G}$
$= \frac{1}{2} \times e^{-2 \times 1/2}$
$= 0.5 \times e^{-1} = 0.183$ (18.3%)
∴ out of 500 frames only 500 × 0.183 = 93 frames can be transmitted

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

4 station → all station together giving 38 frames
for 1st → $\frac{38}{4}$ = ___ frames transmitted
↳ worst case when all are giving

Slotted ALOHA



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

$S = G \times e^{-G}$
$S_{max} = 0.386$ at $G = 1$

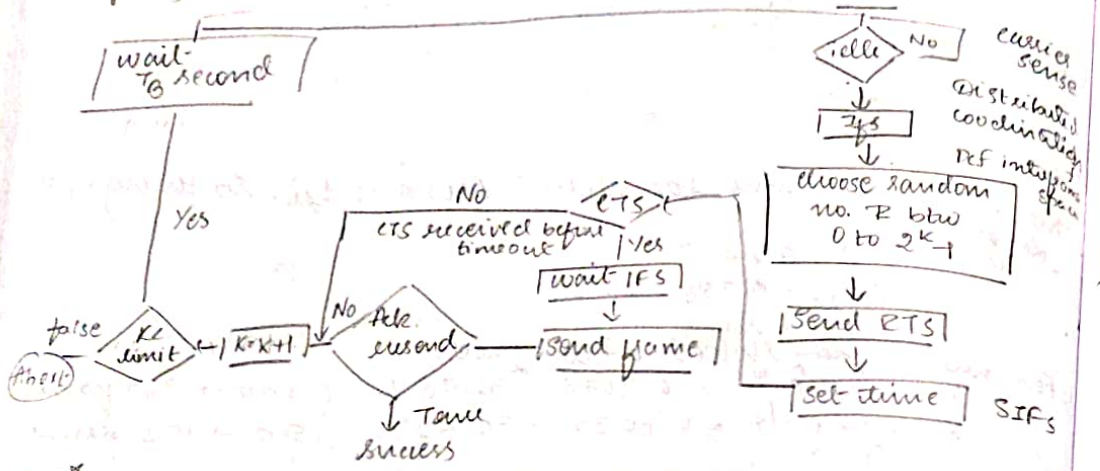
Q) For given Q, solve using slotted ALOHA method
i) $S = 1 \times e^{-1} = 0.3678 = 36.78\%$
ii) $S = \frac{1}{2} \times e^{-1/2} = 0.3032 = 30.32\%$
iii) $S = \frac{1}{4} \times e^{-1/4} = 0.1947 = 19.47\%$

1000 → 367 frames
500 → 152 frames
250 → 48 frames
↳ successful transmission

CSMA

CSMA/CA K = No. of attempts T_B = back off time IFS = inter frame space
RTS = ready to send CTS = clear to send

completely avoids collision



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

$V \geq 0.2 \times 10^5$

$V \geq 20000$ km/s

$BW = 100 \times 10^6$ bps
$D = 1$ km
$L = 8 \times 1250$ bits

$TT \geq 2PD$

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

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

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

$= 0.5$

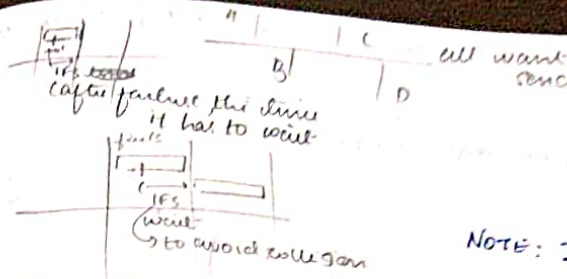
$A = \frac{PD}{TT} = \frac{\frac{D}{V}}{\frac{L}{BW}} = \frac{D \times 5}{V}$

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

$\eta = \frac{1}{1 + 6.44a} = \frac{1}{1 + 3.2} = 0.236$

$\therefore$ efficiency = 23.6%.

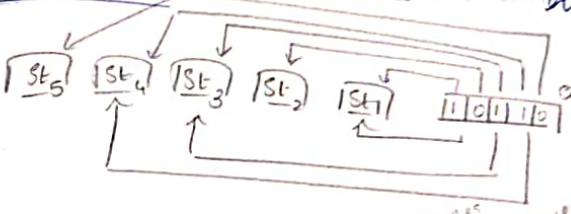
[D = distance - cable dist
L = length of msg
V = velocity
BW = bandwidth

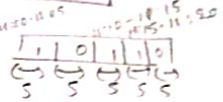


NOTE: In CAP, there is no chance of collision & in all the protocols of CAP, we have round-robin system

CAP(control access protocol)

1) Reservation





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)

4 nothing to transmit - but system still waits to avoid collision

(A wants to send data to B)
System A sends okk to server then sends data
Server sends okk to A
select to check if B is free to receive msgs
if B sends okk to server then data can be transmitted

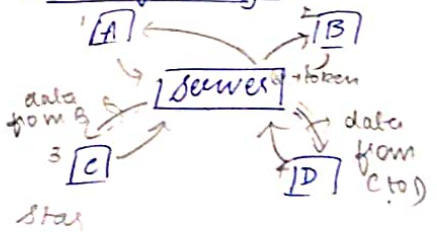
FCFS if multiple systems want to send data.

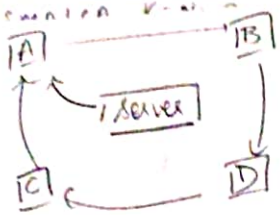
token passed to A first, if A has nothing to transfer then token send back to server

B has data, keeps the token as long as it has to send data, once done send back to server

2) Polling

3) Token passing



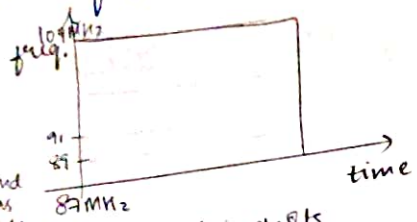


Ring

token to A, keeps token until it transmits data, then token given to B.
A wants to send data to C, so A holds data until C's turn comes.

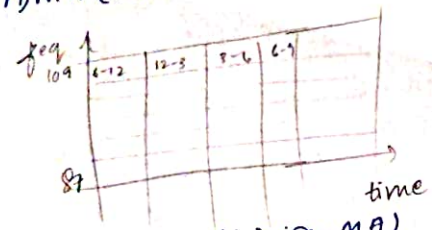
Channelized protocol (collision can happen)

→ FDMA (freq. division multiple access)

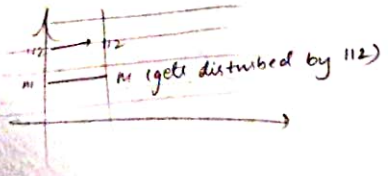


band is assigned 88, new station starts

→ TDMA (time division MA)



→ CDMA (code division MA)



IEEE 802.11

LL | IEEE 802.1 → deals with MAC layer (combination of DLL & NL)

DLL
- contention free service → point coordination fn. (PCF)
- contention service → Distributed coordination fn. DCF → CSMA/CA (wireless)

no competition btw devices
10-11 → Device1
12-1 → D2
competition → al-wt time which device should be given
depends on time

2.4 GHz
5 GHz

physical layer

802.11 B	802.11 G	802.11 N	802.11 ac	802.11 ax

longest byte

when data missing
1 2 3 4
seq. imp

how long data transmitted

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

entire thing is 2bytes

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

newer version taken into consider.

DS = dist. station
↳ AP, same → 1 from to
dev → 0
(frames communicate with each other conc.)

or to from

resending = 1 frame

See to

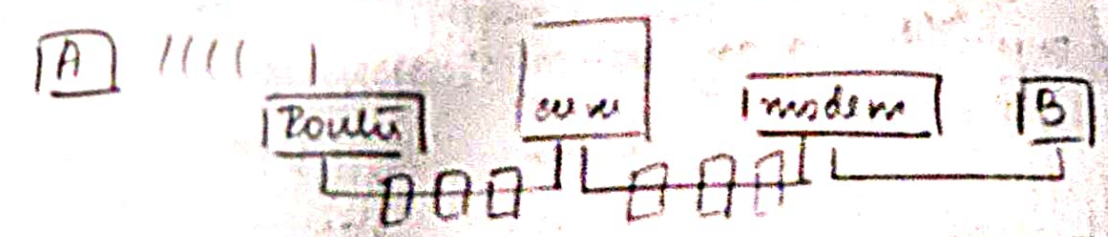
WEP → wired encryption

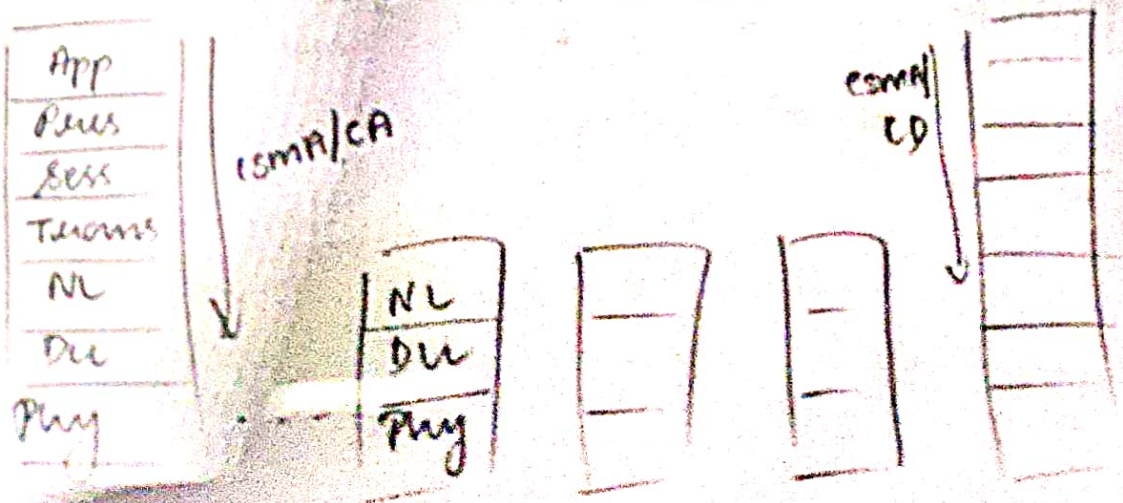
to send in orderly manner
is order

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

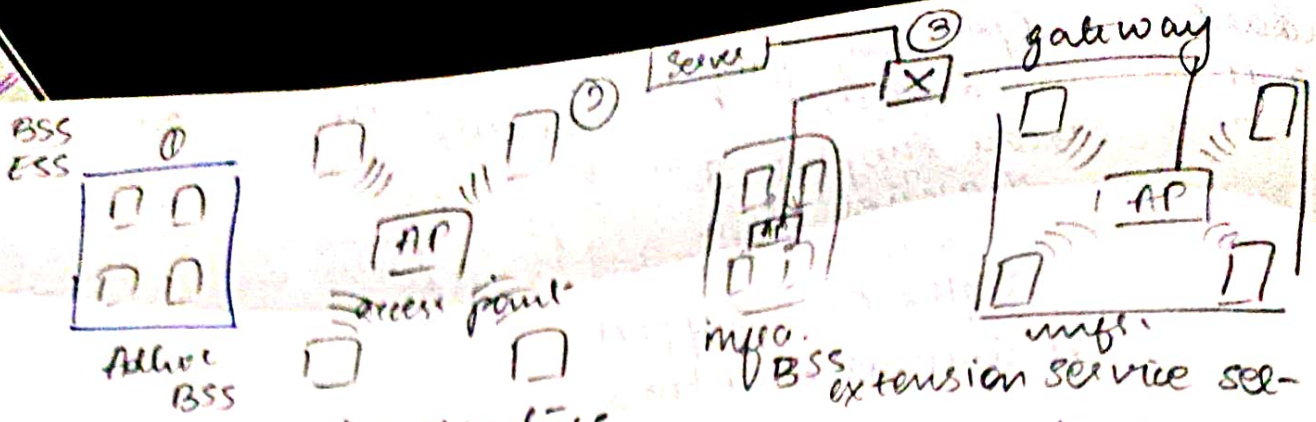
Bits	subtype
1011	RTS (ready to send)
1100	CTS (clear to send)
1101	Ack
0000	Data frame

Standard	Operating freq.	Channel width	Speed
802.11	2.4GHz can support 3 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 2 channels diff	
802.11b	2.4GHz	20MHz (5.5.02, 5.04-5.08)	10Mbps
802.11a	5GHz	20MHz	54Mbps
802.11g	2.4GHz	20MHz	54Mbps
802.11n	2.4GHz & 5GHz supported both in total = 28 devices supported	20MHz & 40 MHz	450 Mbps
802.11ac	5GHz	40MHz	866 Mbps
802.11ac wave2	5GHz	20MHz & 40MHz & 80MHz 10 devices	1.73Gbps
802.11ax	2.4GHz & 5GHz	20MHz & 40MHz & 80MHz & 160MHz	2.4Gbps
802.11az	2.4GHz & 5GHz	20MHz & 40MHz & 80MHz & 160 MHz & 320MHz 500 devices at a time	10Gbps

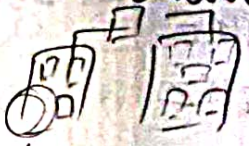




only dest. add
keeps
changing,
source add
remain
Same



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



↳ moved

A's range doesn't know C exists

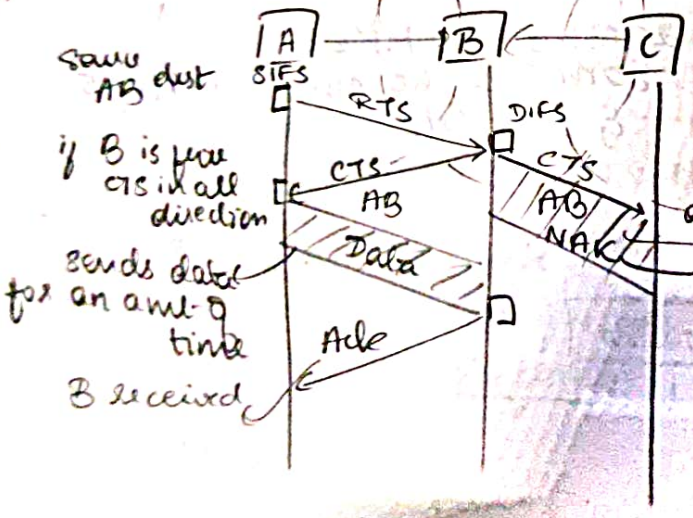
wireless → CSMA/CA

Both want to communicate with B, collision

Hidden node prblm

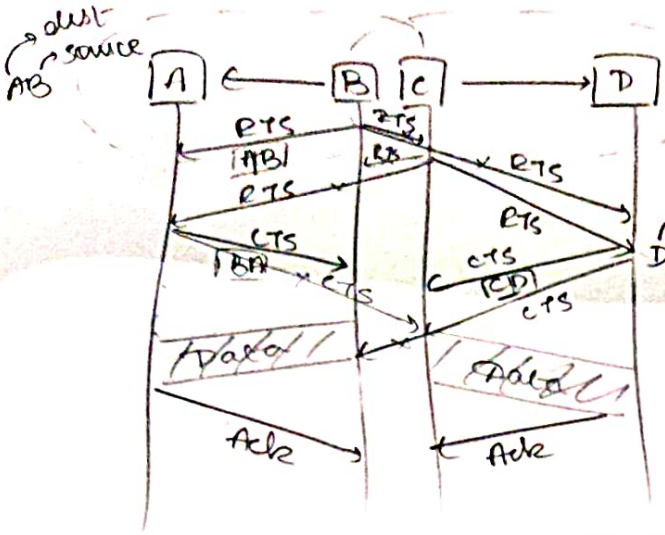
↳ A doesn't know C

↳ C doesn't know A



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

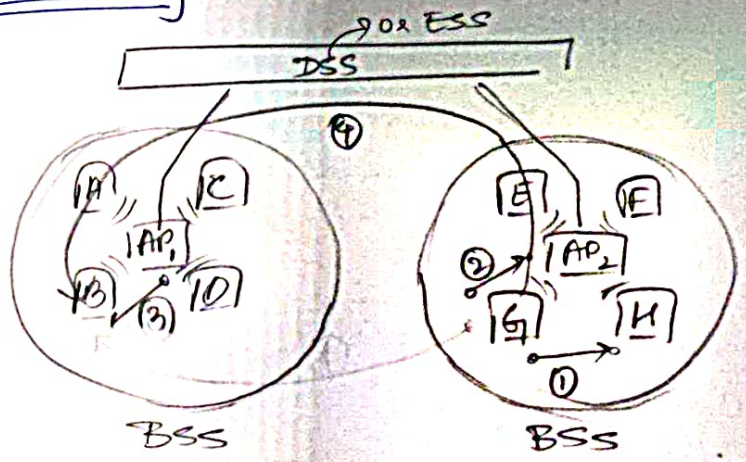
when B → A RTS sent to all
C & D have no prblm, ignore RTS

C → D

ignored

After which
A sends CTS to B, C
A → B CTS A ✗ C CTS
now B sends data to A
& A sends back ack to B
D sends CTS to C
now C sends to D
D sends Ack C

Addressing



AP1 has no data some other st has sent.

some other pt. can be dest. E, F, G, H

st → st comm. no intermediate

in which region your comm..

two stations	To DS	from DS	Add1	Add2	Add 3	Add 4	seq
mobile points (G → H)	0	0	Destn.	source	AP2	NA	
G → AP2 & G sends AP1 → B	1	0	AP2	source	Destn.	NA	
	0	1	Destn.	AP1	source	NA	
all stationary G → B	1	1	Destn.	source	AP1 ↓ source	AP2 ↓ Destn.