Networking Essentials
ITC 2243

KAVINDU CHETHIYA YAKUPITIYA

PHD (IS-READING), MSC (CS), PGD (CS), BSC (IT), DIP (TECH, IT), CCNA, NSE (CERT)

Model vs Protocol vs Port

- **A model** is the specification set by a standards organization as a guideline for designing networks.
- **A protocol** is a set of rules that controls the interaction of different devices in a network or an internetwork. (A protocol is just the language that the two applications on either end of a conversation agree to speak in.
- **Ports** are virtual places within an operating system where network connections start and end.

Networking Models

- There are two basic models in the world of networking.

1) OSI Reference Model (Open System Interconnection)
2) TCP/IP Model

Similarities Between OSI and TCP/IP

1. **Layered Approach:** Both models adopt a layered approach to networking. They divide the complex task of network communication into multiple layers.
2. **Hierarchical Structure:** In both models, the layers are organized hierarchically, with each layer building upon the services provided by the layer below it.
3. **Communication Flow:** Both models define how data is passed from one layer to another.
4. **Protocol Independence:** Both models are designed to be protocol-independent.
5. **Error Handling:** Both models address error handling and recovery at various layers.
6. **Teaching and Learning Tools:** Both models serve as valuable educational tools for teaching networking concepts.

Differences Between OSI and TCP/IP

OSI Model	TCP/IP Model
• Contains 7 Layers	• Contains 4 Layers
• Developed by International Organization for Standardization (ISO) (1970's)	• Developed by ARPANET (1970's)
• Functionalities oriented	• Protocol oriented model
• Used mainly in education	• Widely used in network technologies
• Comprehensive and theoretical model	• Used in real world networking implementation

Protocols

A network protocol is an accepted set of rules that govern data communication between different devices in the network.

The protocols can be broadly classified into three major categories

1. Communication protocols
2. Management protocols
3. Security protocols

Types of Protocols

1) Communication protocols

These protocols formally set out the rules and formats through which data is transferred. These protocols handle syntax, semantics, error detection, synchronization, and authentication.

Examples – HTTP, TCP, UDP, BGP, ARP, IP, DHCP

2) Management protocols

These protocols assist in describing the procedures and policies that are used in monitoring, maintaining, and managing the computer network. These protocols also help in communicating these requirements across the network to ensure stable communication. Network management protocols can also be used for troubleshooting connections between a host and a client.

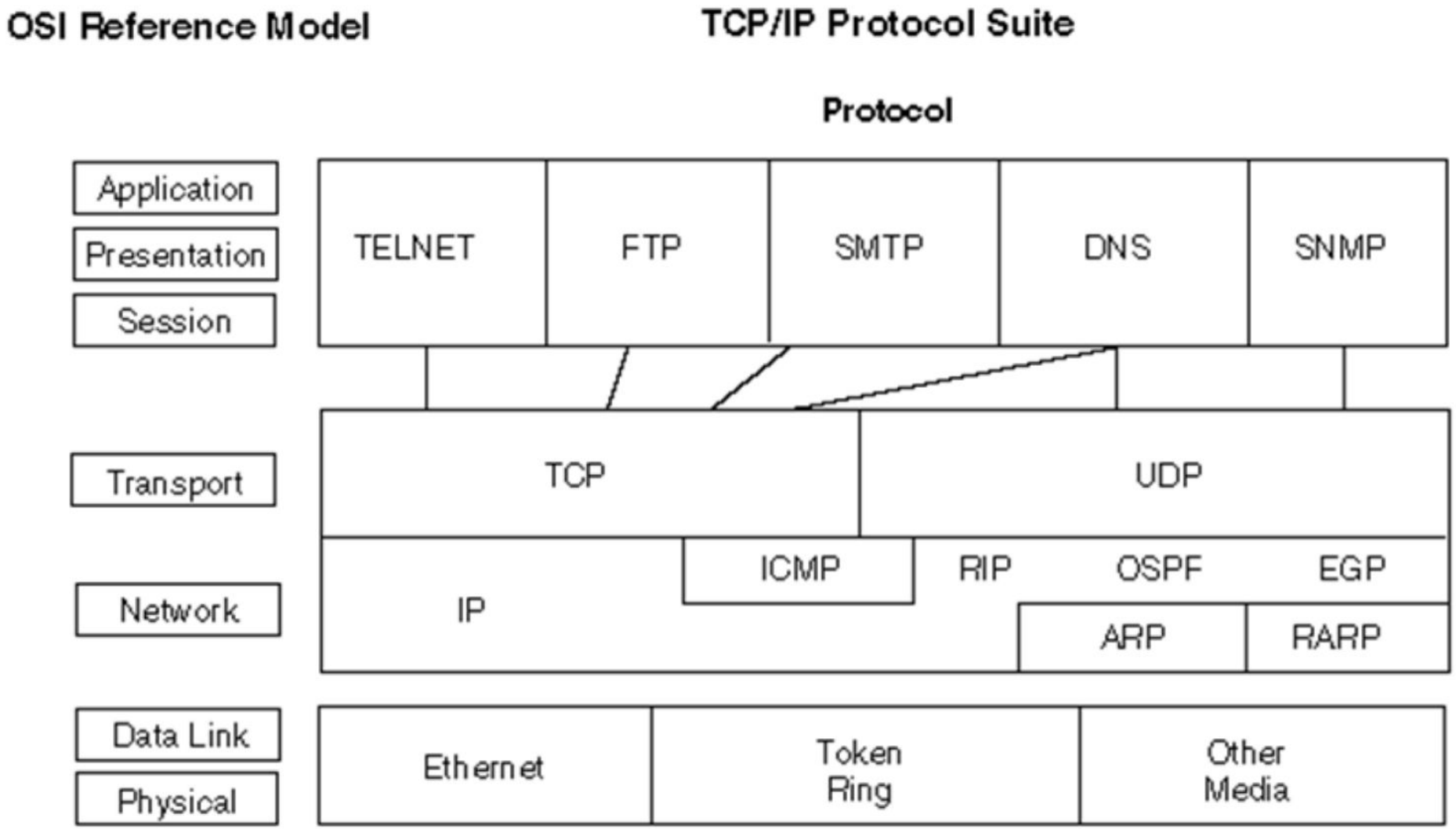
Examples – ICMP, SNMP, FTP, POP3, TELNET

Types of Protocols

3) Security Protocols

These protocols secure the data in passage over a network. These protocols also determine how the network secures data from any unauthorized attempts to extract or review data. These protocols make sure that no unauthorized devices, users, or services can access the network data. Primarily, these protocols depend on encryption to secure data.

Examples – HTTPS, IPsec, SSL, TLS, SSH


OSI Reference Model
TCP/IP Protocol Suite
Protocol
Application
Presentation
Session
Transport
Network
Data Link
Physical
TELNET
FTP
SMTP
DNS
SNMP
TCP
UDP
ICMP
RIP
OSPF
EGP
IP
ARP
RARP
Ethernet
Token Ring
Other Media

Ports

- Ports are virtual places within an operating system where network connections start and end.
- Ports are identified by numbers, which can range from 0 to 65,535.

❑ **There are three categories of ports.**

1) Well-Known Ports: Well-known ports are in the range of **0 to 1023.** These ports are standardized and reserved for specific services and applications that are commonly used in networking.

Examples **-** Port 80: HTTP , Port 443: HTTPS, Port 25: SMTP, Port 110: POP3, Port 143: IMAP, Port 22: SSH, Port 21: FTP, Port 53: DNS, Port 3389: RDP, Port 67/68: DHCP

2) Registered ports: are in the range of **1024 to 49151.** These ports are used for various applications and services, but they are not as standardized as well-known ports. They are typically used by software and applications that require specific port numbers, but they are not globally recognized standards.

Examples **-** Port 1433: Microsoft SQL Server, Port 3306: MySQL database, Port 5432: PostgreSQL database, Port 8080: Alternative HTTP port often used for web proxies

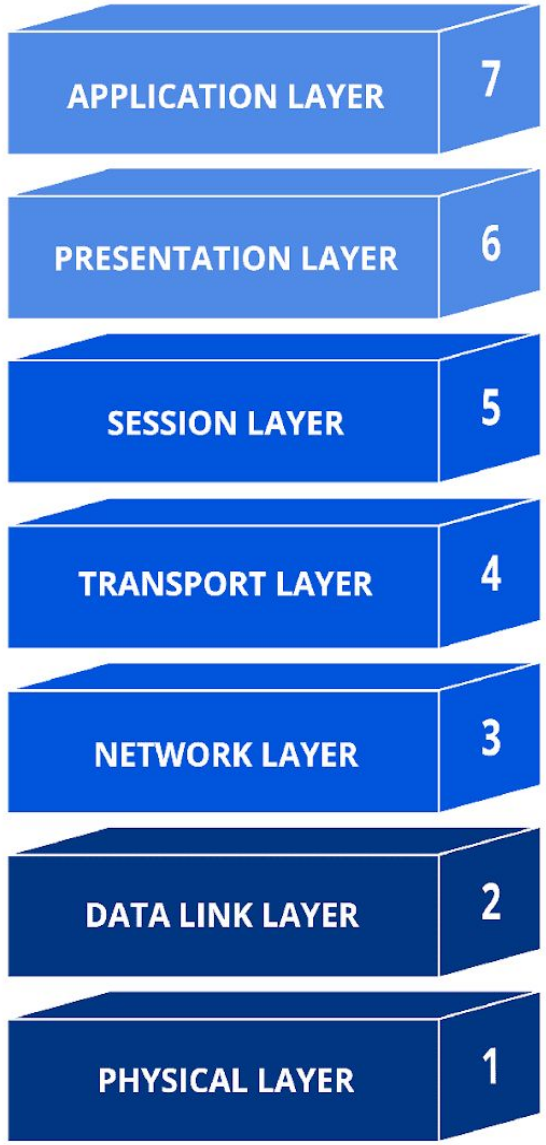
3) Dynamic/Private Ports: Dynamic or private ports are in the range of **49152 to 65535.** These ports are used for temporary and private purposes.

Port #	Application Layer Protocol	Type	Description
20	FTP	TCP	File Transfer Protocol - data
21	FTP	TCP	File Transfer Protocol - control
22	SSH	TCP/UDP	Secure Shell for secure login
23	Telnet	TCP	Unencrypted login
25	SMTP	TCP	Simple Mail Transfer Protocol
53	DNS	TCP/UDP	Domain Name Server
67/68	DHCP	UDP	Dynamic Host
80	HTTP	TCP	HyperText Transfer Protocol
123	NTP	UDP	Network Time Protocol
161,162	SNMP	TCP/UDP	Simple Network Management Protocol
389	LDAP	TCP/UDP	Lightweight Directory Authentication Protocol
443	HTTPS	TCP/UDP	HTTP with Secure Socket Layer

ISO OSI Reference Model

- A reference model is a conceptual blueprint of how communications should take place. It addresses all the processes required for effective communication and divides these processes into logical groupings called layers. When a communication system is designed in this manner, it's known as layered architecture.

- This was created by ISO organization in 1970's

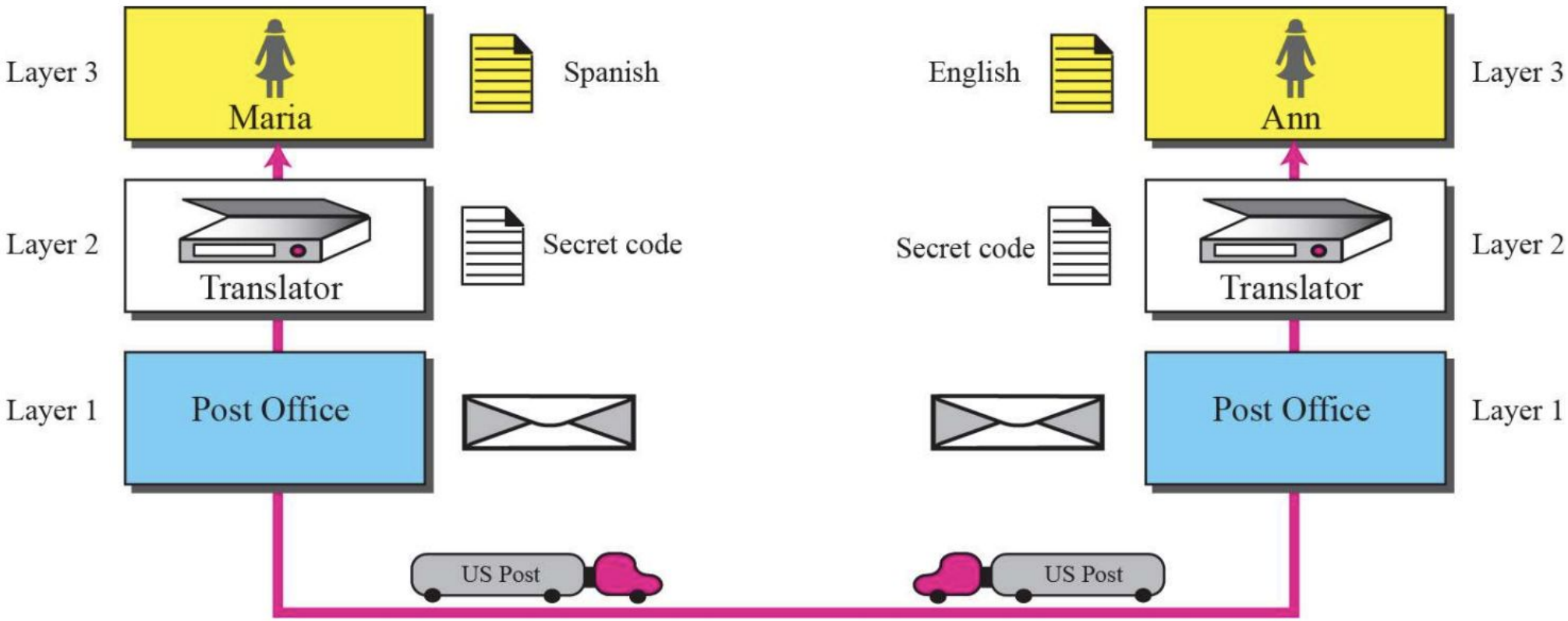


A Practical Example

- Assume Maria and Ann are neighbours with a lot of common ideas.
- However, Maria speaks only Spanish, and Ann speaks only English.
- Since both have learned the sign language in their childhood, they enjoy meeting in a cafe a couple of days per week and exchange their ideas using signs. Occasionally, they also use a bilingual dictionary.
- Communication is face to face and happens in one layer as shown below

A Practical Example Continue….

- Now assume that Ann has to move to another town because of her job. Before she moves, the two meet for the last time in the same cafe.
- Although both are sad, Maria surprises Ann when she opens a packet that contains two small machines.
- The first machine can scan and transform a letter in English to a secret code or vice versa. The other machine can scan and translate a letter in Spanish to the same secret code or vice versa.
- Ann takes the first machine; Maria keeps the second one. The two friends can still communicate using the secret code, as shown in next slide.


Layer 3
Maria
Spanish
English
Ann
Layer 3
Layer 2
Translator
Secret code
Secret code
Translator
Layer 2
Layer 1
Post Office
Post Office
Layer 1
US Post
US Post

ISO OSI Reference Model

HTTP, HTTPS, SMTP, FTP, Telnet, SSH
DNS, NTP, TFTP, SNMP

Data representation – ASCII,
Encryption, Data compassion

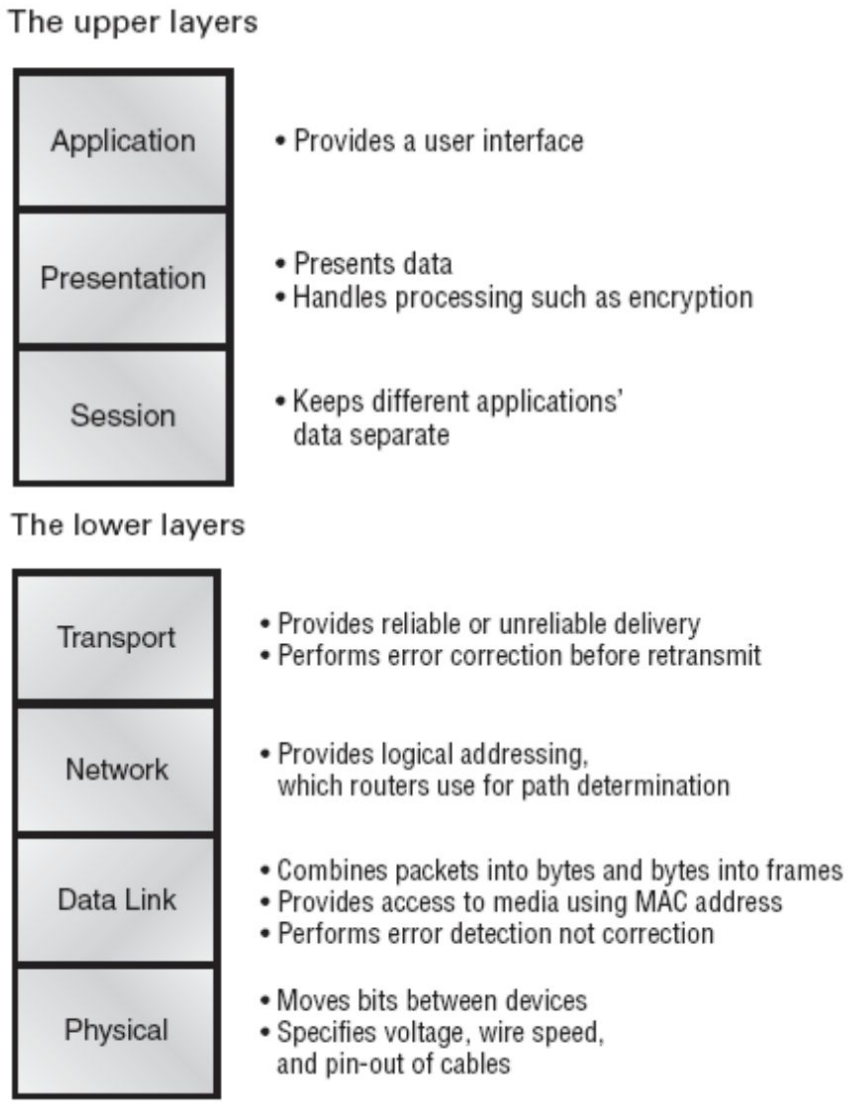
Manages sessions – creating, managing
Terminating (Dialogue Control)

Segments data, TCP | UDP
End-to-End EC

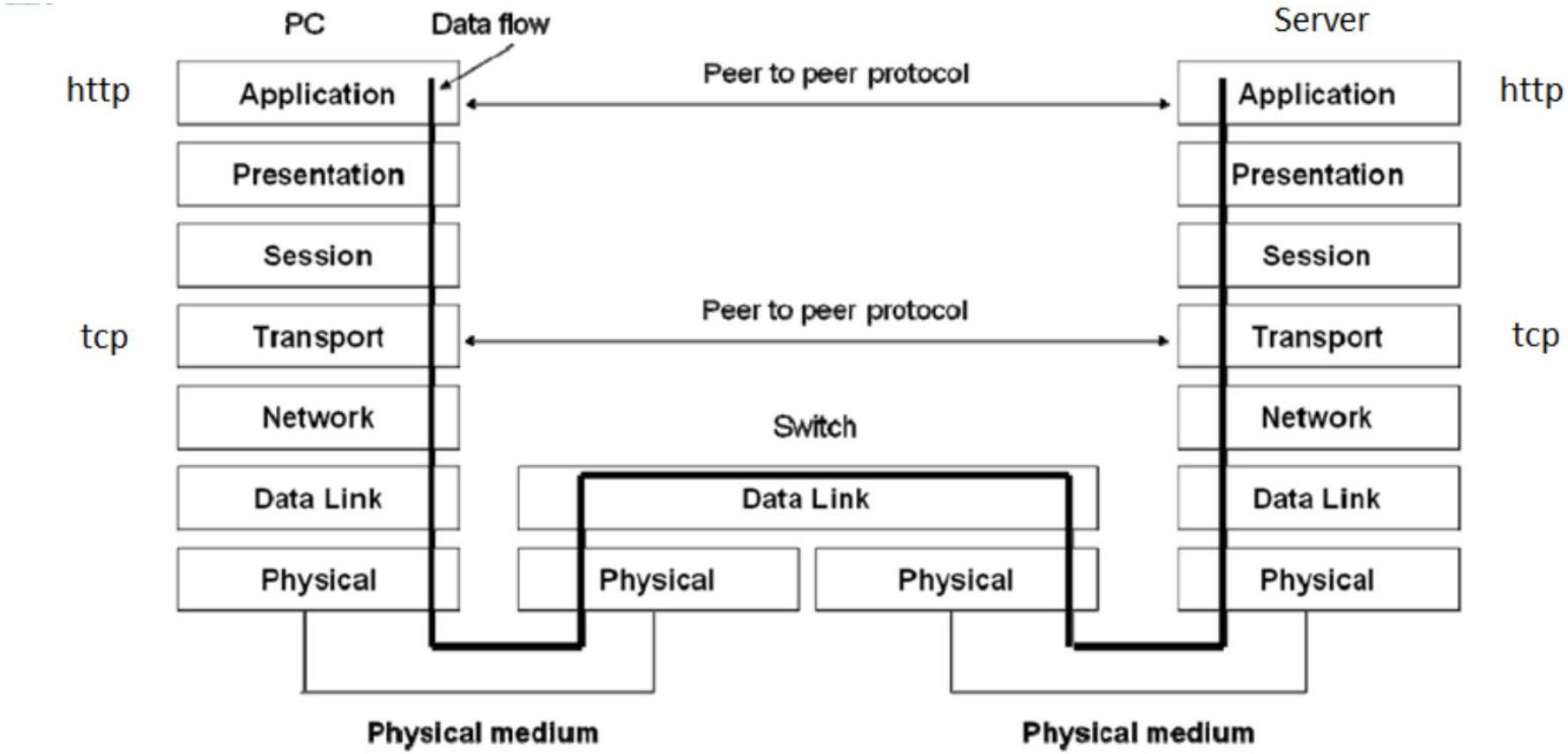
Routers, Routing Pkts (Est. Path)
IP address

Switches, Bridges
Two sub-layers : LLC, MAC
Switching frames, MAC address,
link-to-link EC

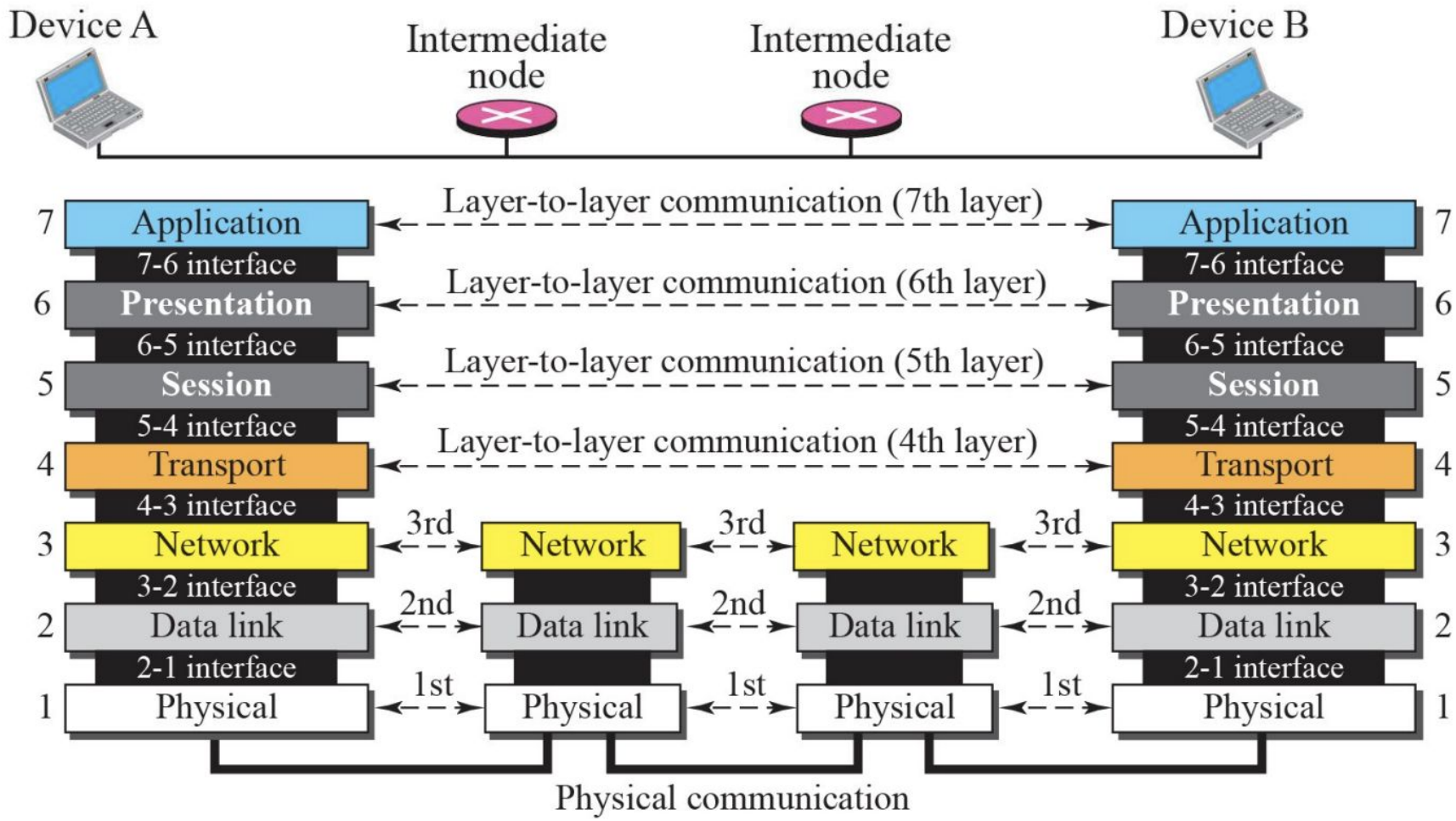
Hub, repeaters
Cables, connectors, voltage,



- A peer to peer protocol is that takes place between two layers at the same level in different end-stations. Conceptually a peer to peer protocol operates horizontally and directly between the two layers as shown below:



Layer to Layer Communication

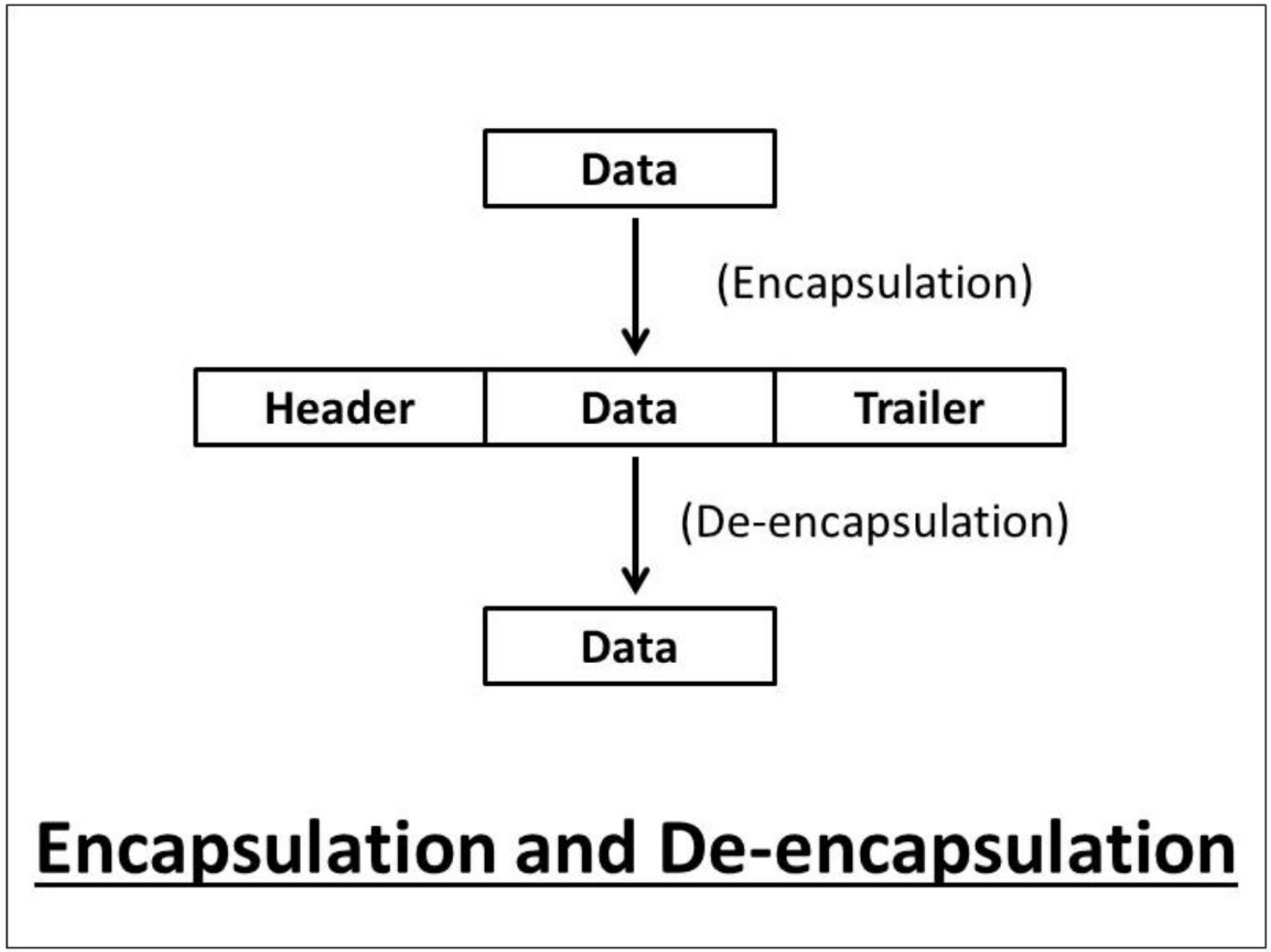


Data Encapsulation

- Data Encapsulation is the process in which some extra information is added to the data item to add some features to it.
- This additional information basically divided into two parts, **Header and Trailer.**
- These are elements attached in order to make the transmission more smoother, on each layer a **PDU (Protocol Data Unit)** is generated.
- The data is encapsulated on the sender's side, starting from the application layer to the physical layer.
- Each layer takes the encapsulated data from the previous layer and adds some more information to encapsulate it and some more functionalities with the data.
- These functionalities may include proper data sequencing, error detection and control, flow control, congestion control, routing information, etc.

Data De-encapsulation

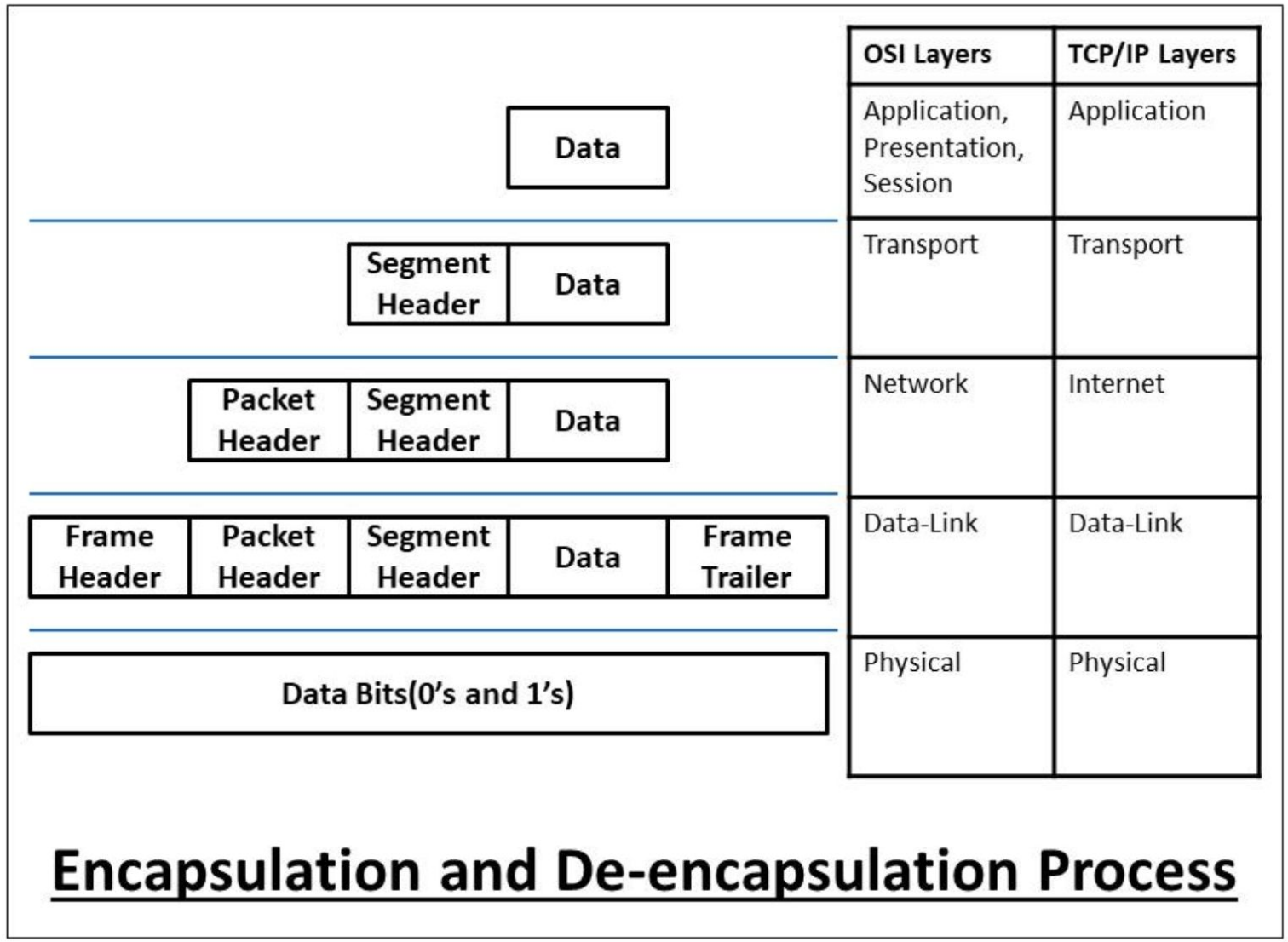
- Data De-encapsulation is the reverse process of data encapsulation.
- The encapsulated information is removed from the received data to obtain the original data.
- This process takes place at the receiver’s end.
- The data is de-encapsulated at the same layer at the receiver’s end to the encapsulated layer at the sender’s end.
- The added header and trailer information are removed from the data in this process.



Encapsulation and De-encapsulation

OSI Layers	TCP/IP Layers	Encapsulated Term
Application	Application	Data
Presentation		Data
Session		Data
Transport	Transport	Segment
Network	Internet	Packet
Data-Link	Data-Link	Frame
Physical	Physical	Bits

Encapsulated Data Term(OSI & TCP/IP Model)



Encapsulation and De-encapsulation Process

Encapsulation Process at Sender's Side

- Step 1: The Application, Presentation, and Session layer in the OSI model , or the Application layer in the TCP/IP model takes the user data in the form of data streams, encapsulates it and forwards the data to the Transport layer. It does not necessarily add any header or footer to the **data.** But it is application-specific and can add the header if needed.
- Step 2: The Transport layer (in the OSI or TCP/IP model) takes the data stream from the upper layers, and divide it into multiple pieces. The Transport layer encapsulates the data by adding the appropriate header to each piece. These data pieces are now called as data **segments.** The header contains the sequencing information so that the data segments can be reassembled at the receiver's end.
- Step 3: The Network layer (in the OSI model) or the Internet layer (in the TCP/IP model) takes the data segments from the Transport layer and encapsulate it by adding an additional header to the data segment. This data header contains all the routing information for the proper delivery of the data. Here, the encapsulated data is termed as a data **packet** or datagram.
- Step 4: The Data-Link layer (in the OSI or TCP/IP model) takes the data packet or datagram from the Network layer and encapsulate it by adding an additional header and footer to the data packet or datagram. The header contains all the switching information for the proper delivery of the data to the appropriate hardware components, and the trailer contains all the information related to error detection and control. Here, the encapsulated data is termed as a data **frame.**
- Step 5: The Physical layer (in the OSI or TCP/IP model) takes the data frames from the Data-Link layer and encapsulate it by converting it to appropriate data **signals or bits (**corresponding to the physical medium).

De-encapsulation Process at Reciever's Side

- Step 1: The Physical layer (in the OSI or TCP/IP model) takes the encapsulated data signals or bits from the sender, and de-encapsulate it in the form of a data frame to be forwarded to the upper layer, i.e., the Data-Link layer.
- Step 2: The Data-Link layer (in the OSI or TCP/IP model) takes the data frames from the Physical layer. It de-encapsulates the data frames and checks the frame header whether the data frame is switched to the correct hardware or not. If the frame is switched to the incorrect destination, it is discarded, else it checks the trailer information. If there is any error in the data, data retransmission is requested, else it is de-encapsulated and the data packet is forwarded to the upper layer.
- Step 3: The Network layer (in the OSI model) or the Internet layer (in the TCP/IP model) takes the data packet or datagram from the Data-Link layer. It de-encapsulates the data packets and checks the packet header whether the packet is routed to the correct destination or not. If the packet is routed to the incorrect destination, the packet is discarded, else it is de-encapsulated and the data segment is forwarded to the upper layer.
- Step 4: The Transport layer (in the OSI or TCP/IP model) takes the data segments from the network layer and de-encapsulate it. It first checks the segment header and then reassembles the data segments to form data streams, and these data streams are then forwarded to the upper layers.
- Step 5: The Application, Presentation, and Session layer in the OSI model , or the Application layer in the TCP/IP model takes encapsulated data from the Transport layer, de-encapsulate it, and the application-specific data is forwarded to the applications.

Seven Layers of OSI Reference Model

- Physical layer
- Data-link layer
- Network layer
- Transport layer
- Session layer
- Presentation layer
- Application layer

Physical Layer

- The physical layer is responsible for transmitting a **bit stream** over a **physical medium.**
- It encodes and decodes bits into **groups of bits.**
- It then transforms a stream of bits into a **signal.**

Data-link Layer

- The data-link layer organizes bits into **logical units** called **frames.**
- The data-link layer is responsible only for **node-to-node delivery** of the frame.
- The data-link layer is often responsible for **error handling** between two adjacent stations.
- **Bit-stuffing** is done at this stage. (Bit stuffing is the insertion of non information bits into data.)

Network Layer

- The network layer is responsible for delivery of a **packet** between the original source and final destination **(Routing and Forwarding).**
- Using **logical addresses (IP addresses)** instead of physical addresses.
- Example of IP address - 140.122.76.121 (4 Bytes)

Transport Layer

- The Transport Layer **manages the flow of data between sender and receiver.** It prevents the sender from overwhelming the receiver with data too quickly, avoiding congestion and potential data loss. Flow control mechanisms help ensure that the receiver can process the incoming data at a rate it can handle.
- It is responsible to **perform error correction and error detection.** It uses mechanisms like checksums and sequence numbers to detect and potentially correct errors that might occur during transmission.
- Further it perform a **reliable delivery.** This ensures that if data is lost or corrupted during transmission, it will be retransmitted until the receiving side successfully receives the correct data.

Session Layer

- The session layer is designed to **control the dialog** between users.
- The synchronization points divides a long message into smaller ones and ensure that each section is received and acknowledged by the receiver.
- Most network implementations today do not use a separate session layer, their services are usually included in the application layer.
- Its primary responsibility is to **establish, manage, and terminate communication sessions between two devices.**
- A session refers to a logical connection established between two devices to enable data exchange for a specific purpose, such as file transfer, remote access, or database synchronization.

Presentation Layer

- The presentation layer is concerned with the syntax and semantics of the information exchanged between two systems.
- It deals with the fact that different systems use different coding methods.

❑ **Compress and decompress data**

❑ **Encrypt and decrypt data**

Application Layer

- The application layer enables the user to access the network.
- It defines common applications that can be implemented to make the job of the user simpler.
- **User authentication and privacy** are considered at this layer.
- This is the layer which provide **interface for user** to interact with the network.

Thank you!!!