**ASSIGNMENT-1**

1. **Prepare R&D Document on working of all the layers in OSI Model.**
2. **Prepare R&D Document on working & functionality of TCP/IP Model.**
3. **Prepare R&D Document working of TCP & UDP Protocols, working of HTTP, HTTPs & ICMP Protocol.**
4. **OSI MODEL :**

What Is the OSI Model?

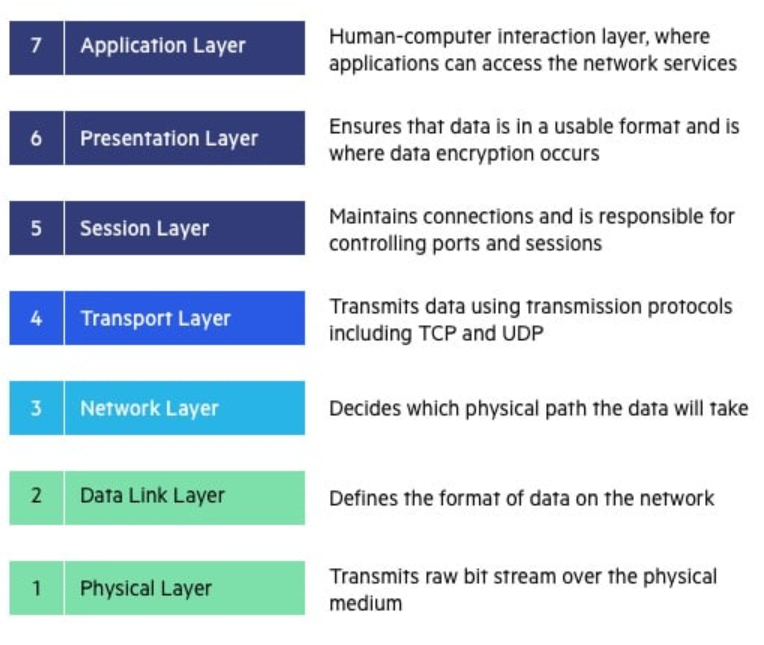
The Open Systems Interconnection (OSI) model describes seven layers that computer systems use to communicate over a network. The OSI model is divided into seven distinct layers, each with specific responsibilities, ranging from physical hardware connections to high-level application interactions.

The OSI model was the first standard model for network communications, adopted by all major computer and telecommunication companies in the early 1980s. It was introduced in 1983 by representatives of the major computer and telecom companies, and was adopted by ISO as an international standard in 1984.

The modern Internet is not based on OSI, but on the simpler TCP/IP model. However, the OSI 7-layer model is still widely used, as it helps visualize and communicate how networks operate.

OSI Model Explained: The OSI 7 Layers

Top-Down Layered Working:



7. Application Layer

The Application Layer serves as the interface between the end-user applications and the underlying network services. This layer provides protocols and services that are directly utilized by end-user applications to communicate across the network. Key functionalities of the Application Layer include resource sharing, remote file access, and network management.

Examples of protocols operating at the Application Layer include Hypertext Transfer Protocol (HTTP) for web browsing, File Transfer Protocol (FTP) for file transfers, Simple Mail Transfer Protocol (SMTP) for email services, and Domain Name System (DNS) for resolving domain names to IP addresses. These protocols ensure that user applications can effectively communicate with each other and with servers over a network.

6. Presentation Layer

The Presentation Layer, also known as the syntax layer, is responsible for translating data between the application layer and the network format. It ensures that data sent from the application layer of one system is readable by the application layer of another system. This layer handles data formatting, encryption, and compression, facilitating interoperability between different systems.

One of the key roles of the Presentation Layer is data translation and code conversion. It transforms data into a format that the application layer can understand. For example, it may convert data from ASCII to EBCDIC. It also includes encryption protocols to ensure data security during transmission and compression protocols to reduce the amount of data for efficient transmission.

5. Session Layer

The Session Layer manages and controls the connections between computers. It establishes, maintains, and terminates connections, ensuring that data exchanges occur efficiently and in an organized manner. The layer is responsible for session checkpointing and recovery, which allows sessions to resume after interruptions.

Protocols operating at the Session Layer include Remote Procedure Call (RPC), which enables a program to execute a procedure on a remote host as if it were local, and the session establishment phase in protocols like NetBIOS and SQL. These services enable reliable communication, especially in complex network environments.

1. Transport Layer

The Transport Layer provides end-to-end communication services for applications. It ensures complete data transfer, error recovery, and flow control between hosts. This layer segments and reassembles data for efficient transmission and provides reliability with error detection and correction mechanisms.

Protocols at this layer include Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP is connection-oriented and ensures reliable data transfer with error checking and flow control, making it suitable for applications like web browsing and email. UDP is connectionless, offering faster, though less reliable, transmission, suitable for applications like video streaming and online gaming.

3. Network Layer

The Network Layer is responsible for data routing, forwarding, and addressing. It determines the best physical path for data to reach its destination based on network conditions, the priority of service, and other factors. This layer manages logical addressing through IP addresses and handles packet forwarding.

Key protocols at this layer include the Internet Protocol (IP), which is important for routing and addressing, Internet Control Message Protocol (ICMP) for diagnostic and error-reporting purposes, and routing protocols like Routing Information Protocol (RIP) that manage the routing of data across networks.

2. Data Link Layer

The Data Link Layer is responsible for node-to-node data transfer and error detection and correction. It ensures that data is transmitted to the correct device on a local network segment. This layer manages MAC (Media Access Control) addresses and is divided into two sublayers: Logical Link Control (LLC) and Media Access Control (MAC).

Protocols and technologies at this layer include Ethernet, which defines the rules for data transmission over local area networks (LANs), and Point-to-Point Protocol (PPP) for direct connections between two network nodes. It also includes mechanisms for detecting and possibly correcting errors that may occur in the Physical Layer.

1. Physical Layer

The Physical Layer is responsible for the physical connection between devices. It defines the hardware elements involved in the network, including cables, switches, and other physical components. This layer also specifies the electrical, optical, and radio characteristics of the network.

Functions of the Physical Layer include the modulation, bit synchronization, and transmission of raw binary data over the physical medium. Technologies such as Fiber Optics and Wi-Fi operate at this layer, ensuring that the data physically moves from one device to another in the network.

1. **TCP/IP Model:**

What is TCP/IP?

Transmission Control Protocol/Internet Protocol (TCP/IP) is a network protocol suite that defines the requirements for safe and efficient data transfer online. This protocol is tasked with finding the destination IP address.

The TCP/IP interconnects network devices by breaking data into smaller packets, assigning each packet an IP address, and routing them through different networks to their destination. In this process, TCP is responsible for breaking down packets, sending them, and reassembling them at the destination, while IP makes sure the data is sent to the correct device.

How does TCP/IP work?

TCP/IP works according to the client-server communication model, where the client device receives services from a server device. The main function of TCP/IP is to accurately transmit messages between devices over a network.

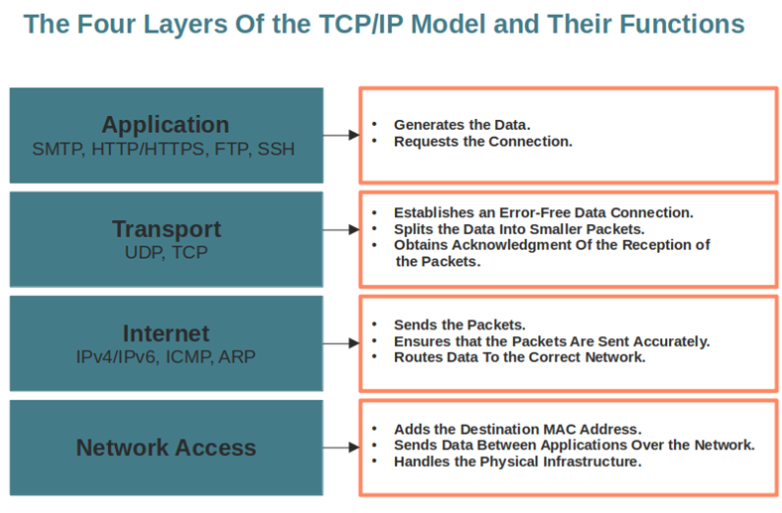
For maximum accuracy, messages are broken down into data packets (small bits of information with headers attached to steer them through the network). If any individual packets are corrupted en route, there’s no need to resend the whole message — the system simply sends the requisite packet again.

The TCP/IP model is responsible for transferring data between two devices. Why only two? Well, unlike radio, internet data isn’t just broadcast to whoever is listening. Even when hundreds of computers are sharing data at the same time, only two devices participate in any given data exchange.

Four layers of TCP/IP

TCP/IP protocols operate on four layers. This system is how the TCP/IP model ensures that different devices and apps can “communicate” and transfer data efficiently.

Four layers of TCP/IP include:



1. Application layer

The top layer includes the application layer protocols. It’s the easiest layer for users to interact with since these protocols are built into their apps (hence the name). Mail programs have SMTP, or Simple Mail Transfer Protocol, Internet browsers use HTTP, or Hypertext Transfer Protocol, and so on. Other common application layer protocols include FTP (File Transfer Protocol), Dynamic Host Configuration Protocol (DHCP), and Simple Network Management Protocol (SNMP).

1. Transport layer

As you might’ve guessed, the transport layer takes care of transporting data. It includes TCP as well as User Datagram Protocol (UDP). UDP is simpler than TCP and is commonly used by real-time applications that don’t need to be as secure as other kinds of data.

TCP establishes the connection between the two networks and chops up the data into smaller pieces (packets) for efficiency. TCP also adds the assembly rules to each packet so the data can be put back together in the correct order after the transfer is complete.

1. Internet layer

Internet layer protocols include IP as well as Address Resolution Protocol (ARP), Internet Group Management Protocol (IGMP), and Internet Control Message Protocol (ICMP). The Internet layer manages the movement of data packets between the networks.

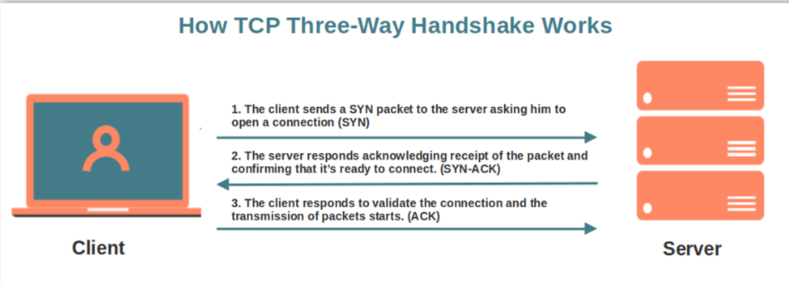
1. Datalink layer

The Datalink layer is the deepest layer of data transfer and can also be called the network interface layer. The job of this layer is to make sure that data not only arrives at the intended IP address (the router) but also the correct device within that network (your phone connected to that router). This involves identifying the MAC address of the intended device and managing data transfer through the Ethernet cables and Wi-Fi.

1. **Working of TCP & UDP Protocols**

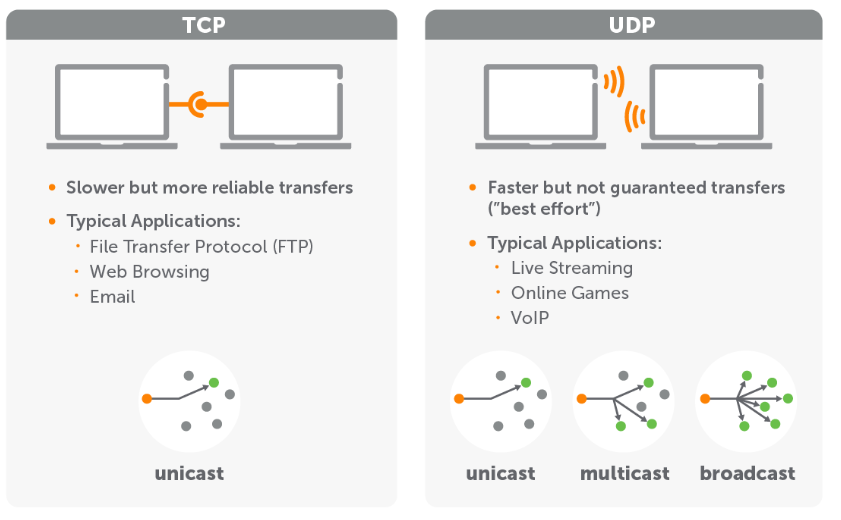
Transmission Control Protocol (TCP)

TCP is a connection-oriented protocol that ensures reliable and ordered delivery of data between applications. It begins with a three-way handshake to establish a connection: the sender sends a SYN (synchronize) message, the receiver responds with SYN-ACK (synchronize-acknowledge), and the sender replies with an ACK (acknowledge), confirming the connection. Once established, TCP segments the data into packets, assigns sequence numbers, and transmits them. The receiver acknowledges each packet, and if a packet is lost or corrupted, TCP retransmits it. This protocol ensures error checking, flow control, and congestion avoidance, making it suitable for tasks like web browsing, file transfers, and email.



User Datagram Protocol (UDP)

UDP is a connectionless protocol that sends data without establishing a connection, making it faster but less reliable than TCP. It does not perform error recovery, sequencing, or flow control. Instead, it transmits datagrams (packets) directly to the receiver, regardless of whether they are received or in order. Because of its low overhead, UDP is used in real-time applications such as video streaming, online gaming, and VoIP, where speed is more important than reliability. While UDP doesn’t guarantee delivery, it’s ideal for time-sensitive data transmission.



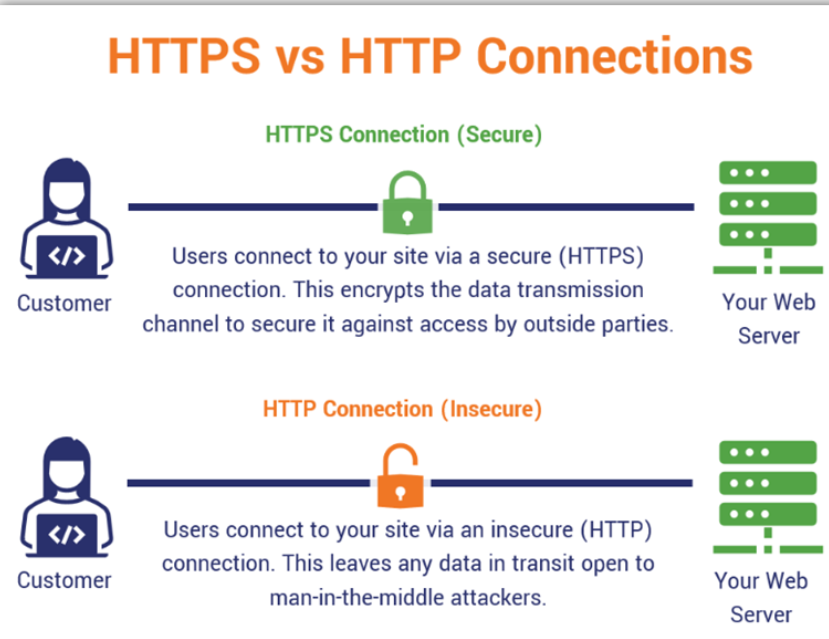
1. **Working of HTTP & HTTPS Protocols**

Hypertext Transfer Protocol (HTTP)

HTTP is an application-layer protocol used to transfer data such as HTML documents between a web browser and a web server. It operates over TCP and uses a request-response model. When a user enters a URL in the browser, the browser sends an HTTP request to the server hosting the website. The server processes the request and returns an HTTP response containing the requested resource (like a webpage or image). HTTP is stateless, meaning each request is treated independently without memory of previous requests. While efficient for basic content delivery, it lacks security, as data is transmitted in plaintext.

Hypertext Transfer Protocol Secure (HTTPS)

HTTPS is the secure version of HTTP. It encrypts data using SSL/TLS (Secure Sockets Layer/Transport Layer Security) protocols, ensuring confidentiality, integrity, and authentication between the client and server. When a user accesses a website via HTTPS, the client and server first perform an SSL/TLS handshake to exchange encryption keys. Once a secure connection is established, HTTP requests and responses are transmitted over this encrypted channel. This protects sensitive data such as login credentials, payment information, and personal details, making HTTPS essential for secure online communication.



HTTP is stateless, meaning each request is treated as a new one without memory of previous interactions. This improves speed but requires cookies or sessions to maintain user data like logins. In HTTPS, the server presents a digital certificate issued by a trusted Certificate Authority (CA) to verify its identity during the SSL/TLS handshake. The client (browser) checks this certificate to ensure the website is genuine and not a fraudulent imitation. If the certificate is valid and trusted, a secure session is established. This certificate-based trust mechanism is essential for preventing man-in-the-middle (MITM) attacks and ensuring that users are safely interacting with authentic websites, especially on public or unsecured networks.

1. **Working of ICMP Protocol**

Internet Control Message Protocol (ICMP)

ICMP is a network-layer protocol used primarily for diagnostic and error-reporting functions in an IP network. Unlike TCP and UDP, ICMP does not transmit application data but instead provides feedback about network conditions. When an error occurs during packet delivery—such as a host being unreachable or a router being unable to forward a packet—ICMP generates a message and sends it back to the source IP. Common tools like ping and traceroute rely on ICMP to test network connectivity and measure latency. For example, when you use ping, your computer sends ICMP Echo Request messages, and if the destination is reachable, it replies with Echo Reply messages. ICMP thus helps in monitoring, diagnosing, and managing network health and reachability.

The ICMP protocol requires no handshakes or formal connections. Instead, it operates as a connectionless protocol. And the data you get back is in numeric form. You'll need to decode it to make sense of it.

Developers often use charts to explain what ICMP messages look like.

Each message is made up of four parts.

Type: This field contains a number that defines the message's general category.

Code: This field contains a number that offers more information about the message's type.

Checksum: This field contains a numeric representation of security. Errors or mismatches here indicate that the data has been tampered with.

Content: Additional information may appear here, depending on your type and code content.

Imagine that your message contains this data:

Type: 3

Code: 3

Checksum: 3 3

You've been told that your destination is unreachable due to an unavailable destination port, and the message hasn't been altered in transit.

