the sender and the receiver is done by the carrier. Something which is not obvious immediately is that the tasks must be done in the order given in the hierarchy. At the sender site, the letter must be written and dropped in the mailbox before being picked up by the letter carrier and delivered to the post office. At the receiver site, the letter must be dropped in the recipient mailbox before being picked up and read by the recipient.

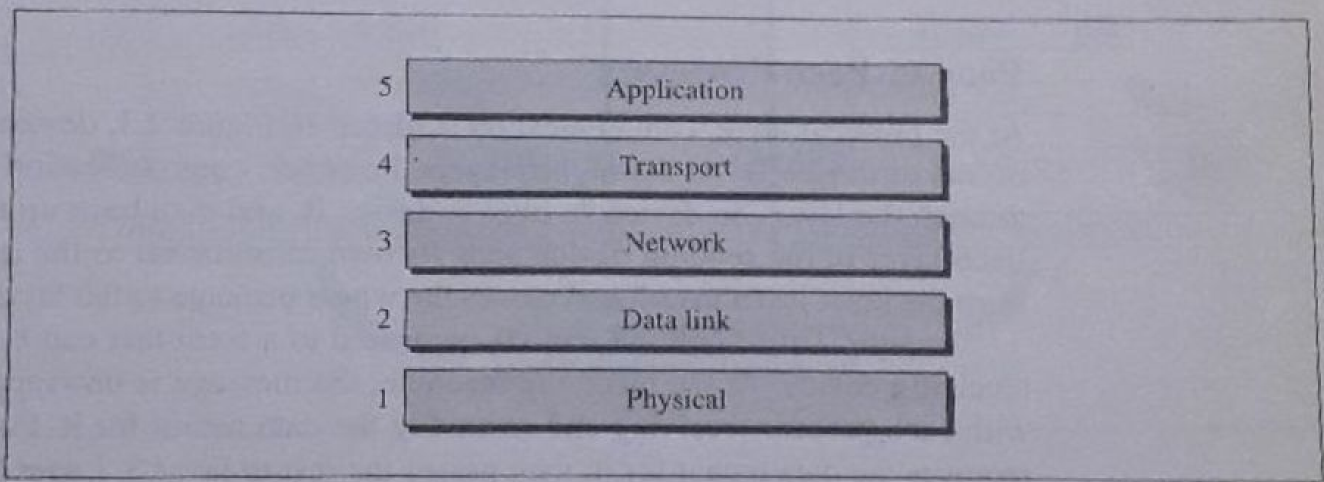
Services

Each layer at the sending site uses the services of the layer immediately below it. The sender at the higher layer uses the services of the middle layer. The middle layer uses the services of the lower layer. The lower layer uses the services of the carrier.

2.2 INTERNET MODEL

The layered protocol stack that dominates data communications and networking today is the five-layer **Internet model,** sometimes called the **TCP/IP protocol suite** (see Fig. 2.2). The model is composed of five ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), and application (layer 5). Figure 2.3 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 model.

Figure 2.2 *Internet layers*



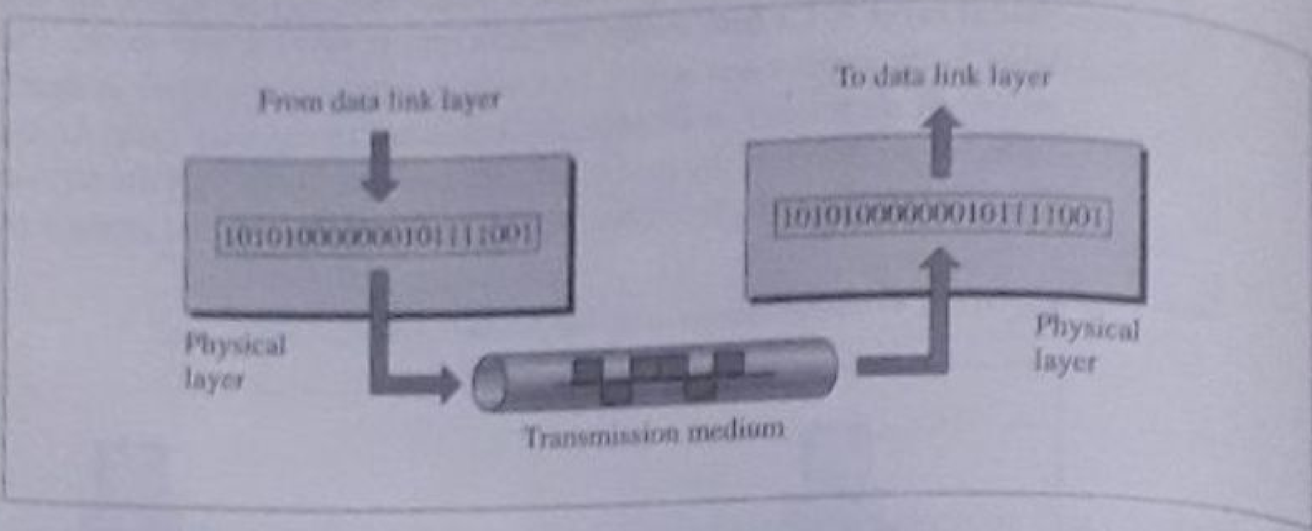
In developing the model, the designers distilled the process of transmitting data to its most fundamental elements. They identified which networking functions had related uses and collected those functions into discrete groups that became the layers. Each layer defines a family of functions distinct from those of the other layers. By defining and localizing functionality in this fashion, the designers created an architecture that is both comprehensive and flexible.

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

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 media. It also defines the procedures and functions that physical devices and interfaces have to perform for transmission to occur. Figure 2.5 shows the position of the physical layer with respect to the transmission media and the data link layer.

Figure 2.5 *Physical layer*



The physical layer is responsible for transmitting individual bits from one node to the next.

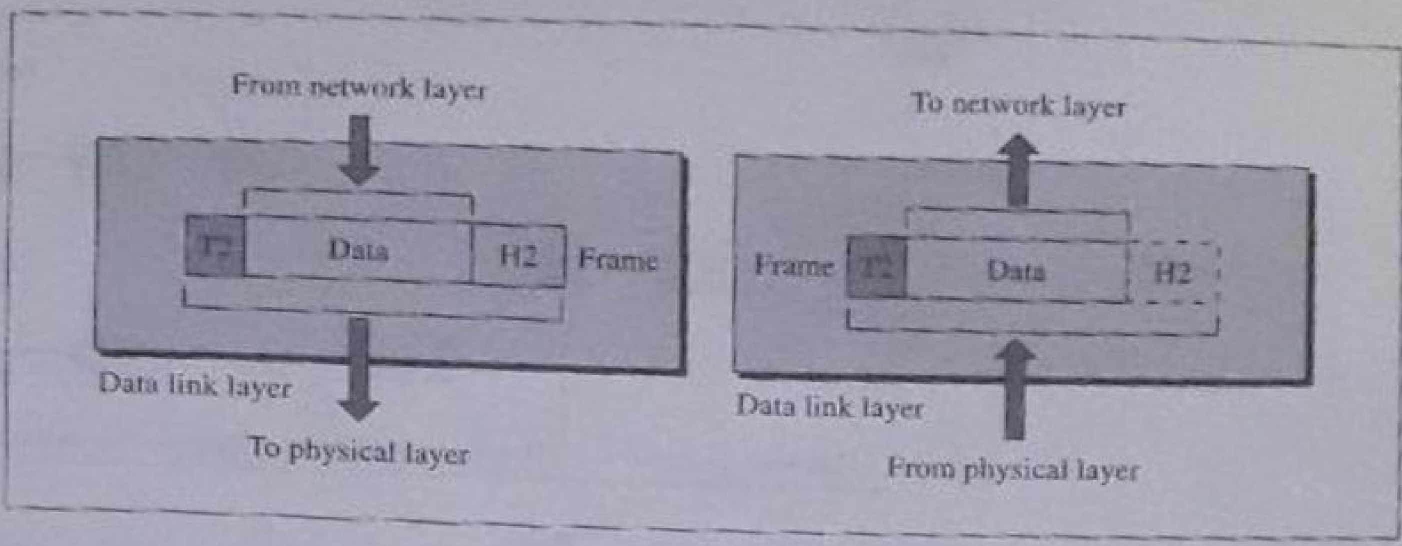
We discuss the physical layer in Part II of this book and include the dominant protocols designed for this layer. The major duties of the physical layer are as follows:

- **Physical characteristics of interfaces and media.** The physical layer defines the characteristics of the interface between the devices and the transmission media. It also defines the type of transmission medium (see Chapter 7).
- **Representation of bits.** The physical layer data consists of a stream of **bits** (sequence of 0s or 1s) without any interpretation. To be transmitted, bits must be encoded into signals—electrical or optical. The physical layer defines the type of representation (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 not only must use the same bit rate but also must be synchronized at the bit level. In other words, the sender and the receiver clocks must be synchronized.

Data Link Layer

The **data link layer** transforms the physical layer, a raw transmission facility, to a reliable link. It makes the physical layer appear error-free to the upper layer (network layer). Figure 2.6 shows the relationship of the data link layer to the network and physical layers.

Figure 2.6 *Data link layer*



The data link layer is responsible for transmitting frames from one node to the next.

We discuss the data link layer in Part III of this book and include the dominant protocols designed for this layer. The major duties of the data link layer are as follows:

- **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 sender and/or receiver of the frame. If the frame is intended for a system outside the sender's network, the receiver address is the address of the connecting device that connects the network to the next one.
- **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.

Figure 2.7 illustrates **hop-to-hop (node-to-node) delivery** by the data link layer.

Example 1

In Figure 2.8 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 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 2.7 *Node-to-node delivery*

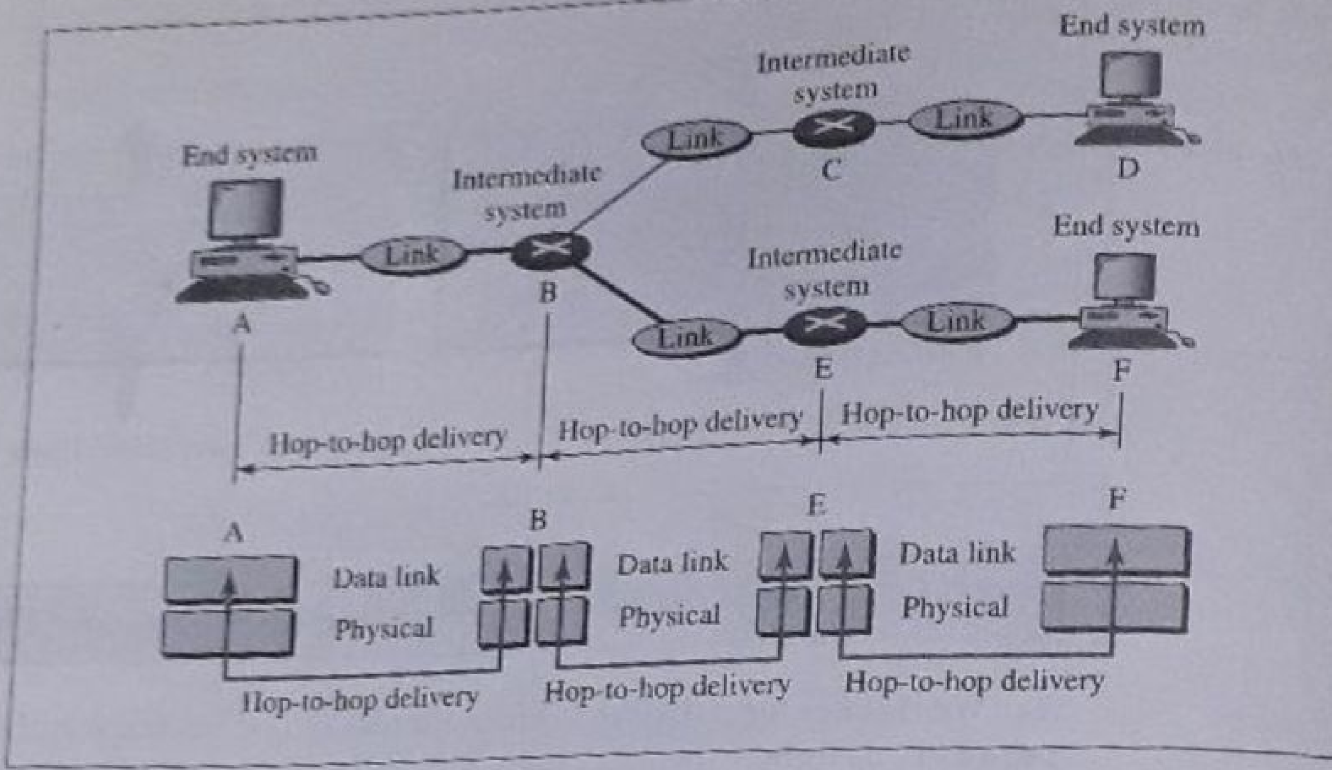
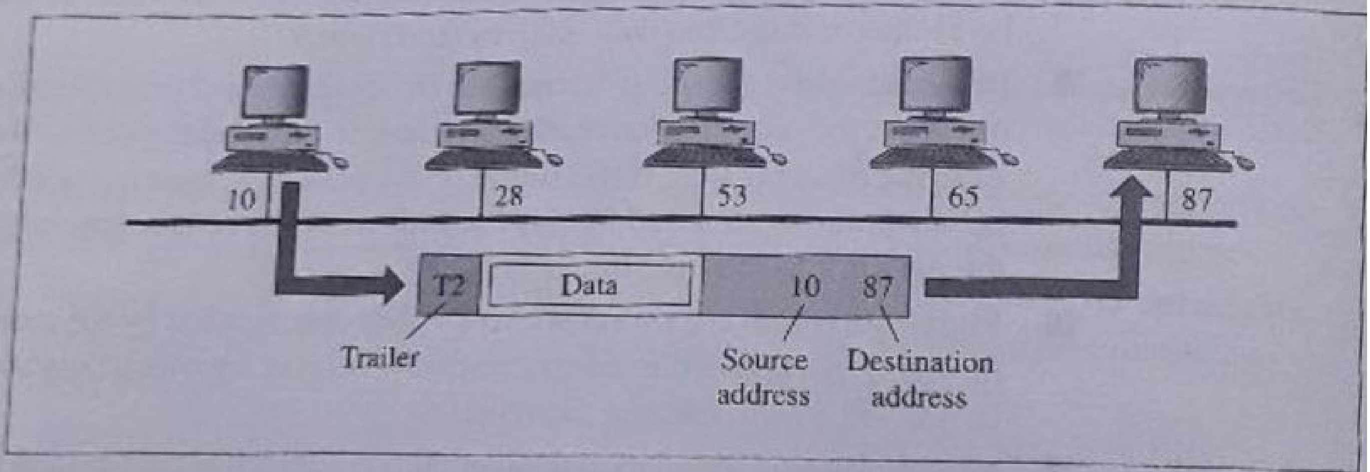


Figure 2.8 *Example 1*



Network Layer

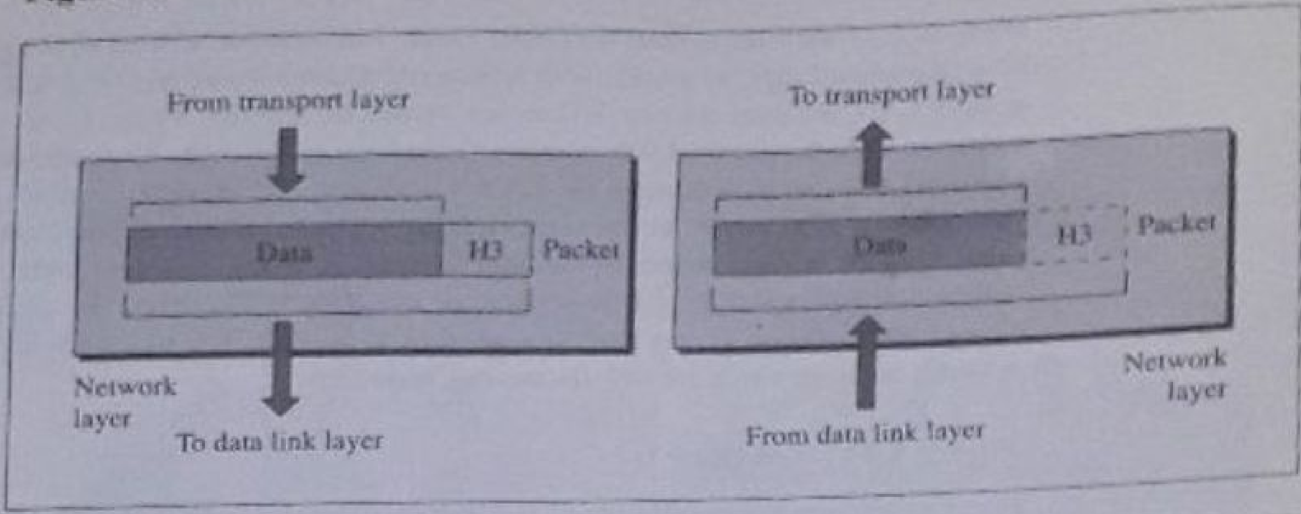
The **network layer** is responsible for the **source-to-destination delivery** of a packet possibly across multiple networks. Whereas the data link layer oversees the delivery of the packet between two systems on the same network, the network layer ensures that each packet gets from its point of origin 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 with connecting devices between the networks, there is often a need for the network layer to accomplish source-to-destination delivery. Figure 2.9 shows the relationship of the network layer to the data link and transport layers.

We discuss the network layer in Part IV of this book and include the dominant protocols designed for this layer. The major duties of the network layer are as follows:

- **Logical addressing.** The physical addressing implemented by the data link layer handles the addressing problem locally. If a packet passes the network boundary,

Figure 2.9 *Network layer*



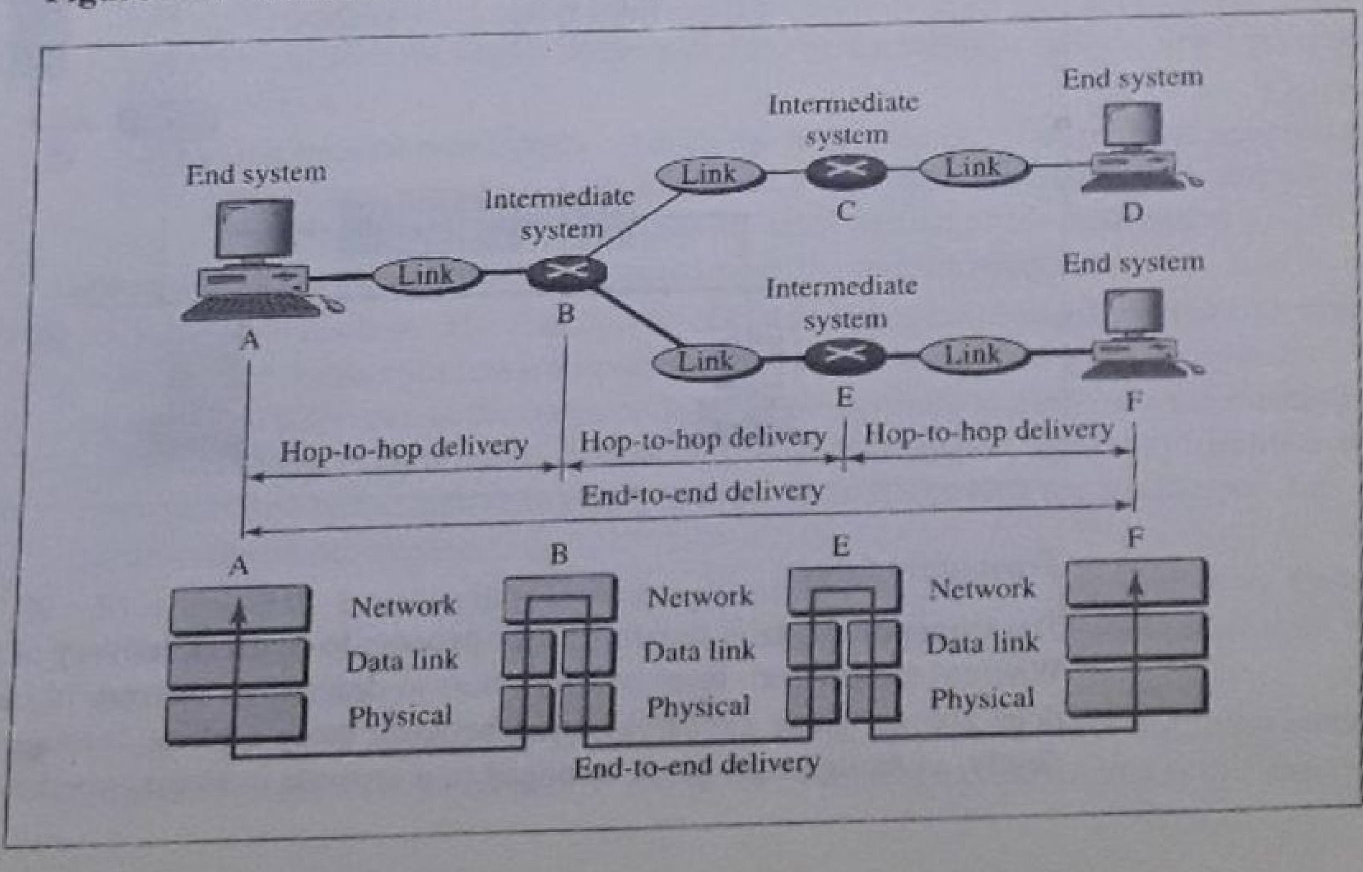
The network layer is responsible for the delivery of packets from the original source to the final destination.

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 to create an **internetwork** (network of networks) or a large network, the connecting devices (called *routers* or *switches*) route or switch the packets to their final destination. One of the functions of the network layer is to provide this mechanism.

Figure 2.10 illustrates source-to-destination delivery by the network layer.

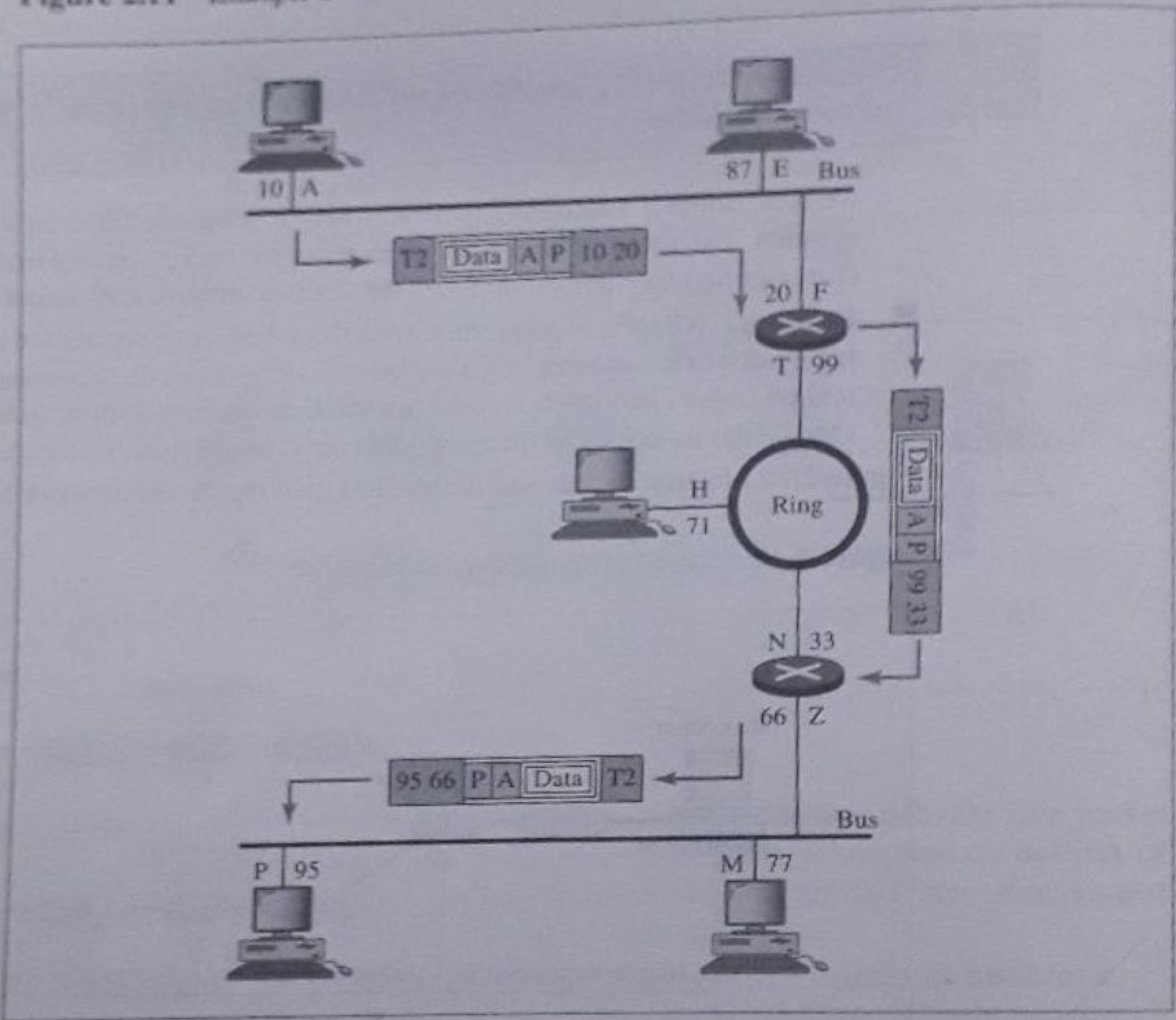
Figure 2.10 *Source-to-destination delivery*



Example 2

In Figure 2.11 we want to send data from a node with network address A and physical address 10, located on one LAN, to a node with a network address P and physical address 95, located on another LAN. Because the two devices are located on different networks, we cannot use physical addresses only; the physical addresses only have local jurisdiction. What we need here are universal addresses that can pass through the LAN boundaries. 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 as the packet moves from one network to another. The box with the R is a router (internetwork device), which we will discuss in Chapter 16.

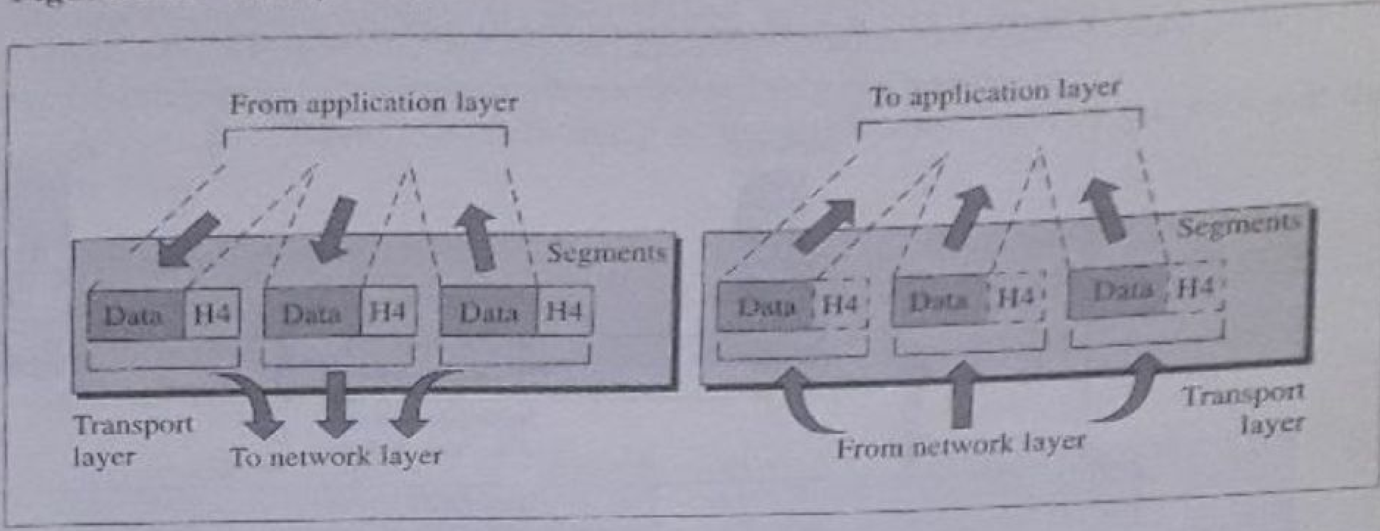
Figure 2.11 *Example 2*



Transport Layer

The **transport layer** is responsible for **process-to-process delivery** of the entire message. Whereas the network layer oversees host-to-destination 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 process-to-process level. Figure 2.12 shows the relationship of the transport layer to the network and session layers.

Figure 2.12 *Transport layer*



The transport layer is responsible for delivery of a message from one process to another.

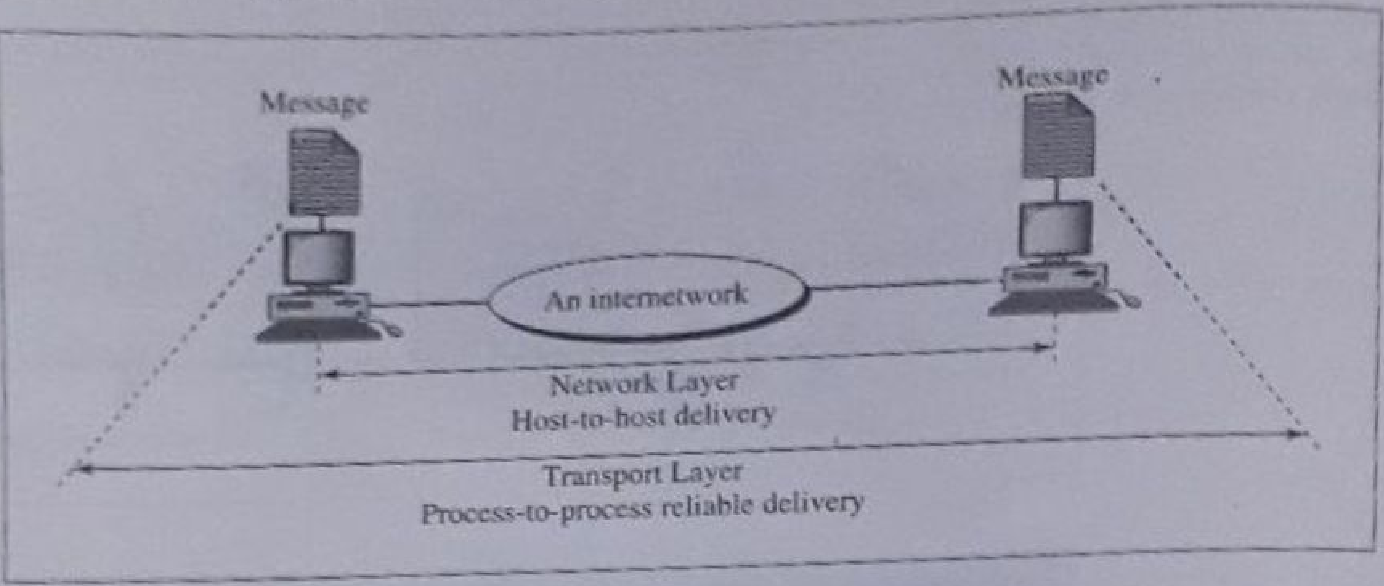
We discuss the transport layer in Part V of this book and include the dominant protocols designed for this layer. The major duties of the transport layer are as follows:

- **Port addressing.** Computers often run several processes (running programs) at the same time. For this reason, process-to-process delivery means delivery not only from one computer to the next but also from a specific process on one computer to a specific process on the other. The transport layer header must therefore include a type of address called a **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.
- **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 arrival 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.

Figure 2.13 illustrates a process-to-process delivery by the transport layer.

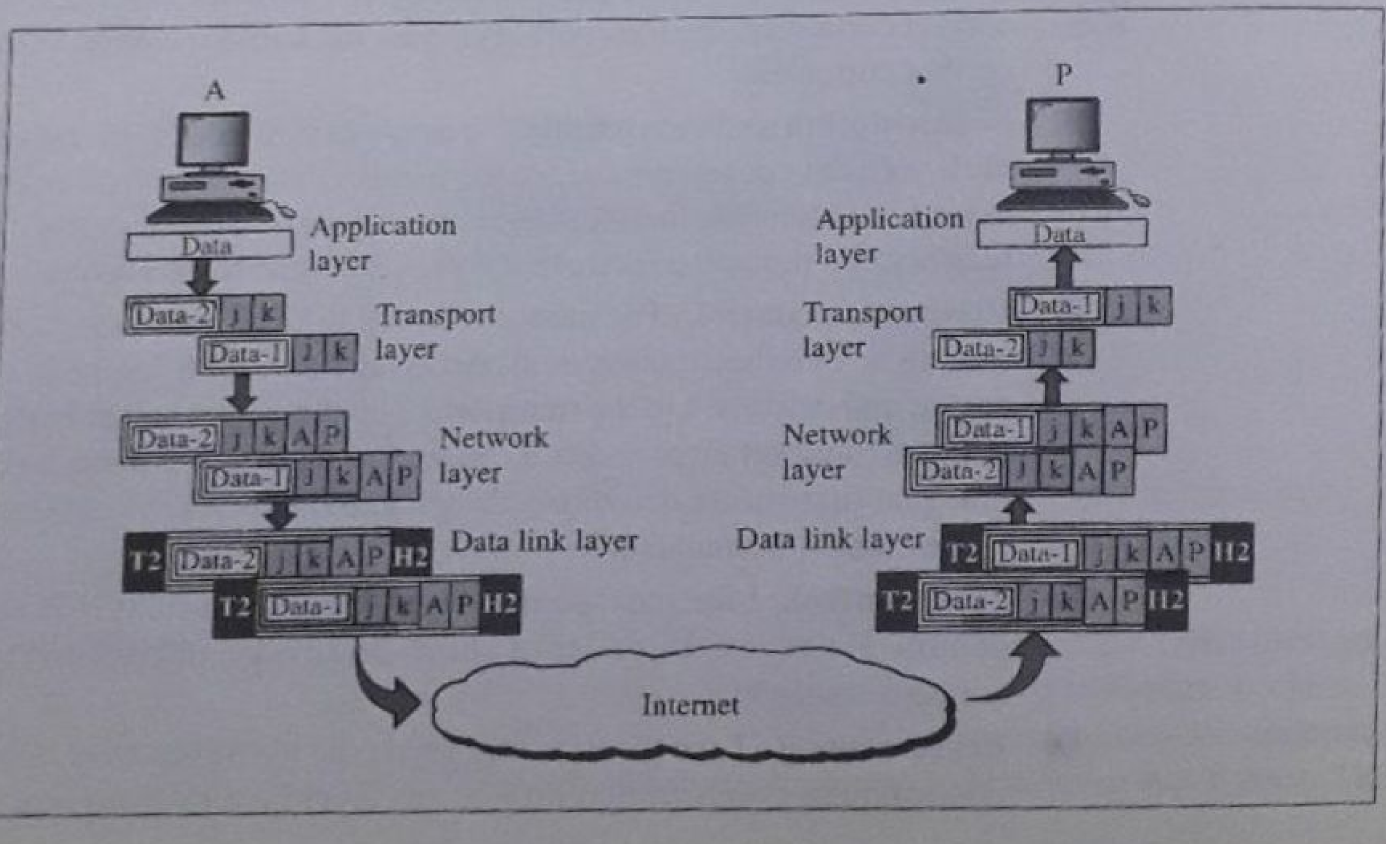
Figure 2.13 *Reliable process-to-process delivery of a message*



Example 3

Figure 2.14 shows an example of transport layer communication. Data coming from the upper layers have port addresses *j* and *k* (*j* is the address of the sending process, and *k* is the address of the receiving process). Since the data size is larger than the network layer can handle, the data are split into two packets, each packet retaining the port addresses (*j* and *k*). Then in the network layer, network addresses (A and P) are added to each packet. The packets can travel on different

Figure 2.14 *Example 3*



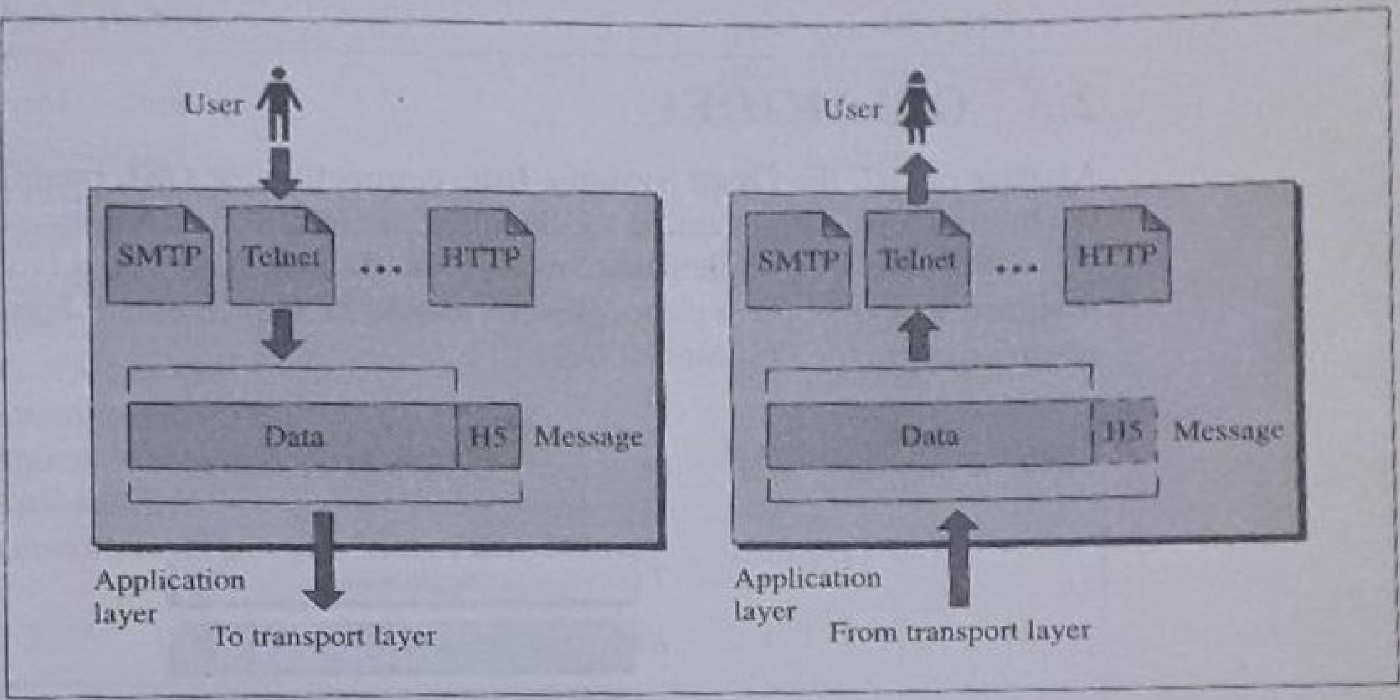
paths and arrive at the destination either in order or out of order. The two packets are delivered to the destination transport layer, which is responsible for removing the transport layer headers and combining the two pieces of data for delivery to the application layer.

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, access to the World Wide Web, and so on.

Figure 2.15 shows the relationship of the application layer to the user and the transport layer.

Figure 2.15 *Application layer*



The application layer is responsible for providing services to the user.

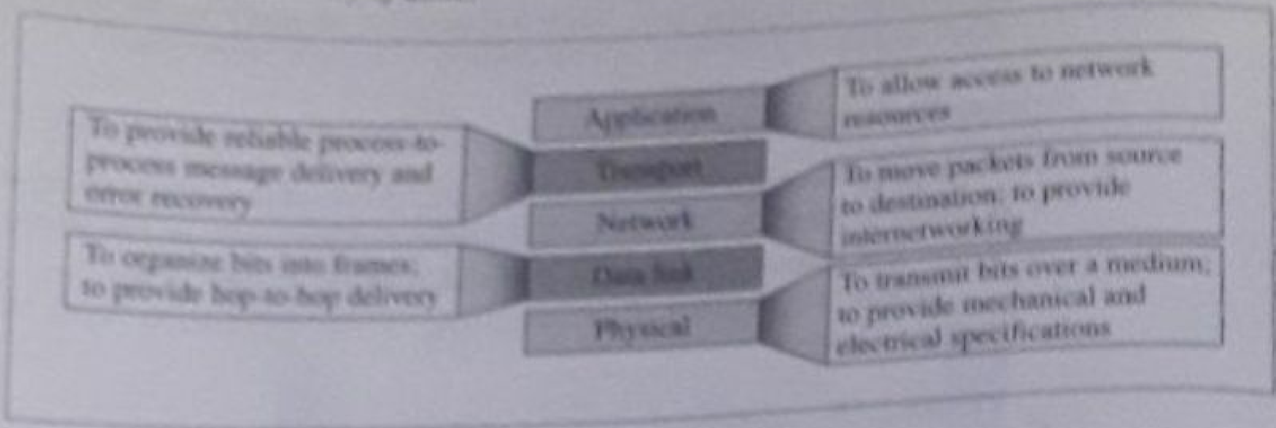
We discuss the application layer in Part VI of this book and include the dominant protocols designed for this layer. The major duties of the application layer are as follows:

- **Mail services.** This application is the basis for email forwarding and storage.
- **File transfer and access.** This application allows a user to access files in a remote host (to make changes or read data), to retrieve files from a remote computer for use in the local computer, and to manage or control files in a remote computer locally.
- **Remote log-in.** A user can log into a remote computer and access the resources of that computer.
- **Accessing the World Wide Web.** The most common application today is the access of the World Wide Web (WWW).

Summary of Layers

Figure 2.16 summarizes the duties of each layer.

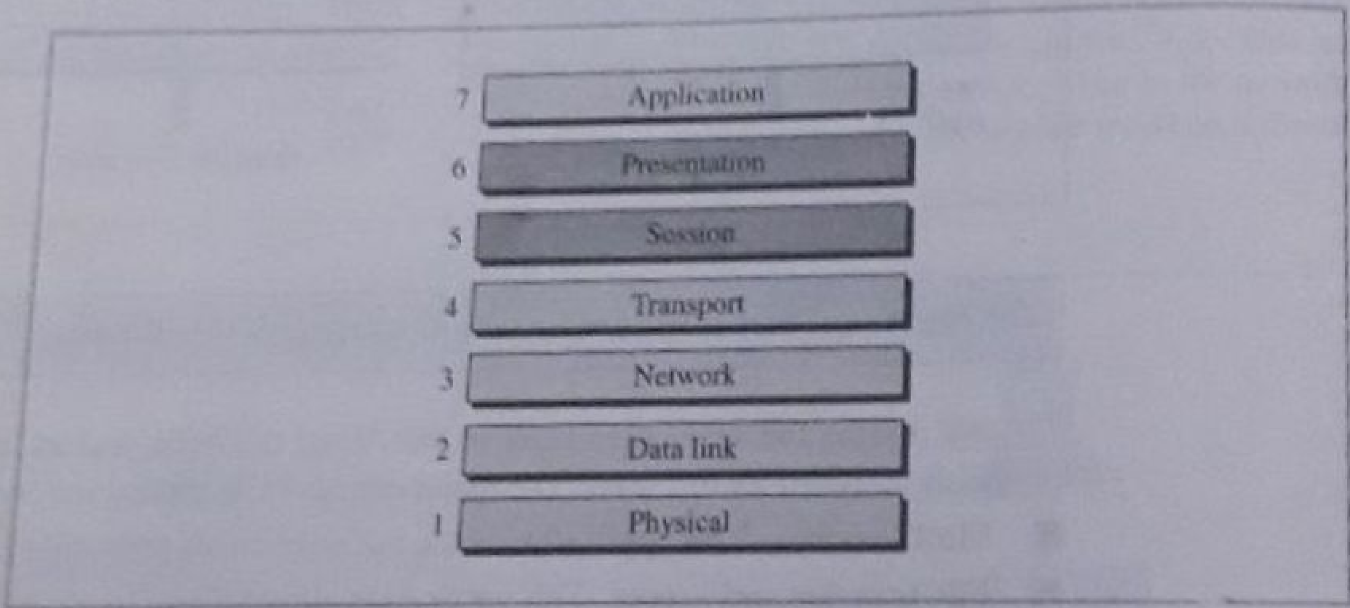
Figure 2.16 *Summary of duties*



2.3 OSI MODEL

Another model, the **Open Systems Interconnection**, or **OSI**, model, was designed by the International Organization for Standardization (ISO). It is a seven-layer model. OSI was never seriously implemented as a protocol stack, however; it is a theoretical model designed to show how a protocol stack should be implemented. Figure 2.17 shows the seven layers in the OSI model.

Figure 2.17 *OSI model*



As Figure 2.17 shows, OSI defines two extra layers: the session and presentation layers. The **session layer** is the network *dialog controller*. It was designed to establish, maintain, and synchronize the interaction between communicating systems.

The **presentation layer** was designed to handle the syntax and semantics of the information exchanged between the two systems. It was designed for data translation, encryption, decryption, and compression.

The OSI model is briefly discussed in Appendix C.