**COMPUTER NETWORKING**

1. **What is an OSI model? Explain all its layers with diagram.**

**Ans.** Before 1990’s data communication model had been used, in 1990’s open system Interconnection (OSI) was used. Later TCP/ Ip protocol suit has been developed such that it was easy to use and efficient to other. Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement on International standards. An ISO standard that covers all aspects of network communications is the OSI model, which was first introduced in 1970s.

**Open System:** A set of protocols that allows any two different systems to communicate regardless of their underlying architecture.

**Purpose of OSI Model:** It shows how to facilitate communication between different systems without requiring changes to the logic of underlying hardware and software.

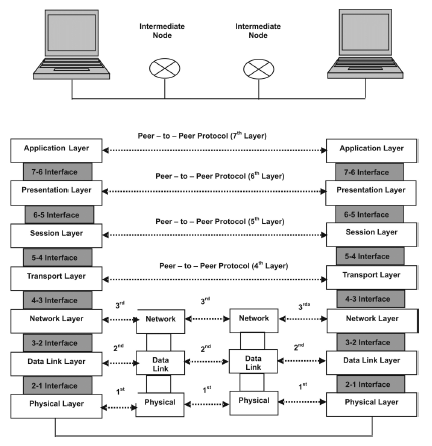
The OSI model is not a protocol; it is a model for understanding and designing a network architecture that is flexible, robust, and interoperable. The OSI Model is a layered framework for the design of network systems that allows communication between all types of computer systems. It consists of seven separate but related layers, each of which defines a part of the process of moving information across a network.

**Layered Architecture:**

The OSI Model is composed of seven ordered layers:

* + Layer 1 – The Physical Layer
  + Layer 2 – The Data Link Layer
  + Layer 3 – The Network Layer
  + Layer 4 – The Transport Layer
  + Layer 5 – The Session Layer
  + Layer 6 – The Presentation Layer
  + Layer 7 – The Application Layer

Below shows the layers involved when a message is sent from device A to device B. As the message travels from one device to another, it may pass through several intermediate nodes or devices. These intermediate nodes or devices usually involve only the first three layers of the OSI model.



1. **What is Message and Packet switching?**

**Ans.** **Message switching**

Message switching was the precursor of packet switching, where messages were routed in their entirety and one hop at a time. It was first introduced by Leonard Kleinrock in 1961. Message switching systems are nowadays mostly implemented over packet-switched or circuit-switched data networks.

Hop-by-hop Telex forwarding are examples of message switching systems. E-mail is another example of a message switching system. When this form of switching is used, no physical path is established in advance in between sender and receiver. Instead, when the sender has a block of data to be sent, it is stored in the first switching office (i.e. router) then forwarded later at one hop at a time.

Each block is received in its entity form, inspected for errors and then forwarded or re-transmitted. It is a form of store-and-forward network. Data is transmitted into the network and stored in a switch. The network transfers the data from switch to switch when it is convenient to do so, and as such the data is not transferred in real-time. Blocking cannot occur, however, long delays can happen. The source and destination terminal need not be compatible, since conversions are done by the message switching networks.

Again consider a connection of a network shown in Figure 2.23. For instance, when a telex (or email) message is sent from A to D, it first passes over a local connection (AB). It is then passed at some later time to C (via link BC), and from there to the destination (via link CD). At each message switch, the received message is stored, and a connection is subsequently made to deliver the message to the neighboring message switch. Message switching is also known as store-and-forward switching

since the messages are stored at intermediate nodes en route to their destinations.

**Packet switching**

Packet switching splits traffic data (for instance, digital representation of sound, or computer data) into chunks, called packets. Packet switching is similar to message switching. Any message exceeding a network-defined maximum length is broken up into shorter units, known as packets, for transmission. The packets, each with an associated header, are then transmitted individually through the network. These packets are routed over a shared network. Packet switching networks do not require a circuit to be established and allow many pairs of nodes to communicate almost simultaneously over the same channel. Each packet is individually addressed precluding the need for a dedicated path to help the packet find its way to its destination.

Packet switching is used to optimize the use of the channel capacity available in a network; to minimize the transmission latency (i.e. the time it takes for data to pass across the network), and to increase robustness of communication.

1. **List the design issues related to Data Link Layer.**

**Ans.** To accomplish the above goals, the data link layer takes the packets from the network layer and encapsulates bit stream into units called *frames* for transmission. Note that frames are nothing more than “packets'' or “messages''. By convention, we'll use the term ``frames'' when discussing DLL packets. Each frame contains a frame header, a payload field for holding the packet, and a frame trailer as illustrated in below.

|  |  |  |
| --- | --- | --- |
|  | Sending machine |  |
|  | Packet |  |
|  |  |  |
| Header | Payload field | Trailer |

In general the DLL design issues are listed below:

1. In general, the Data Link Layer provides services to the network layer. The network layer should be able to send packets to its neighbors without worrying about the details of getting it there in one piece.
2. The DLL does the process of framing the bits, i.e. encapsulating the packets
3. Sender checksums the frame and sends checksum together with data. The checksum allows the receiver to determine when a frame has been damaged in transit.
4. Receiver recomputes the checksum and compares it with the received value. If they differ, an error has occurred and the frame is discarded.
5. Perhaps return a *positive* or *negative acknowledgment* to the sender. A positive acknowledgment indicates the frame was received without errors, while a negative acknowledgment indicates the opposite.
6. **Flow control:** A data link protocol discusses tasks like error control and flow control but these tasks are also dealt at transport layer along with some other protocols.
7. **Briefly explain Point-to-Point Protocol.**

**Ans. Point-to-Point Protocol (PPP)**

Point-to-Point Protocol (PPP) is a network-specific standard protocol with STD number 51. Its status is elective, and it is described in RFC 1661 and RFC 1662. The standards defined in these RFCs were later extended to allow IPv6 over PPP, defined in RFC 2472.

There are a large number of *proposed standard protocols*, which specify the operation of PPP over different kinds of point-to-point links. Each has a status of elective. We advise you to consult STD 1 – Internet Official Protocol Standards for a list of PPP-related RFCs that are on the Standards Track.

Point-to-point circuits in the form of asynchronous and synchronous lines have long been the mainstay for data communications. In the TCP/IP world, the de facto standard SLIP protocol has served admirably in this area, and is still in widespread use for dial-up TCP/IP connections. However, SLIP has a number of drawbacks that are addressed by the Point-to-Point Protocol.

PPP has three main components:

* A method for encapsulating datagrams over serial links.
* A *Link Control Protocol (LCP)* for establishing, configuring, and testing the data-link connection.
* A family of *Network Control Protocols* (NCPs) for establishing and configuring different network-layer protocols. PPP is designed to allow the simultaneous use of multiple network-layer protocols.

Before a link is considered to be ready for use by network-layer protocols, a specific sequence of events must happen. The LCP provides a method of establishing, configuring, maintaining, and terminating the connection. LCP goes through the following phases:

1. **Briefly explain five parts of Multipurpose Internet Mail Extensions.**

**Ans.** Electronic mail is probably the most widely used TCP/IP application. However, SMTP is limited to 7-bit ASCII text, with a maximum line length of 1000 characters. This results in a number of limitations, including:

 SMTP cannot transmit executable files or other binary objects. There are ad hoc methods of encapsulating binary items in SMTP mail items, such as:

– Encoding the file as pure hexadecimal

– The UNIX **uuencode** and **uudecode** utilities, used to encode binary data in the UNIX-to-UNIX Copy (UUCP) mailing system

However, none of these can be described as a *de facto* standard (though **uuencode**/**uudecode** is perhaps the most pervasive, due to the pioneering role of UNIX systems in the Internet).

 SMTP cannot transmit text data that includes national language characters, because these are represented by code points with a value of 128 (decimal) or higher in all character sets based on ASCII.

 SMTP servers might reject mail messages over a certain size. Any given server can have permanent or transient limits, or both, on the maximum amount of mail data it can accept from a client at any given time.

 SMTP gateways that translate from ASCII to EBCDIC and vice versa do not use a consistent set of code page mappings, resulting in translation problems.

 Some SMTP implementations or other mail transport agents (MTAs) in the Internet do not adhere completely to the SMTP standards defined in RFC 2821. Common problems include:

– Removal of trailing white-space characters (tabs and spaces).

– Addition of white-space characters to make all lines in a message the same length.

– Wrapping of lines longer than 76 characters.

– Changing of new line sequences between different conventions. (For example, <CRLF> characters might be converted to <CR> or <LF> sequences.)

– Conversion of tab characters to multiple spaces.

MIME is a draft standard that includes mechanisms to resolve these problems in a manner that is highly compatible with existing RFC 2822 standards. Because mail messages are frequently forwarded through mail gateways, it is not possible for an SMTP client to distinguish between a server that manages the destination mailbox and one that acts as a gateway to another network. Because the mail that passes through a gateway might be tunneled through further gateways, some or all of which can be using a different set of messaging protocols, it is not possible in general for a sending SMTP to determine the lowest common denominator capability common to all stages of the route to the destination mailbox. For this reason, MIME assumes the worst: 7-bit ASCII transport, which might not strictly conform to or be compatible with RFC 2821. It does not define any extensions to RFC 2821, but limits itself to extensions within the framework of RFC 2822. Therefore, a MIME message is one which can be routed through any number of networks that are loosely compliant with RFC 2821 or are capable of transmitting RFC 2821 messages.

MIME can be described in five parts:

 Protocols for including objects other than US ASCII text mail messages within the bodies of messages conforming to RFC 2822. These are described in RFC 2045.

 General structure of the MIME media typing system, which defines an initial set of media types. This is described in RFC 2046.

 A protocol for encoding non-U.S. ASCII text in the header fields of mail messages conforming to RFC 2822. This is described in RFC 2047.

 Various IANA registration procedures for MIME-related facilities. This is described in RFC 2048.

 MIME conformance criteria. This is described in RFC 20410.

1. **Distinguish IPV4 and IPV6 addressing schemes**

**Ans.** **Comparing IPV4 and IPV6 Addressing**

IPV4 addresses and addressing concepts and their IPV6 equivalents.

|  |  |
| --- | --- |
| IPV4 addresses | IPV6 addresses |
| Internet address classes | Not applicable in IPV6 |
| Broadcast address: Network Broadcast, Subnet Broadcast, all Subnet directed Broadcast, Limited Broadcast | Not applicable in IPV6 |
| IPV Multicast addresses(224.0.0.0/4) | IPV Multicast addresses(ff00::/8) |
| Unspecified address is 0.0.0.0 | Unspecified address is:: |
| Loopback address is 127.0.0.1 | Loopback address is::1 |
| Public IPV4 Addresses | Global unicast addresses |
| Private IPV4 addresses (10.0.0/80172.16.0.0/12, and 192.168.0.0/16) | Site-local addresses(FEC0::/10) |
| APIPA addresses(169.254.0.0/16) | Link-local addresses(FE80::/64) |
| Address syntax : clotted decimal notation | Address syntax : colon hexadecimal format with suppression of leading zero and zero compression. Embedded IPV4 addresses are expressed in clotted decimal notation |
| Address prefix syntax: prefix length of clotted decimal(subnet mask)notation | Address prefix syntax: prefix length notation only |