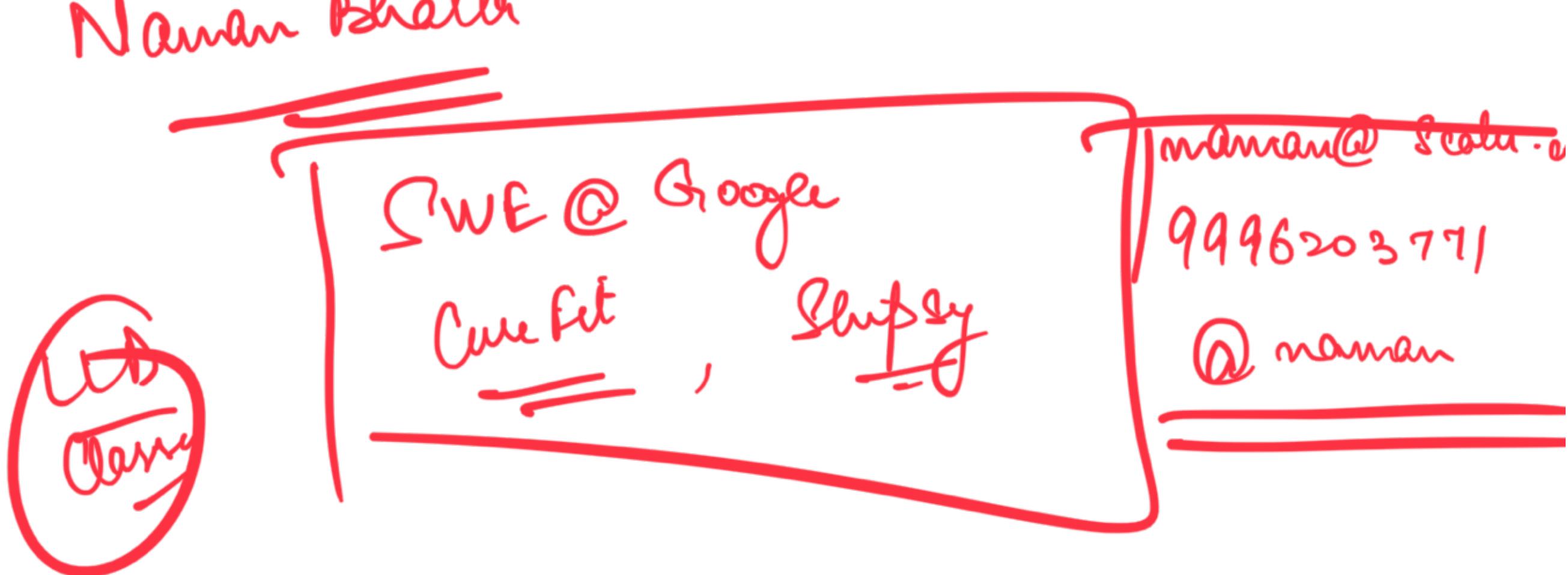


Naman Bhalla



→ CN Terminology → Port
→ OSI Model → IP

Agenda

→ OSI (Open System Interconnection)



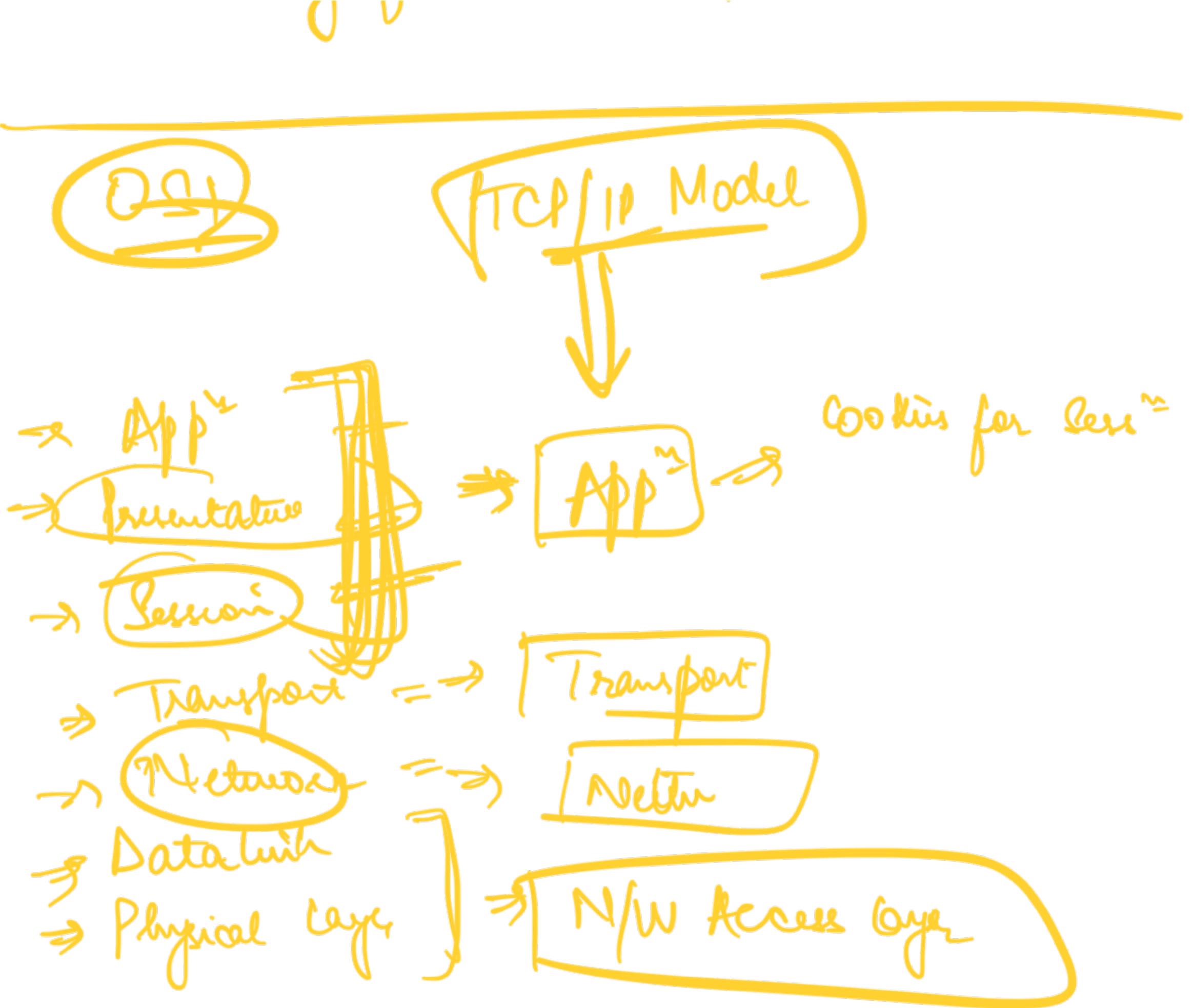
⇒ rebranding
of OSI

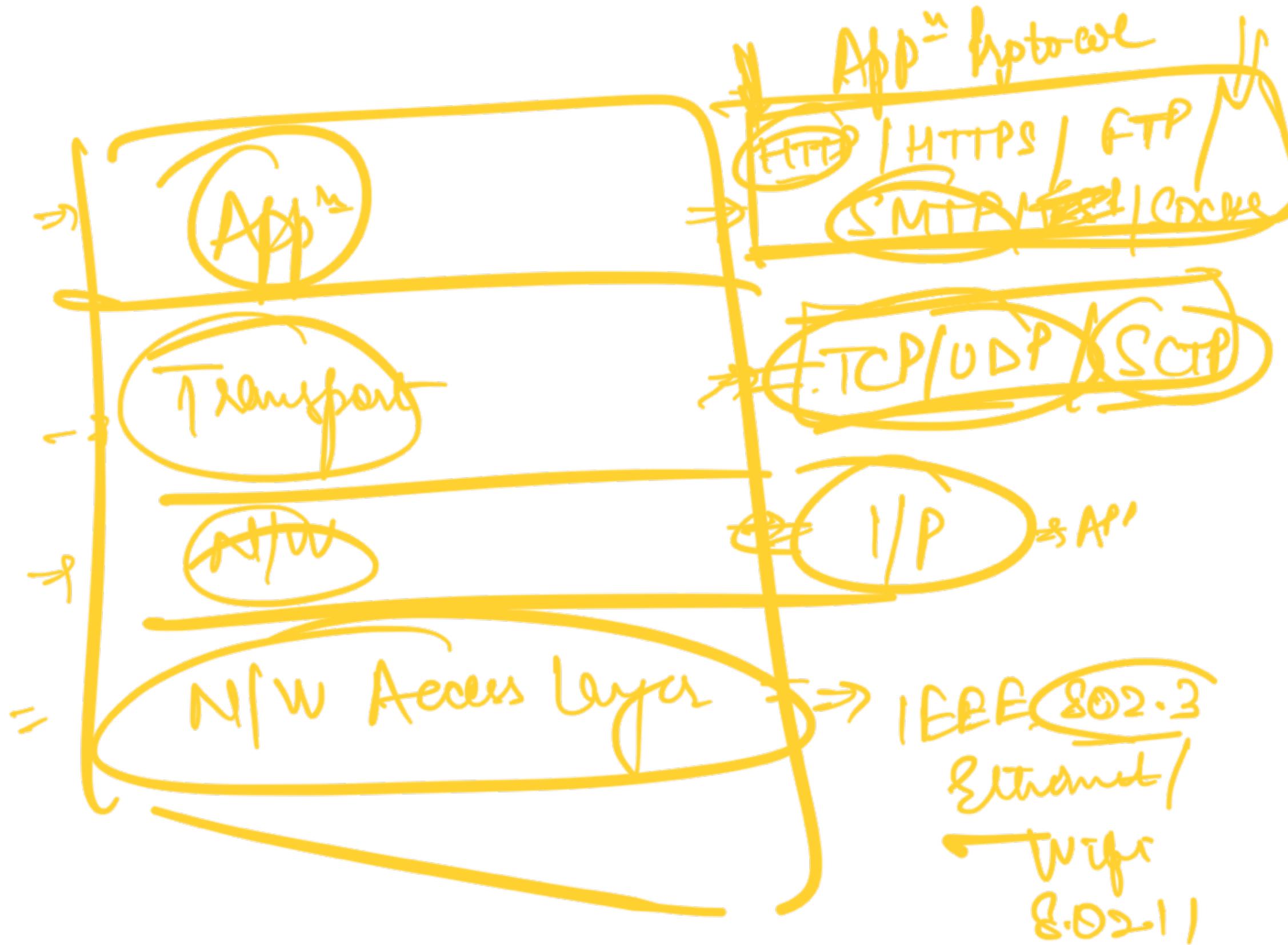
Model on which Internet
works

→ Protocols at different layers



⇒ Journey of a web request





Huge set of protocols to allow app of

^ a diverse set to work

RFC ≈ Request for Comments



HTTP \Rightarrow :80

~~SSH~~

FTP \Rightarrow :21 :23

SSH \Rightarrow :22

SMTP \Rightarrow :485

to identify a particular app

\Rightarrow Uniquely identify a process to transmit data over the net

:80 \Rightarrow

\Rightarrow in terminal \rightarrow all bound ports

Node TS



Any-Server-Name: 80 → Any-Server-name

google.com:80 → googl.cn

:443 → **HTTPS**

any-website



Browser



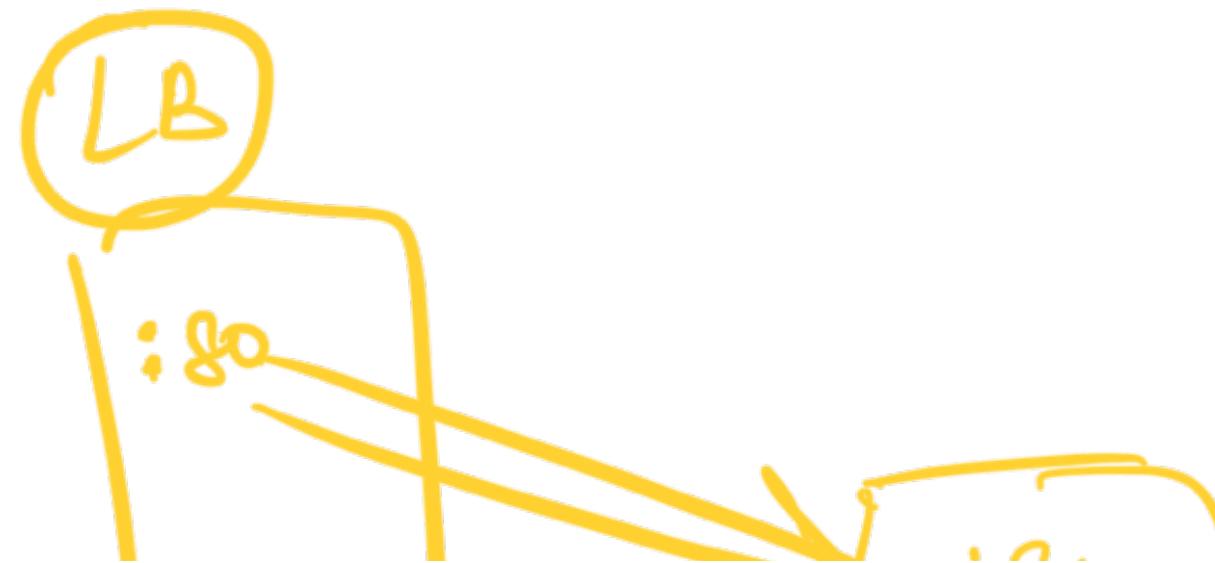
→ Most websites use this protocol by default

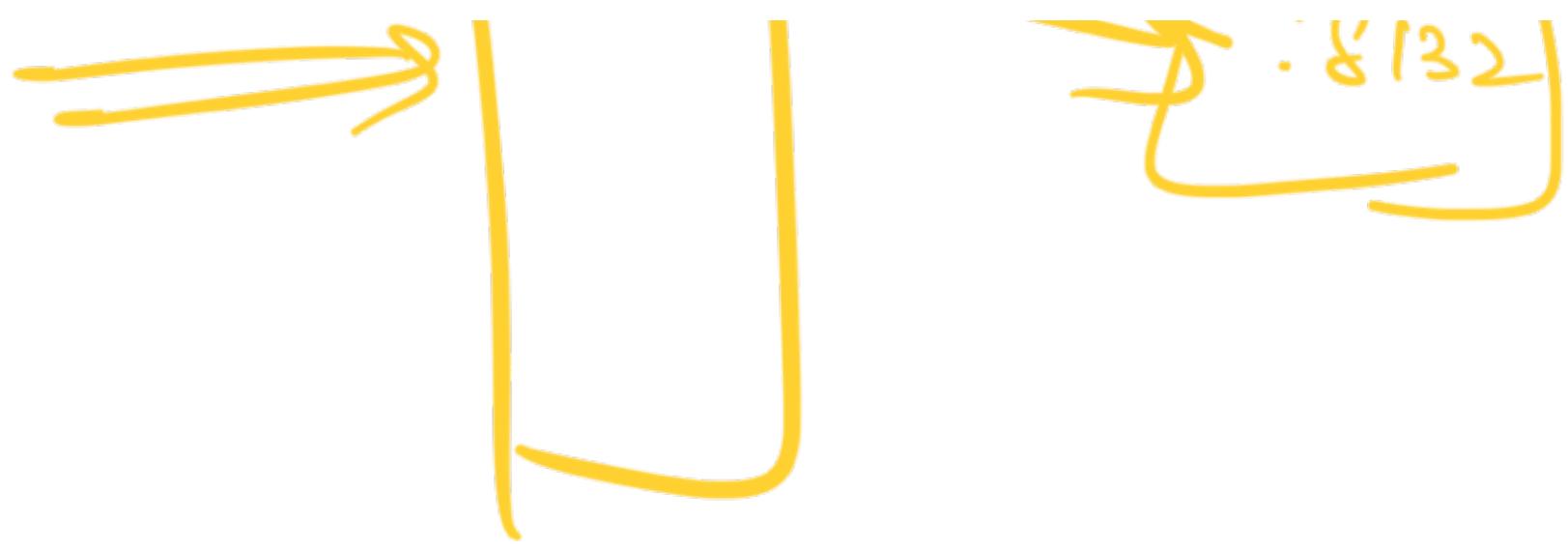
→ my IP is going to map to that port

~~http~~ : 111



NodeJS → : 5000
Spring Boot → : 8008





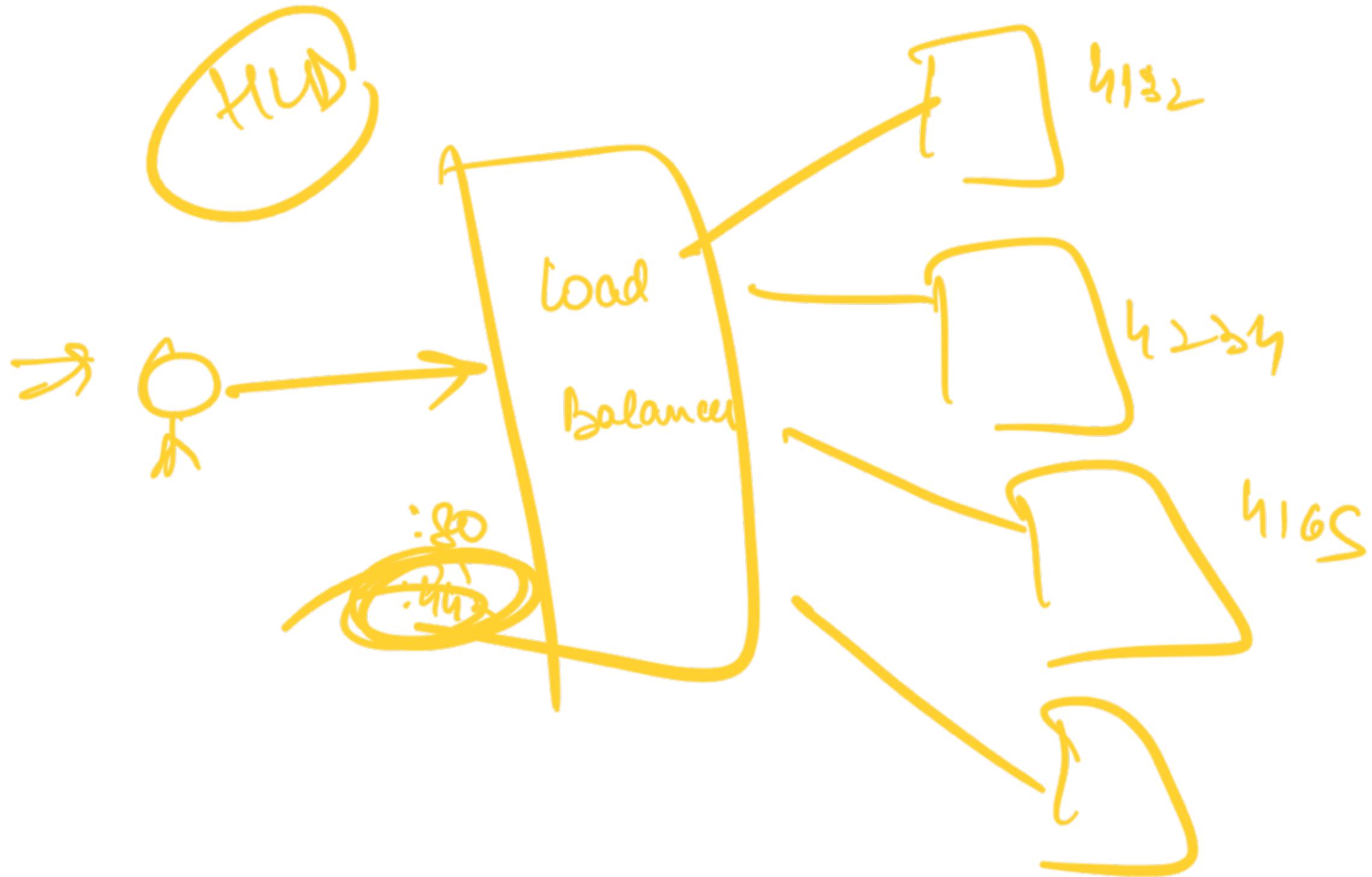
A hand-drawn diagram showing a connection path. A yellow oval encloses the text "1.2.3.4", which is followed by a yellow arrow pointing to the right, labeled with the number "1".

http: 1.2.3.4: 80

https://1.2.3.4



https://111.2.3.4:443



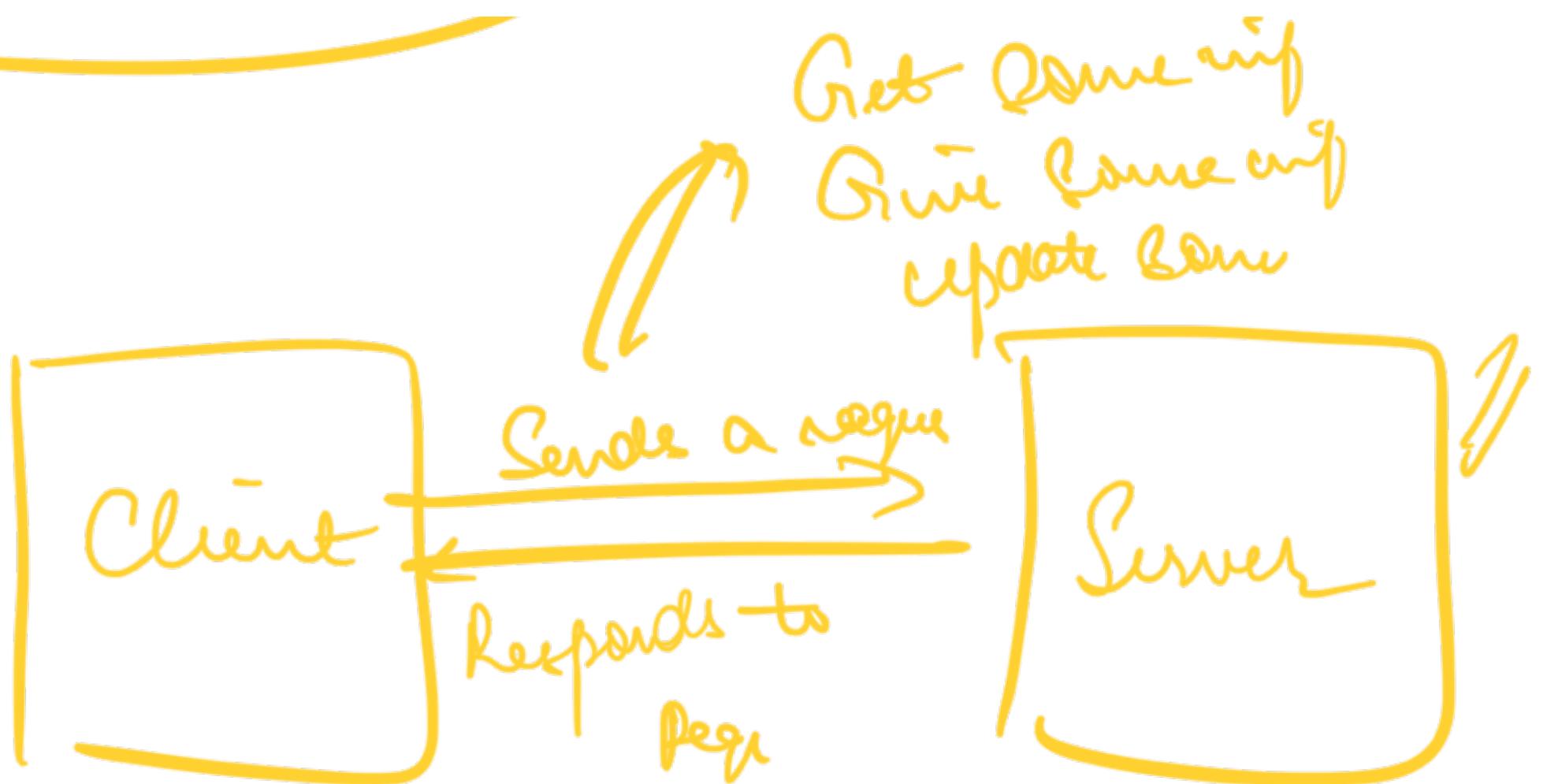
HTTPs → HTTP +  (Secure Socket Layer)

→ HTTP: Hyper Text Transfer Protocol

HTML
HyperText
Markup
Language



Request - Response protocol



Messaging \Rightarrow Websocket

~~Send No Emission~~

\rightarrow Websocket

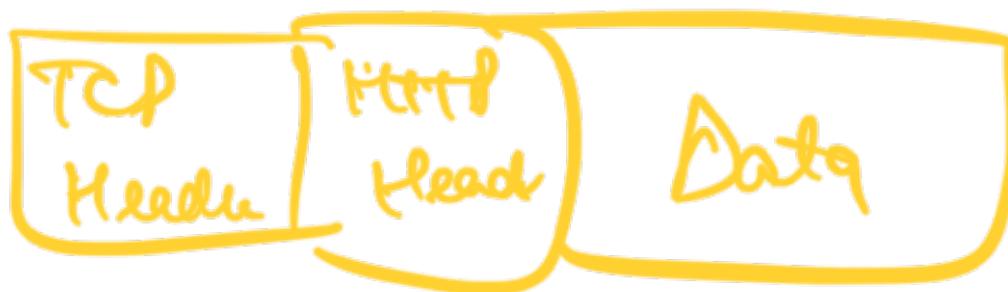
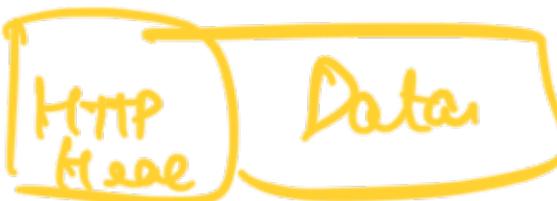
→ X MPP



Special rep in the seq of bits that are received or sent



↓ APP²
Data





So that receiver can make
sense of the data following
the header

↗ All the features that are supported by fields of header
of a particular ~~Protocol~~



: 80

: 443

(SSL/TLS)

- Request Response Behavior
- Client Dev

GET ⇒ Client want to get some data
get from a file lc

POST

- ↳ Client wants Server to store file
- ↳ Upload
- ↳ Sign up / Create a
- ↳ fill up a form

Request to search clothes on Magento

GET /POST/

GET

upload on img

JS



URI



Send a post

requests . createRequest (POST,
celle,
post

↳ Dev Classes

HTTP Status Codes

States of the request that you send

2XX = Success Has Happen

3XX ⇒ Resource Has Moved

4XX → Reasons doesn't exist

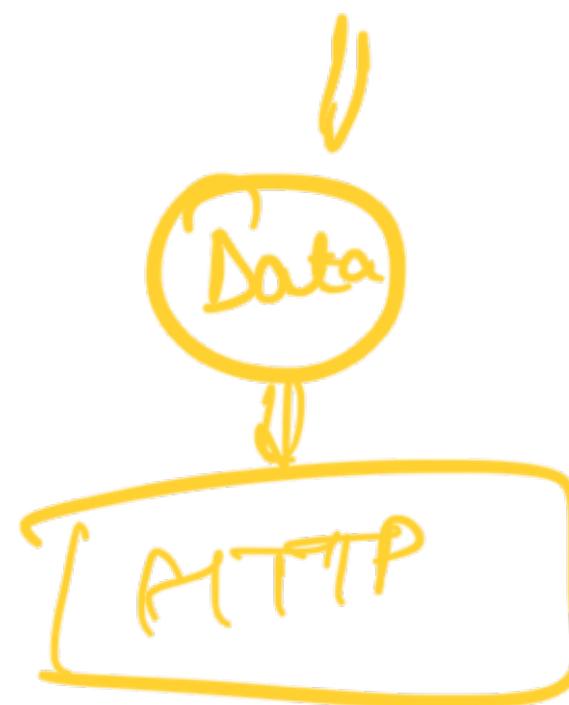
5XX ← Server is not able to req

10:42 AM

TCP , UDP, IP

2KB
- Images

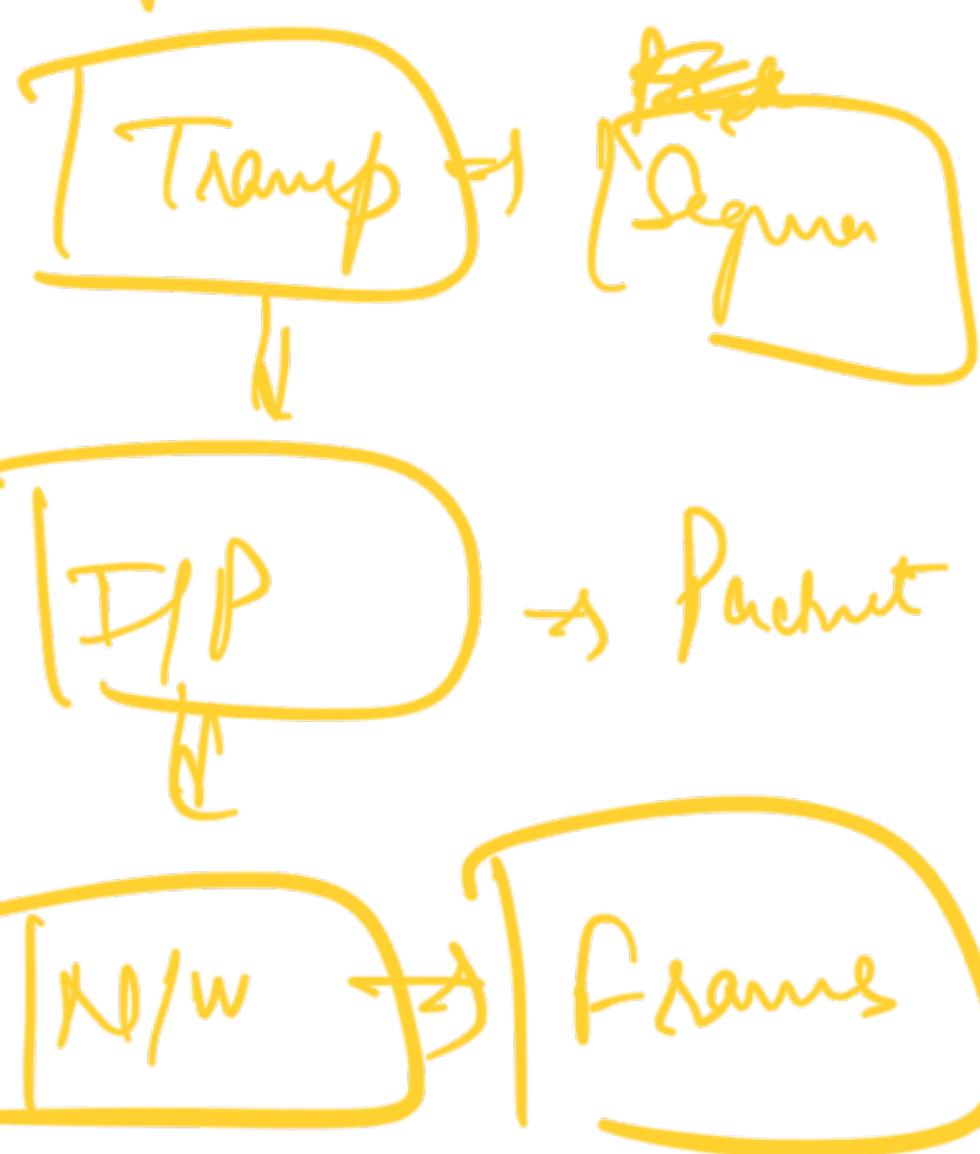
6MB



Broken down into multiple

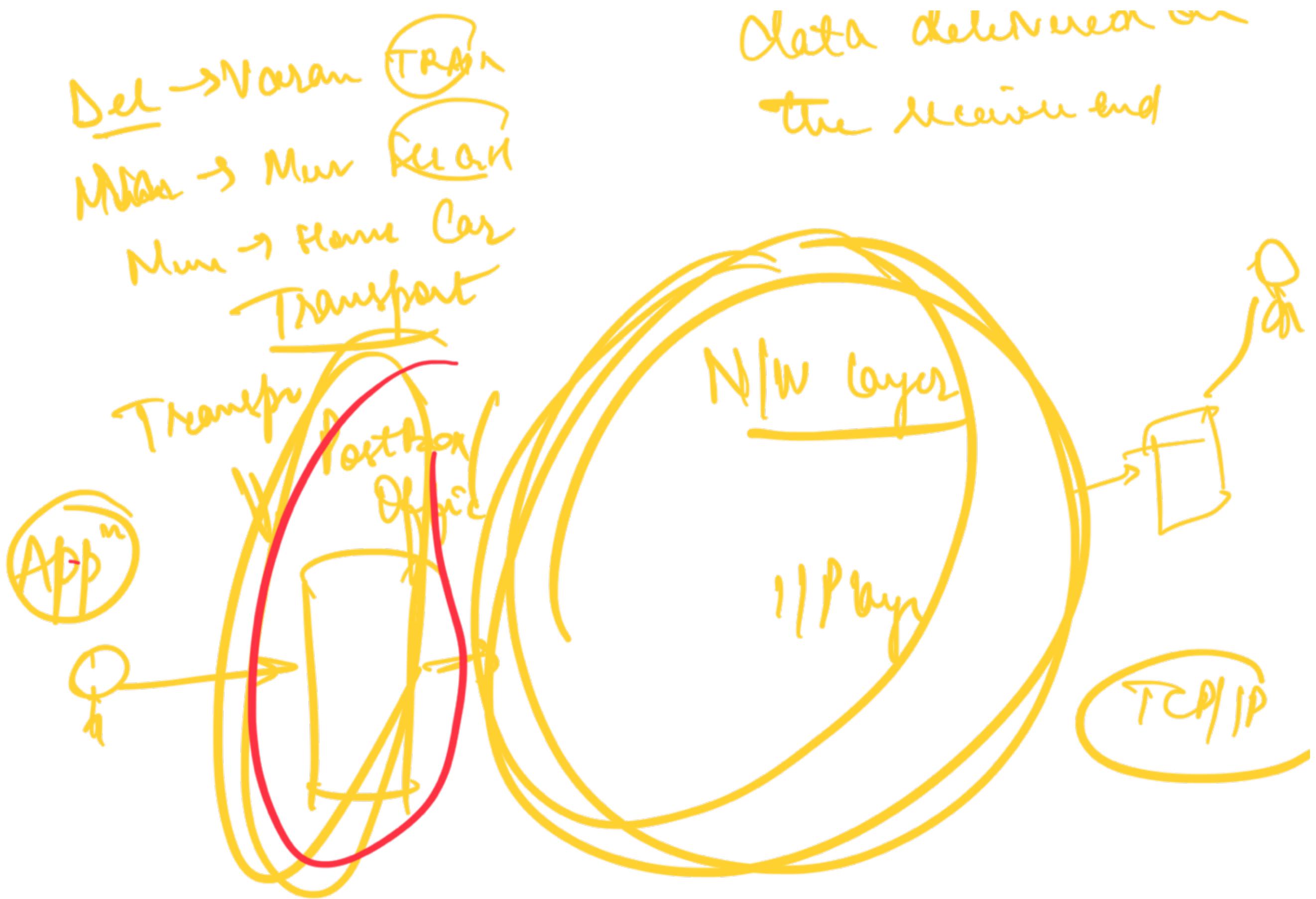
HTTP
2MB

chunks or accepted
by the ~~lower~~ protocol

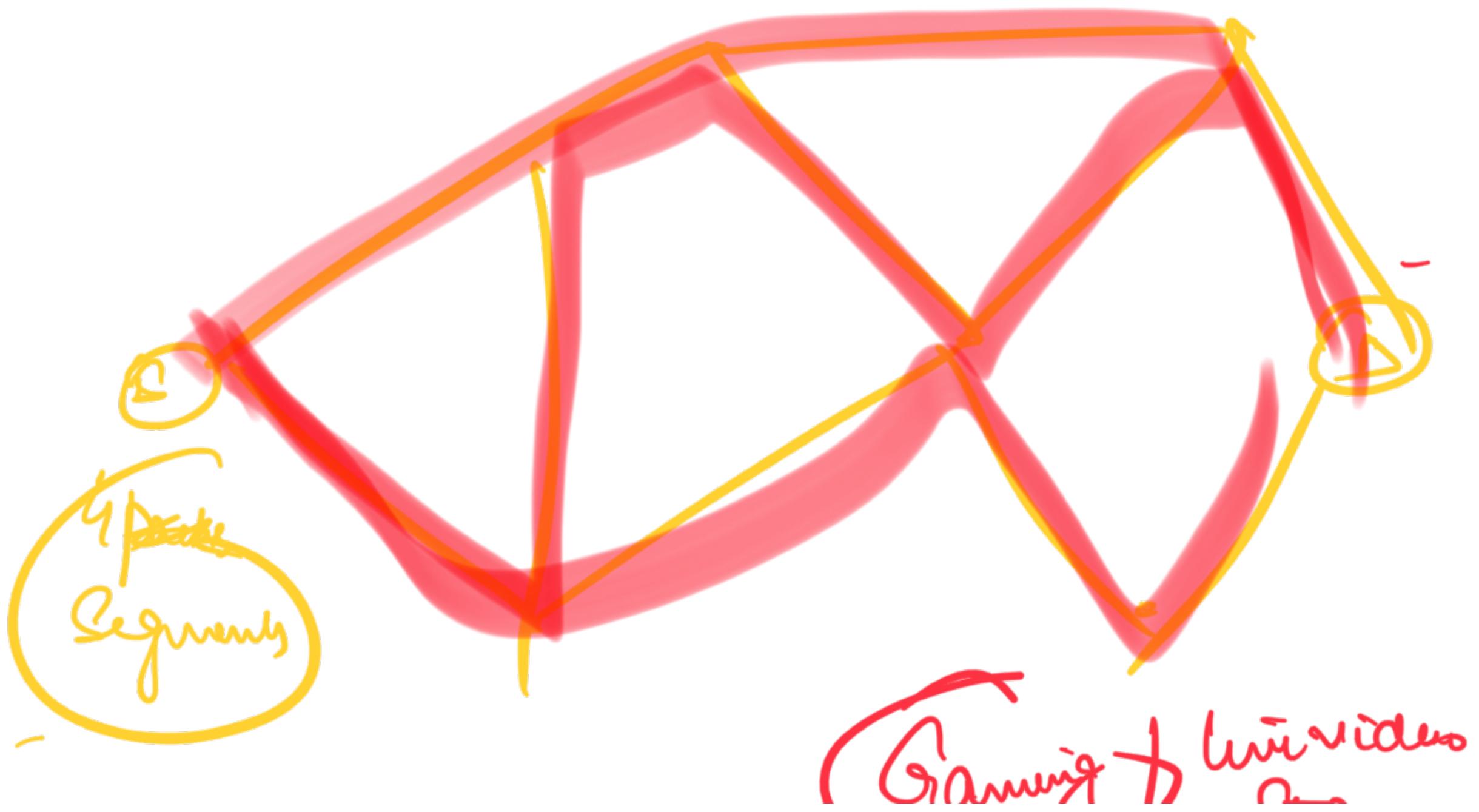


Transport \Rightarrow works with the
N/w layer to get the

Data delivered on
the receiver end



⇒ Break into multiple segments



TCP

→ Transmission Control
Protocol

→ Guaranteed ordered
delivery of data

→ Data delivery guarantee

→ More fair, less fast

UDP

User Datagram Protocol

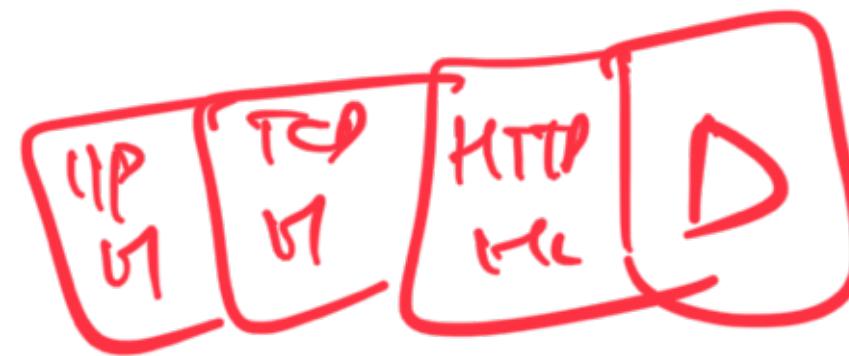
→ Don't guarantee of order

→ Don't even guarantee. But
will make best effort

→ Efficient. Less Header Size



→ Gaming Video Game



↳ Size of UDP Header << Size
of TCP Head.

→ Connection Oriented → Connection less

Protocol

No to esr

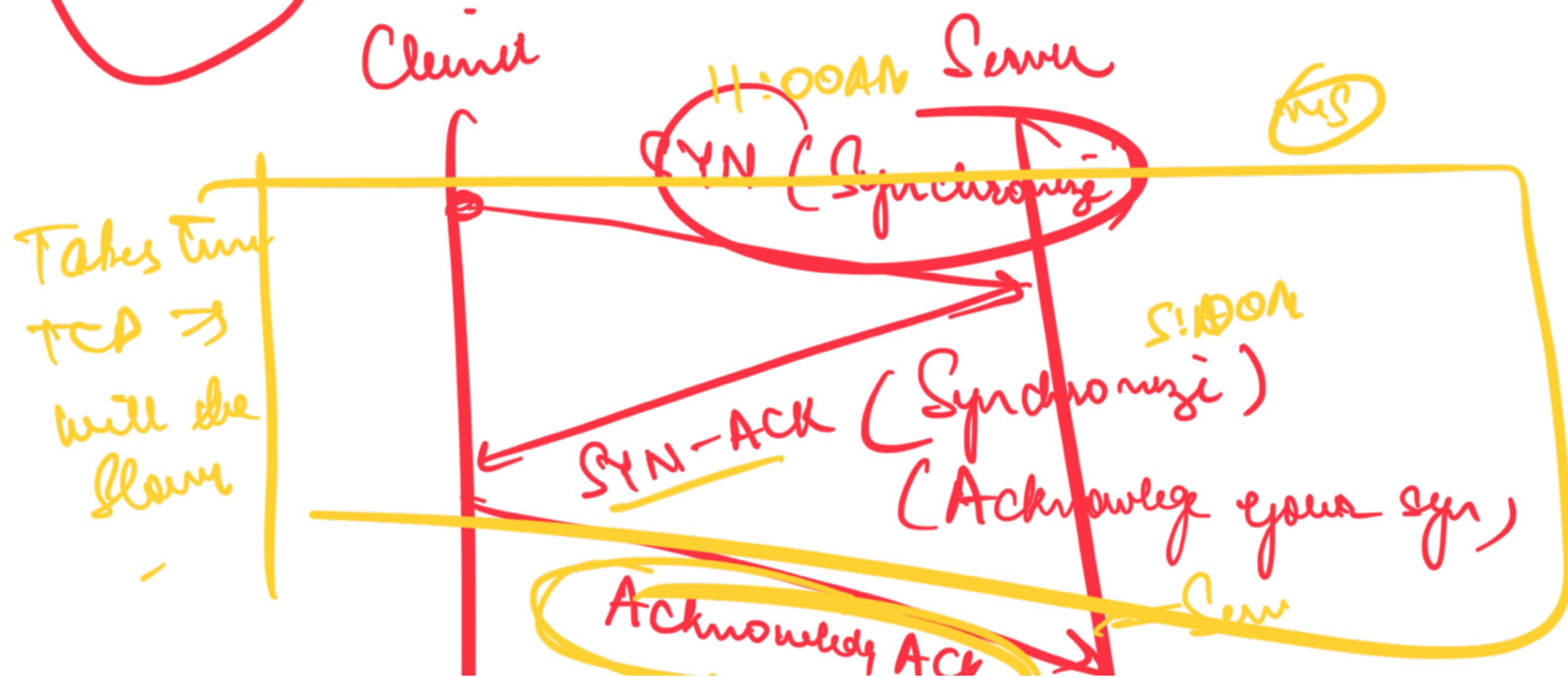
① TCP will set up a conn with
the receiver

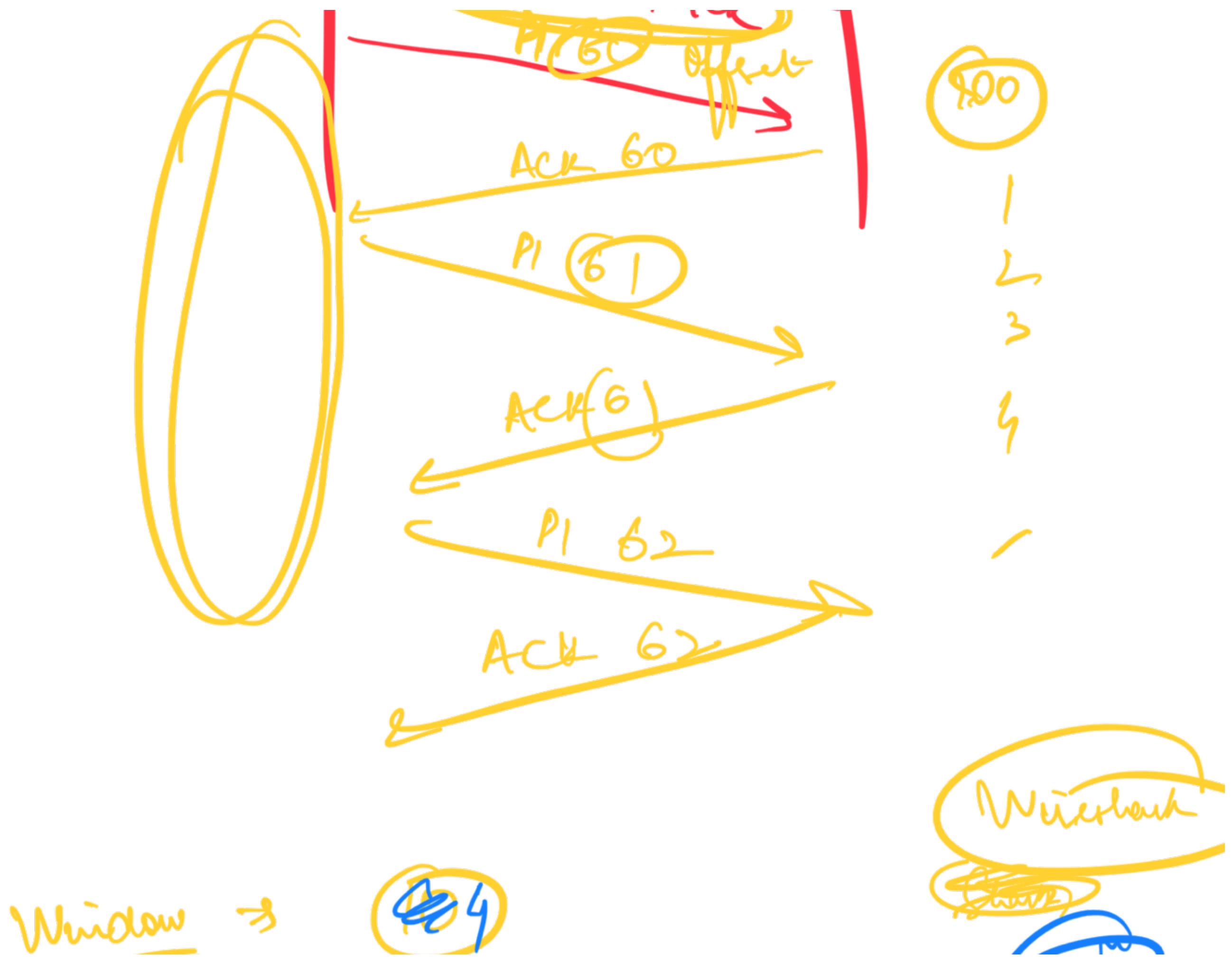
② TCP will send data over that conn
↳ TCP 1 step

→ No need to setup conn in UDP

→ Just send every data with

3 Way Handshake



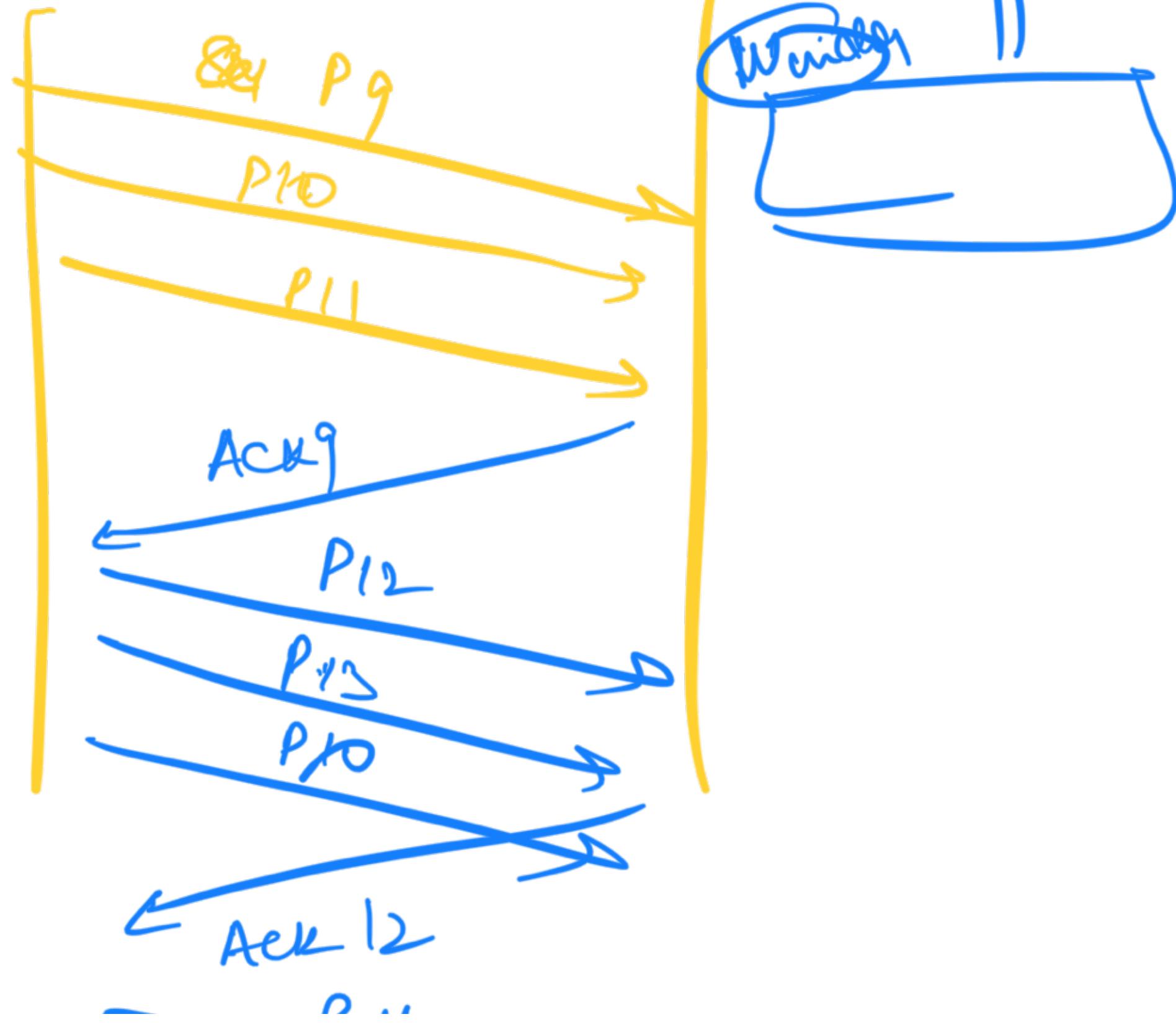


Segment 9

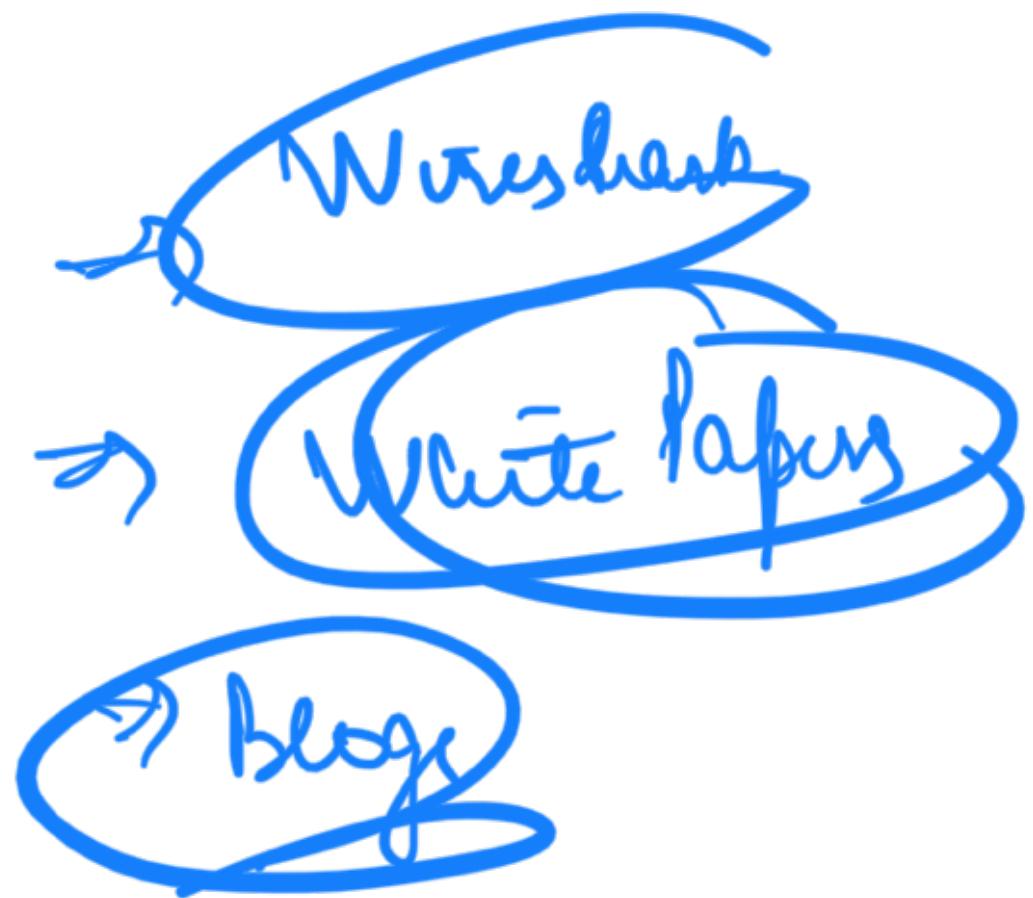
seq 10 11 12 13

Client

Server



R14



Video Streaming \Rightarrow UDP
Without video calls \Rightarrow TCP



IP



SMTI, DNS,

⇒ TCP vs UDP

⇒ 3 way Handsh

⇒ Diff type of HTTP n

⇒ Diff type of HTTP w