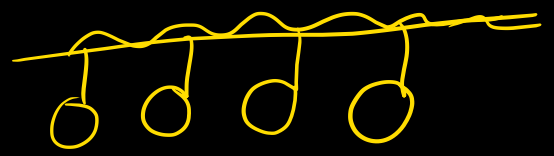
1

A - AL
P - PL
S - SL
T - TL
N - NL
D - DLL
P - PL

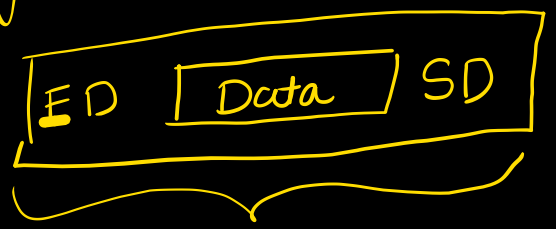
Data link layer:

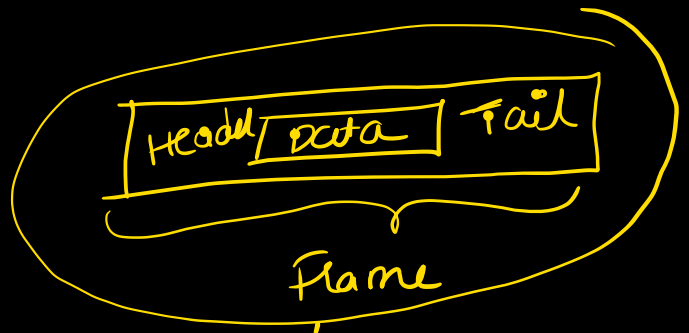
1) Flow control → (Stop and wait, GBN, SR)
2) Error control → (CRC, checksum)
3) Framing
4) Physical addressing
5) Access control

Framing and PA

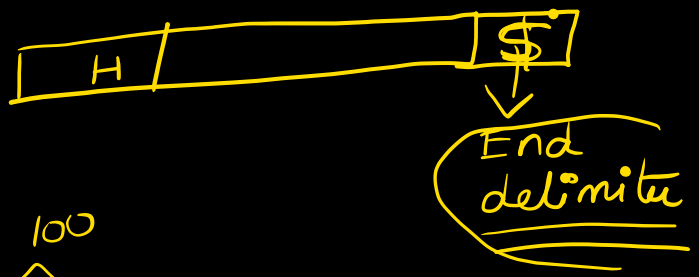


Framing:

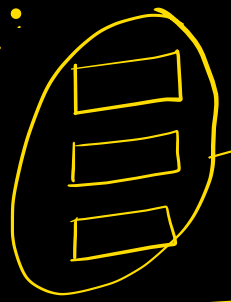




Framing



FL:



→ Same.

Frames

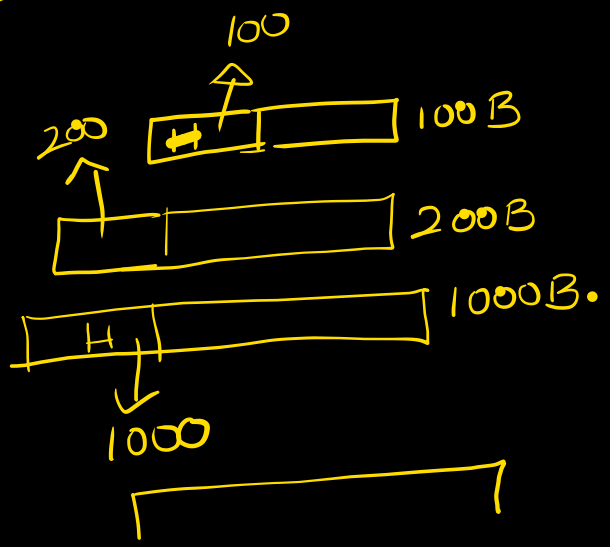
Fixed length

→ No need to specify length

Variable length

→ Specify length ✗

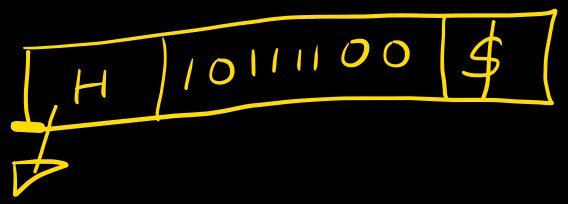
→ End delimiter ✗



Packet

- AL — message
- TL — segment
- NL — Datagram
- DLL — Frame
- PL — single protocol data unit

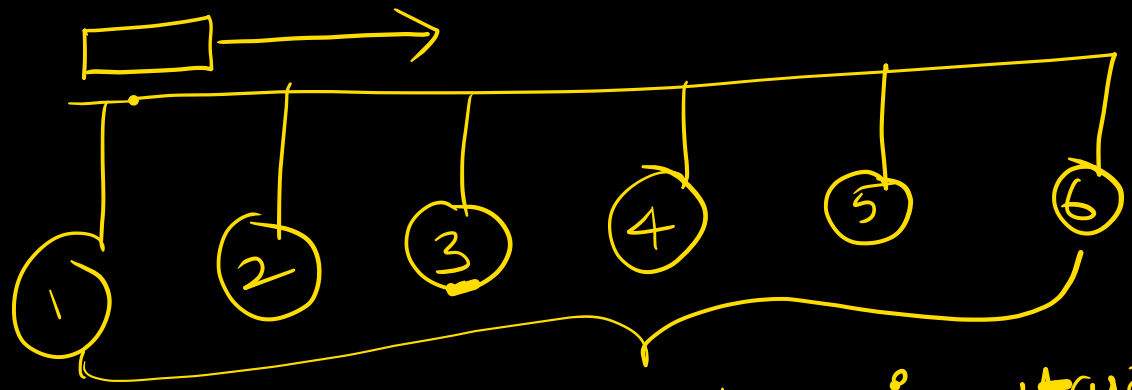
IPDU



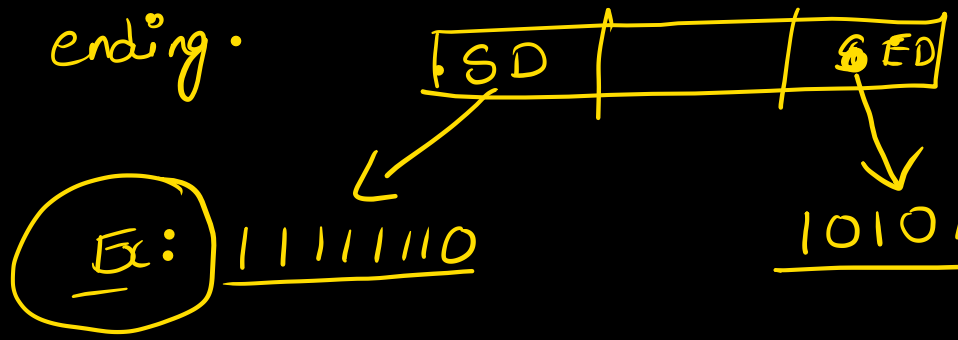
SD → start delimiter

End delimiter

Frame is variable size



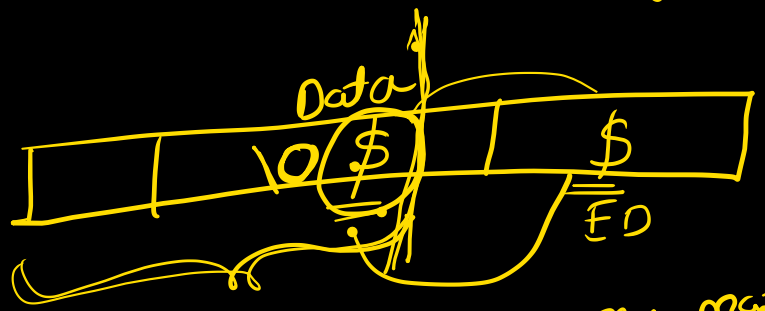
where the data is starting and where the data is ending.



1980's: In internet only text data used to be sent.

→ special character is used.

Now, => we are sending → Text, pics, code, videos, audios.

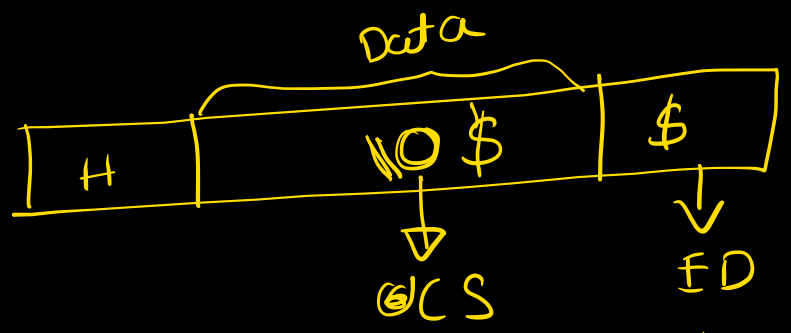


Pattern may match

character stuffing

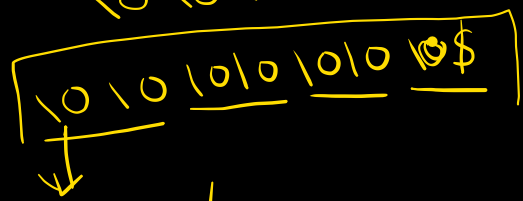
$ → Data
add (\0) → escape.

Data → $
add → \0
Data has \0$
add \0\0

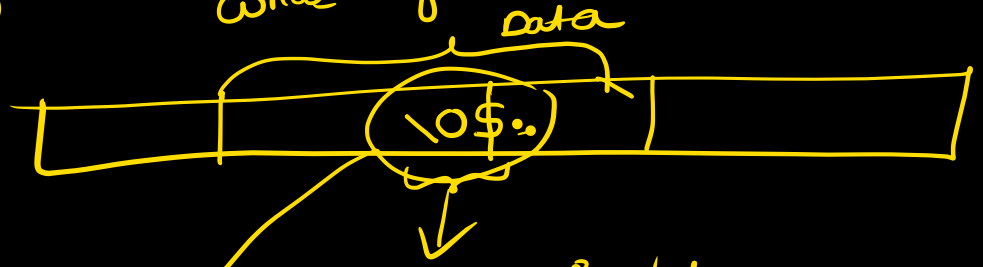


\0\0\0$
↓ one more

every dollar
- add \0
every escape
- add \0

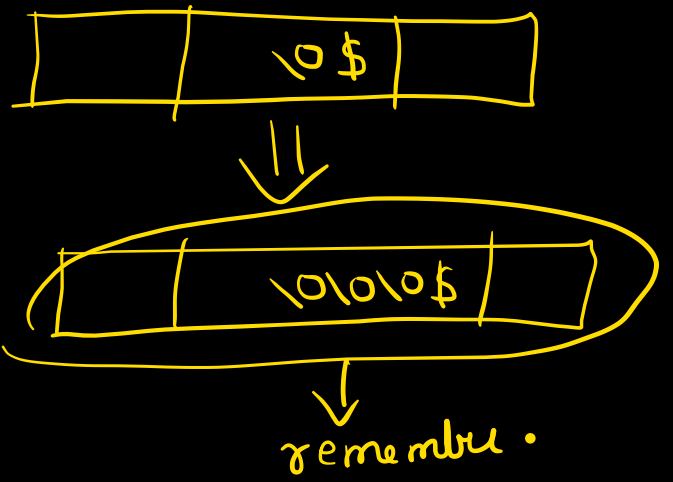


what if data has '\0$'



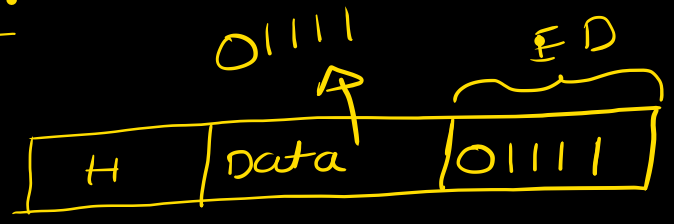
Present in data.

Rec → $ → and '\0' is add by sender



add two '\0' characters,
one for '\0' one for $

Bit Stuffing:



Ex:

If ED = 01111

what if data has 01111 → R think end of the flame

01111

Break the sequence.

Best way.

∴ Data 01111 → Send → 011101 → Rec → 01111$\cancel{0}$1

Removes zero

But if data Contains 01111001 → Rec will think '0' is added → 01111$\cancel{0}$1

Removes

If ED = 01111

If D Contain 01111 → Send → 011101 → Rec → 0111(0)1

If Data in 0111(0)1 → Send / Rec → 011101

Removed

Removed

Data Containing:

01111 → 01110

01110 → 01110

01110 (0) X

At Rec:

0111(0)(0)(1) → Rule 0

Remove

ED → 01111 → Then in data add '0' after

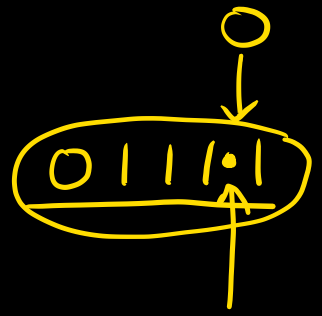
01111

01110 → 011100

01111 (0) (0)

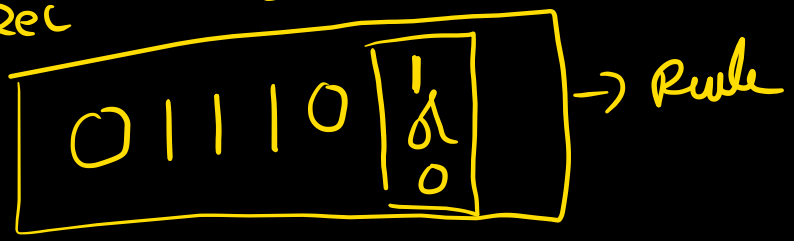
If ED = 01111

Data matches with



Data contains → 0 1 1 1 0 1

Rec this 3 bits in addr.



→ Rule

If ED = 0 1 1 1 1 1 1 → ED

Data = 0 1 0 1 0 1 1 1 1 0 1 1 1 1 1 0 0

0 → 5 1's

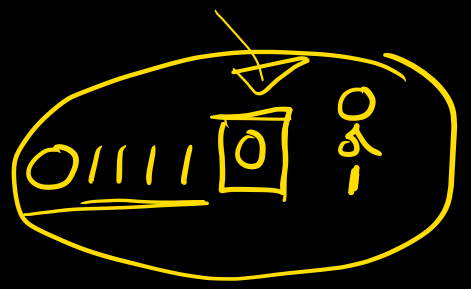
→ After 0 - 4 1's but '0'

0 → 6 1's

5 1's → 0

what is transmitted

ED = 0 1 1 1 1 1 → rule



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

ED: 0 1 1 1 1 1 1 1 1 1

rule 0 1 1 1 1 1 1 1 1 0

5 min Break.

Full stack course with Job guarantee

↓

Fee would be little high around 10k

↓

Because it is Job guarantee

↓

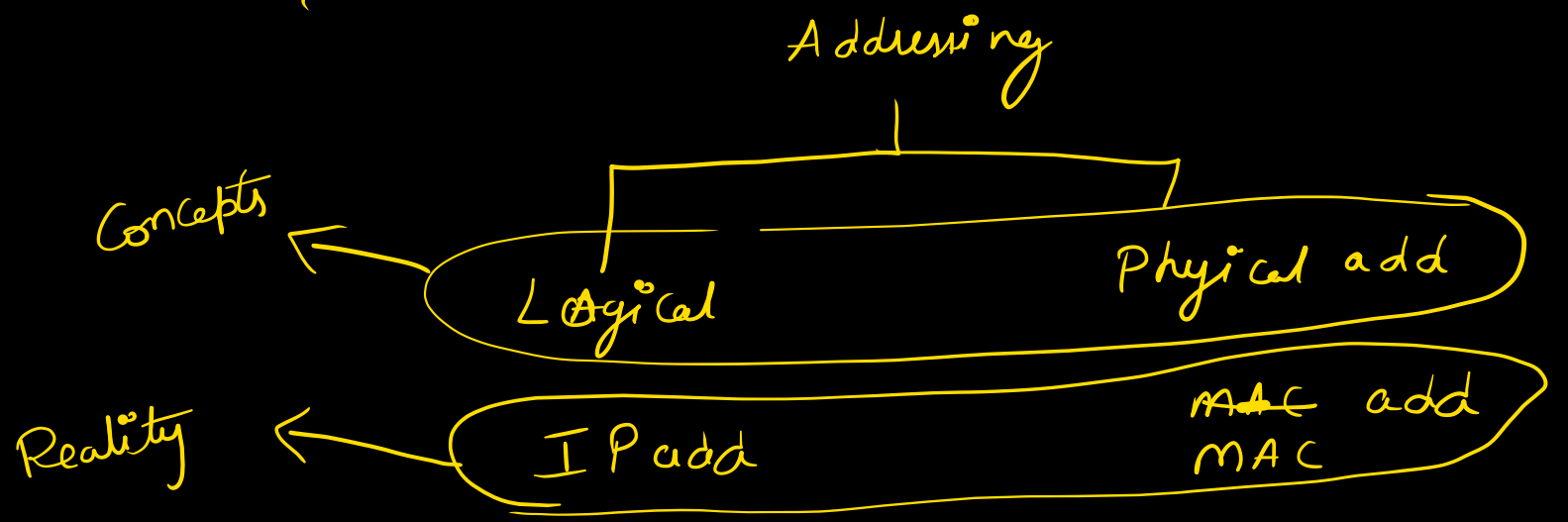
Big team is needed

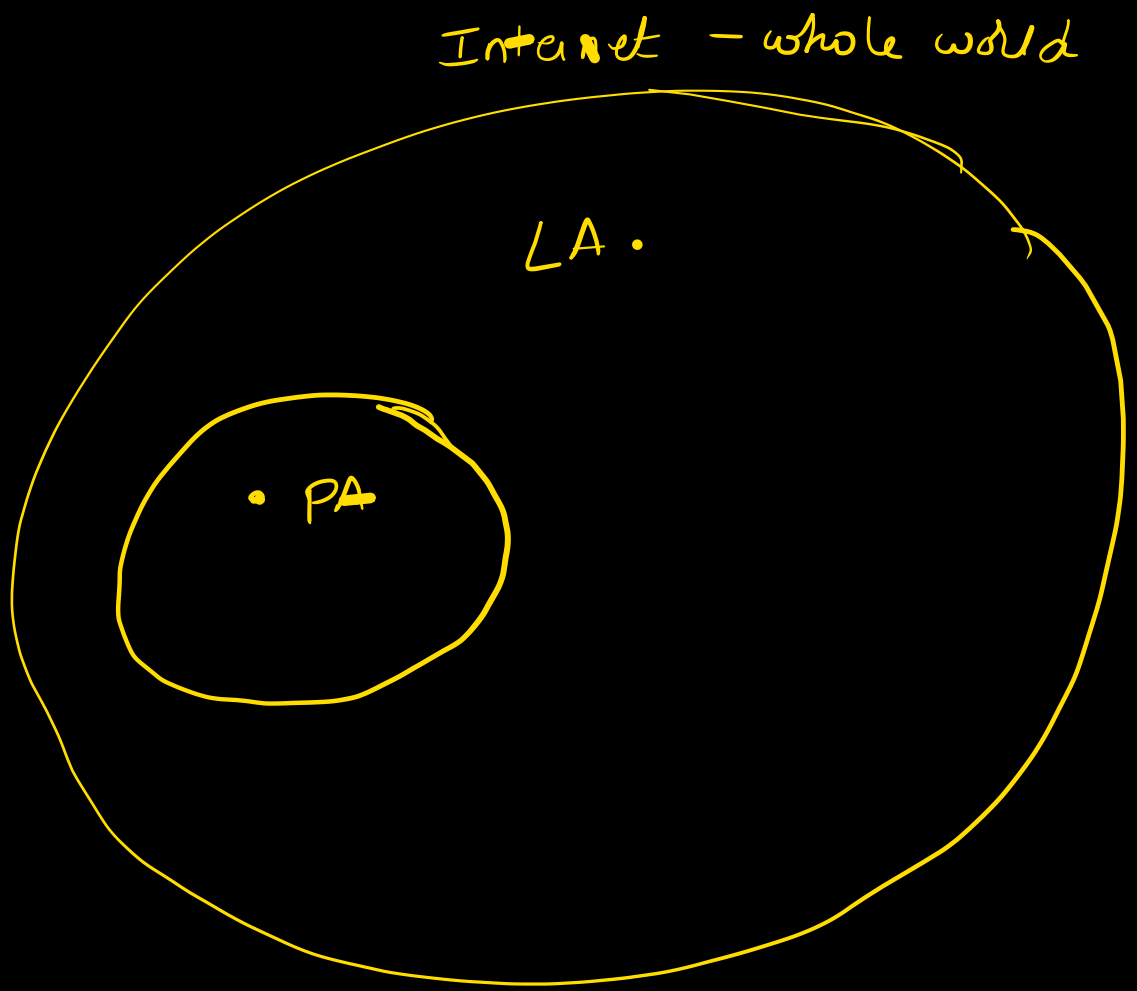
How is job guarantee?

↓

If no job → Then 100% refund

Physical addressing: → DLL

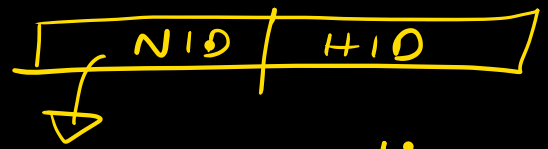




LA → any address that is unique in the entire world

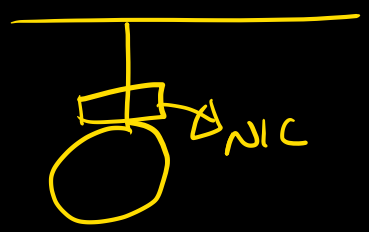
PA → Any add that is unique in a network of lan of subnet

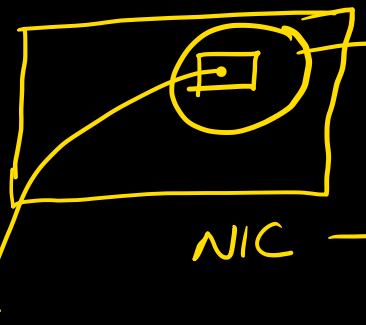
IP - add → 32 bit → S/w numbu → not permanent



useful for routing

MAC - add → 48 bit numbu → H/w numbu → permanent





IP → MAC add:

→ x location.

Vendor ID → DELL

Date → 04.01.2023

S.NO → 100 → unique → Globally.

Can I use mac as logical add → NO, Because it is not useful for routing

IP: → unique globally.

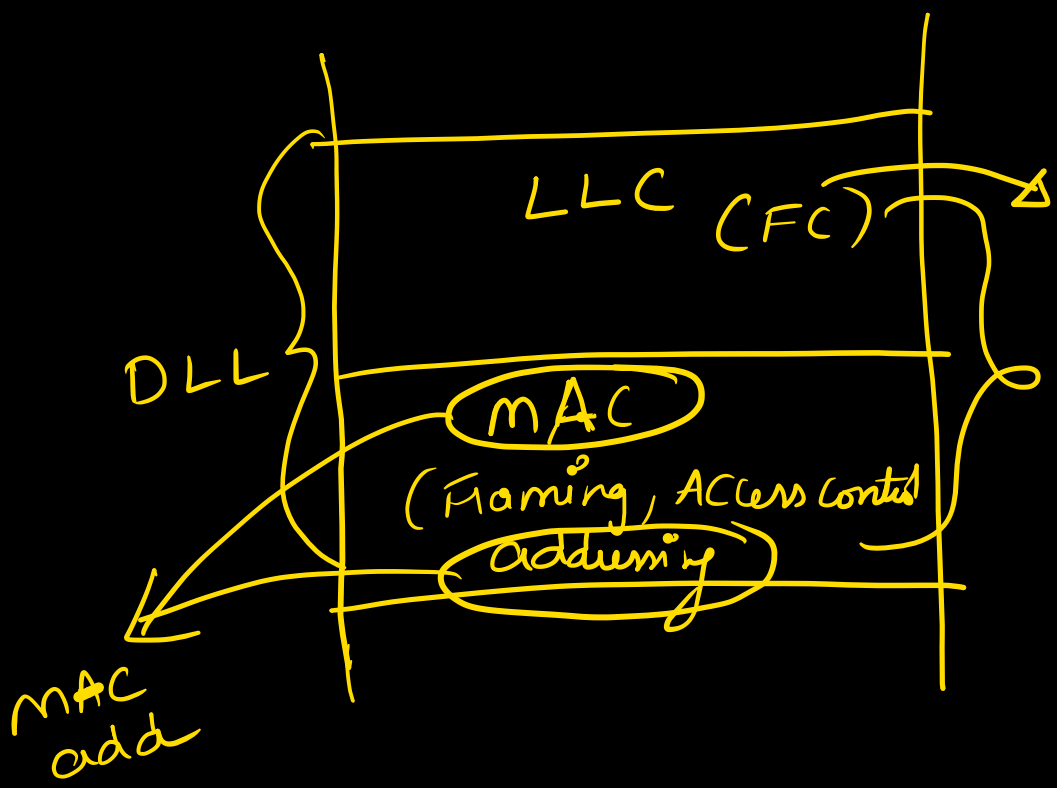
logical add → unique globally.

we can use → MAC → PA

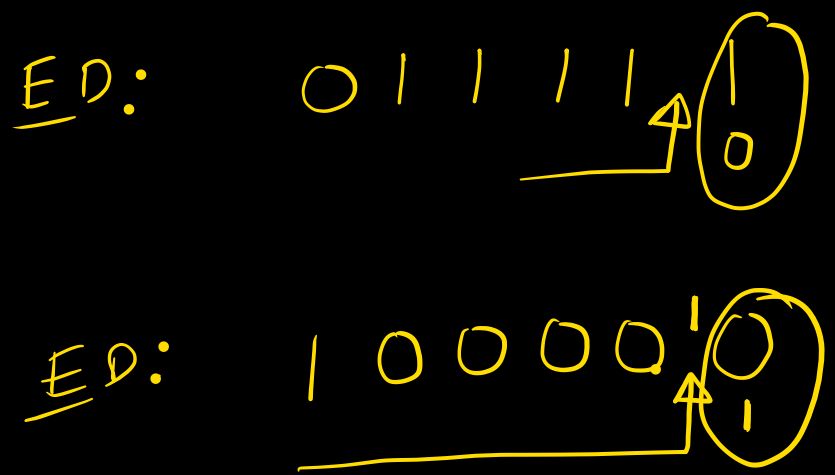
Can I use IP add as logical add?

Yes → using

Why is it called MAC add.?

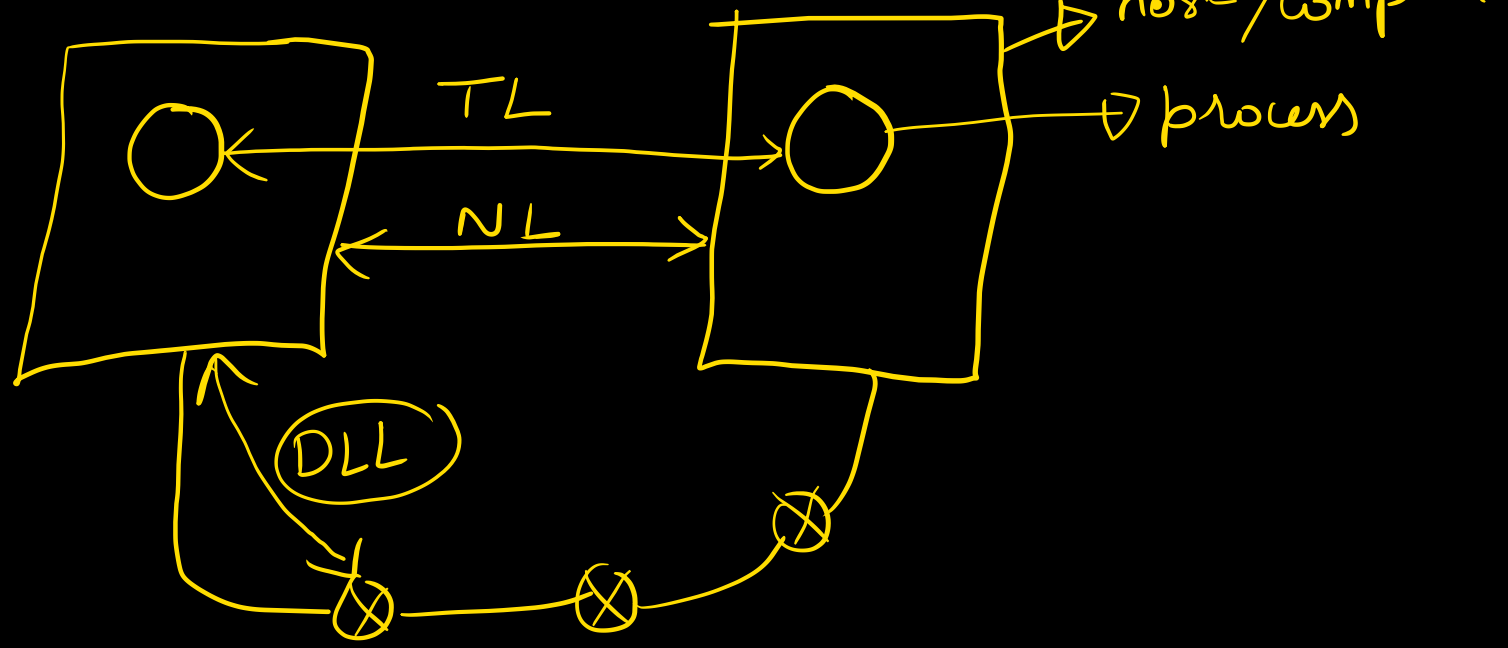


LLC - Logical link control
MAC - Medium access control
flow control
Error control


ED:
0 1 1 1 1 1 0
ED:
1 0 0 0 0 0 1 0

Network layer : (L-3) L1 - PL L2 - DLL L3 - NL

1) Host to host Connectivity

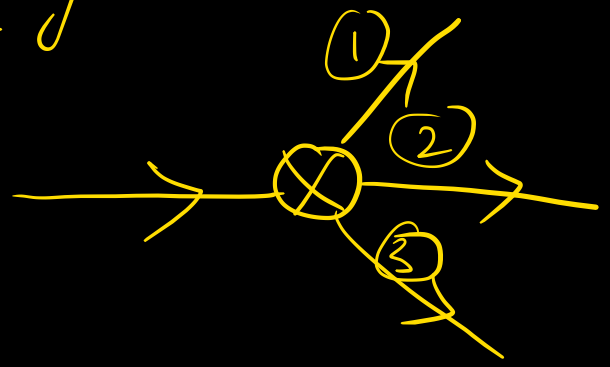


→ host / Computer
→ process

DLL - Hop to hop
NL ← Host to host
TL → end to end

2) logical addressing

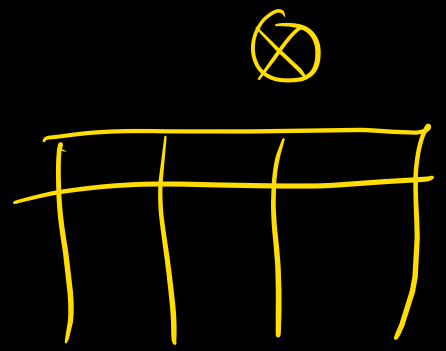
3) Switching



? decision of which link to use

↘ switching

4) Routing.



Process of Creating Routing table in Routing

5) Congestion control

IP doesnt do congestion control
we will not study congestion control

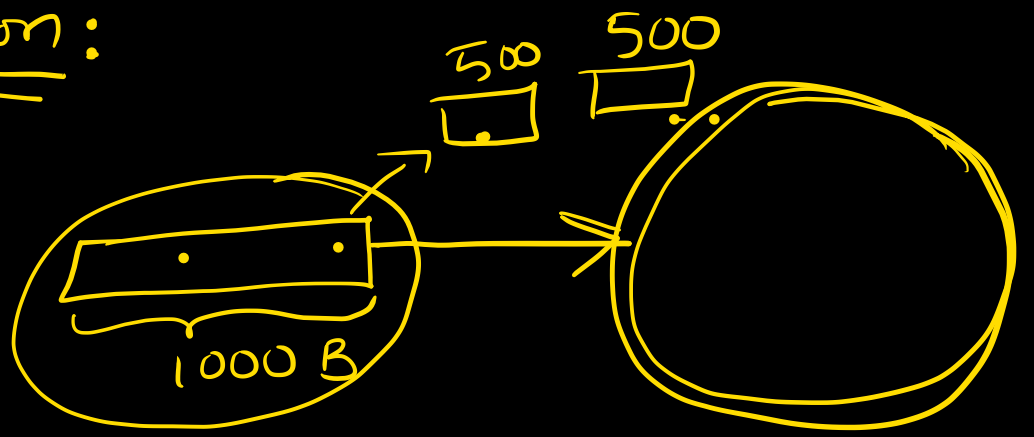
IP in internet protocol

↓

used at N/w layer

6) Fragmentation:

At IP, we will study this in detail

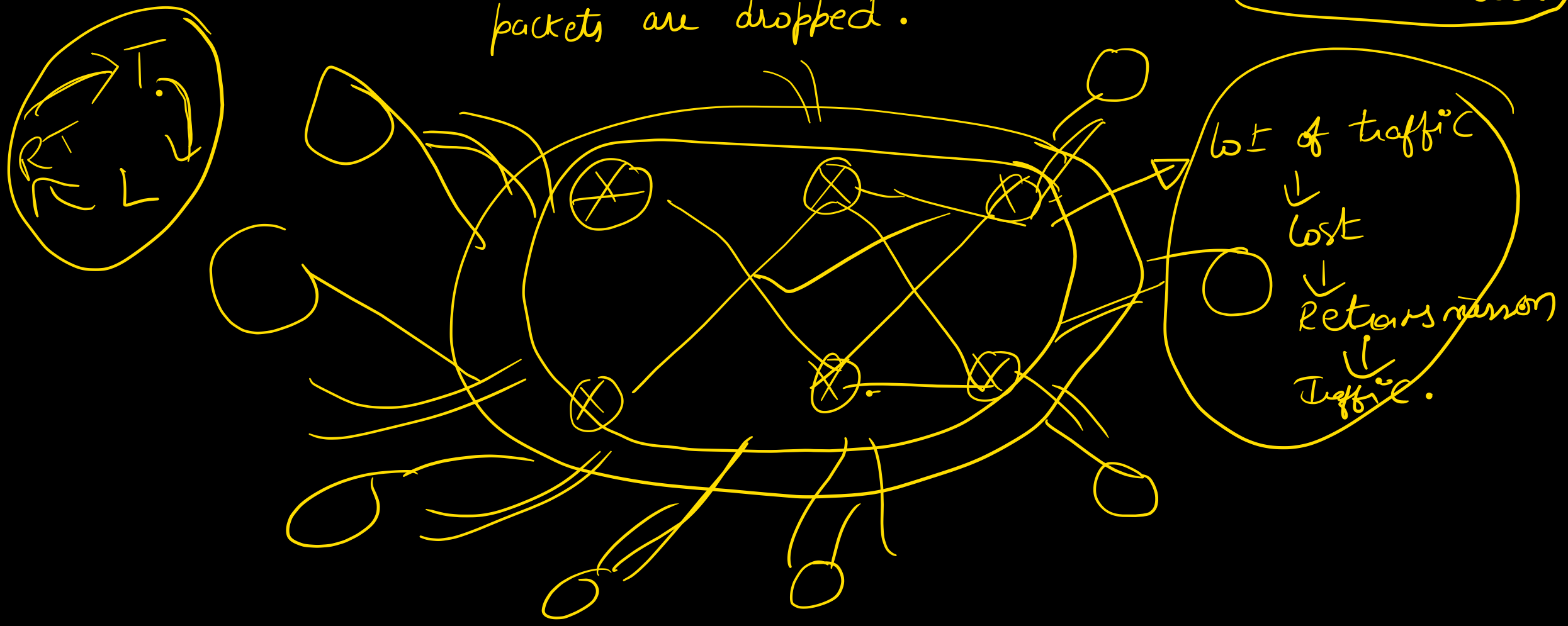


Maximum Transmittable unit (MTU) = 500

Congestion → traffic becomes too heavy and packets are dropped.

FC → S and R

CC → Network

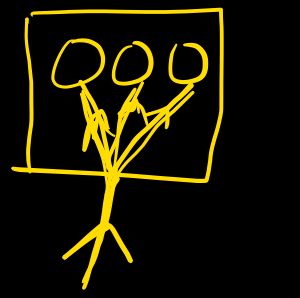


using RT → Switching

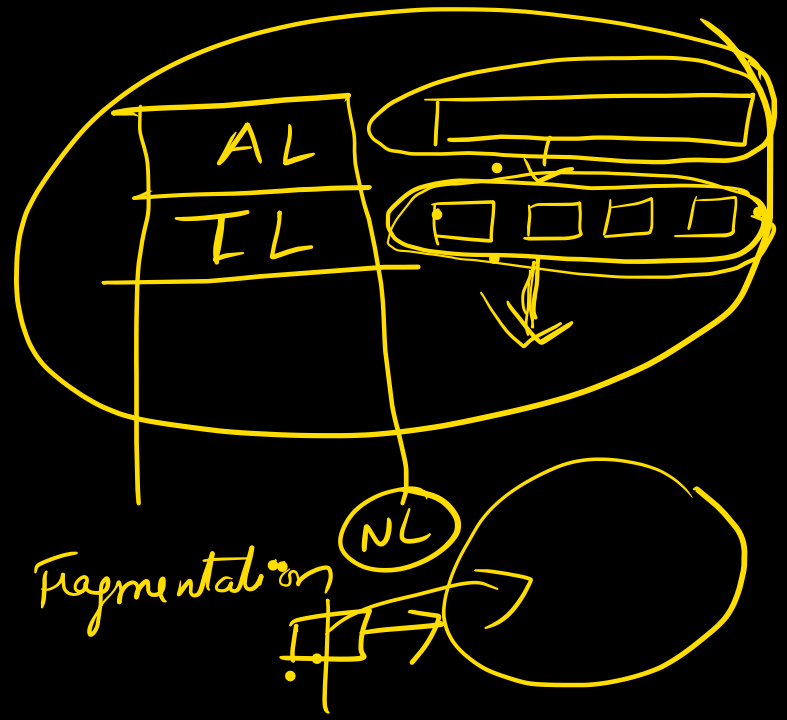
Preparing RT → Routing

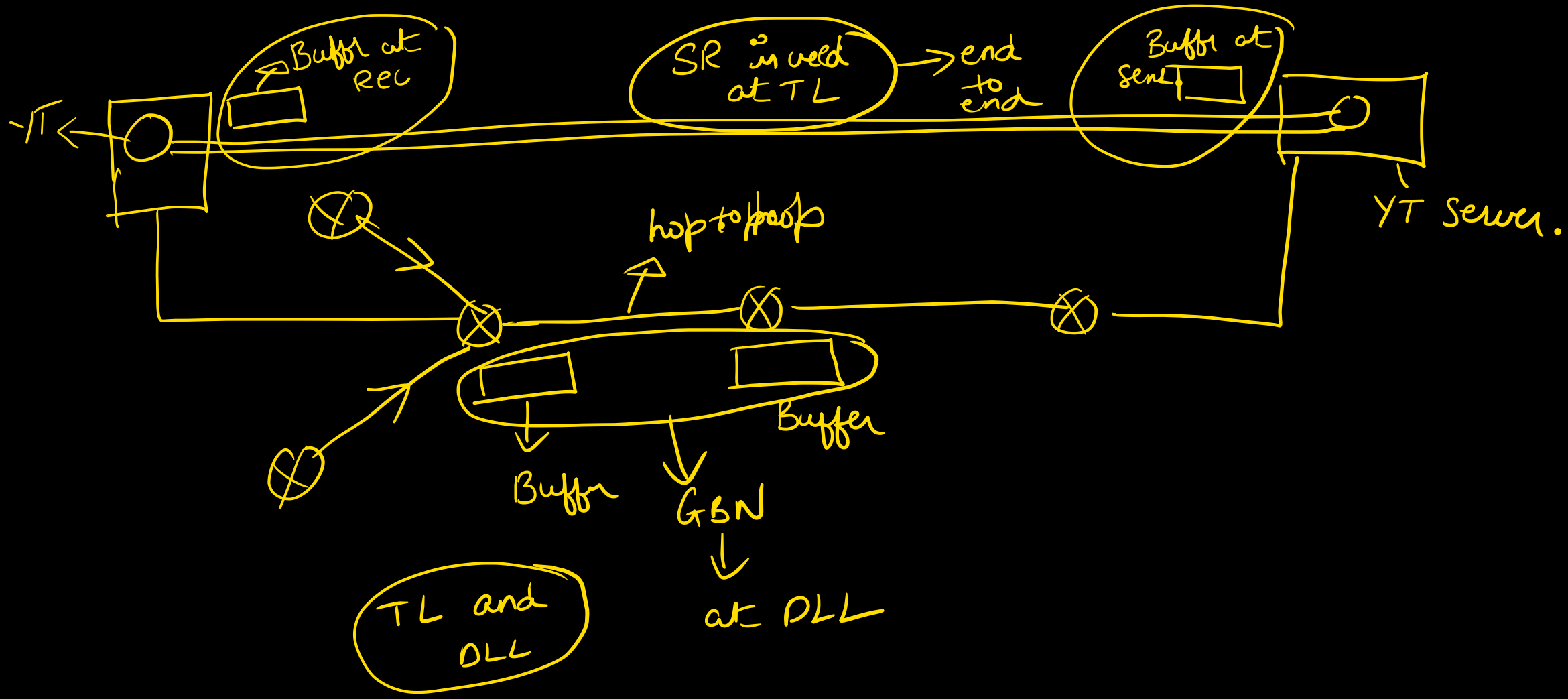
Transport layer:

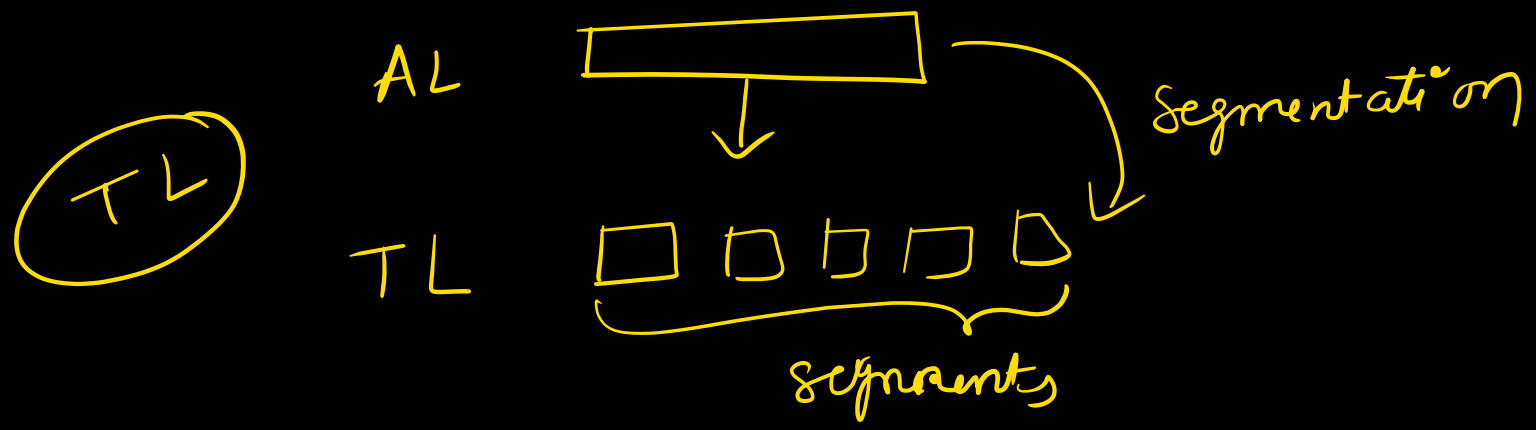
1) End to end connectivity
2) Flow control → DLL (Stop and wait ✗, GBN ✗, SR ✓)
3) Error Control (CRC ✗ and checksum ✓)
4) Segmentation
5) multiplexing and demultiplexing

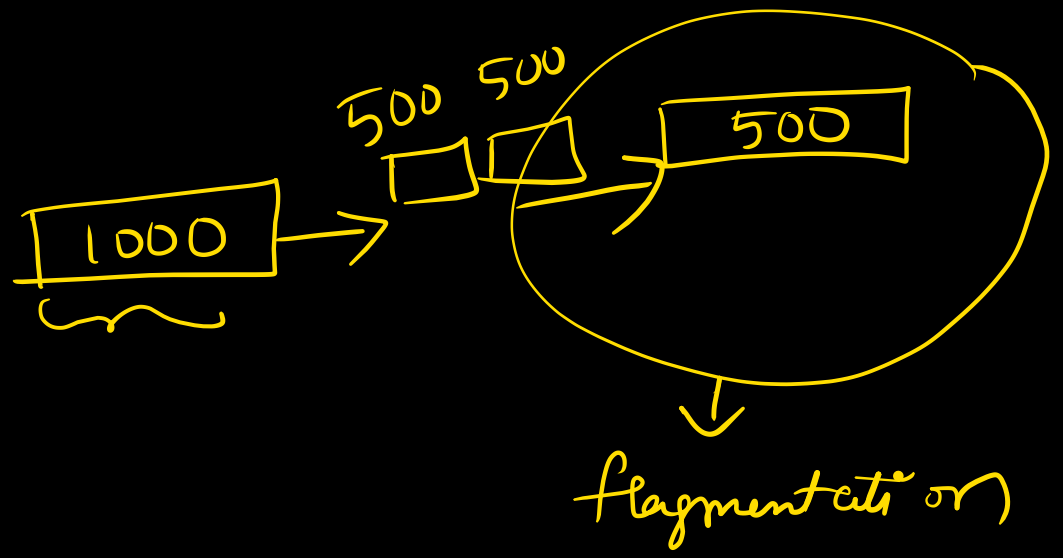


6) Congestion Control → TCP → CC




Buffer at Rec
SR in used at TL
end to end
Buffer at Sent
YT
YT Server.
hop to hop
Buffer
Buffer
GBN
at DLL
TL and DLL


TL
AL
TL
segmentation
segments


1000
500
500
500
fragmentation

TL → end to end → TCP.

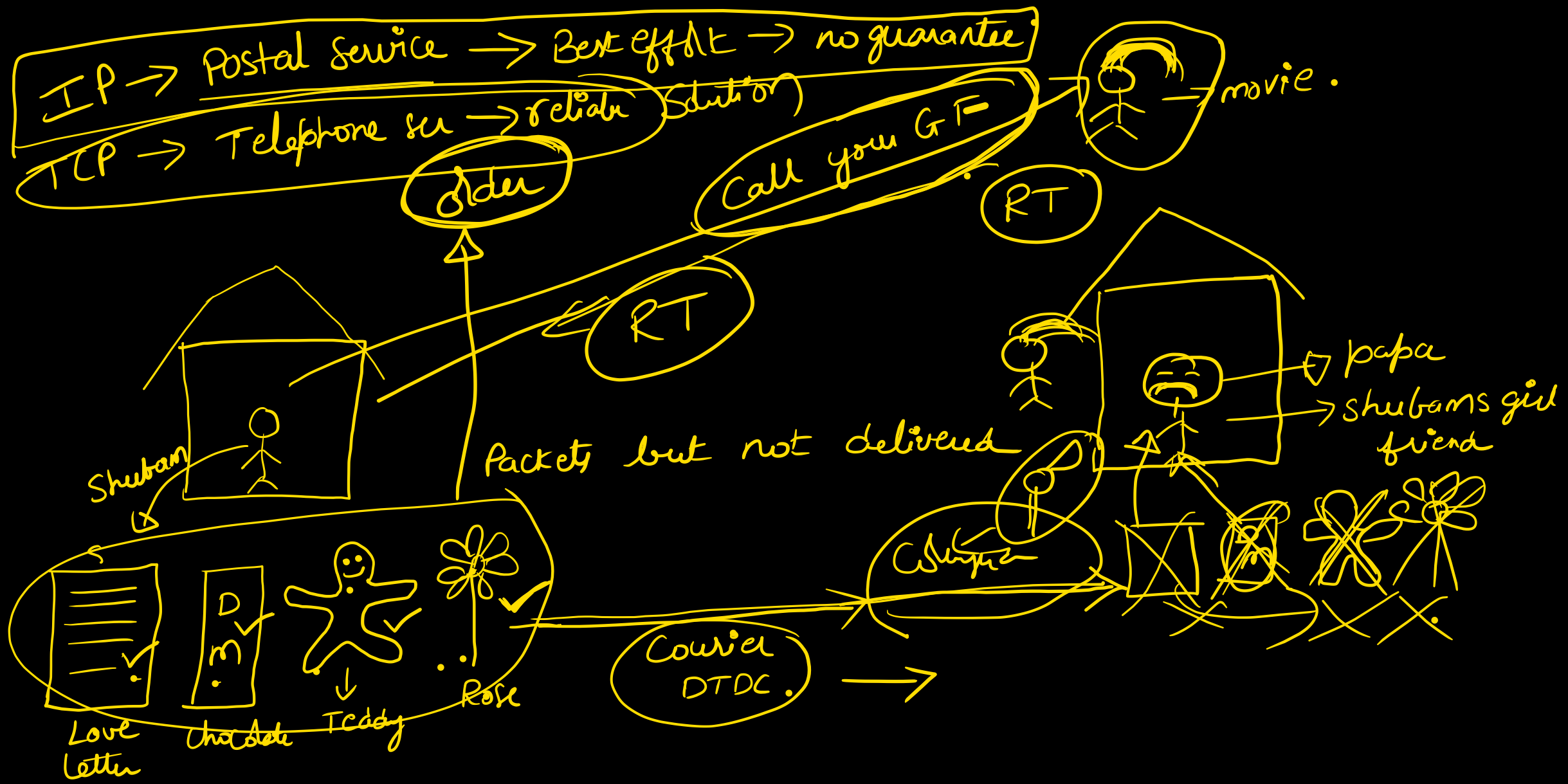
NL → hop to hop. → IP.

Why TCP is needed when IP can deliver packets?

Break 5 min.

example.

Intuition


IP → Postal service → Best effort → no guarantee
TCP → Telephone service → reliable
Order
Call you GF
movie.
RT
RT
Shubam
Packets but not delivered
papa
shubams girl friend
Courier
DTDC
Love letter
Chocolate
Teddy
Rose

TCP → Reliable - Connection oriented

↓

Telling that you are sending data and asking to be ready with buffers

IP → unreliable → connectionless

↓

Rec is not ready

UDP → Nothing → Not doing anything at transport layer.

↓

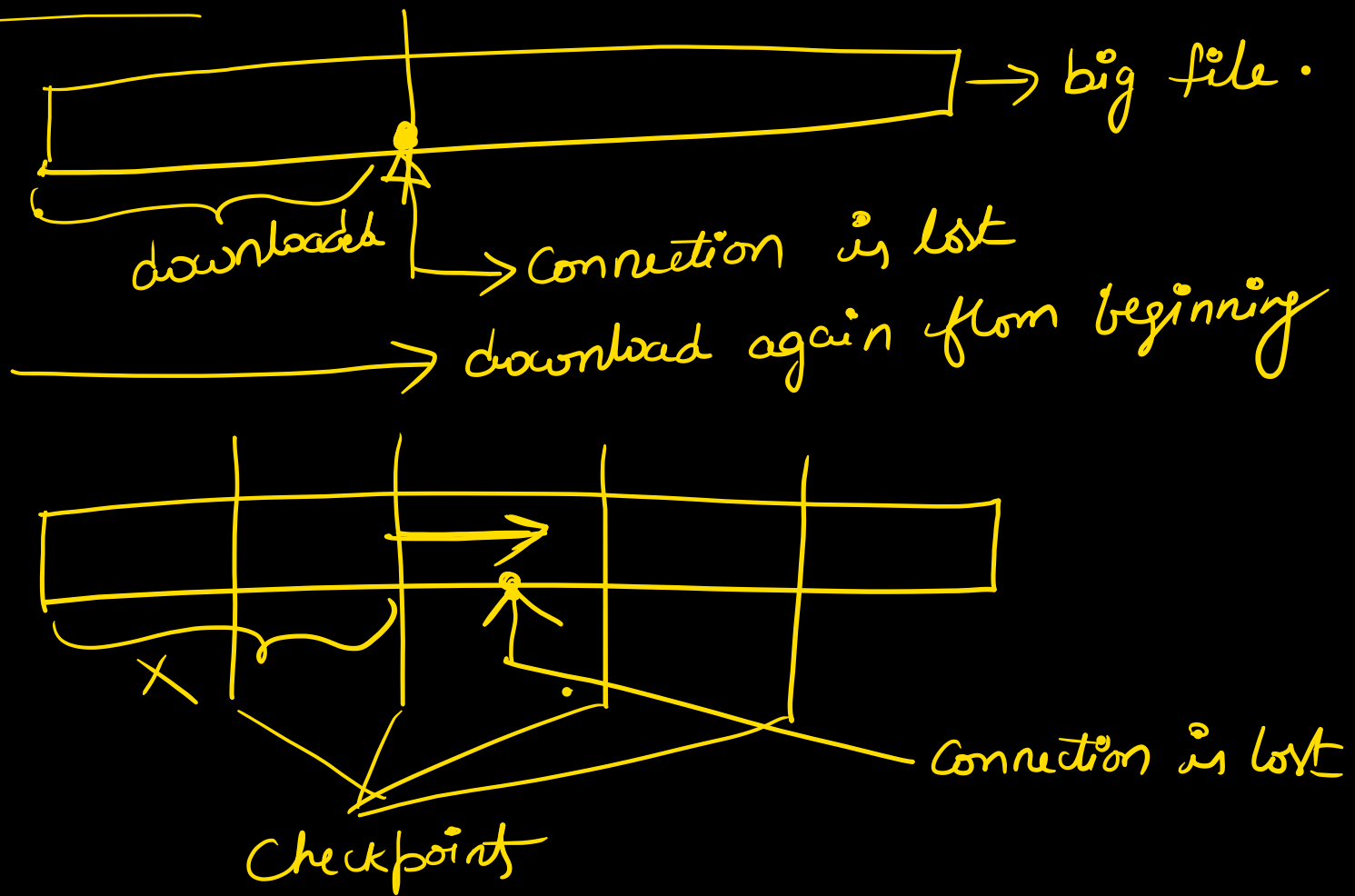
we will see this

Session layer:

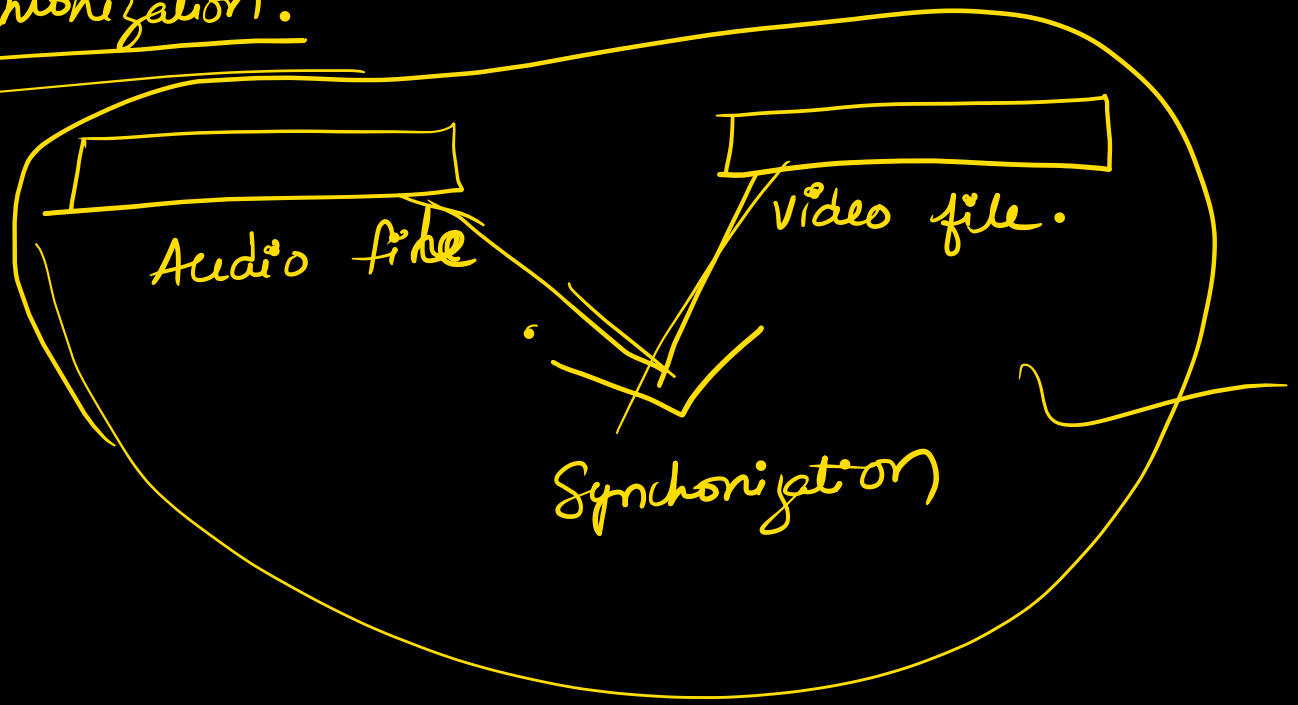
1) Authorization and authentication

- Authorization → Resource access → like permissions
- authentication → username, password

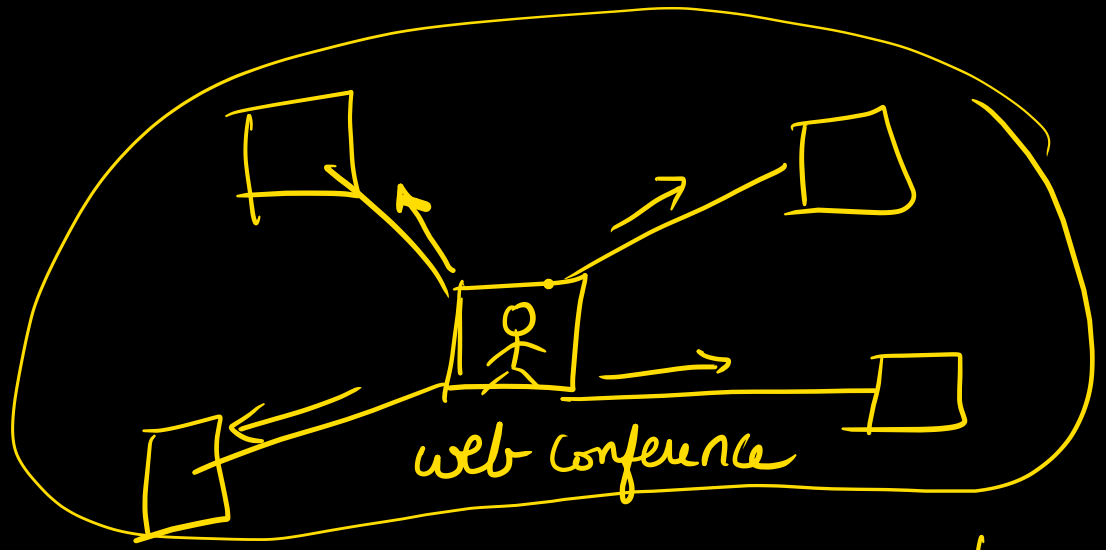
2) check points : Ex: Torrents



3) Synchronization:



4) dialog control



at any time only one person can speak.

5) Logical grouping:

grouping the operations together.

~~atomic~~ atomic operation

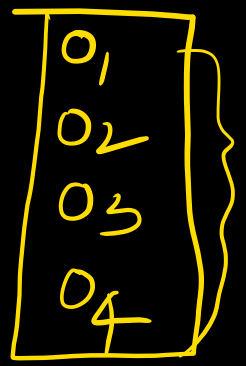
&

Acid properties.

↓

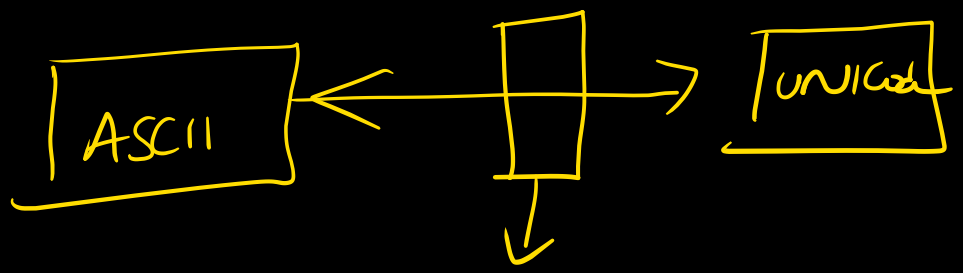
Grouping is required

DBMS



Presentation layer:

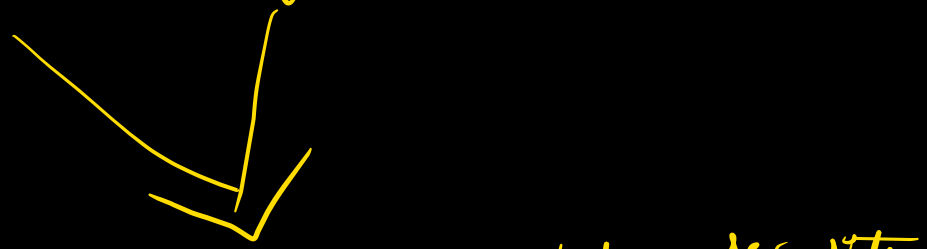
1) Character translation (ASCII, unicode, EBCDIC)



A → U

U → A

2) Encryption / Decryption



Privacy & data security

3) Compression:

→ Reduce the size

Application layer → depends on the application not related to networking