OSI MODEL

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 **Open Systems Interconnection (OSI)** model. An **open system** is a model that allows any two different systems to communicate regardless of their underlying architecture. Vendor-specific protocols close off communication between unrelated systems. The purpose of the OSI model is to open communication between different systems without requiring changes to the logic of the 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.

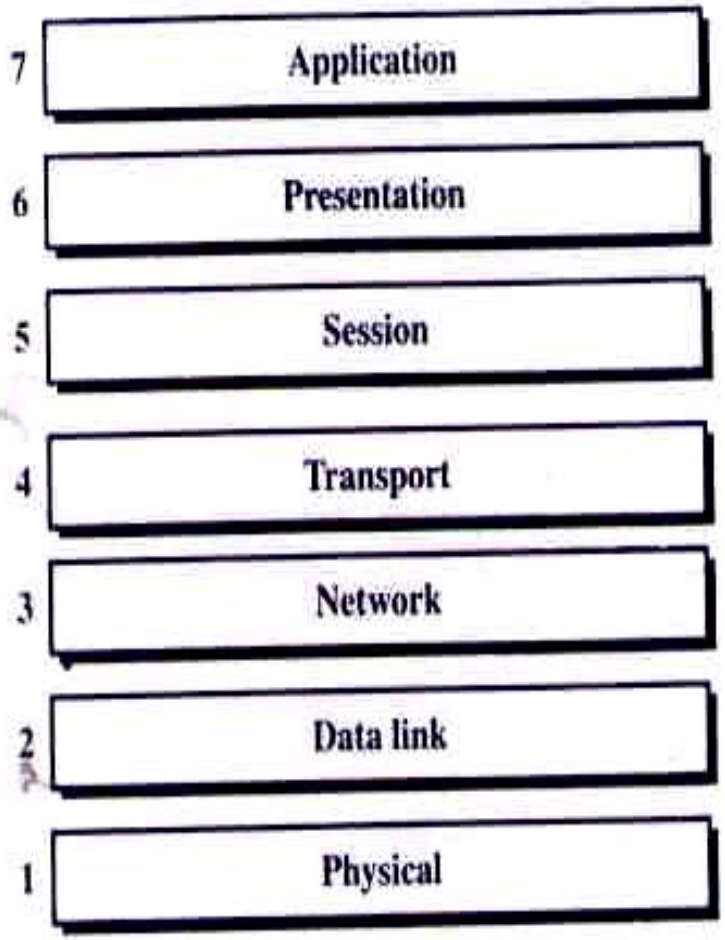
3.1 THE MODEL

The Open Systems Interconnection model is a layered framework for the design of network systems that allows for communication across all types of computer systems. It consists of seven separate but related layers, each of which defines a segment of the process of moving information across a network (see Figure 3.1). Understanding the fundamentals of the OSI model provides a solid basis for exploration of data communication.

Layered Architecture

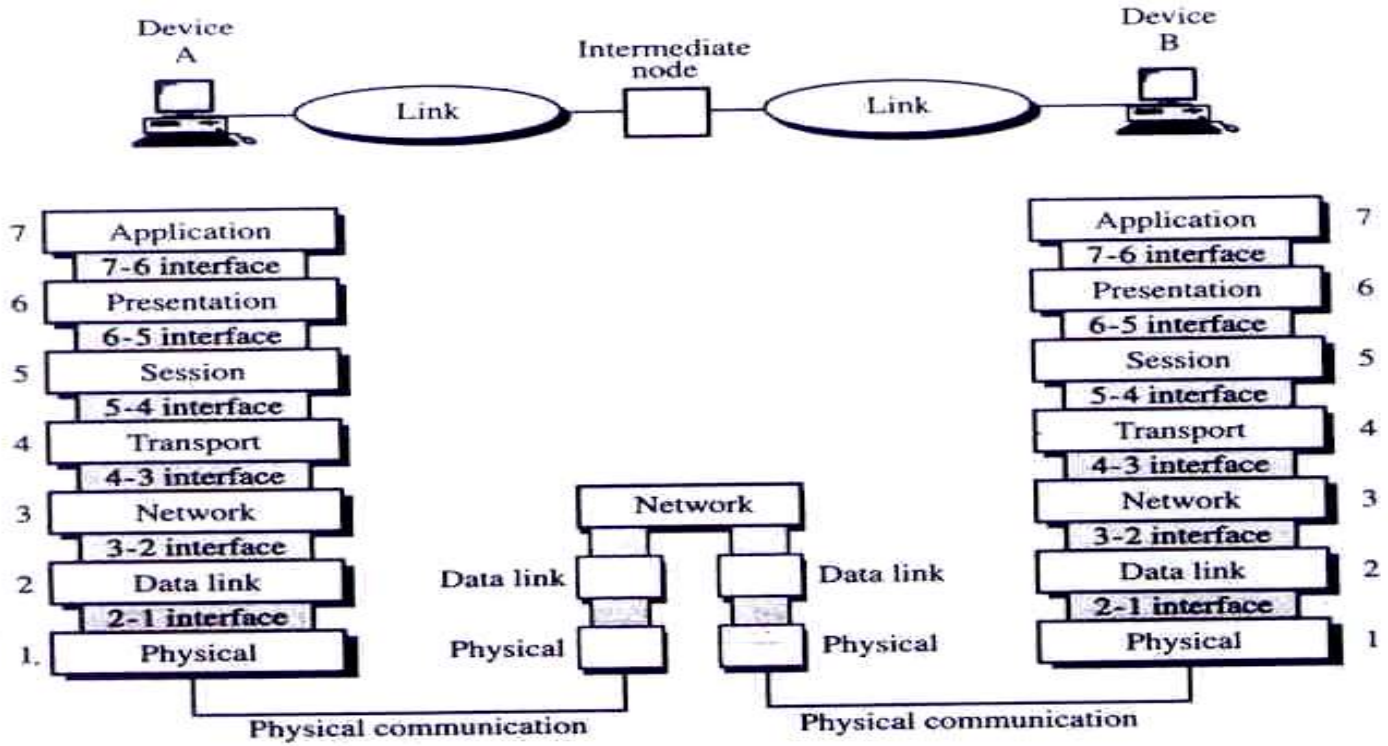
The OSI model is built of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7). Figure 3.2 shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model. In developing the model, the designers distilled the process of transmitting data down to its most fundamental elements. They identified which networking

Functions had related uses and collected those functions into discrete groups that became the layers. Each layer defines a family of functions distinct from those of the other layers. By defining and localizing functionality in this fashion, the designers



The OSI Model

OSI Layers


Device A
Intermediate node
Device B
Link
Link
7 Application
7-6 interface
6 Presentation
6-5 interface
5 Session
5-4 interface
4 Transport
4-3 interface
3 Network
3-2 interface
2 Data link
2-1 interface
1. Physical
Network
Data link
Physical
Data link
Physical
Physical communication
Physical communication

created an architecture that is both comprehensive and flexible. Most important, the OSI model allows complete transparency between otherwise incompatible systems.

A mnemonic for remembering the layers of the OSI model is: **Please Do Not Touch Steve's Pet Alligator** (Physical, Data Link, Network, Transport, Session, Presentation, Application).

Peer-to-Peer Processes

Within a single machine, each layer calls upon the services of the layer just below it. Layer 3, for example, uses the services provided by layer 2 and provides services for layer 4. Between machines, layer *x* on one machine communicates with layer *x* on another machine. This communication is governed by an agreed-upon series of rules and conventions called protocols. The processes on each machine that communicate at a given layer are called **peer-to-peer processes.** Communication between machines is therefore a peer-to-peer process using the protocols appropriate to a given layer.

At the physical layer, communication is direct: Machine A sends a stream of bits to machine B. At the higher layers, however, communication must move down through the layers on machine A, over to machine B, and then back up through the layers. Each layer in the sending machine adds its own information to the message it receives from the layer just above it and passes the whole package to the layer just below it. This information is added in the form of **headers** or **trailers** (control data added to the beginning or end of a data parcel). Headers are added to the message at layers 6, 5, 4, 3, and 2. A trailer is added at layer 2.

Headers are added to the data at layers 6, 5, 4, 3, and 2. Trailers are usually added only at layer 2.

At layer 1 the entire package is converted to a form that can be transferred to the receiving machine. At the receiving machine, the message is unwrapped layer by layer, with each process receiving and removing the data meant for it. For example, layer 2 removes the data meant for it, then passes the rest to layer 3. Layer 3 removes the data meant for it and passes the rest to layer 4, and so on.

Interfaces between Layers

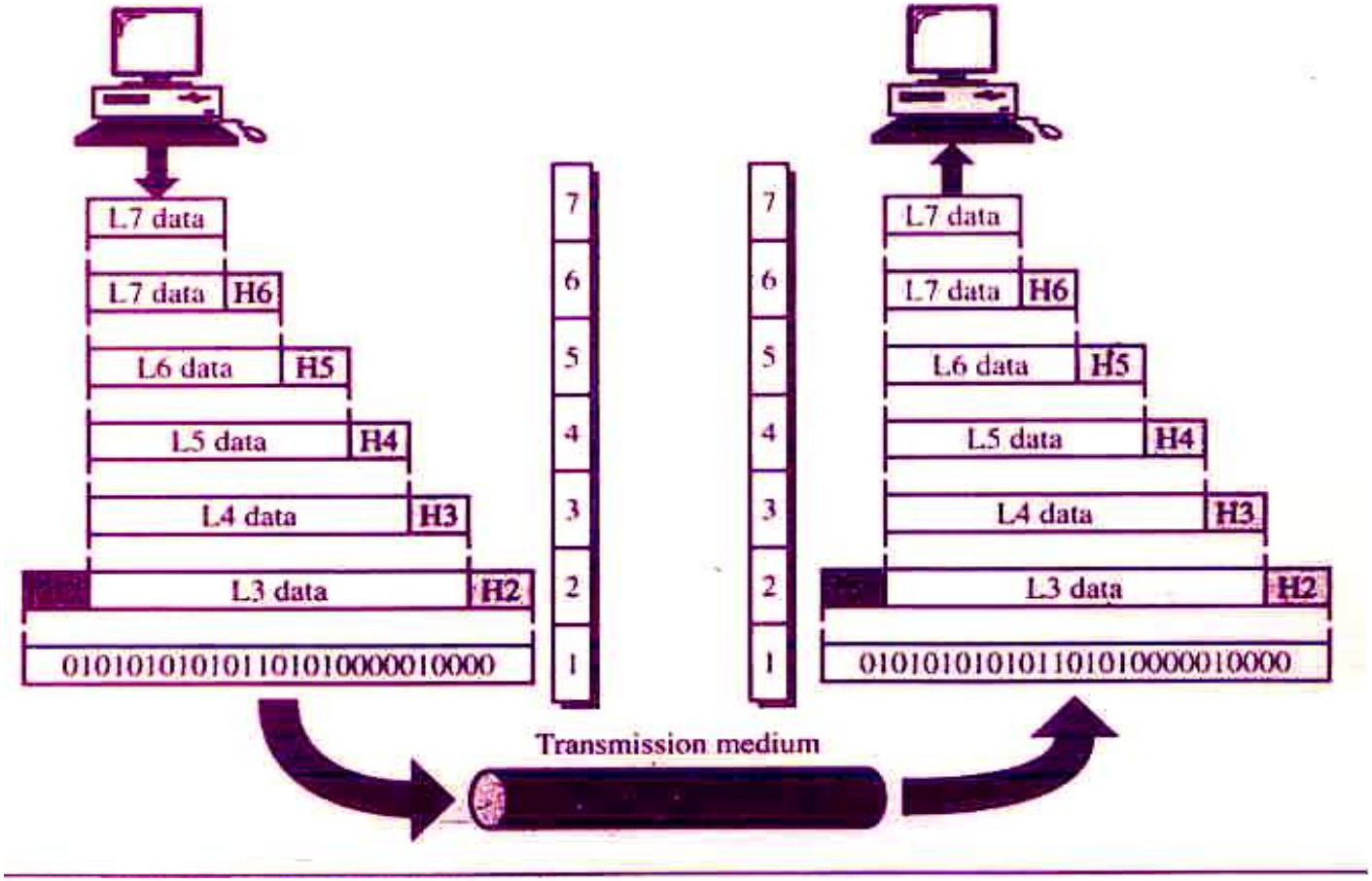
The passing of the data and network information down through the layers of the sending machine and back up through the layers of the receiving machine is made possible by an **interface** between each pair of adjacent layers. Each interface defines what information and services a layer must provide for the layer above it. Well-defined interfaces and layer functions provide modularity to a network. As long as a layer still provides the expected services to the layer above it, the specific implementation of its functions can be modified or replaced without requiring changes to the surrounding layers.

Organization of the Layers

The seven layers can be thought of as belonging to three subgroups. Layers 1, 2, and 3—physical, data link, and network—are the network support layers; they deal with the physical aspects of moving data from one device to another (such as electrical

specifications, physical connections, physical addressing, and transport timing and reliability). Layers 5, 6, and 7—session, presentation, and application—can be thought of as the user support layers; they allow interoperability among unrelated software systems. Layer 4, the transport layer, ensures end-to-end reliable data transmission while layer 2 ensures reliable transmission on a single link. The upper OSI layers are almost always implemented in software; lower layers are a combination of hardware and software, except for the physical layer, which is mostly hardware.

In Figure 3.3, which gives an overall view of the OSI layers, L7 data means the data unit at layer 7, L6 data means the data unit at layer 6, and so on.The process starts out at layer 7 (the application layer), then moves from layer to layer in descending sequential order. At each layer (except layers 7 and 1), a header is added to the data unit. At layer 2, a trailer is added as well. When the formatted data unit passes through the physical layer (layer 1), it is changed into an electromagnetic signal and transported along a physical link.



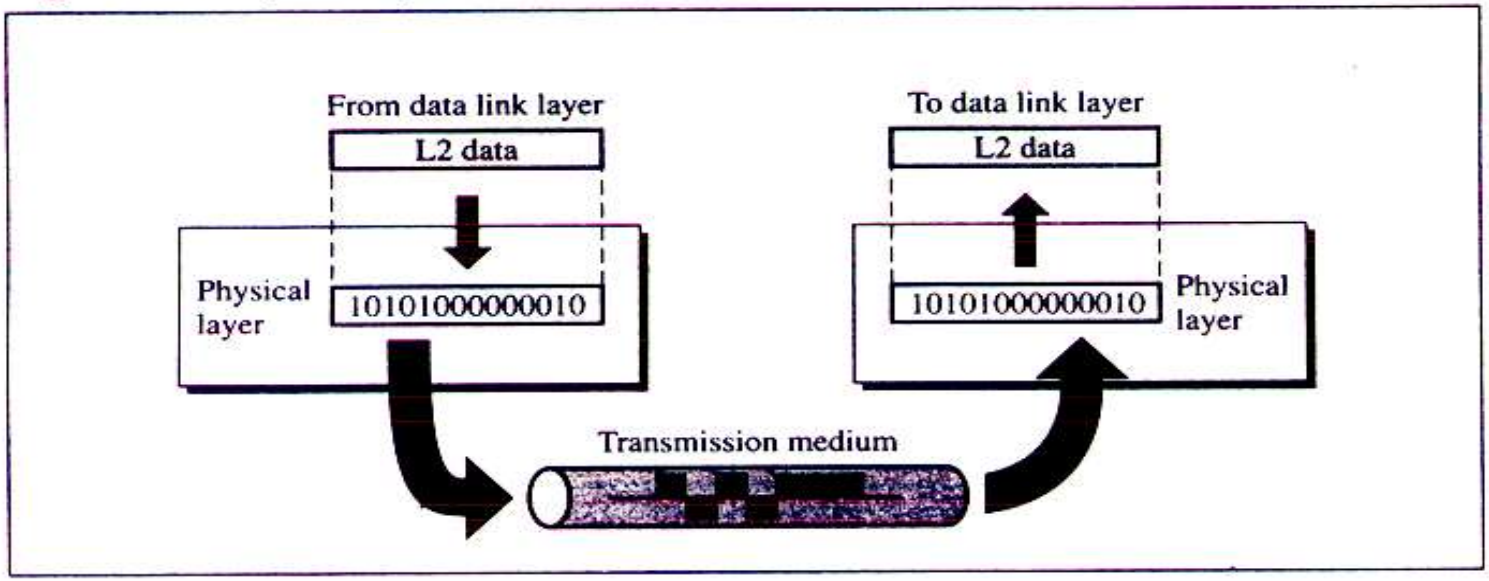
An exchange using the OSI model

Functions of the Layers :

Physical Layer

The **physical layer** coordinates the functions required to transmit a bit stream over a physical medium. It deals with the mechanical and electrical specifications of the interface and transmission medium. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur. Figure 3.4 shows the position of the physical layer with respect to the transmission medium and the data link layer.

Figure 3.4 *Physical layer*



The physical layer is concerned with the following:

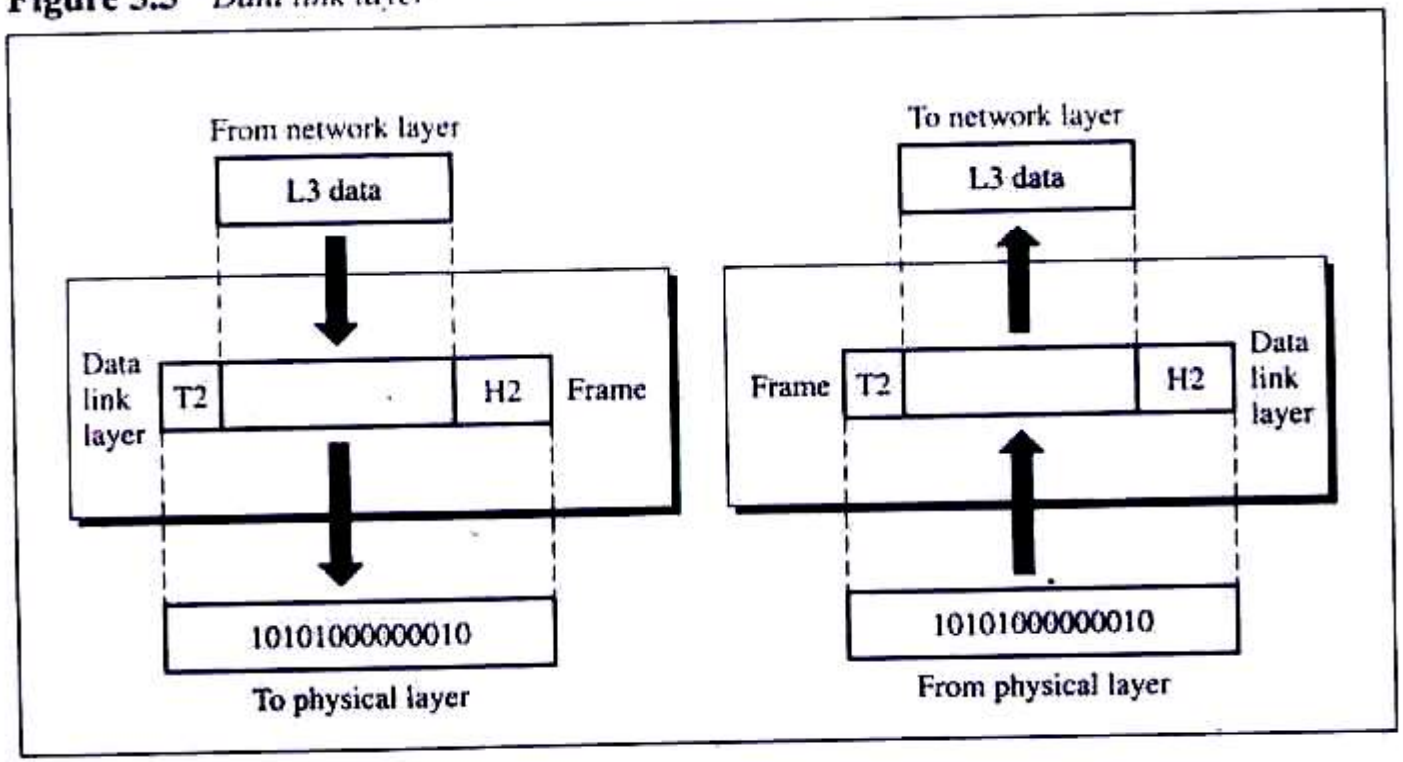
- **Physical characteristics of interfaces and media.** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium (see Chapter 7).
- **Representation of bits.** The physical layer data consist of a stream of **bits** (sequence of 0s and 1s) without any interpretation. To be transmitted, bits must be encoded into signals—electrical or optical. The physical layer defines the type of **encoding** (how 0s and 1s are changed to signals).
- **Data rate.** The **transmission rate**—the number of bits sent each second—is also defined by the physical layer. In other words, the physical layer defines the duration of a bit, which is how long it lasts.
- **Synchronization of bits.** The sender and receiver must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.
- **Line configuration.** The physical layer is concerned with the connection of devices to the medium. In a *point-to-point configuration*, two devices are connected together through a dedicated link. In a *multipoint configuration*, a link is shared between several devices.

- **Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected using a *mesh topology* (every device connected to every other device), a *star topology* (devices are connected through a central device), a *ring topology* (every device is connected to the next, forming a ring), or a *bus topology* (every device on a common link).
- **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In the *simplex mode*, only one device can send; the other can only receive. The simplex mode is a one-way communication. In the *half-duplex mode*, two devices can send and receive, but not at the same time. In a *full-duplex* (or simply duplex) *mode*, two devices can send and receive at the same time.

Data Link Layer

The **data iink layer** transforms the physical layer, a raw transmission facility, to a reliable link and is responsible for **node-to-node** delivery. It makes the physical layer appear error free to the upper layer (network layer). Figure 3.5 shows the relationship of the data link layer to the network and physical layers.

Figure 3.5 *Data link layer*



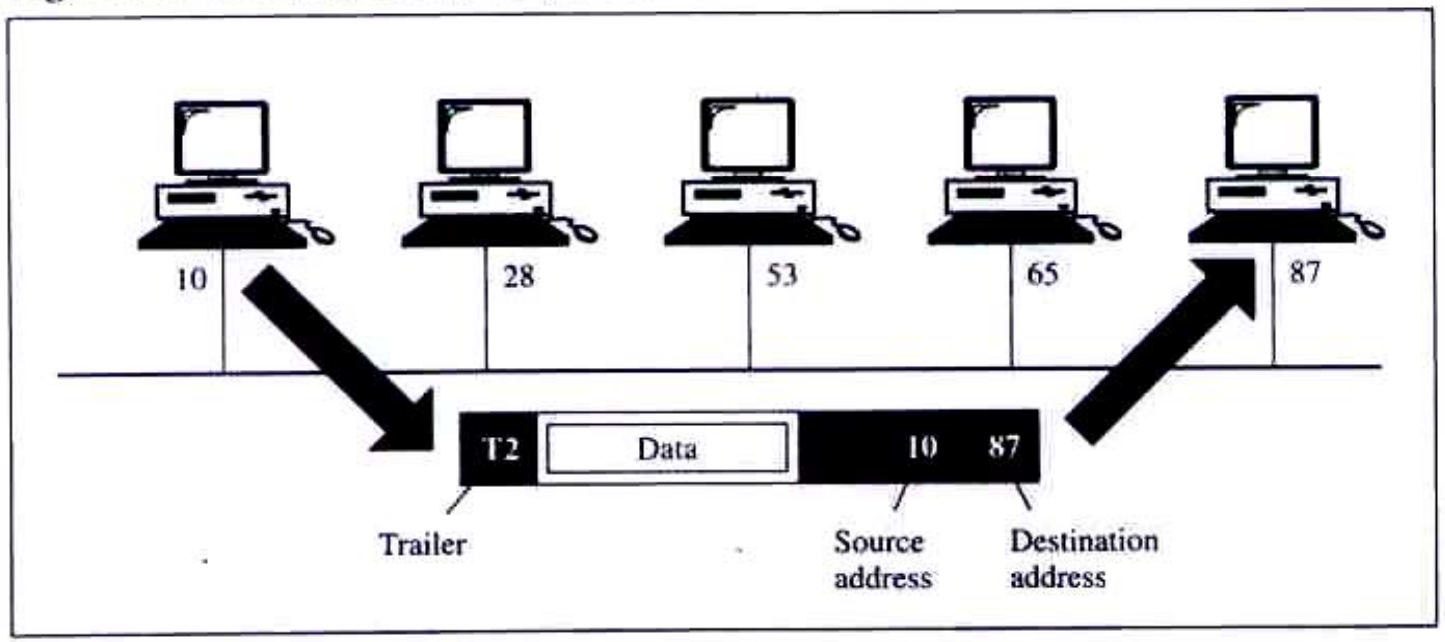
Specific responsibilities of the data link layer include the following:

- **Framing.** The data link layer divides the stream of bits received from the network layer into manageable data units called **frames.**
- **Physical addressing.** If frames are to be distributed to different systems on the network, the data link layer adds a header to the frame to define the **physical address** of the sender **(source address)** and/or receiver **(destination address)** of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the device that connects one network to the next.
- **Flow control.** If the rate at which the data are absorbed by the receiver is less than the rate produced in the sender, the data link layer imposes a flow control mechanism to prevent overwhelming the receiver.
- **Error control.** The data link layer adds reliability to the physical layer by adding mechanisms to detect and retransmit damaged or lost frames. It also uses a mechanism to prevent duplication of frames. Error control is normally achieved through a trailer added to the end of the frame.
- **Access control.** When two or more devices are connected to the same link, data link layer protocols are necessary to determine which device has control over the link at any given time.

Example 3.1

In Figure 3.6 a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link. At the data link level this frame contains physical (link) addresses in the header. These are the only addresses needed. The rest of the header contains other information needed at this level. The trailer usually contains extra bits needed for error detection.

Figure 3.6 *Data link layer (Example 3.1)*

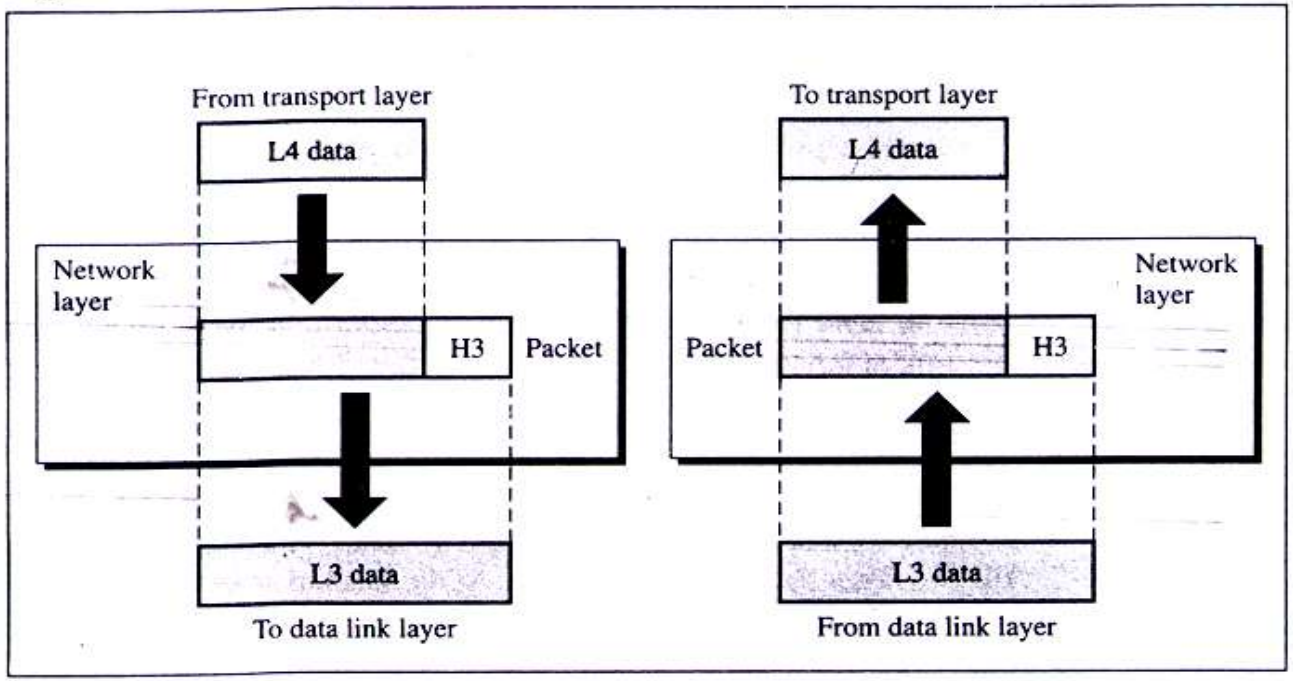


Network Layer

The **network layer** is responsible for the source-to-destination delivery of a packet possibly across multiple networks (links). Whereas the data link layer oversees the delivery of the packet between two systems on the same network (links), the network layer ensures that each packet gets from its point of origin to its final destination.

If two systems are connected to the same link, there is usually no need for a network layer. However, if the two systems are attached to different networks (links) with connecting devices between the networks (links), there is often a need for the network layer to accomplish source-to-destination delivery. Figure 3.7 shows the relationship of the network layer to the data link and transport layers.

Figure 3.7 *Network layer*

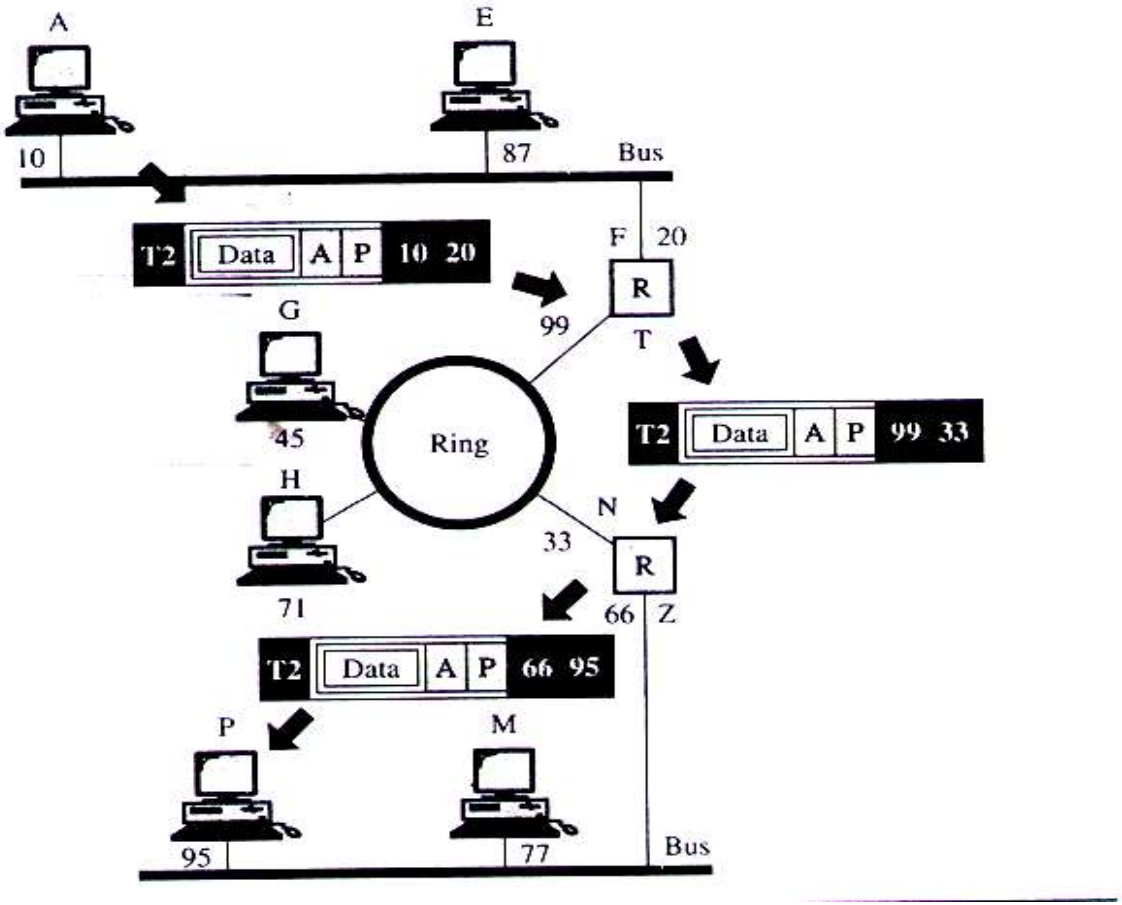


Specific responsibilities of the network layer include the following:

- **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary, we need another addressing system to help distinguish the source and destination systems. The network layer adds a header to the packet coming from the upper layer that, among other things, includes the **logical addresses** of the sender and receiver.
- **Routing.** When independent networks or links are connected together to create an *internetwork* (a network of networks) or a large network, the connecting devices (called *routers* or *gateways*) route the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Example 3.2

Now imagine that in Figure 3.8 we want to send data from a node with network address A and physical address 10, located on one local area network, to a node with a network address P and physical address 95, located on another local area network. Because the two devices are located on different networks, we cannot use physical addresses only; the physical addresses have only local jurisdiction. What we need here are universal addresses that can pass through the boundaries of local area networks. The network (logical) addresses have this characteristic. The packet at the network layer contains the logical addresses, which remain the same from the original source to the final destination (A and P, respectively, in the figure). They will not change when we go from network to network. However, the physical addresses will change when the packet moves from one network to another. The box with the R is a router (internetwork device), which we will discuss in Chapter 21.



Network Layer example

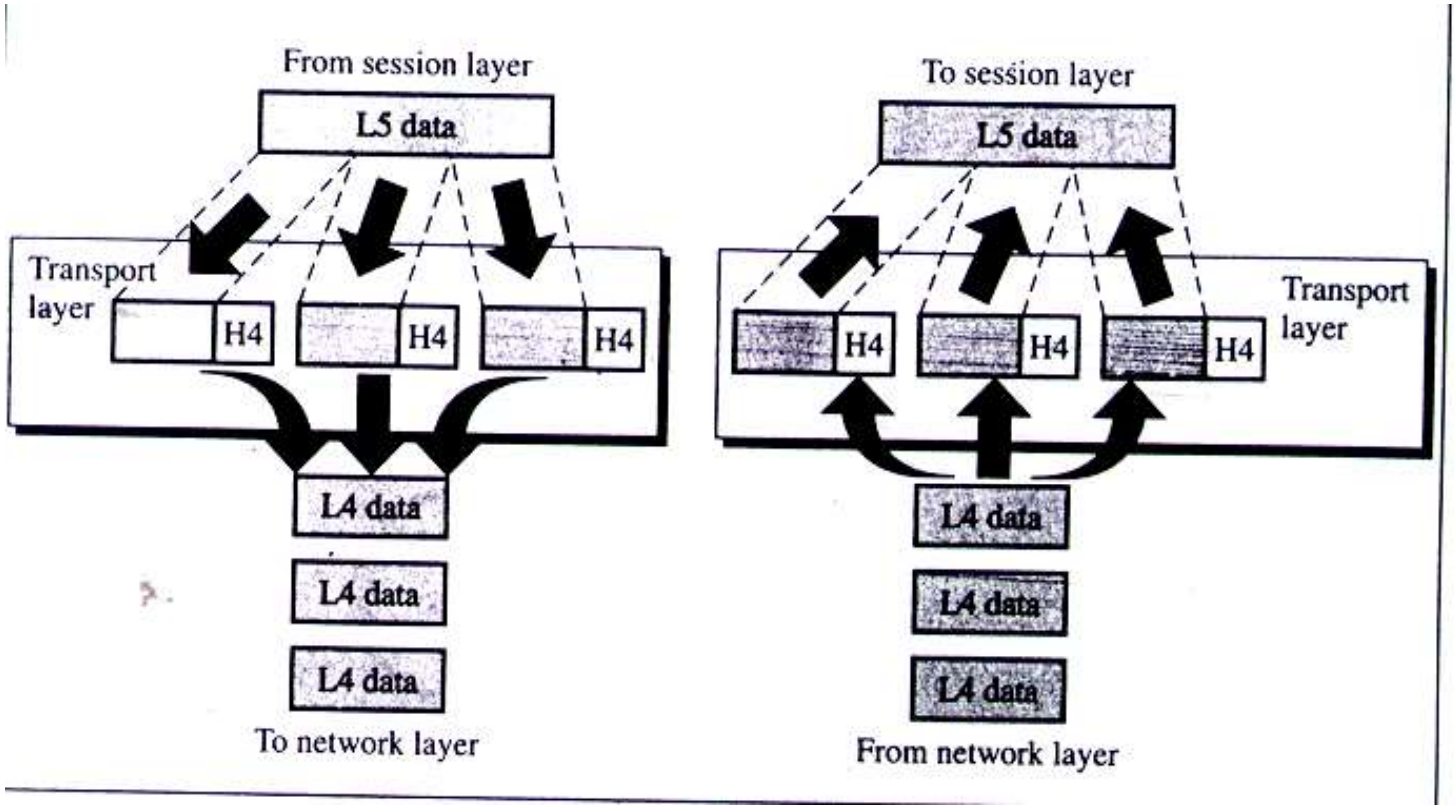
Transport Layer

The **transport layer** is responsible for **source-to-destination** (end-to-end) **delivery** of the entire message. Whereas the network layer oversees end-to-end delivery of individual packets, it does not recognize any relationship between those packets. It treats each one independently, as though each piece belonged to a separate message, whether or not it does. The transport layer, on the other hand, ensures that the whole message arrives intact and in order, overseeing both error control and flow control at the source-to-destination level. Figure 3.9 shows the relationship of the transport layer to the network and session layers.

For added security, the transport layer may create a *connection* between the two end ports. A connection is a single logical path between the source and destination that is associated with all packets in a message. Creating a connection involves three steps: connection establishment, data transfer, and connection release. By confining transmission of all packets to a single pathway, the transport layer has more control over sequencing, flow, and error detection and correction.

Specific responsibilities of the transport layer include the following:

- **Service-point addressing.** Computers often run several programs at the same time. For this reason, source-to-destination delivery means delivery not only from one computer to the next but also from a specific process (running program) on one computer to a specific process (running program) on the other. The transport layer header therefore must include a type of address called a *service-point address* (or **port address**). The network layer gets each packet to the correct computer; the transport layer gets the entire message to the correct process on that computer.



TRANSPORT LAYER

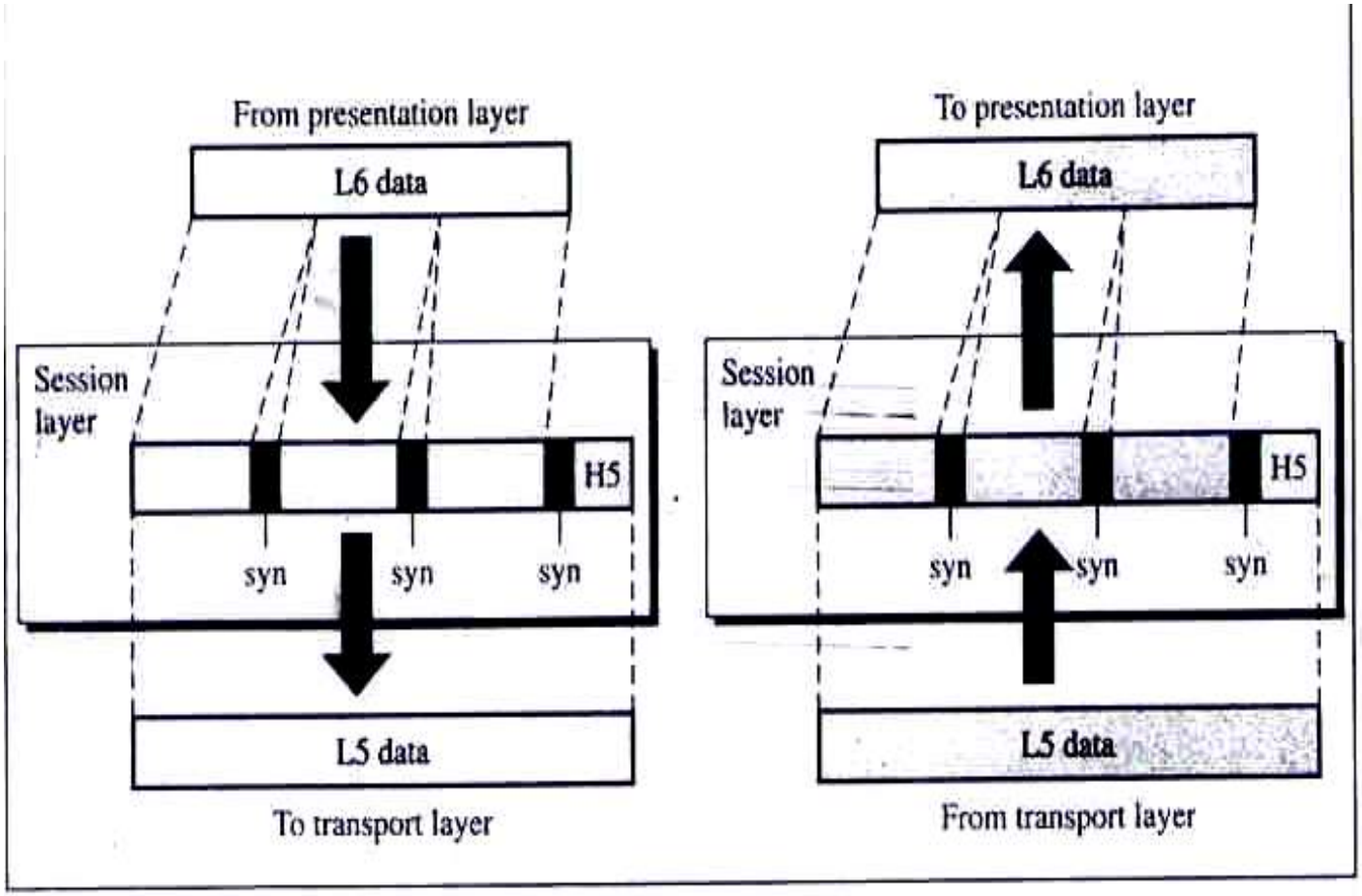
- **Segmentation and reassembly.** A message is divided into transmittable segments, each segment containing a sequence number. These numbers enable the transport layer to reassemble the message correctly upon arriving at the destination and to identify and replace packets that were lost in the transmission.
- **Connection control.** The transport layer can be either connectionless or connection-oriented. A connectionless transport layer treats each segment as an independent packet and delivers it to the transport layer at the destination machine. A connection-oriented transport layer makes a connection with the transport layer at the destination machine first before delivering the packets. After all the data are transferred, the connection is terminated.
- **Flow control.** Like the data link layer, the transport layer is responsible for flow control. However, flow control at this layer is performed end to end rather than across a single link.
- **Error control.** Like the data link layer, the transport layer is responsible for error control. However, error control at this layer is performed end to end rather than across a single link. The sending transport layer makes sure that the entire message arrives at the receiving transport layer without **error** (damage, loss, or duplication). Error correction is usually achieved through retransmission.

Session Layer

The services provided by the first three layers (physical, data link, and network) are not sufficient for some processes. The **session layer** is the network *dialog controller.* It establishes, maintains, and synchronizes the interaction between communicating systems.

Specific responsibilities of the session layer include the following:

- **Dialog control.** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place either in half-duplex (one way at a time) or full-duplex (two ways at a time). For example, the dialog between a terminal connected to a mainframe can be half-duplex.
- **Synchronization.** The session layer allows a process to add checkpoints (synchronization points) into a stream of data. For example, if a system is sending a file of 2000 pages, it is advisable to insert checkpoints after every 100 pages to ensure that each 100-page unit is received and acknowledged independently. In this case, if a crash happens during the transmission of page 523, retransmission begins at page 501: pages 1 to 500 need not be retransmitted. Figure 3.11 illustrates the relationship of the session layer to the transport and presentation layers.

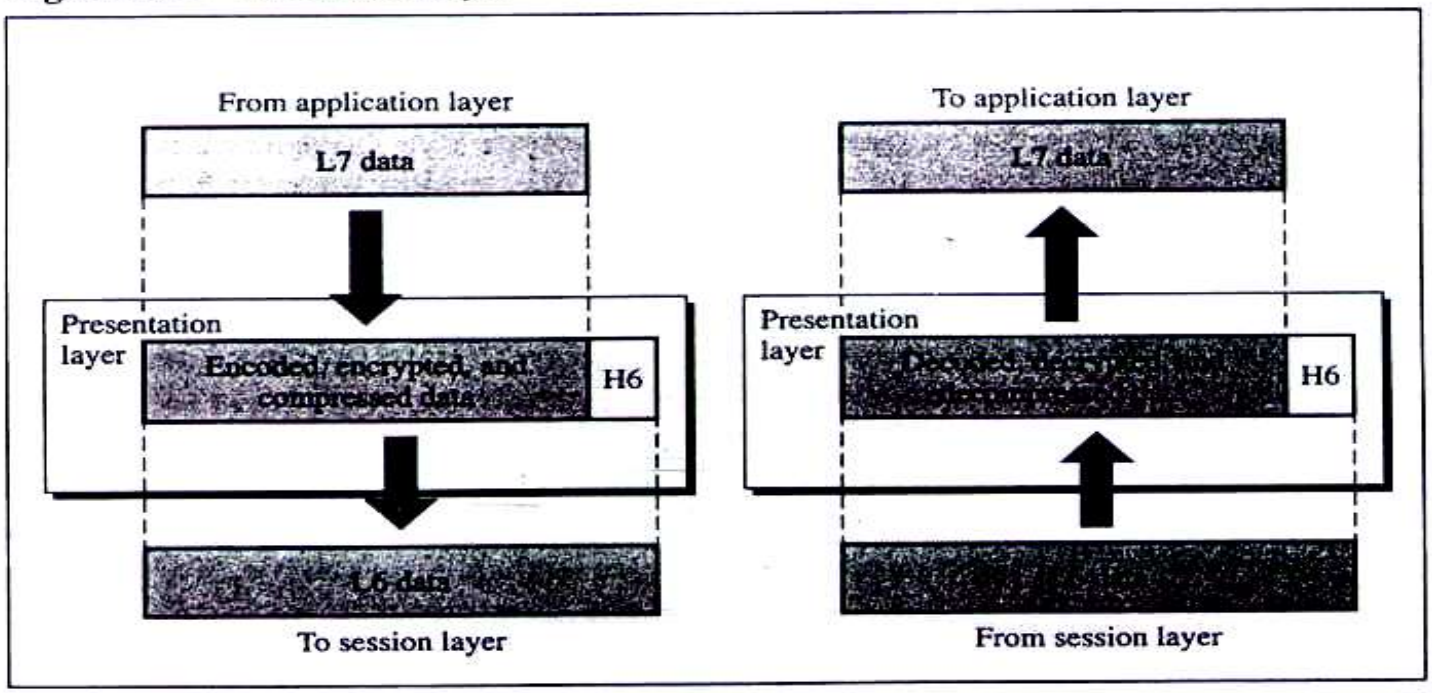


SESSION LAYER

Presentation Layer

The **presentation layer** is concerned with the syntax and semantics of the information exchanged between two systems. Figure 3.12 shows the relationship between the presentation layer and the application and session layers.

Figure 3.12 *Presentation layer*



Specific responsibilities of the presentation layer include the following:

- **Translation.** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The

information should be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation layer is responsible for interoperability between these different encoding methods. The presentation layer at the sender changes the information from its sender-dependent format into a common format. The presentation layer at the receiving machine changes the common format into its receiver-dependent format.

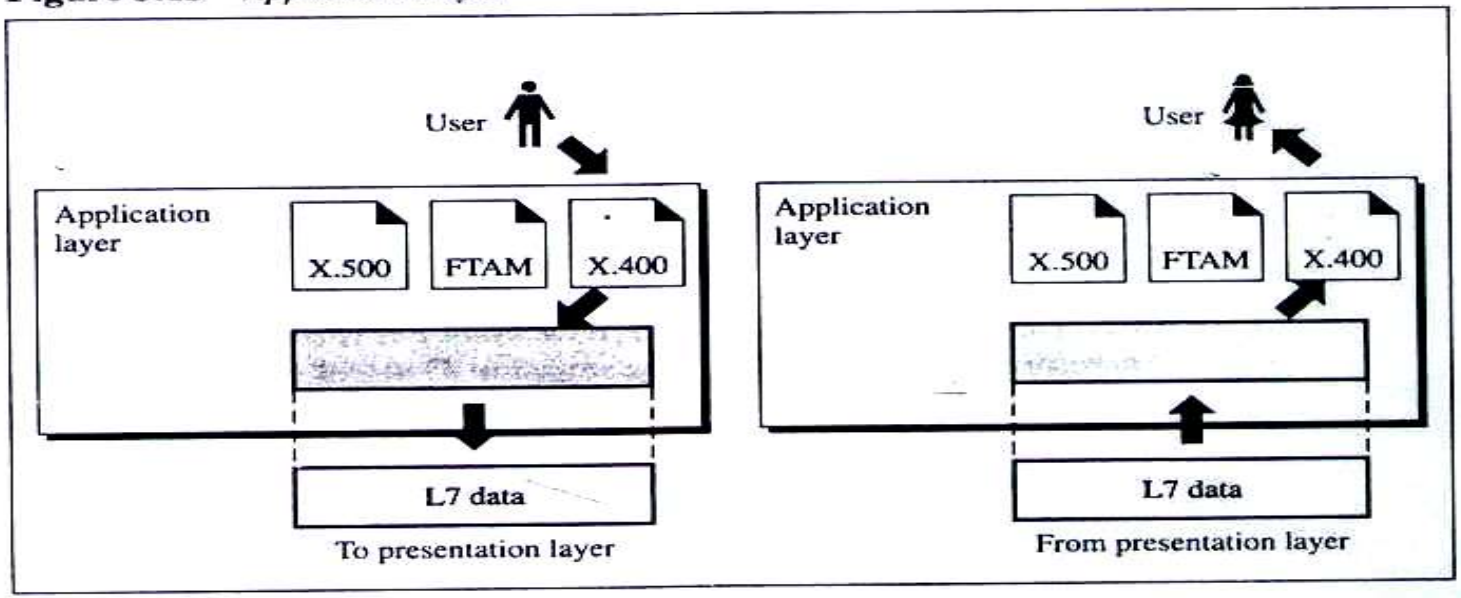
- **Encryption.** To carry sensitive information, a system must be able to assure privacy. Encryption means that the sender transforms the original information to another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- **Compression.** Data compression reduces the number of bits to be transmitted. Data compression becomes particularly important in the transmission of multimedia such as text, audio, and video.

Application Layer

The **application layer** enables the user, whether human or software, to access the network. It provides user interfaces and support for services such as electronic mail, remote file access and transfer, shared database management, and other types of distributed information services.

Figure 3.13 shows the relationship of the application layer to the user and the presentation layer. Of the many application services available, the figure shows only three: X.400 (message-handling services); X.500 (directory services); and file transfer, access, and management (FTAM). The user in this example uses X.400 to send an e-mail message. Note that no headers or trailers are added at this layer.

Figure 3.13 *Application layer*



Specific services provided by the application layer include the following:

- **Network virtual terminal.** A network virtual terminal is a software version of a physical terminal and allows a user to log on to a remote host. To do so, the appli-

cation creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal, which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows you to log on.

- **File transfer, access, and management (FTAM).** This application allows a user to access files in a remote computer (to make changes or read data), to retrieve files from a remote computer; and to manage or control files in a remote computer.
- **Mail services.** This application provides the basis for e-mail forwarding and storage.
- **Directory services.** This application provides distributed database sources and access for global information about various objects and services.

Summary of Layer Functions

The functions of the seven layers are summarized in Figure 3.14.

Figure 3.14 *Summary of layer functions*