14/12/2025

L-4 TCP/IP Protocol Stack

Concepts covered   ①

- TCP/IP protocol stack
- Basic functions of TCP/UDP & IP
- Data Encapsulation

Introduction

- TCP/IP is the most fundamental protocol used in the internet
  - allows Computer to Communicate / share resources
  - used as a Standard
  - The bridge the gap between non Compatible platform ↕

  Various other devices

  Various other platforms like mac, windows, linux, ATMs. etc.

- Work on TCP/IP Started in 1970s.

  - funded by US military

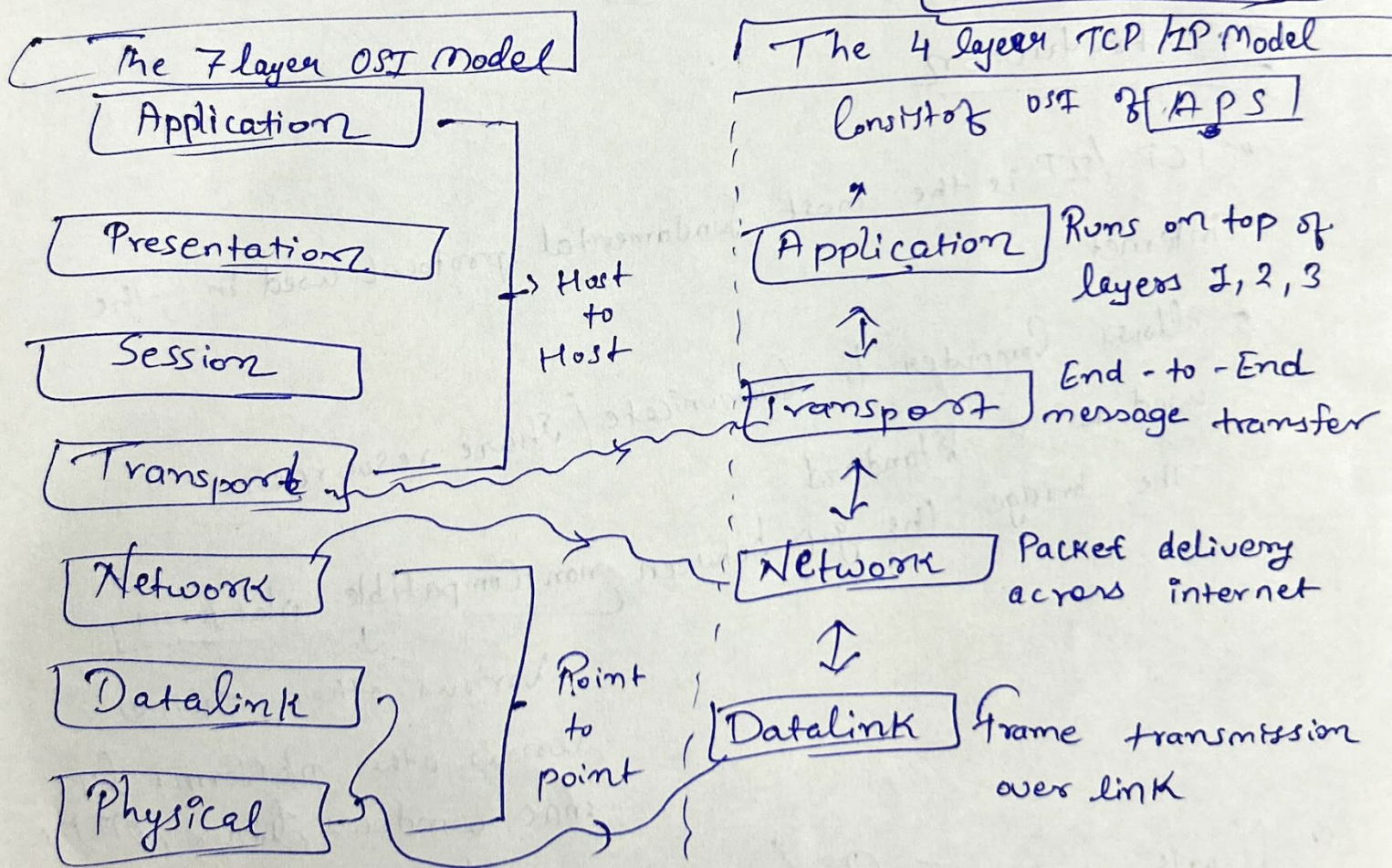 - Advanced Research Project Agency.

Network layer in TCP/IP

- In 1978, International Standard organization (ISO) proposed the 7-layer OSI reference model for network services & protocols.

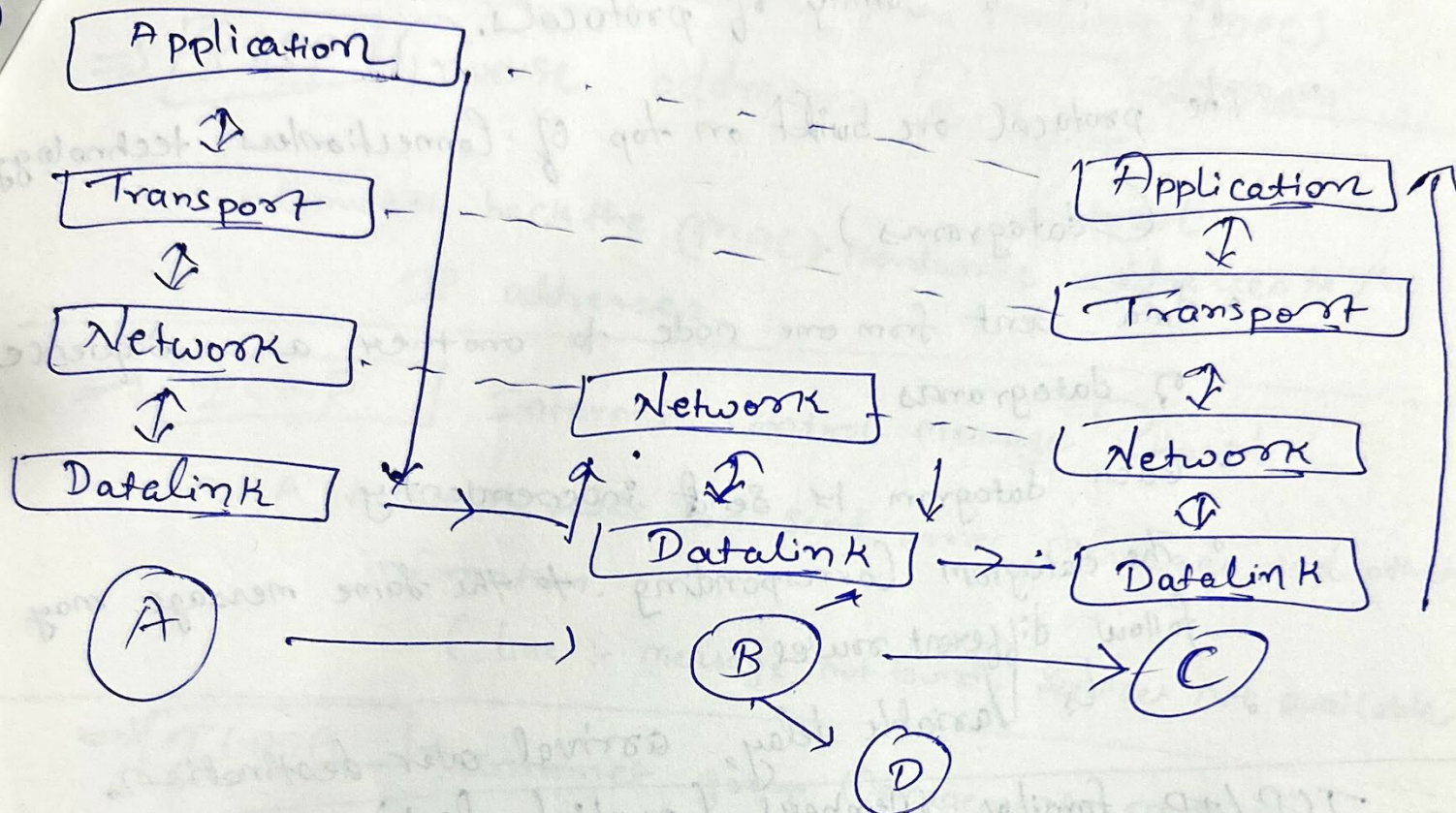- TCP/IP does not strictly follow the OSI model

- It follows the simple, 4-layer model.

The 7 layer OSI Model	The 4 layer TCP/IP Model
Application	Consists of OSI 3 A P S
Presentation	Application — Runs on top of layers 1, 2, 3
Session	
Transport	Transport — End-to-End message transfer
Network	Network — Packet delivery across internet
Datalink	Datalink — frame transmission over link
Physical	

Host to Host

Point to point

Data flow in 4-layer Model. ③

Application ─────┐
   ↕   │
Transport ─   │
   ↕   │
Network ──   │
   ↕   ↓
Datalink ───────→ Network
                   ↕
(A) ──────────→ Datalink ──→
                (B)

Application ↗
   ↕
Transport
   ↕
Network
   ↕
Datalink ←─ · · ─

(C)

(D)

↳ more data link points
here so it selects which
route is better or is this
point or node valid for
sending message if not it will
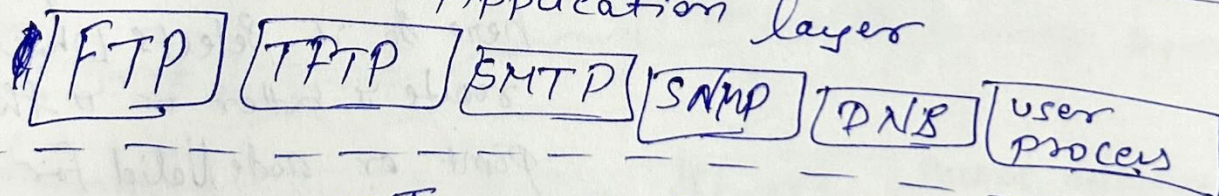check next point/node.
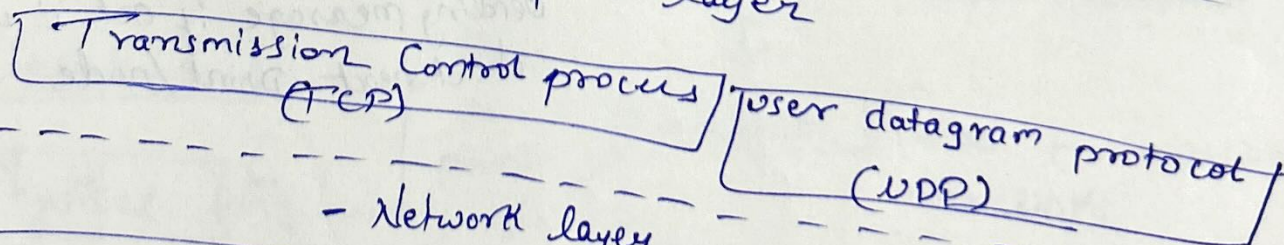
TCP/IP protocol Suites

- refers to a family of protocols.
- The protocol are built on top of Connectionless technology (datagrams).
  - data sent from one node to another as a sequence of datagrams.
  - Each datagram is send independently.
  - The datagram Corresponding to the same message may follow different routes
    - ↳ Variable delay, arrival over destination.
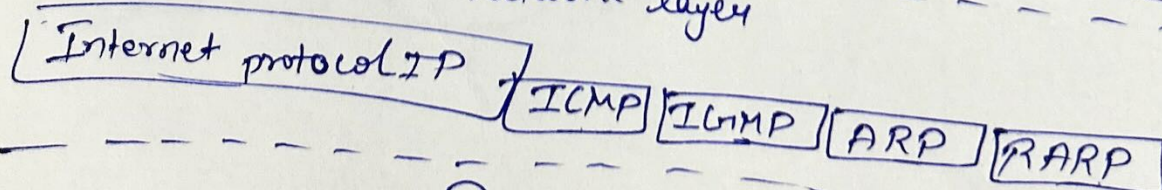
TCP/IP family Members (partial list)

——— Application layer

| FTP | | TFTP | | SMTP | | SNMP | | DNS | | user process |

— Transport layer

| Transmission Control process (TCP) | | user datagram protocol (UDP) |

— Network layer

| Internet protocol IP | | ICMP | | IGMP | | ARP | | RARP |

— Datalink layer

| Datalink & Hardware layer (eg :- Ethernet) |
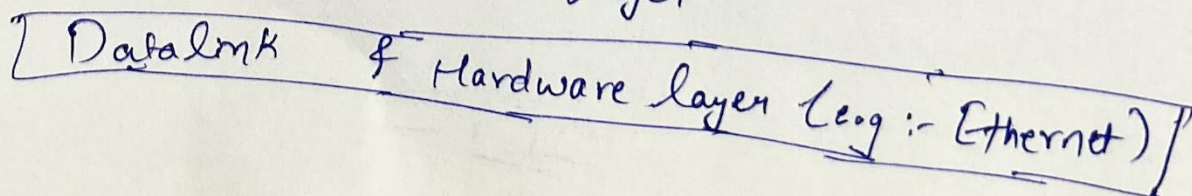
⇒ **ARP** → Address Resolution Protocol

    ↳ Converts the IP address to the Hardware (MAC) addresses.

⇒ **RARP** Reverse addresses Resolution Protocol

- Converts back the (MAC) Hardware addresses to the IP addresses.

⇒ **ICMP** Internet Control message Protocol

- A network device can send error messages and other information.

      (Like : message not found, Service not available)

⇒ **IGMP** → Internet group Management protocol.

- A node Can Send its <u>multicast group membership</u> to adjacent routers.

    (whenever it want to Send data to selected groups it will Send signal to nearest routers to get a selected data. or where to get that data from.)

Typical Scenario

| user process | | user process |

TCP ↕

UDP ↕

IP ↕

Data link & Hardware layer
(eg. Ethernet)
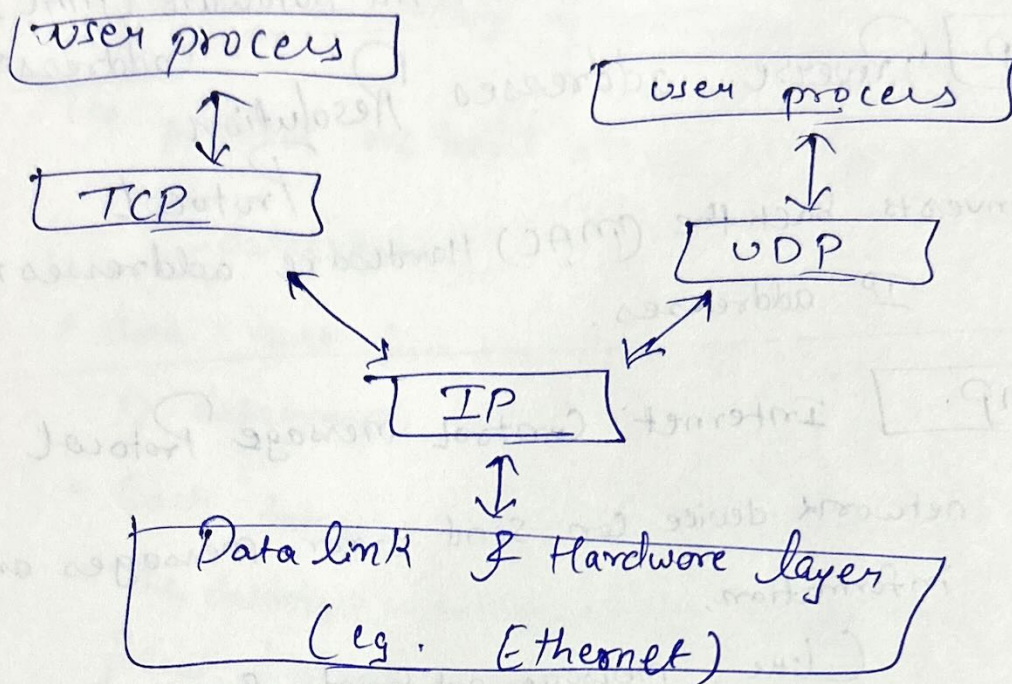
What does IP do?

- IP transports datagram (packets) from a Source node to destination node.
- Responsible for routing the packets
  → Breaks a packet into smaller packets, if whenever the bigger data of packet is received required
- Unreliable Service
  ↳ A packet may be lost in transit
  ↳ packets may arrive out of order
  ↳ Duplicate packets may be generated.

[what does the TCP do?]

- TCP provides a connection-oriented, reliable service for sending messages,
  - Split a message into packets.
  - Reassemble packets at destination
  - Resend packets that were lost in transit.

- Interface with IP
  - each packet forwarded to IP for delivery
  - Error Control is done by TCP.

[what does UDP do?]

- UDP provides a [connectionless, unreliable] service for sending datagrams (packets),
  - Message small enough to fit in a packets (e.g. DNS query).
  - Similar (and faster) than TCP.

  - Never split data into smaller pieces,

  - Does not care about error control
    if there is an error let it be.

- Interface with IP.
  - each UDP packet sent to IP for delivery.

Addresses in TCP/IP

```
┌─────────────────────┐              ┌─────────────────────┐
│ user     process    │              │  user  process      │
└─────────────────────┘              └─────────────────────┘
         ↕                                     ↕
┌─────────────────┐                   ┌──────────────┐       Port address
│   TCP           │                   │  TCP         │        (16 bit).
└─────────────────┘                   └──────────────┘
              ↖         ┌────────────┐    ↗
               ↘        │   IP       │       IP address
                        └────────────┘        (32 bits)
                             ↕
┌──────────────────────────────────────────┐  physical Address
│  Data link   & Hardware  layer            │
│          eg :- (Ethernet)                 │      (48 bits)
└──────────────────────────────────────────┘
```
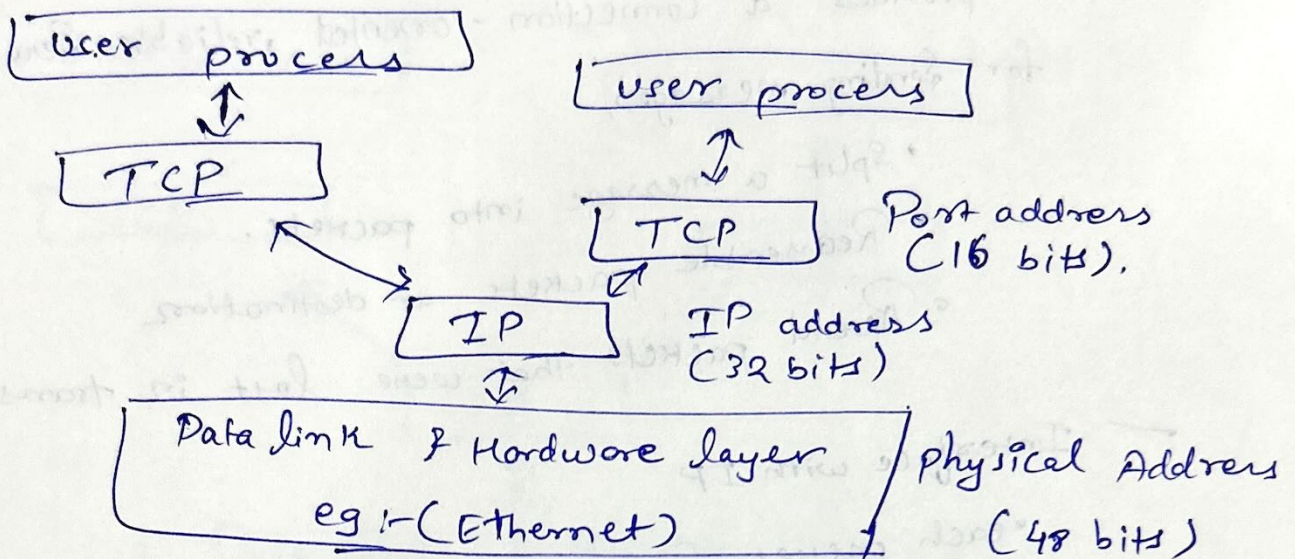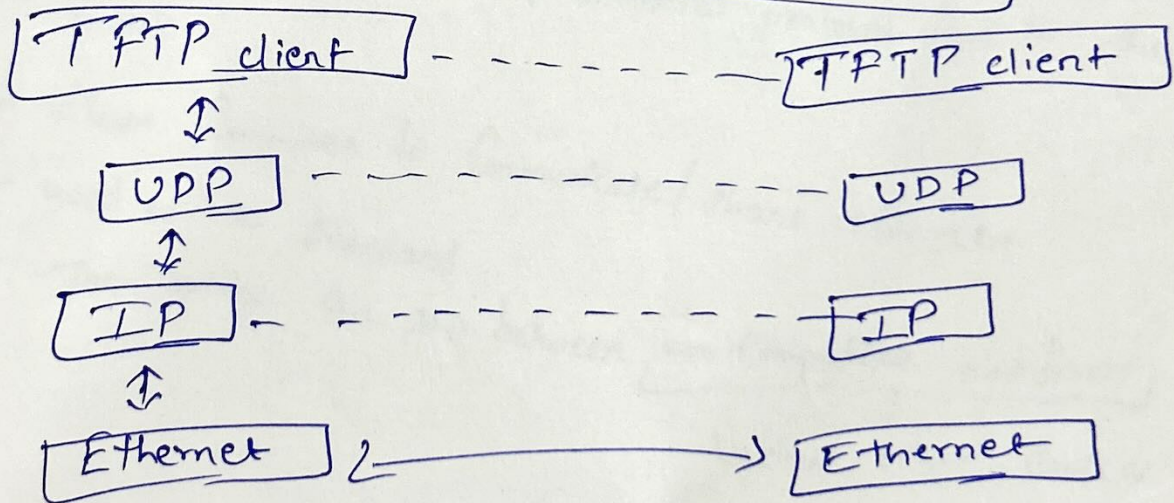
Encapsulation

o Basic Concept

- As data flows down the protocol hierarchy, headers (and trailers) get appended to it

- As data moves up the hierarchy, headers (and trailers) get stripped off.

- As example to illustrate :- [TFTP uses UDP not TCP]

  - Trival file transfer protocol (TFTP).
  - TFTP client transfers 200 bytes of data

  [4 bytes of TFTP header gets added]

[TFTP client] - - - - - - - - - [TFTP client]

↕

[UDP] - - - - - - - - - - - - - [UDP]

↕

[IP] - - - - - - - - - - - - - - [IP]

↕

[Ethernet] ⤶ - - - - - - - - → [Ethernet]

Encapsulation in TFTP.

Sender ↓

[Data] receiver.

[H-TFTP] [Data] TFTP message

[H-UDP] [H-TFTP] [Data] UDP message

[H-IP] [H-UDP] [H-TFTP] [Data] IP message

[H-ETH] [H-IP] [H-UDP] [H-TFTP] [Data] [T-Eth] Ethernet frame

14   20   8   4   200