01 - Protocols

▼ Layering in Communication

- Communication tasks are broken into layers to manage complexity.
- Using a layered model helps in the design of complex, multi-use and multi-vendors networks.
- Individual parts of the systems can be designed independently, still work together seamlessly.
- **Benefits:**
    - Allows **competition and innovation**.
    - Changes in one layer **do not affect others**.
    - **Defined functions** per layer.

Example: Sending a letter

- Write & envelope (high-level)
- Transfer to post office (mid-level)
- Physical delivery (low-level)

▼ Protocols

- **Protocols = rules for communication**
- **Ensure:**
    - Identified sender & receiver
    - Common language/grammar
    - Speed & timing
    - Confirmation & Acknowledgment
- **Protocols define:**
    - Message encoding

    - Formatting/encapsulation
    - Size, timing, delivery options

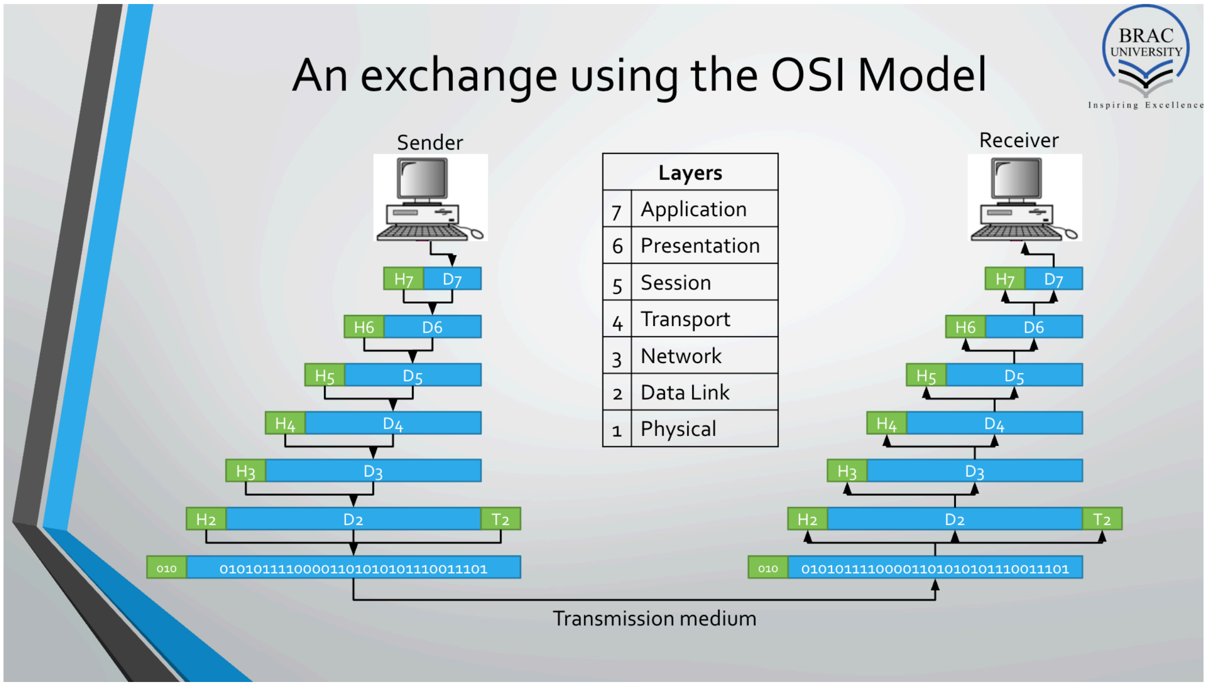
▼ Protocol Suites

- **TCP/IP Model:**
    - Open *de facto* standard, IETF governed
- **OSI Model:**
    - *De jure* standard, theoretical reference model

▼ OSI Model

Serial	Layer	Packet Data Unit (PDU)
7	Application	Data
6	Presentation	Data
5	Session	Data
4	Transport	Segment
3	Network	Packet
2	Data Link	Frame
1	Physical	Bits

- Layer 3 to Layer 7 - Communications between applications
- Layer 1 to Layer 2 - Moving raw data across the network



An exchange using the OSI Model

Layers	
7	Application
6	Presentation
5	Session
4	Transport
3	Network
2	Data Link
1	Physical

Data Link Layer

The **header** is added to provide addressing and control information for frame delivery, while the **trailer** is added for **error detection**. Together, they ensure that data is delivered **accurately and to the correct destination** over the physical network.

Component	Location	Key Contents	Purpose
Header	Beginning	MAC addresses, control info	Identifies sender/receiver, manages flow
Trailer	End	Error-checking info (e.g., CRC)	Detects transmission errors

▼ At the Sender

Data encapsulation happens from Layer 7 down to Layer 1

- **Layer 7 (Application Layer):** Generates **H7** header and **D7** (user data).
- **Layer 6 (Presentation Layer):** Adds **H6** header to **D6(H7+D7)**.
- **Layer 5 (Session Layer):** Adds **H5** header to **D5(H6+D6)**.
- **Layer 4 (Transport Layer):** Adds **H4** header to **D4(H5+D5)**.
- **Layer 3 (Network Layer):** Adds **H3** header to **D3(H4+D4)**.

- **Layer 2 (Data Link Layer):** Adds **H2** header and T**2** trailer to **D2(H3+D3)**.
- **Layer 1 (Physical Layer):** Converts the entire frame into **binary bits (010...)** for transmission over the medium.

This layered encapsulation ensures that each layer only communicates with its peer layer at the receiver.The Physical Layer carries all headers and data as a bitstream during transmission.

▼ At the Receiver

Data decapsulation happens from Layer 1 up to Layer 7.

Layer 1 (Physical Layer): Receives the binary bits.

Layer 2 (Data Link Layer): Reads and removes **H2, T2**.

Layer 3 (Network Layer): Reads and removes **H3**.

Layer 4 (Transport Layer): Reads and removes **H4**.

Layer 5 (Session Layer): Reads and removes **H5**.

Layer 6 (Presentation Layer): Reads and removes **H6**.

Layer 7 (Application Layer): Finally, **D7** (user data) is delivered to the application for user use.

Each layer only processes and removes its own header and then passes the data upward.

▼ Visual Analogy

- At the **sender (post office)**:
    - Your parcel (data) gets multiple **wrappings (headers)** with instructions at each stage (sorting, transport, routing).
- The parcel is **shipped**.
- At the **receiver (post office)**:
    - Each layer removes its corresponding **wrapping** to access the final parcel.
- Finally, **the receiver gets the original content.**

OSI Layers Detailed Functions

7. Application Layer

```
- User Interface
- Provide user services
```

6. Presentation Layer

```
- Translation
Analogy:
You and your friend speak different languages, so before sending the letter, you translate your message into your friend’s language so they can understand it when they receive it.
In networking:
Converts data formats between sender and receiver (e.g., ASCII ↔ EBCDIC, JSON ↔ XML).

- Compression
Analogy:
Your letter is long and bulky, so you fold it tightly or compress it into a smaller envelope to save postage and make it easier to transport.
In networking:
Reduces the size of the data for efficient transmission (e.g., zipping files before sending).

- Encryption
Analogy:
You are sending sensitive information, so you lock your letter in a small safe or use a secret code so that only your friend with the key or code can read it.
In networking:
Encrypts data for security during transmission, ensuring only the intended recipient can read it (e.g., SSL/TLS encryption).

The Presentation Layer is like preparing your letter before sending: you
```

translate it (translation), fold it small (compression), and lock it safely (encryption) for your friend.

5. Session Layer

- Session management
- Dialog control

4. Transport Layer

- Process to process delivery
- Port addressing (16-bit port numbers)
- Segmentation & reassembly
- Sequence numbering to maintain order of sequence
- Connection control (e.g., TCP handshake)
- Flow & error control
- Multiplexing
- PDU: Segment

Segmentation & Reassembly
Analogy: Shipping furniture in parts

- You bought a large wardrobe, but it is too big for delivery in one piece.
- The seller disassembles it into smaller parts (segmentation) to ship easily.
- When it arrives, you reassemble it at home (reassembly) to get your complete wardrobe.

In networking:
The Transport Layer breaks large data into smaller segments for transmission and reassembles them at the receiver.

Sequence Numbering
Analogy: Page numbers in a book shipment

- Your friend mails you a long story split into 5 envelopes.
- Each envelope has a page number on the pages:

So even if the envelopes arrive out of order, you can arrange the pages correctly.

Importance:
- Ensures correct order of data reassembly.
- Detects missing segments if a page (segment) does not arrive.

Connection Control
Analogy: Phone call setup
Before speaking to your friend:
- You call (connection setup).
- You talk (data transfer).
- You hang up (connection termination).

In networking:
Establishing, maintaining, and terminating connections before and after data transfer, ensuring reliable communication.

Flow & Error Control
Analogy: Drinking water from a tap
Flow control: If the water flows too fast, you cannot drink properly. You ask to slow down the flow to a manageable rate.
Error control: If dirt gets into the water, you ask for a clean refill to get clean water.

In networking:
Flow Control: Prevents the sender from overwhelming the receiver with too much data too quickly.
Error Control: Detects errors in data and requests retransmission for correctness.

Multiplexing
Analogy: Multiple TV channels on one cable

```
- A single cable connection carries multiple TV channels simultaneously.
- You choose which channel (data stream) you want to watch at your TV.

In networking:
Multiple applications (videos, emails, downloads) share the same network connection, each identified and separated at the receiver.
```

3. Network Layer

```
- Host to host delivery
- Logical addressing (IP addresses)
- 32-bit IP address
- Routing decisions
- Packet delivery
- PDU: Packet
```

2. Data Link Layer

```
- Hop to hop delivery
- Physical addressing (MAC)
- 48-bit MAC address, represented using hexa-decimal number
- Framing, error control, flow control
- PDU: Frame
```

1. Physical Layer

```
- Transmission of bits, physical medium characteristics
- Topologies: Bus, Ring
- Modes: Simplex, Half-Duplex, Full-Duplex
```

Why bit synchronization is important?
Bit synchronization is important in the Physical Layer because it helps the receiver know exactly when each bit (0 or 1) starts and ends while receiving data.

If the timing is not matched, the receiver may get confused and read the wrong bits, causing errors in the data.

It is like clapping at the same time in a group song, if you are not in sync, it will sound wrong. In the same way, without bit synchronization, data will be wrong.

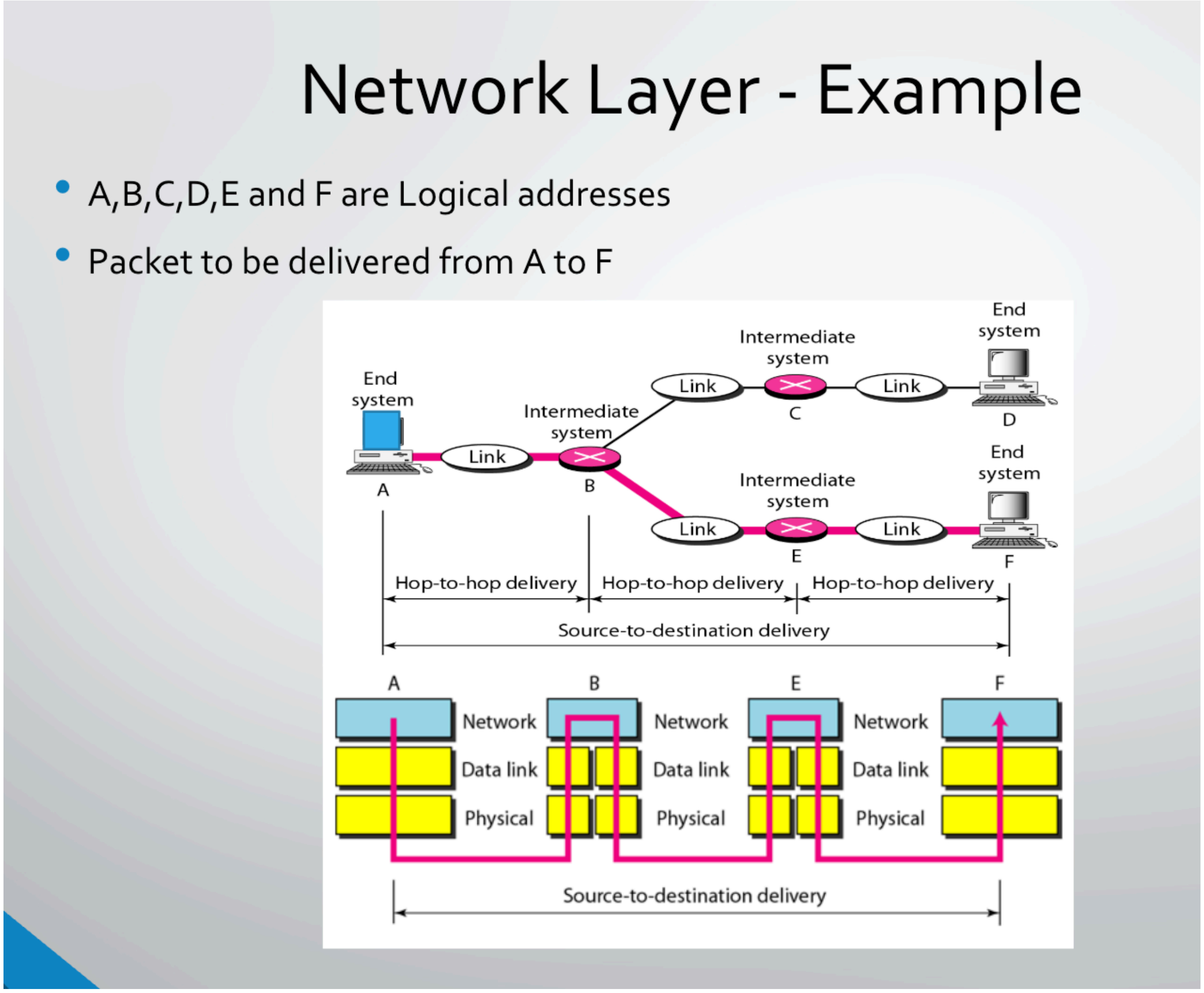
▼ TCP/IP Model

Serial	OSI Model	TCP/IP Model
4/5	Application	Application
4/5	Presentation	Application
4/5	Session	Application
3/4	Transport	Transport
2/3	Network	Internet
1/2	Data Link	Network Access / Data Link
1/1	Physical	Network Access / Physical

Encapsulation:

- Email data → Segments → Packets → Frames → Bits

▼ Network Layer



Network Layer - Example

- A,B,C,D,E and F are Logical addresses
- Packet to be delivered from A to F

Lower diagram represents

- It shows **how a packet travels from A to F** across **multiple nodes (hops)** in a network.
- It explicitly shows **layer invocation at each hop**.

▼ Why the same layers are repeated

OSI Layers are **implemented at each device**:

- Every **end system (A, F)** and **intermediate system (B, E)** has its **own stack** (Network, Data Link, Physical layers).
- **Each device processes the packet independently.**

Layer invocation per **hop**

- To deliver **hop-by-hop**, **Data Link + Physical layers are used at each link.**
- To deliver **end-to-end**, the **Network layer is involved at each device** for routing decisions.

▼ Step-by-step flow

At A (Source):

- Data starts at **Network layer** (adds logical address).
- Goes to **Data Link layer** (adds frame header for A ➔ B).
- Goes to **Physical layer** (converts to bits for transmission).

At B (Router):

- Bits received at **Physical layer**.
- Frame extracted at **Data Link layer**.
- Packet checked at **Network layer** to decide next hop.
- Sent back down:
  - Network ➔ Data Link (new frame for B ➔ E).
  - Data Link ➔ Physical (send bits to E).

At E (Router):

- Same process:
  - Physical ➔ Data Link ➔ Network ➔ Data Link ➔ Physical.

At F (Destination):

- Bits received at **Physical layer**.
- Frame removed at **Data Link layer**.
- Packet extracted at **Network layer** for final delivery to upper layers (Transport, Application).

Summary:

- **Layers repeat because each node processes and forwards the packet independently.**

- The **Data Link + Physical layers repeat for each hop** (to enable hop-to-hop delivery).
- The **Network layer repeats for each device** to handle routing and addressing until the destination is reached.

▼ Visual Analogy

Imagine **postal mail**:

- **Your parcel (packet) has the destination address (Network Layer) written on it, which stays until it reaches the recipient.**
- At **each post office (router)**:
  - Staff check the address to decide where it goes next (Network Layer).
  - They repackage it for the next truck/train/flight (Data Link Layer).
  - It travels to the next post office (Physical Layer).
- This process **repeats at every post office (hop)** until the parcel reaches the recipient.

In essence:

Your packet is like a parcel,

addressed once but repackaged and transported repeatedly at each post office (hop) until it reaches the final destination.

▼ Addressing in Networking

4 address levels in TCP/IP:

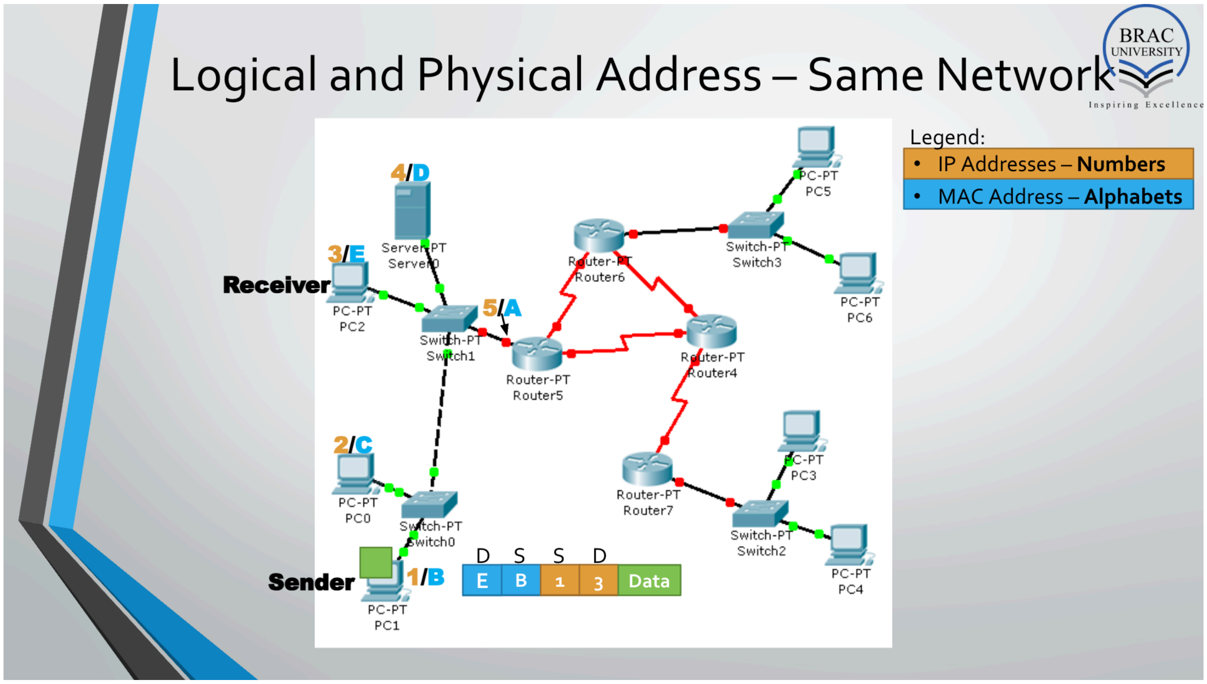
1. **Specific (Application-level):** URLs, emails address
2. **Port Address:** Identifies application/service (e.g., 80 for Web/HTTP, 23 for TCP).
3. **Logical Address:**
   a. IP Address, remains same across network.
   b. Address changes based on location.

c. Used to identify host/end devices.

4. **Physical Address:** MAC Address, changes hop-to-hop.

Key Points:

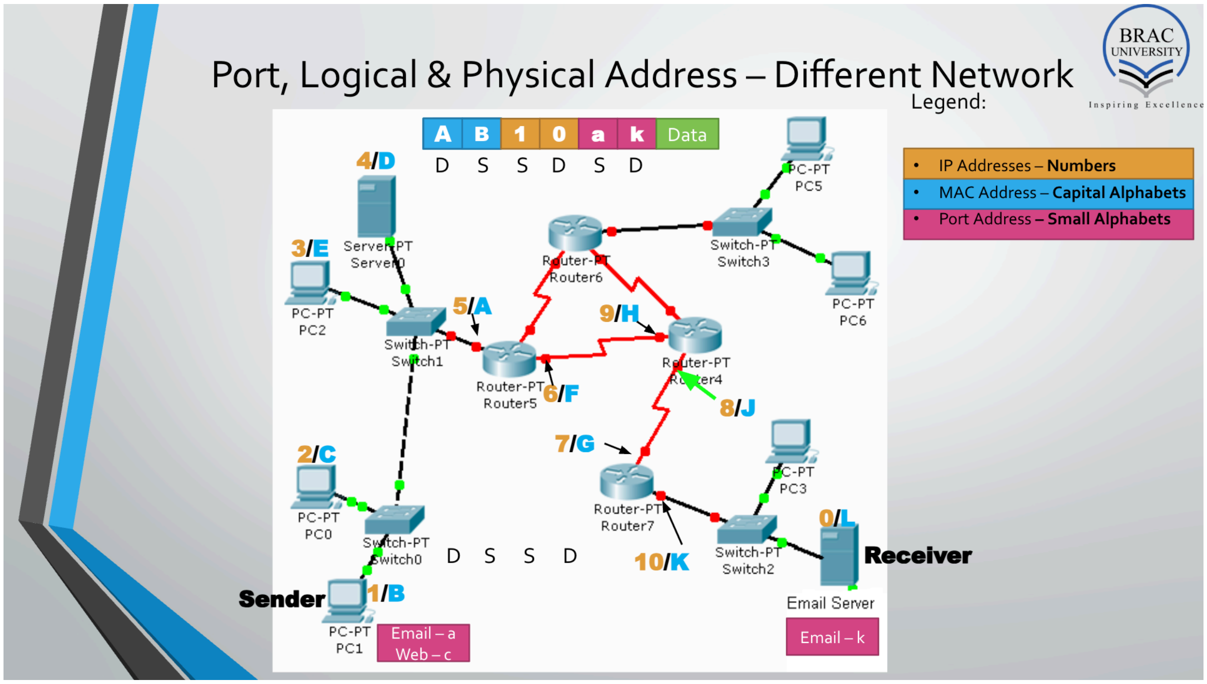
- **MAC/Physical address changes per hop.**
- **Logical and port addresses remain constant source → destination.**



Logical and Physical Address – Same Network

Legend:

- IP Addresses – **Numbers**
- MAC Address – **Alphabets**



Port, Logical & Physical Address – Different Network

Legend:

- IP Addresses – **Numbers**
- MAC Address – **Capital Alphabets**
- Port Address – **Small Alphabets**