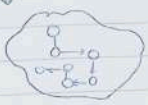
-Computer Networks-

- computers connected together (in layman terms)
  ↳ network

A — B (computer nodes)

⇓

and now interconnection of all networks = Internet



general form of internet as connection of computer networks

→ How internet started?

↳ All started with ARPA (Advanced research projects Agency)

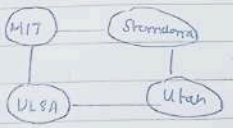
↓ to do scientific & research etc

to communicate with each other ← 4 ports

↓

ARPANET

TCP/IP



→ to share files or documents using TCP protocols and while transferring, no data should be lost (main task) with secured network

↓

∴ for this different rules must be required for the same (how particular data is been sent)

protocols (eg. TCP, UDP etc)

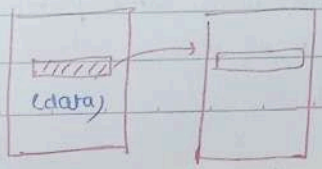
(ARPANET)

↓

was a research project

↓

but linkage b/w the files to transfer our focus to other document was missing



URL — Uniform resource Locator

∴ WWW (world wide web) came into picture

↓

stores the documents and we can access documents via the

↓

connection of all webpages on web servers

↓

but the problem of "search" engine was there which was not there in WWW.

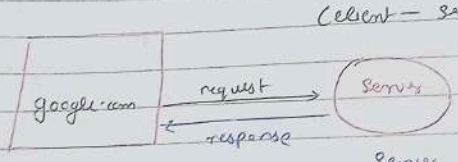
↓

Further modifications were after that mode

↓

Search engines were developed (first was YAHOO)

(client — server Based architecture)



Client

Server

for different countries, communication is been made via wires beneath the ground or ocean bays

(Client sends request to server to request some of its documents or stuff and expects back a response to get all resources)

Protocols - rules decided by internet society on how data is been transferred, manipulated etc.

↓

TCP (transmission control protocol)

→ data will be sent to the receiver without any support or data been corrupted

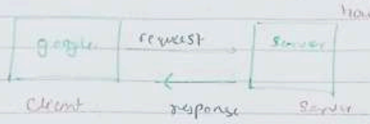
user datagram protocol

→ UDP – we don't care about the reliability of data (100% secured or not), we just care if even a small segment is reachable to receiver or not

→ HTTP (hypertext transfer protocol) – used generally by web browsers

↳ used to define the format of the data that is to be transferred b/w websites (client → server)

how is data sent through also (represented in HTTP)



→ How data is been sent over a network:

Data → divided into segments or chunks → packets been distributed

↓

call (individual been made) → According to the call, the data is been sent

↓

now how to find which server to be connected etc.? → Computer / Server identified by an IP address for individual recognization

↓

like a phone book directory

IP address (Basic idea)

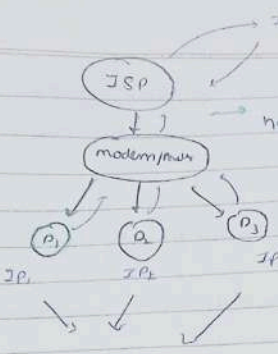
X . X . X . X

↓

range (0–255)

Jack → 9910765

↳ if we call google.com, we are sent to this IP address



has global IP address

all devices connected to this modem would have same IP address in the world

(Local IP addresses)

(Router/modem uses DHCP protocol to assign local IP address

(Also a set of rules and regulations) ← DHCP → dynamic host control protocol

Suppose P_1 requests google.com, then response is been sent from P_1 → modem/router → ISP → Internet

(now here modem decides who has actually requested for service (P_1, P_2, P_3))

(detects using IP address) ← (done using NAT (network address translator)

but can we check for which application in that particular device (say P_1) are we making a request?

↓

(MONGODB or google's application's ports)

↓

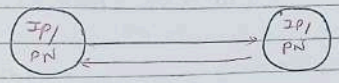
this is done using ports

(As a single computer / device might be working on numerous no. of applications)

DELTA
Date / /
Pg No.

→ port numbers are specifically beenused to differ the application on single device

Ip → decides where our device is been located



(Device1) (Device2)

PN - Port number (decides on which application we need to establish communications)

Ports → 16 bit number

↓

16 cells and each cell can contain 1 bit data (either 0 or 1)

∴ Total port no = 2^{16} = 65,000 port nos

(webpages using HTTP and its work's done on port = 80)

HTTP related work - at port 80

for mongodB = 27017 port no

Now from (0 - 1023) → reserved port no's

↳ reserved for HTTP stuff

(1024 - 49151) → registered port no's

↳ registered for specific applications like mongodB, mySql etc.

SQL (that we run on our systems) - has port no as 1433

After 49151, remaining port no's we can use for our own purpose accordingly

→ When we are sending data from one computer → upload
** when we download data from (receive) one computer to another → Download

How do we determine the speed of ISP (Internet service provider)

↳ for eg. 1 mbps ? ⇒ mega bits per second
1 (10^6 x bits per second)

1 gbps = 10^9 x bits per second
1 kbps = 10^3 x bits per second → very slow

How computers interact with each other?

guided way — Unguided way

guided way: Set of path already defined (pre-defined)
↓
for eg. - Nodes connected via wires (physical medium)

Unguided way: There's no fixed path for data transfer for eg wifi, Bluetooth
From one country to another via cables under the in oceans

Bigger entities → Smaller entities → Internet Service Provider (ISP) → Local people

Physical transmission → optical fibre cables, Co-axial cables

Wireless transmission → bluetooth, wifi, 3G, 4G LTE, 5G

DELTA
Date / /
Pg No

* Local Area Network (LAN) : Small range network, data transmission rate is very high due to short range
↳ for homes / offices etc
↓
via ethernet cables / adaptors / switches
→ can connected via WiFi

* Metropolitan Area network (MAN)
↳ network across a city

* WAN (wide Area network) - Network made across countries
↓
we would be using optical fibre cables for this network

→ Internet is a collection of all 3 above mentioned networks (LAN, MAN, WAN)

① SONET - Synchronous optical networking
Carry data using optical fibre cables ∴ can carry longer distances

② Frame relay - Used to connect our LAN to WAN

* What is a modem?
↳ Basically used to convert digital signals to analog signals and vice versa
↓
0 1 0 1 1 0 1 0 → (Electrical Signal)
(So that we can transfer it over a telephone line)

ISP provides services to locals via routers

↳ Tier 1 internet providers subdivide internet via ISPs

or another modem on receiving side]

that would recover digital data encoded in image

* What is a router?

↓

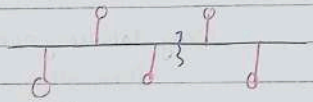
It's a networking device that forwards data packets between computer networks. One or more packet-switched networks or sub networks can be connected using a router

↓

By sending packets to their intended IP addresses, it manages traffic b/w different networks and permits several devices to share an internet connection

→ Network Topologies:-

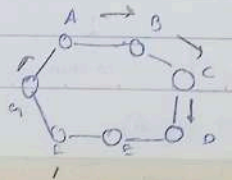
① Bus topology → every node connected to a single transmission path (cord or channel)



If any linkage in between gets broken, then it will spoil entire network

and also only 1 person can send data at a time

② Ring topology- Computers connected in a ring, eg: systems communicate with one another



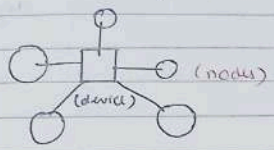
* If cable breaks in between, no transmission further possible

+ Lots of unnecessary calls are been made for eg we need to send info from A to E ∴ we only have necessity to go via B → C → D ∴ more no of calls been made

(3) Star topology -

A central device connected to all nodes (computers) for eg if 2 nodes need to communicate (send data), they would have to go through central node (device)

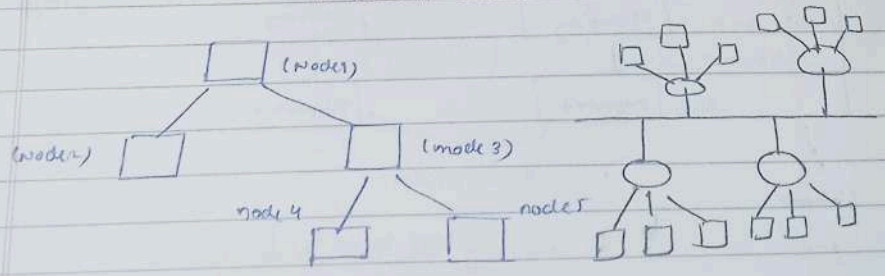
→ If central device fails, then network as overall fails



(4) Tree topology - A type of network topology in which the nodes are organised in design of tree (Sort of combination of Bus + Star topology)

↓

Tree topology - Having a root node from which branching cable starts and branching cables connects root node with all nodes in hierarchical manner

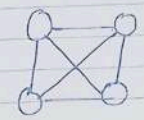


(Tree topology)

(5) Mesh topology - A type of network topology in which each node is connected to every other node fully through dedicated link in the network.

No of links = $n * \frac{(n-1)}{2}$ n = no of nodes

→ Very expensive because of high cabling cost

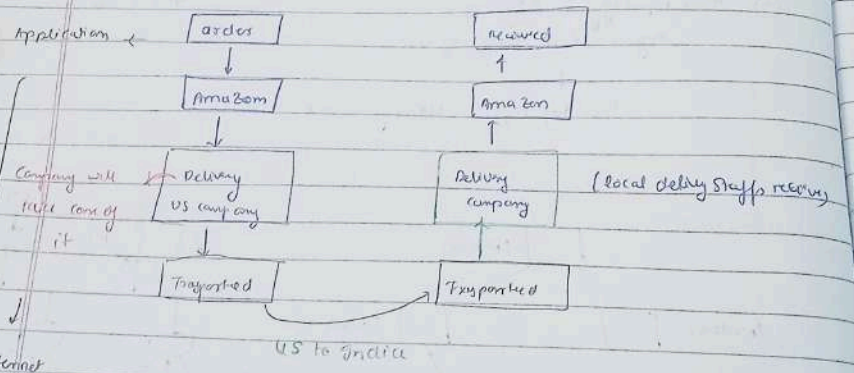


(Mesh)

→ Scalability issues
(if we need to add 1 more node, then difficult to add)

* Structure of a network :-

Lets take an example of ordering a product from Amazon



In same way, we can think of a particular network

DELTA
Date / /
Pg No.

Application → Application layer

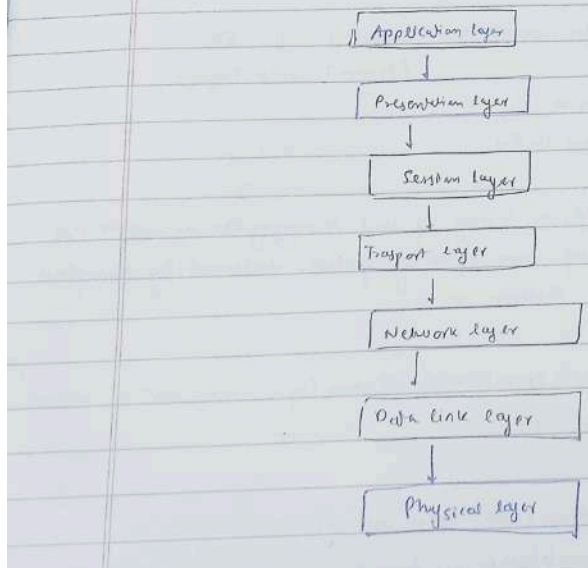
↳ Just like we order something from which we are interacting for eg - WHATSAPP MESSENGER

We don't know the internal functionality of layers at which it is been working

OSI Model (Open Systems interconnection)

* A standard way to express how two or more computers in a particular network try to establish their communication with each other

* We have 7 layers in this model to get a depth knowledge on our internet network



* Every layer will be having different protocols, devices being used

↓

they interact with each other

↳ (work at individual levels)

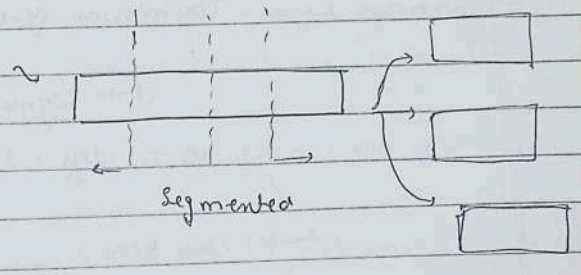
DELTA
Date / /
Pg No.

Session layer assumes that the layers below it will do their work
↓
i.e session layer establishes a session → the trasport layer tries to trasport our data
(assumes its

⇒ Trasport layer - To work with the data ensure that this data reaches to the right person exactly
↳ how data is to be trasported → managed by some protocols
↙ ↘
TCP UDP
(Trasmission control Protocol) (User datagram protocol)

Trasportation of data is done as follows

1. Segmentation - data received from the session layer is divided into smaller chunks known as segments



↓
every segment is given port no's i.e

→ Source and Destination port no's and a Sequence no's
↳
Source → from where we recived our data
Destination → to the place where we want to trasfer our data

* Sequence number
↓
helps to reassemble the segments in correct order
↳ done because not all the data is sent serially but in chunks → ∴ to maintain an order

DELTA
Date / /
Pg No

→ Protocols - HTTPS, FTP etc.

* Application layer - Implemented in a software, users will interact with their application respectively (send messages, files, emails) etc.

↳ Data from application layer is sent to presentation layer

↓

* Presentation layer - Data that it receives is in form of words, characters etc. This layer converts our accepted data into machine based (representable) binary format. ↳ Also known as Translation

from ASCII ⟶ EBCDIC

↳ This layer encrypts our binary data, encodes it into a particular form

↳ abstraction is done and compression so that it is easier to send data further → Data Compression

↓

Protocols for encryption, encoding stuff = SSL
(secured Socket Layer)

* Compression can be
Lossy and Lossless

* Session layer - It helps in setting up and managing the connections and enables the sending and receiving of data followed by terminated of connected sessions

↳ Before a session is established, Session layer does authentication (like username / password)

↓

→ Authentication is done after that ↓

Whether we have the permission to access files or not etc.

* Flow Control - Transport layer controls the amount of data to flow across the network
(For eg: Sender is having 40mbps and receiver has 2mbps)
↳ to regulate this flow control is used

* Error Control - Some data packets might get lost or corrupted data get in between
↳ uses a checksum added to every data segment

TCP - Connection-oriented buffer (when receiver receives data, acknowledgement is sent)

UDP - Connectionless buffer

UDP - Its faster cause it does not have any sense of feedback, due to which some data packets might get lost

=> Network layer - Transmission of data received from transport layer
↳ data segments from one computer to another located in a different network

(Router lives here) → IP addressing done in this layer also called as logical addressing

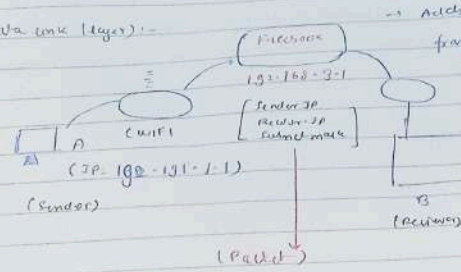
Sender → Receiver

Source IP add.	Main Data	Destination IP add.

↳ Routing helps to send the data to destination IP address
↳ finding the best path to send the data packets via various routing protocols and Algorithms (is done in this layer)

+ Load balancing also done in network layer to make sure it is not overloaded

+ Data link (layer):-



→ Adds MAC address to a frame and then passes to access for the next layer

Data link layer decides which application do we need to send data to

(Physical addressing done in this not the Logical addressing)

↳ MAC Addresses assigned to form a frame

↓

data unit of a data link layer

↙

It is a 12 digit Alpha-numeric

Number of network interface of computer

Computer may not have a single MAC address

↳ 1. Computer's bluetooth - may have different MAC address

2. Computer's wifi - may have different MAC address

+ Allows the upper link layers to access the frames stuff

+ Controls how the datas from & and received by the media Using Media access Control

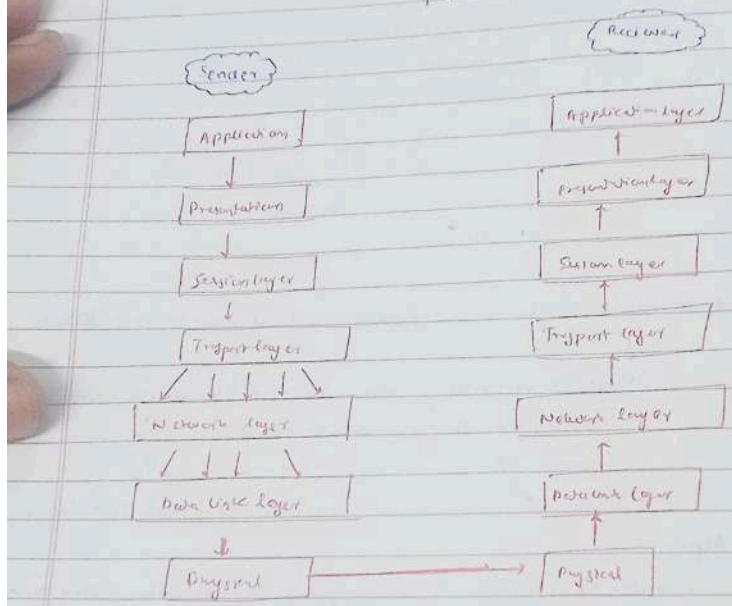
(used to get media from on and off the stuff)

DELTA
Date / /
Pg No

1 Physical layer - Consists of hardware components and cables stuff to transmit the data bits to transfer it to local media via cables across the network

↓

Electrical signal, light signal etc over an optical fibre cable / radio signal in case of wifi



(OSI model)

→ TCP- IP model (another model in computer networks)

↳ internet protocol suite → developed by ARPA

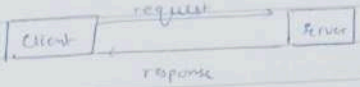
↳ similar to OSI model almost with less no of layers

TCP/IP model - having 5 layers

(i) Application layer
(ii) Transport layer
(iii) network layer
(iv) Data link layer
(v) Physical layer

① Application layer:-
Layer where users interact with having applications like web browsers
↳ or simply what app, FB etc
this applications live on our devices

Set of rules and regulations ↔ it also has some protocols to interact with user
(based on Client - Server architecture)



Server - it is like a system to control our website

(i application layer has these 2 parts (also known as processes)
→ communicating each other

* Collection of Servers in a data company - data centres
↓
Collection of large amount of computers having either static IP address (that do not change)

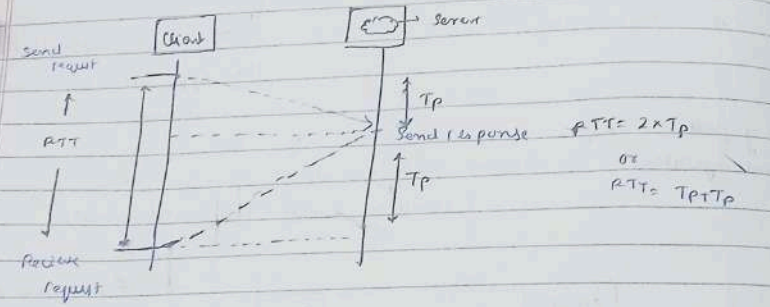
ping time in computer network - time it takes for a small data set to be transmitted from our device to a server on internet and back to our device again → measured in milliseconds (ms)
↳ Same we can write for Round trip time (RTT)

DELTA
Date / /
Pg No

↳ we cannot change ping time because it is already working at its best

RTT → duration in milliseconds it takes for a network request to go from a starting point to a destination and back again to starting point

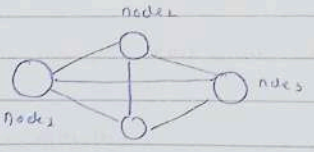
↳ important metric to check health of our connection on a local n/w or larger internet → check speed and reliability



→ Peer to peer (P2P) architecture –

Applications on various devices get communicated each other → no large server or data center etc.

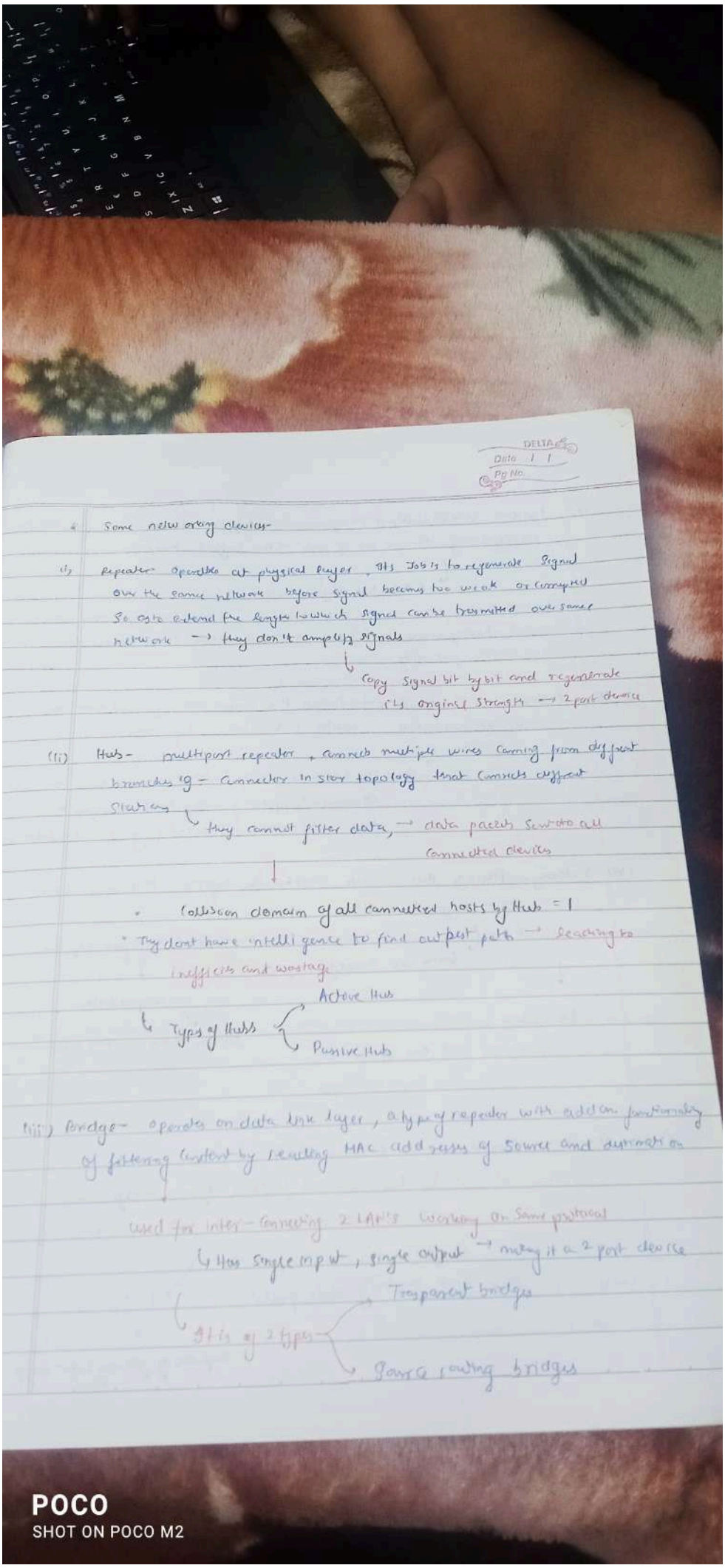
eg- Bit torrent

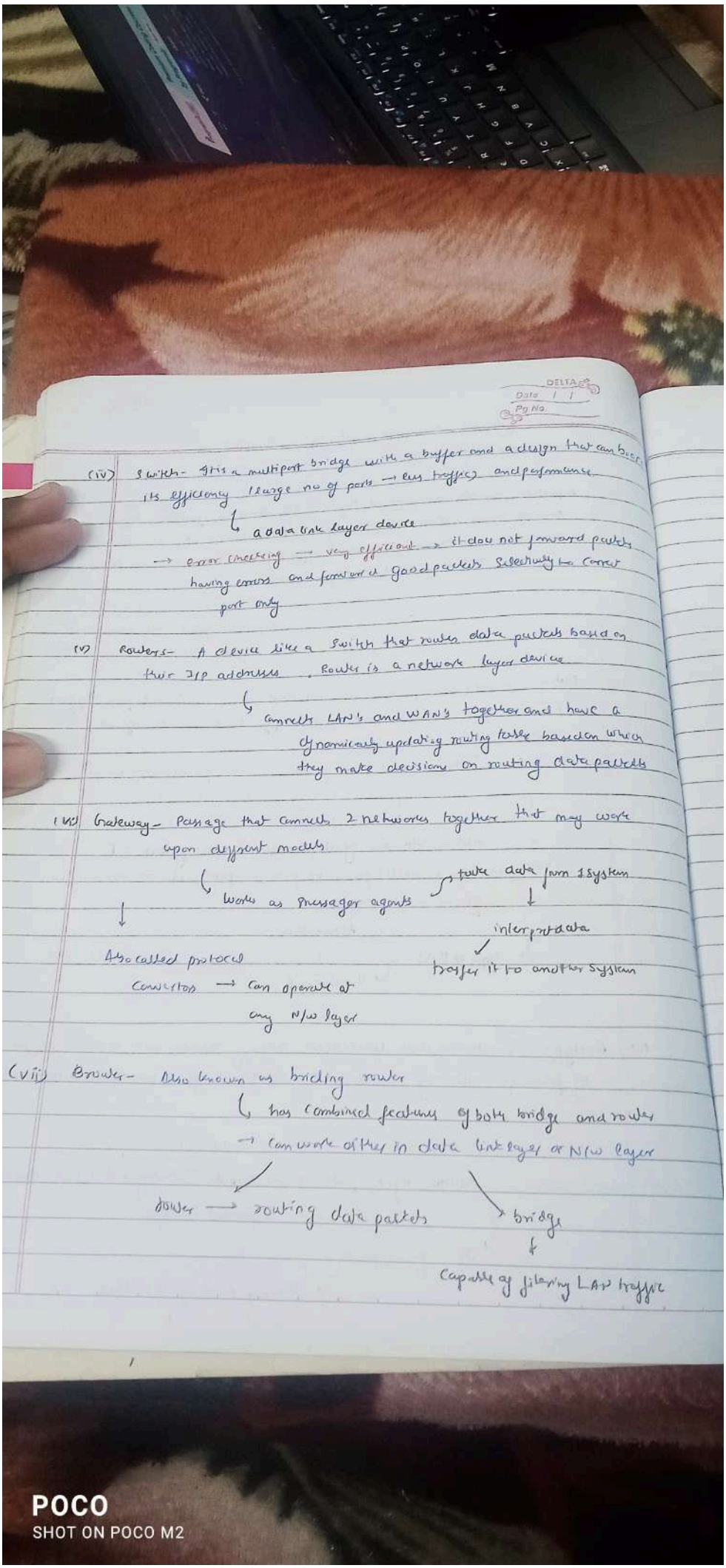


(Peer to peer)

→ Scalability can be done rapidly and it is a decentralized network

→ Every single node (or client) in P2P network can be treated as a client or server respectively





(iv) Switch - It is a multiport bridge with a buffer and a design that can boost its efficiency (large no of ports → less traffic) and performance.

↳ a data link layer device

→ error checking → very efficient → it does not forward packets having errors and forwards good packets selectively to correct port only

(v) Routers - A device like a switch that routes data packets based on their IP addresses. Router is a network layer device

↳ connects LAN's and WAN's together and have a dynamically updating routing table based on which they make decisions on routing data packets

(vi) Gateway - Passage that connects 2 networks together that may work upon different models

↳ works as messenger agents → take data from 1 system → interpret data → forward it to another system

Also called protocol converters → can operate at any N/w layer

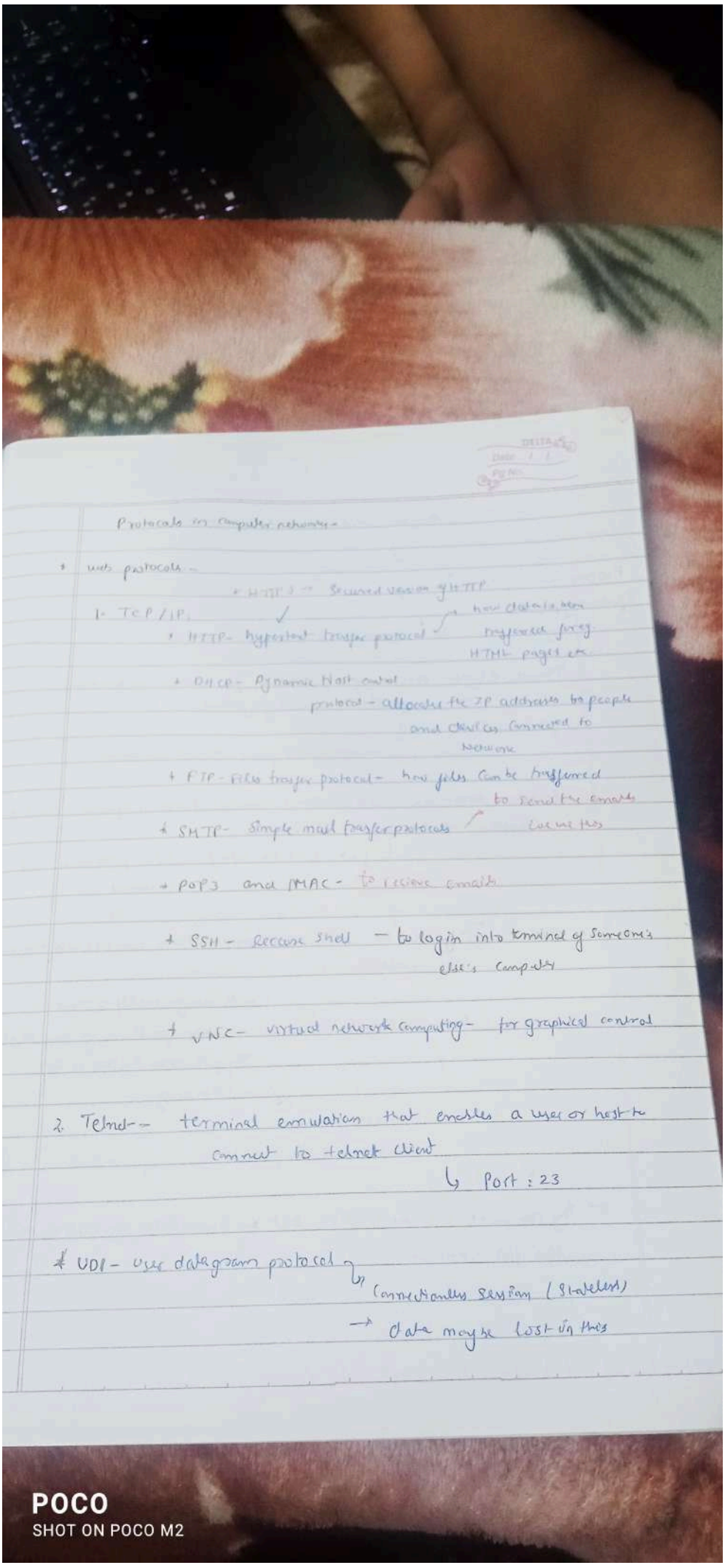
(vii) Bridge - Also known as bridging router

↳ has combined features of both bridge and router

→ can work either in data link layer or N/w layer

router → routing data packets

bridge → capable of filtering LAN traffic



Protocols in computer networks -

* web protocols -

* HTTPS → Secured version of HTTP

1. TCP/IP. ✓

* HTTP- hypertext transfer protocol → how data is sent transferred from HTML pages etc

* DHCP- Dynamic Host control protocol - allocates the IP addresses to people and devices connected to network

* FTP- File transfer protocol- how files can be transferred to send the emails

* SMTP- Simple mail transfer protocols

* POP3 and IMAC- to receive emails

* SSH- Secure shell - to login into terminal of someone's else's computer

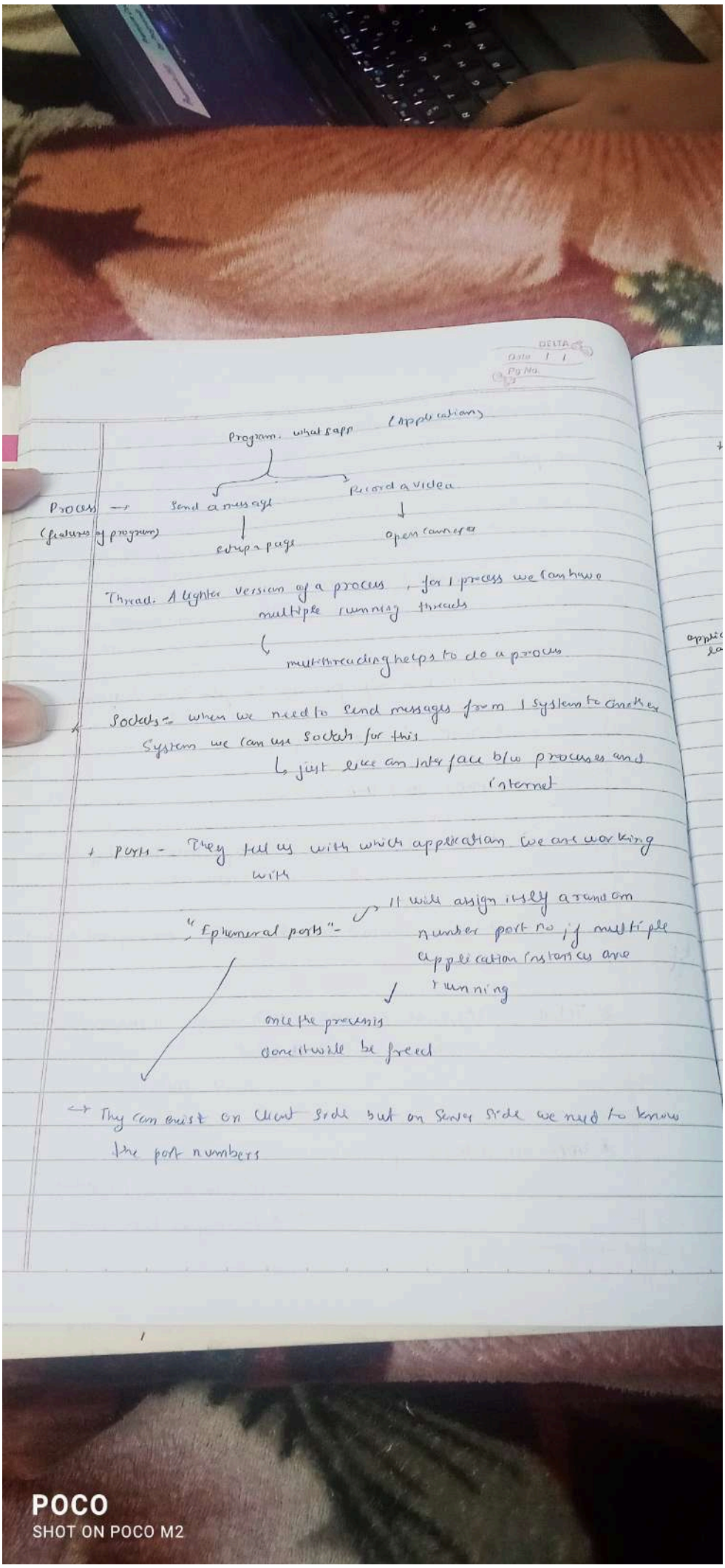
* VNC- virtual network computing- for graphical control

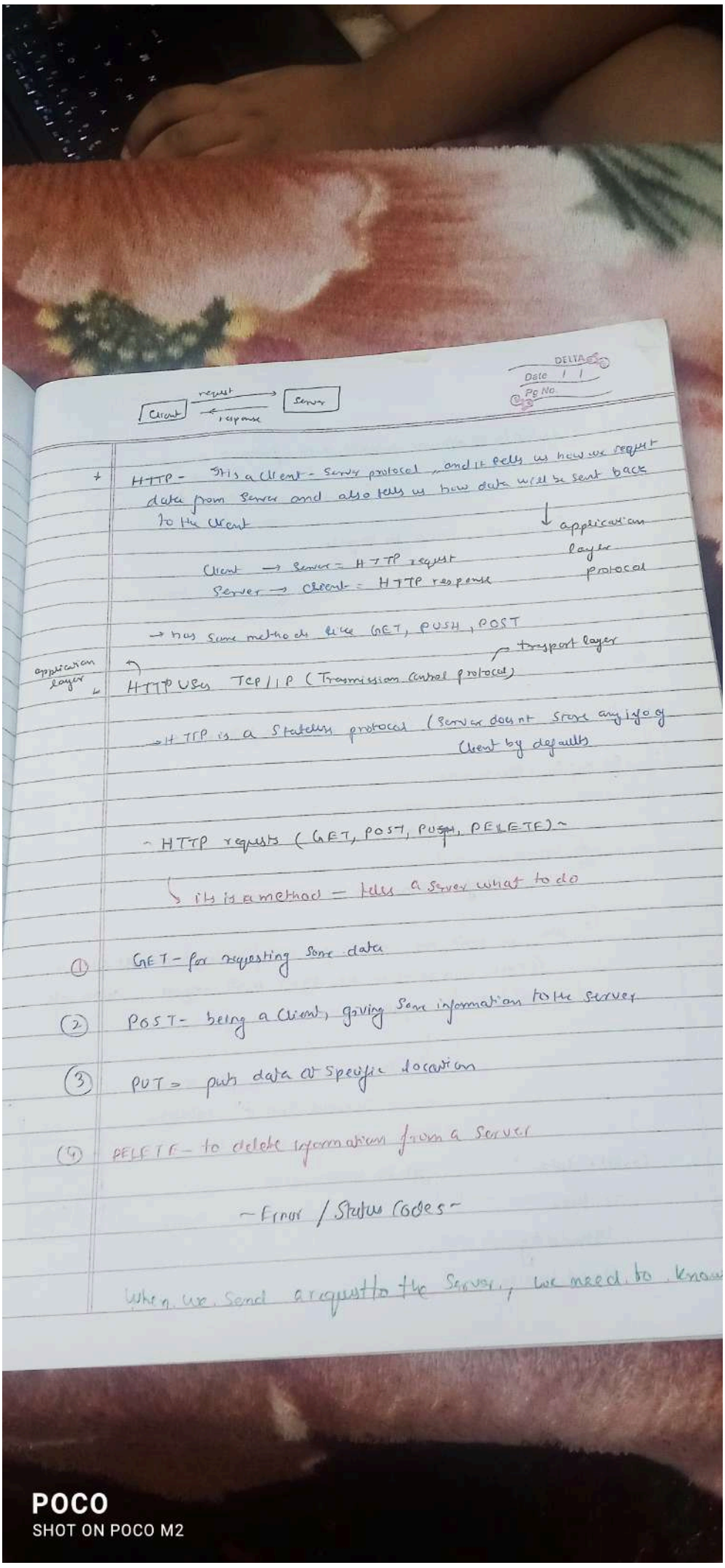
2. Telnet- terminal emulation that enables a user or host to connect to telnet client

↳ Port : 23

* UDP- User datagram protocol → Connectionless session (Stateless)

→ data may be lost in this





* HTTP - It is a Client - Server protocol and it tells us how we request data from server and also tells us how data will be sent back to the Client

↓ application layer protocol

Client → Server = HTTP request
Server → Client = HTTP response

→ has some methods like GET, PUSH, POST

application layer → HTTP uses TCP/IP (Transmission Control protocol) → transport layer

→ HTTP is a Stateless protocol (Server doesn't store any info of Client by default)

- HTTP requests (GET, POST, PUSH, PELETE) -

↳ it is a method = tells a server what to do

① GET - for requesting some data

② POST - being a Client, giving some information to the server

③ PUT = puts data at specific location

④ PELETE - to delete information from a Server

- Error / Status Codes -

When we send a request to the Server, we need to know

if failed to reach as required successfully

for that, we have status / error codes

Status codes classes (for ranges) -

1. For 1xx (100 types) — Information category codes (related)
2. 2xx → Success codes
3. 3xx → for redirecting purposes
4. 4xx → Client - error (like we pressed a false url etc)
5. 5xx → Server error

~ Cookies ~

↳ Unique String, stored in user's browser

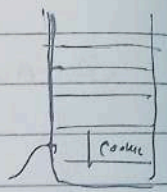
↳ When we visit our website for the first time, it will setup a cookie and whenever we again make request, cookie will be sent to request's header

↳ Server will look for that cookie, searches it in its database

↓

Gets to know who is contacting it

Cookie's data is been reserved by Server



Third party cookies - Cookies that are set for the URL that we don't visit

How email works -

{ for application layer protocol }

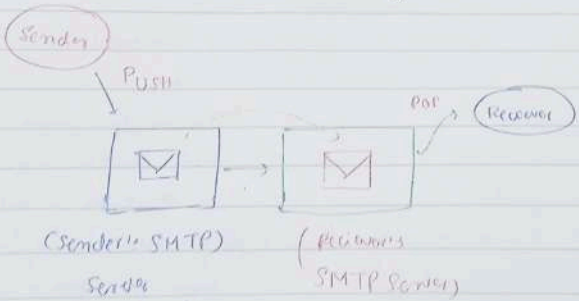
for sending mails - SMTP (simple mail transfer protocol) to people

for receiving mails - POP3 from other users

→ After the application layer, we want our gmail to be transported so what transport layer protocol are going to use → ? (TCP/IP or UDP)

↓

TCP because we cannot afford any loss of data

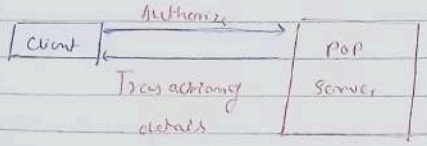


but if we have same account logged in both sender and reciever, no connection established

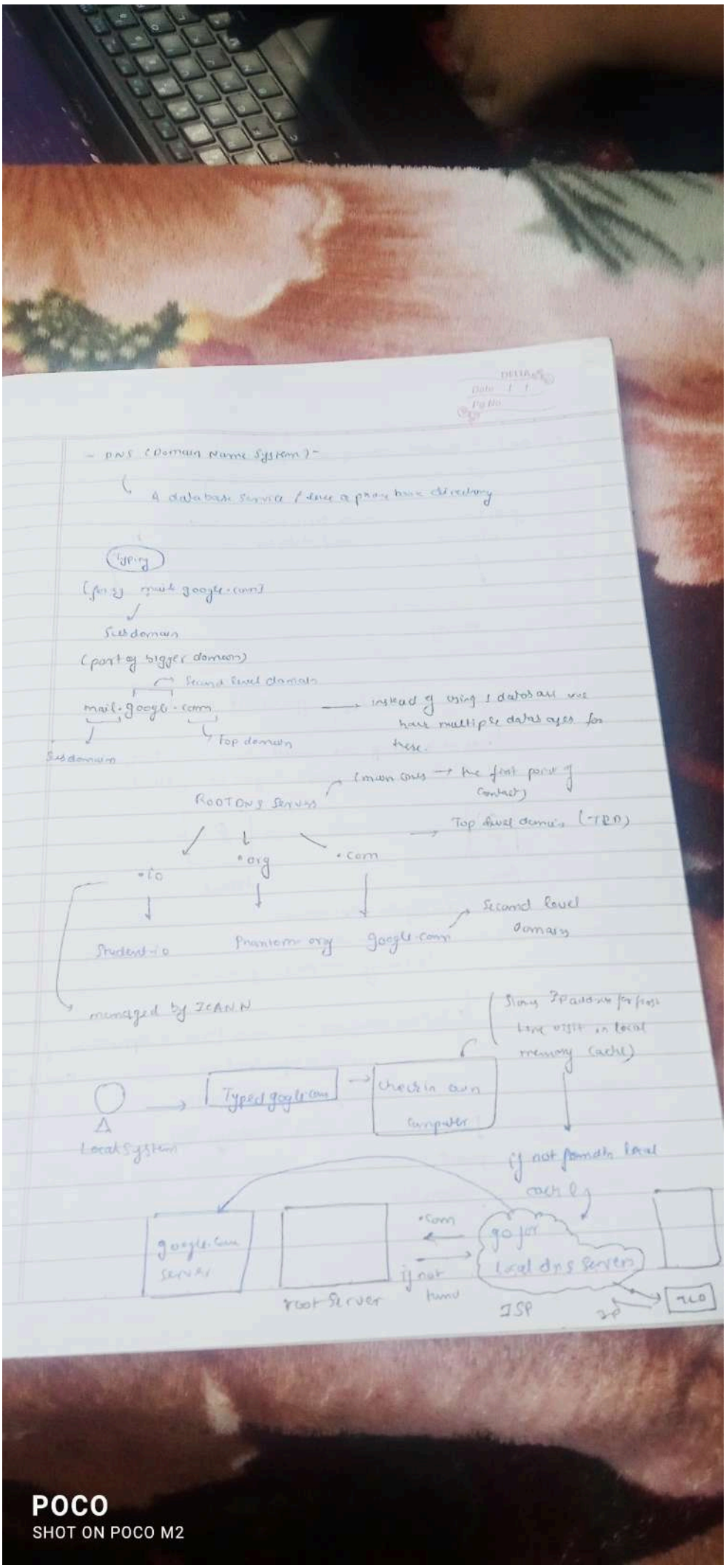
DELTA
Date / /
Pg No.

* How downloading of emails work -

1) POP - post office protocol
   ↳ client connects to the POP server by port no
   ↳ doing all the authorization
   ↓
   then client asks the server to provide all the Emails (that are unread)



2) IMAP - also used to get the mails (internet message access protocol)
   ↓
   mails can be accessed simultaneously on multiple devices
   ↓
   ↳ mails → having local copies
   ↓
   these copies any changes made locally changes in the devices accessed
   (for eg deleting from one device leads to delete of mails from other devices too)



* Congestion occurs when capacity of storage < incoming data at any
→ all the whole segments get lost and leads to ↓ in overall quality

* Congestion control Algorithms are built in TCP

→ Now how to know whether the data is been transmitted in a sequential order or not → we maintain a checksum for the same (to ensure that is not corruption of data)

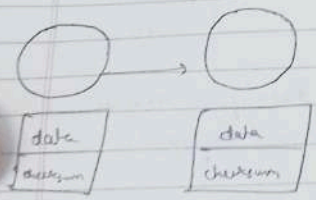
↓
the calculated checksum according to the data is sent from sender side to receiver side

(checksum to check data is corrupted or not)

↳ reciever would already have calculated checksum

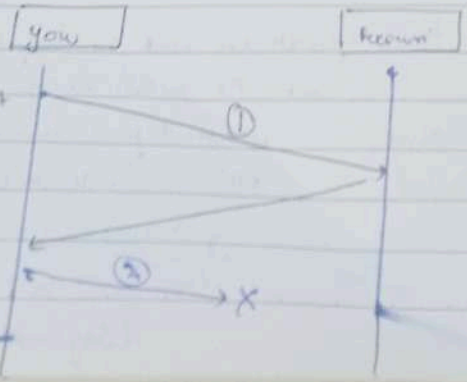
if checksum (sender) != checksum (reciever)

↳ some kind of error is there
else we have successfully transmitted data in ordered manner



* Now how can we know whether the data is successfully sent to reciever side without any corruption in between

↳ we use Timers



→ as soon as we transmit the packets → our timer will start

↓

again reads the receiver

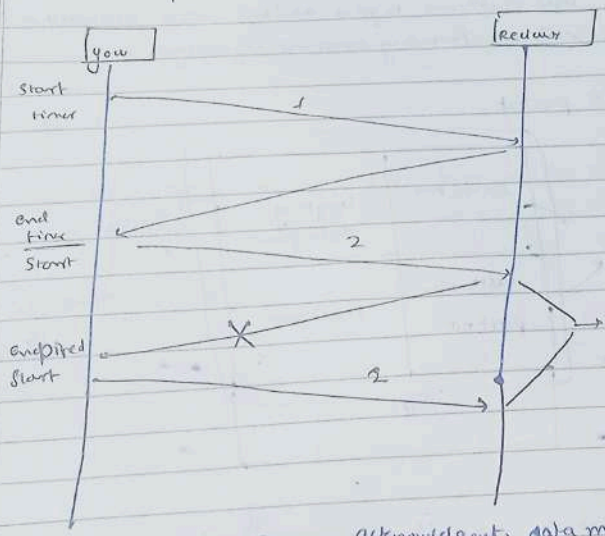
When the sender side receives our acknowledgment → sends back an acknowledgment

↓

it stops the timer

(sender doesn't get acknowledgment)

By chance, while transmitting the packets if receiver doesn't receive, timer expires and retransmission done based on new timer



Sometimes during sending acknowledgment, data may get lost, due to which retransmission is done with new timer → leading to duplicate packets at receiver side

↳ To avoid this every packets given a unique sequence no to distinguish our incoming packets

~TCP- transmission control protocol~

* A Report layer protocol * Reliable

* Application layer sends lots of raw data

↓

TCP segmentizes the data → divides in chunks, adds headers and check sums

↓

It may also collect the data from network layer (divided into smaller chunks)

* Connection based protocol providing Congestion Control
  ↳ oriented

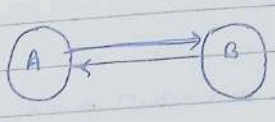
* Takes care of :-

- when data does not arrive (retransmission can be when no acknowledge)
- maintains the order of data (using sequence no)

→ emails, files transfer etc.

* Error control provided, Congestion control

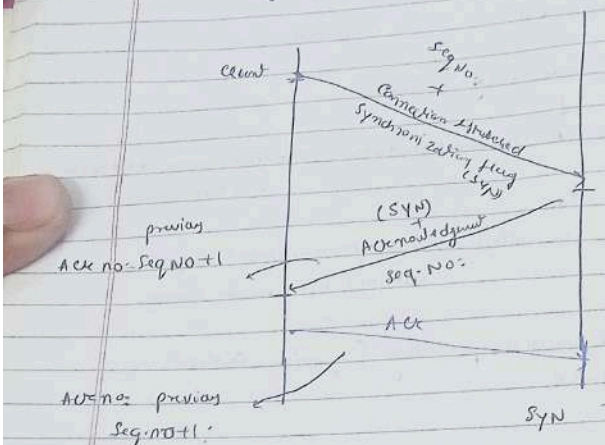
* Full duplex (Simultaneous transmission of files is possible)



(only 2 End points possible)

User A User B

* 3way handshaking



* Dummy connection establishment, Synchronization flag (SYN) along with a sequence no is been sent

* Acknowledgment flag sent along with a mathematical calculation on previous sequence no to generate new seq. no

→ Acknowledgement again sent to Server and connection is established

* In transport layer - we worked with segments

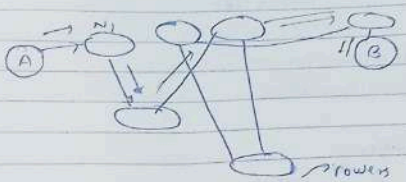
* In Network layer - we worked with packets

* In data link layer - we worked with frames

DELTA
Date / /
Pg No.

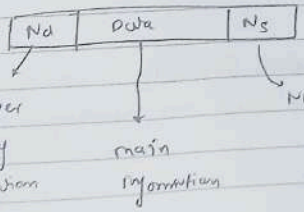
Network layer -

↳ we create with Routers in this case



* Every single router has its own router (n/w) address

→ Data packets are sent from source address to routers where they would check in their routing tables and verify our destination part address / forwarding table



forwarding table would send the packet to the nearest router, this is called as Hop-by-Hop forwarding

↓

done till we reach the correct router

Routers have both

↳ forwarding and routing tables

DELTA
Date / /
Pg.No

I/P address → 192-168-2-30

192-168-2 ↓ network address (IP) ↓ tells in which network our device resides in

30 → device address

* Who creates these (routing) tables? → control plane
↓
just like a graph data structure

* Routers → nodes, edges ← links b/w routers

① Static routing
* adding addresses manually
* Non-adaptive and time consuming

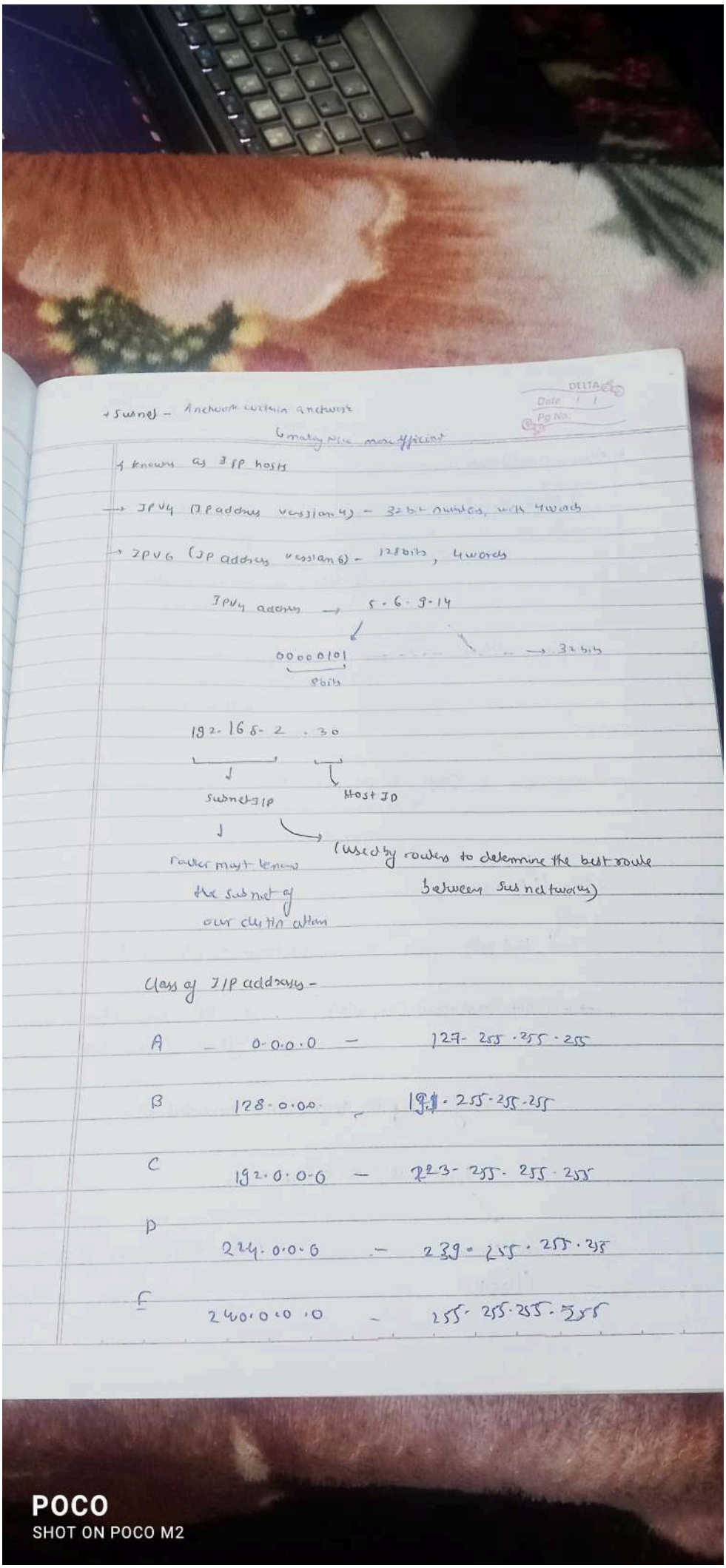
② Dynamic routing

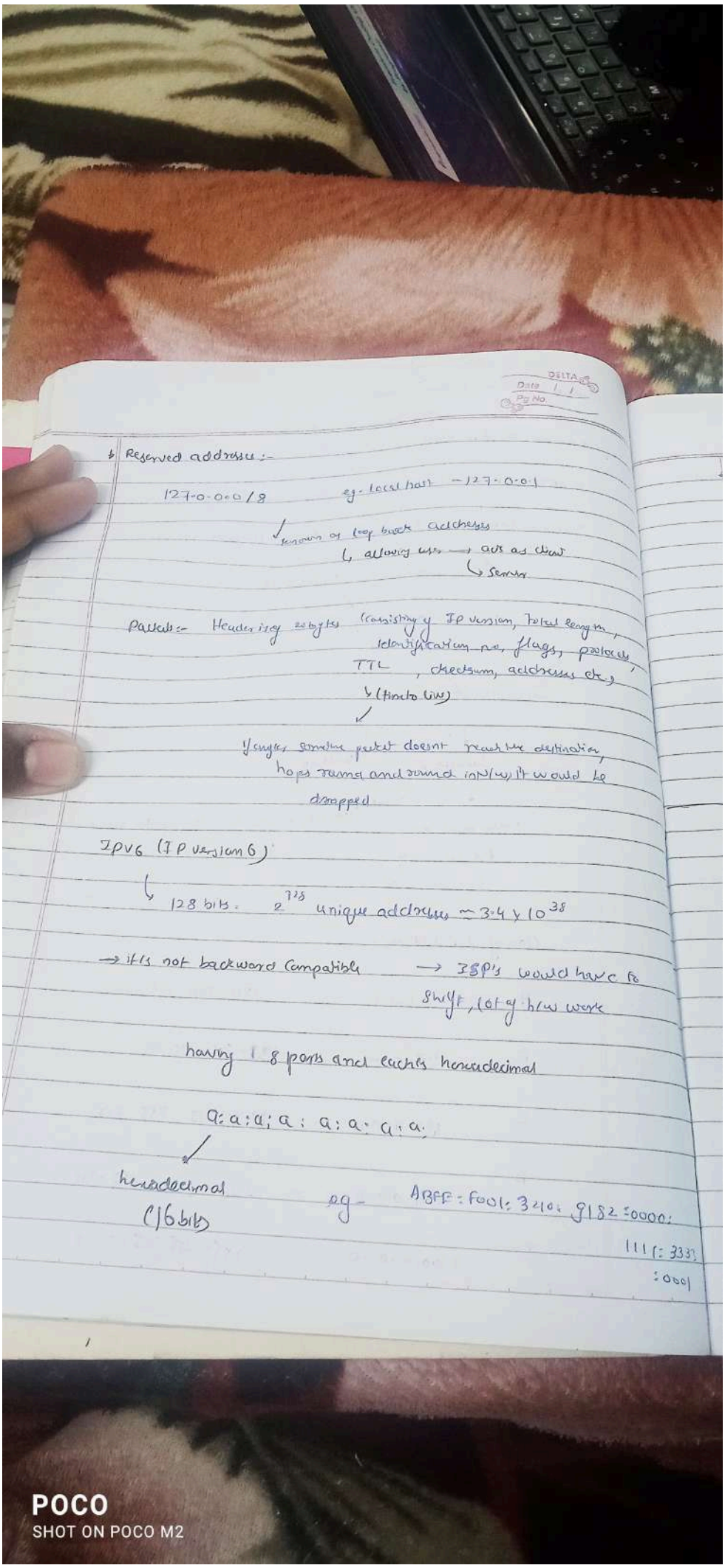
* whenever there is change in N/w like topology change etc, it adapts itself and addresses are assigned to it accordingly

* uses of Algorithms like → Djikstra, Bellman-ford to find shortest paths among routers

~ Internet protocol (IP) ~

protocol present in N/w layer that controls routing (Static / Dynamic) and the control panels along with I/P address





DELTA
Date / /
Pg No.

b) Reserved address :-

127.0.0.0/8 eg. local host - 127.0.0.1

↓ known as loop back addresses

↳ allowing us → act as client
↳ server

Packet :- Header is of 20 bytes (consisting of IP version, total length, identification no, flags, protocol, TTL, checksum, addresses etc.)

↳ (time to live)

↓

If suppose sender packet doesn't reach the destination, hops turned and turned in N/w, it would be dropped

IPv6 (IP version 6)

↳ 128 bits = 2^{128} unique addresses $\approx 3.4 \times 10^{38}$

→ it is not backward compatible → ISP's would have to shift, lot of h/w work

having 8 ports and caches hexadecimal

a:a:a:a:a:a:a:a

↓

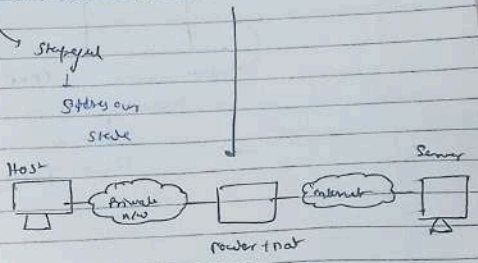
hexadecimal

(16 bits)

eg - ABFF:F001:3210:9182:0000:1111:3333:0001

modify, accept, reject packets etc

- middleboxes – extra devices that also interact with IP packets
  (can be found in N/w as well as transport layer)
  - eg – Firewall, Network address translation (NAT)
    - stateless → not maintaining state
    - stateful → storing our state

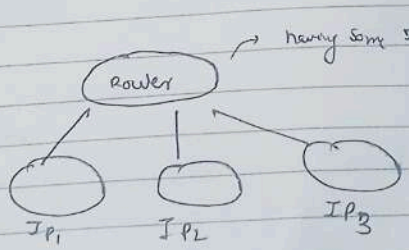


(modification of IP address)

- Data link layer –

Data packets that we receive from n/w layer → we need to send it over a physical link

↳ Send data b/w connected devices



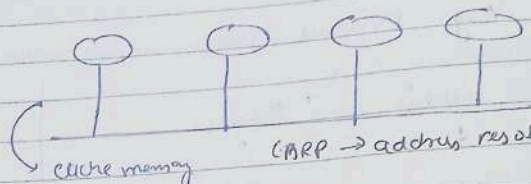
new devices → get some IP address → from DHCP Server

[having pool of addresses]

DELTA
Date / /
Pg No.

(MAC addressing device)

many devices may be connected in LAN, the devices will communicate and share data based on Data link address (layer2)



(ARP → address resolution protocol)

cache memory to check

↓

Revise properly (not much required but see)

~User datagram protocol~

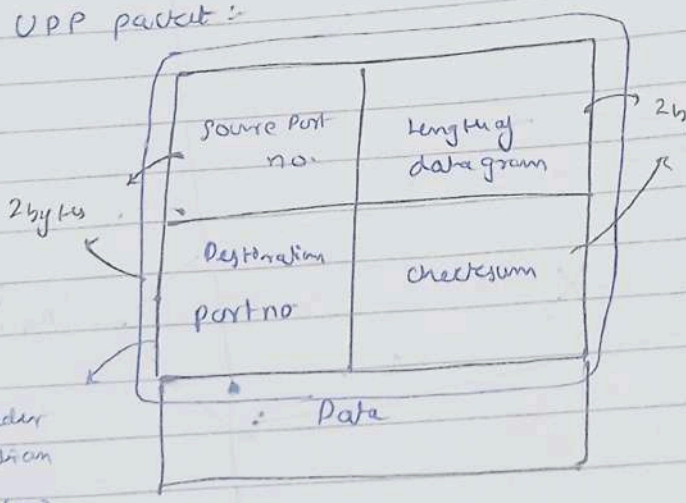
↳ Connectionless protocol in transport layer used to transmit data from N/w layer to Application layer and from Application layer to n/w layer.

* Data may or may not be delivered (not reliable)
* Data may change * Data may not be in order

* It is faster than TCP as it does not need the acknowledgement from receiver every time we send data

UDP uses checksum to find whether data is corrupted or not but will not take any measures regarding it.

UDP packet :-



∴ Data = 2^{16} - 8 = 65536 bytes
(Size of data that can be sent in 1 packet).

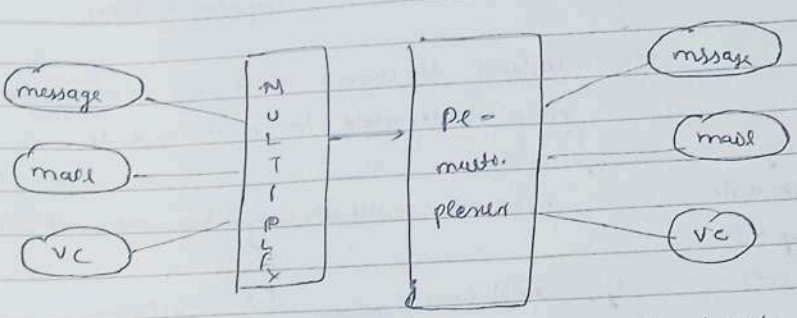
* In Video Conferencing and game Streaming it can be used

↳ DNS uses UDP due to its fast capability

DELTA
Date / /
Pg No.

* TCP protocol - 100% efficient protocol to send the messages over different applications in application layer from N/w layer

- Connection oriented and reliable



(Sender's side)

(reciever's side)

(multiplexer → recieves all the data from various services on system to a single entity

(demultiplexer → unwraps the content and parts to various applications as per requirements

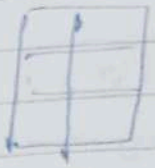
* Data recieved from various applications are in form of packets → these packets are sent to transport layer with the sockets having port no's attached from Sender's and reciever's side due to which they transmit to another application layer and out (Segments)

* Transport layer takes care of congestion control

DELTA
Date / /
Pg No

* DNS → Domain name of our local system → Sent to DNS Server

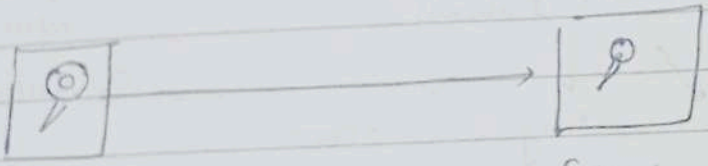
gives back IP address ← checks first in its local database



connects to our visiting server

~ Transport layer of TCP/IP model ~

↳ It is a part of our model where we transport our data from the network layer to application layer



(data from one computer to another

↳ routing → done by N/w layer)

↳ but within our system N/w to application → done by Transport layer

(which application of system do we need to send message to)

provides abstraction