

CHAPTER 2

Network Models

A network is a combination of hardware and software that sends data from one location to another. The hardware consists of the physical equipment that carries signals from one point of the network to another. The software consists of instruction sets that make possible the services that we expect from a network.

We can compare the task of networking to the task of solving a mathematics problem with a computer. The fundamental job of solving the problem with a computer is done by computer hardware. However, this is a very tedious task if only hardware is involved. We would need switches for every memory location to store and manipulate data. The task is much easier if software is available. At the highest level, a program can direct the problem-solving process; the details of how this is done by the actual hardware can be left to the layers of software that are called by the higher levels.

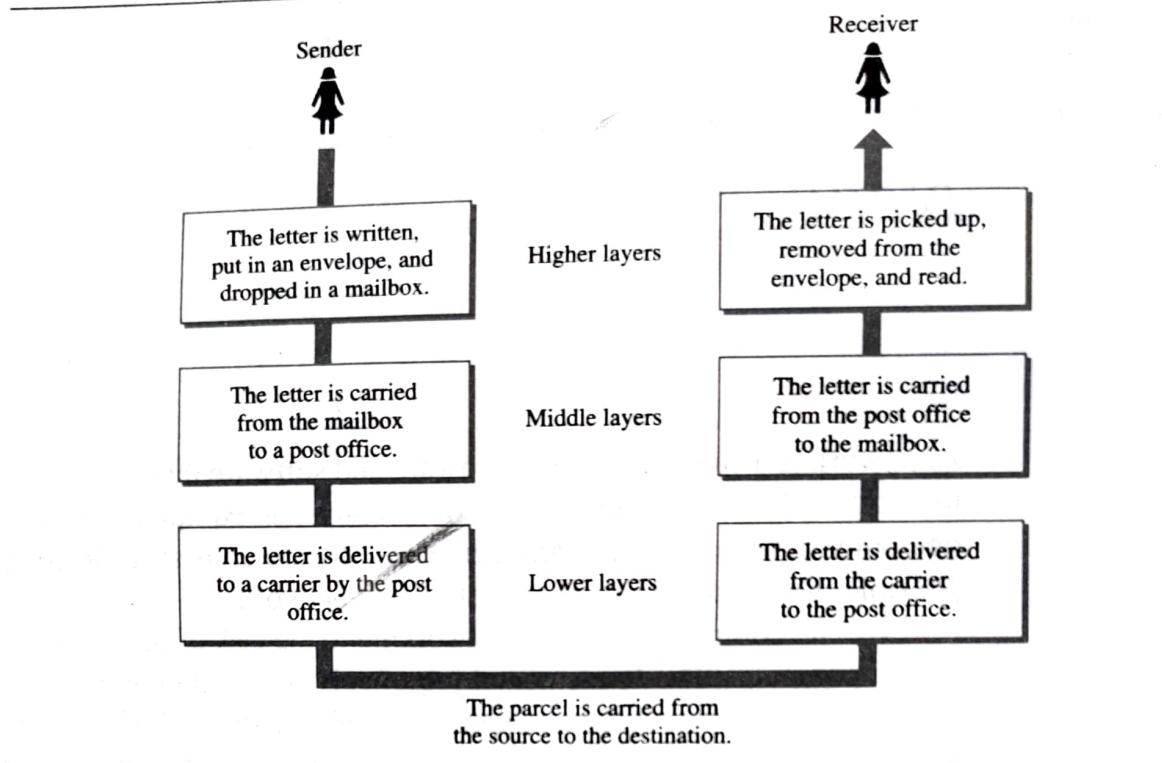
Compare this to a service provided by a computer network. For example, the task of sending an e-mail from one point in the world to another can be broken into several tasks, each performed by a separate software package. Each software package uses the services of another software package. At the lowest layer, a signal, or a set of signals, is sent from the source computer to the destination computer.

In this chapter, we give a general idea of the layers of a network and discuss the functions of each. Detailed descriptions of these layers follow in later chapters.

2.1 LAYERED TASKS

We use the concept of layers in our daily life. As an example, let us consider two friends who communicate through postal mail. The process of sending a letter to a friend would be complex if there were no services available from the post office. Figure 2.1 shows the steps in this task.

Figure 2.1 Tasks involved in sending a letter



Sender, Receiver, and Carrier

In Figure 2.1 we have a sender, a receiver, and a carrier that transports the letter. There is a hierarchy of tasks.

At the Sender Site

Let us first describe, in order, the activities that take place at the sender site.

- Higher layer.** The sender writes the letter, inserts the letter in an envelope, writes the sender and receiver addresses, and drops the letter in a mailbox.
- Middle layer.** The letter is picked up by a letter carrier and delivered to the post office.
- Lower layer.** The letter is sorted at the post office; a carrier transports the letter.

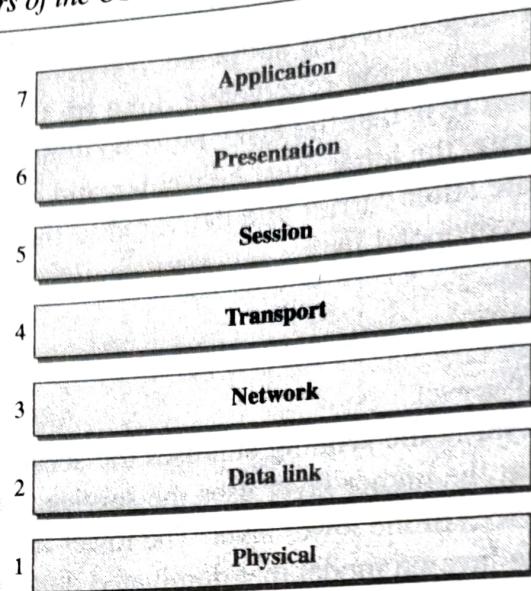
On the Way

The letter is then on its way to the recipient. On the way to the recipient's local post office, the letter may actually go through a central office. In addition, it may be transported by truck, train, airplane, boat, or a combination of these.

At the Receiver Site

- Lower layer.** The carrier transports the letter to the post office.
- Middle layer.** The letter is sorted and delivered to the recipient's mailbox.
- Higher layer.** The receiver picks up the letter, opens the envelope, and reads it.

Figure 2.2 Seven layers of the OSI model



Layered Architecture

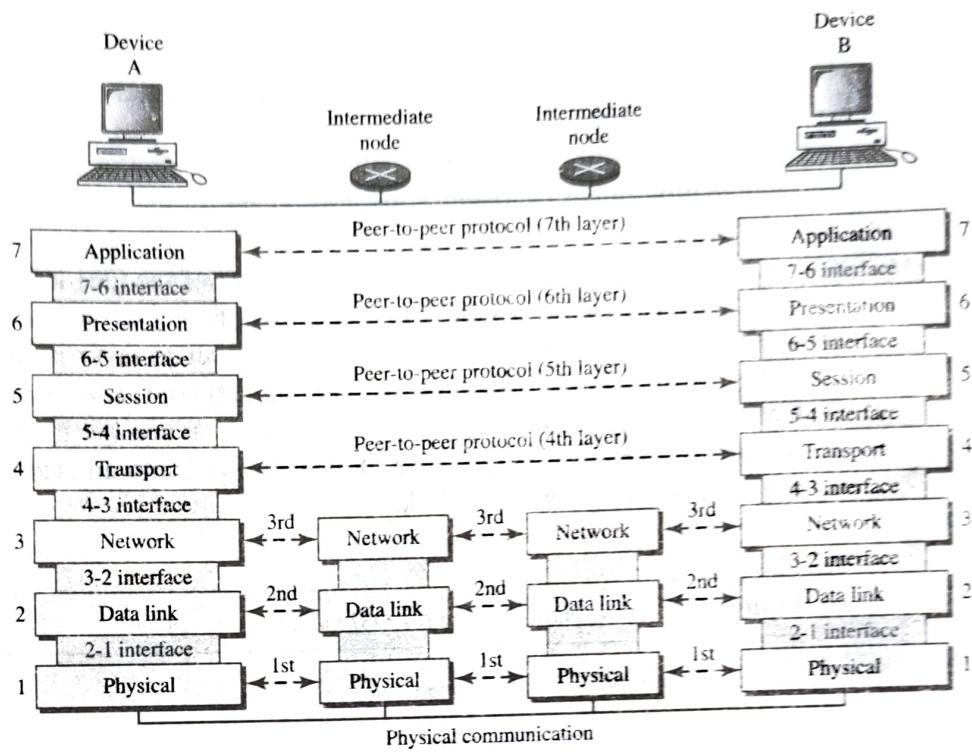
The OSI model is composed of seven ordered layers: physical (layer 1), data link (layer 2), network (layer 3), transport (layer 4), session (layer 5), presentation (layer 6), and application (layer 7). Figure 2.3 shows the layers involved when a message is sent from device A to device B. As the message travels from A to B, it may pass through many intermediate nodes. These intermediate nodes usually involve only the first three layers of the OSI model.

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. Most importantly, the OSI model allows complete interoperability between otherwise incompatible systems.

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.

Peer-to-Peer Processes

At the physical layer, communication is direct: In Figure 2.3, device A sends a stream of bits to device B (through intermediate nodes). At the higher layers, however, communication must move down through the layers on device A, over to device B, and then

Figure 2.3 The interaction between layers in the OSI model

back up through the layers. Each layer in the sending device 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.

At layer 1 the entire package is converted to a form that can be transmitted to the receiving device. 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 then 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 device and back up through the layers of the receiving device is made possible by an **interface** between each pair of adjacent layers. Each interface defines the 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 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

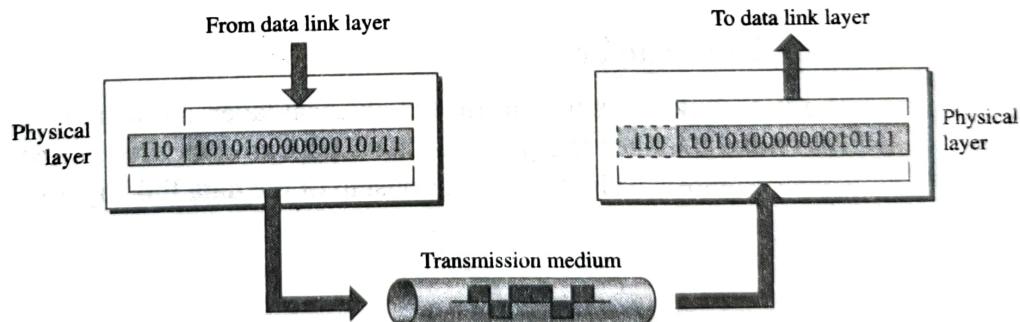
2.3 LAYERS IN THE OSI MODEL

In this section we briefly describe the functions of each layer in the OSI model.

Physical Layer

The **physical layer** coordinates the functions required to carry 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 2.5 shows the position of the physical layer with respect to the transmission medium and the data link layer.

Figure 2.5 Physical layer



The physical layer is responsible for movements of individual bits from one hop (node) to the next.

The physical layer is also concerned with the following:

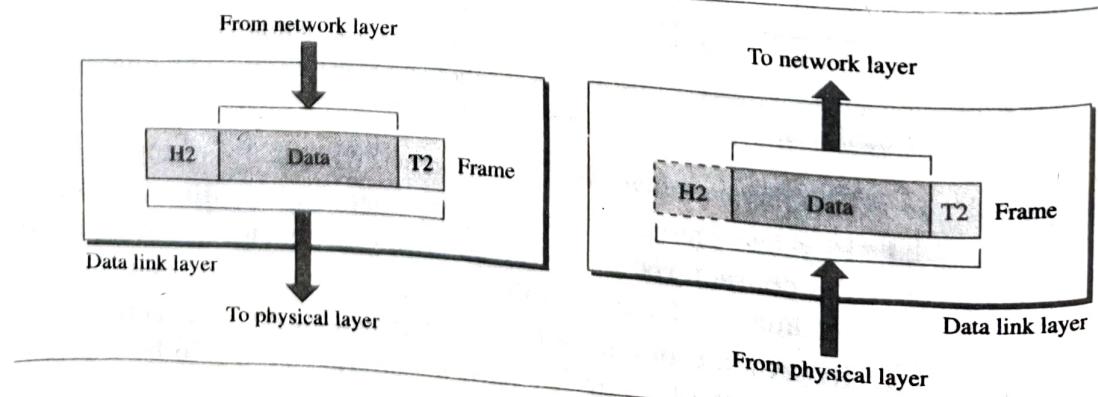
- Physical characteristics of interfaces and medium.** The physical layer defines the characteristics of the interface between the devices and the transmission medium. It also defines the type of transmission medium.
- Representation of bits.** The physical layer data consists of a stream of **bits** (sequence of 0s or 1s) with no 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 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.
 - **Line configuration.** The physical layer is concerned with the connection of devices to the media. In a point-to-point configuration, two devices are connected through a dedicated link. In a multipoint configuration, a link is shared among several devices.
 - **Physical topology.** The physical topology defines how devices are connected to make a network. Devices can be connected by using a mesh topology (every device is connected to every other device), a star topology (devices are connected through a central device), a ring topology (each device is connected to the next, forming a ring), a bus topology (every device is on a common link), or a hybrid topology (this is a combination of two or more topologies).
 - **Transmission mode.** The physical layer also defines the direction of transmission between two devices: simplex, half-duplex, or full-duplex. In 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 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



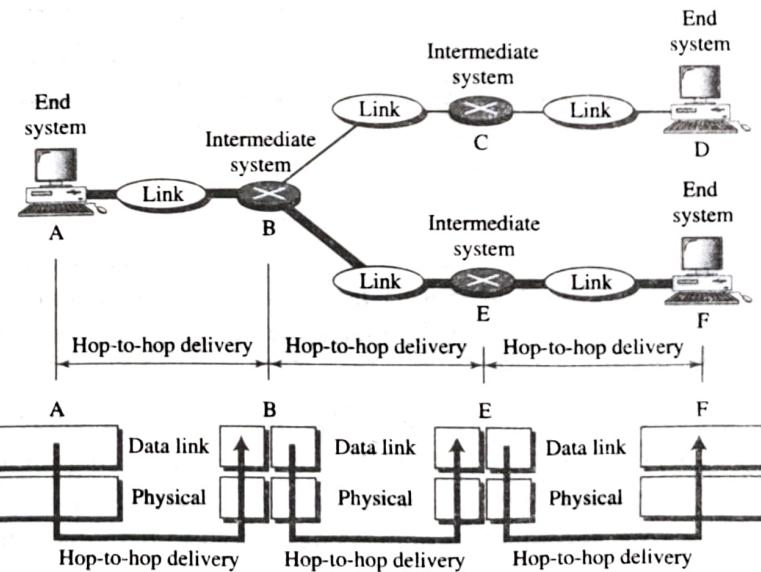


The data link layer is responsible for moving frames from one hop (node) to the next.

- Other 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 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 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 at which data are produced in the sender, the data link layer imposes a flow control mechanism to avoid 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 recognize duplicate 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.

Figure 2.7 Hop-to-hop delivery



As the figure shows, communication at the data link layer occurs between two adjacent nodes. To send data from A to F, three partial deliveries are made. First, the data link layer at A sends a frame to the data link layer at B (a router). Second, the data link layer at B sends a frame to the data link layer at E. Third, the data link layer at E sends a frame to the data link layer at F.

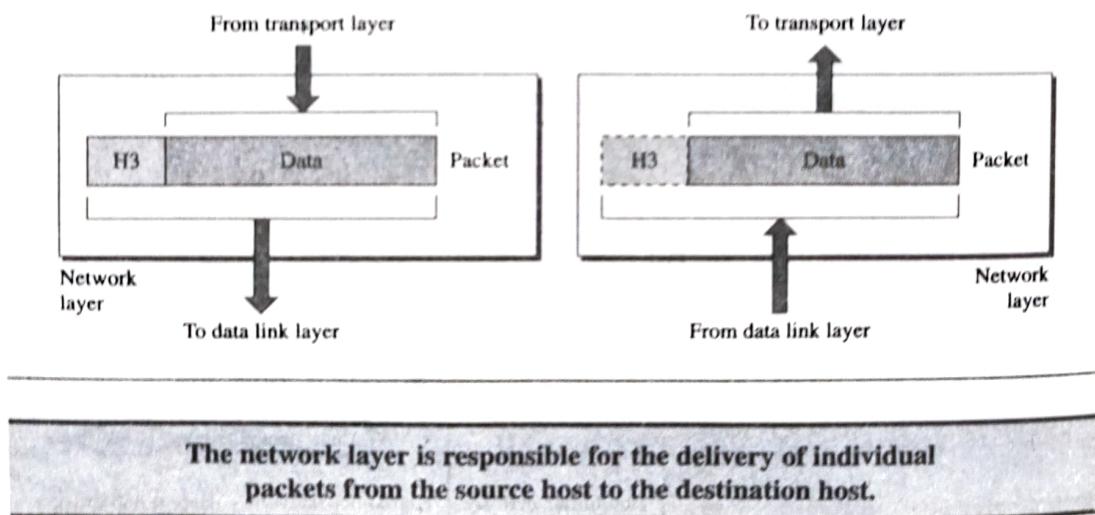
link layer at B sends a new frame to the data link layer at E. Finally, the data link layer at E sends a new frame to the data link layer at F. Note that the frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address and A as the source address. The frame from B to E has E as the destination address and B as the source address. The frame from E to F has F as the destination address and E as the source address. The values of the trailers can also be different if error checking includes the header of the frame.

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 2.8 shows the relationship of the network layer to the data link and transport layers.

Figure 2.8 Network layer



Other 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. We discuss logical addresses later in this chapter.
- Routing.** When independent networks or links are connected to create *internetworks* (network of networks) or a large network, the connecting devices (called *routers*)

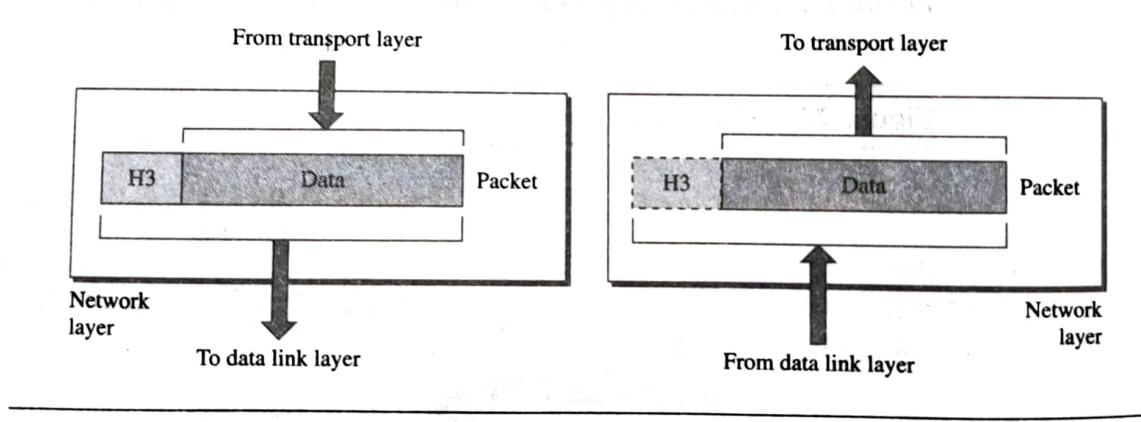
link layer at B sends a new frame to the data link layer at E. Finally, the data link layer at E sends a new frame to the data link layer at F. Note that the frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address and A as the source address. The frame from B to E has E as the destination address and B as the source address. The frame from E to F has F as the destination address and E as the source address. The values of the trailers can also be different if error checking includes the header of the frame.

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 2.8 shows the relationship of the network layer to the data link and transport layers.

Figure 2.8 Network layer



The network layer is responsible for the delivery of individual packets from the source host to the destination host.

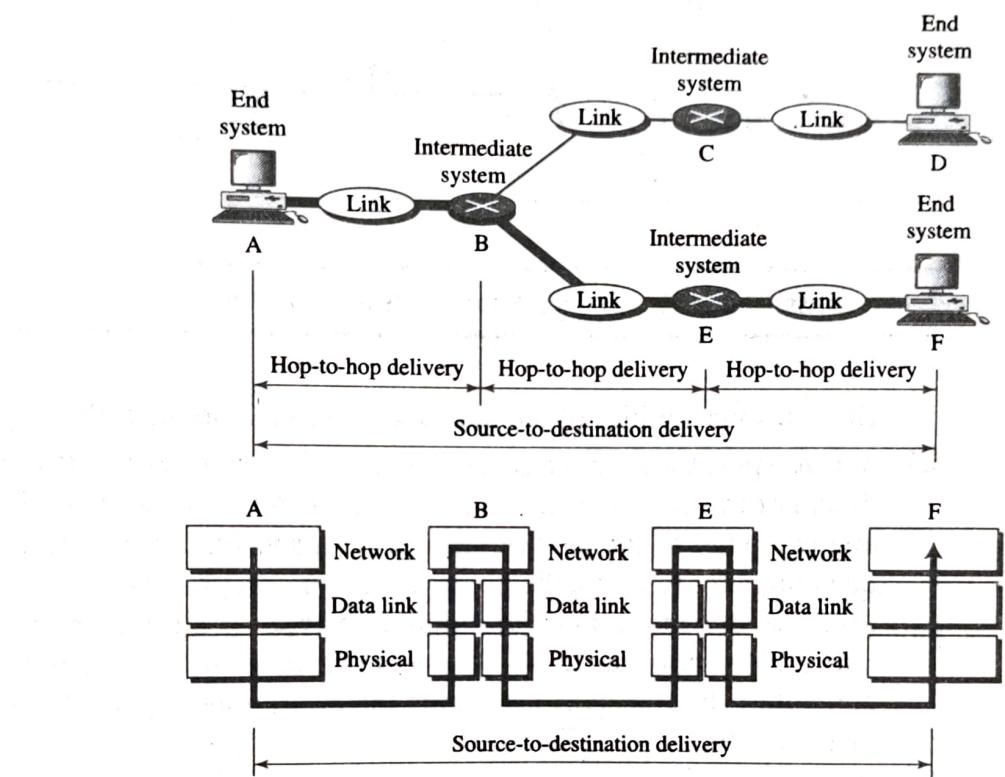
Other responsibilities of the network layer include the following:

- ❑ **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. We discuss logical addresses later in this chapter.
- ❑ **Routing.** When independent networks or links are connected to create *internetworks* (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 2.9 illustrates end-to-end delivery by the network layer.

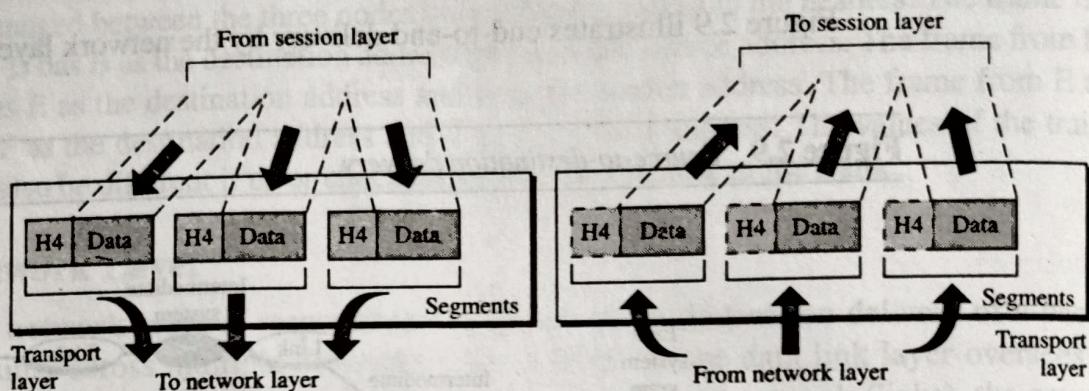
Figure 2.9 Source-to-destination delivery



As the figure shows, now we need a source-to-destination delivery. The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet. As we will see in later chapters, router B uses its routing table to find that the next hop is router E. The network layer at B, therefore, sends the packet to the network layer at E. The network layer at E, in turn, sends the packet to the network layer at F.

Transport Layer

The **transport layer** is responsible for **process-to-process delivery** of the entire message. A process is an application program running on a host. Whereas the network layer oversees **source-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 source-to-destination level. Figure 2.10 shows the relationship of the transport layer to the network and session layers.

Figure 2.10 Transport layer

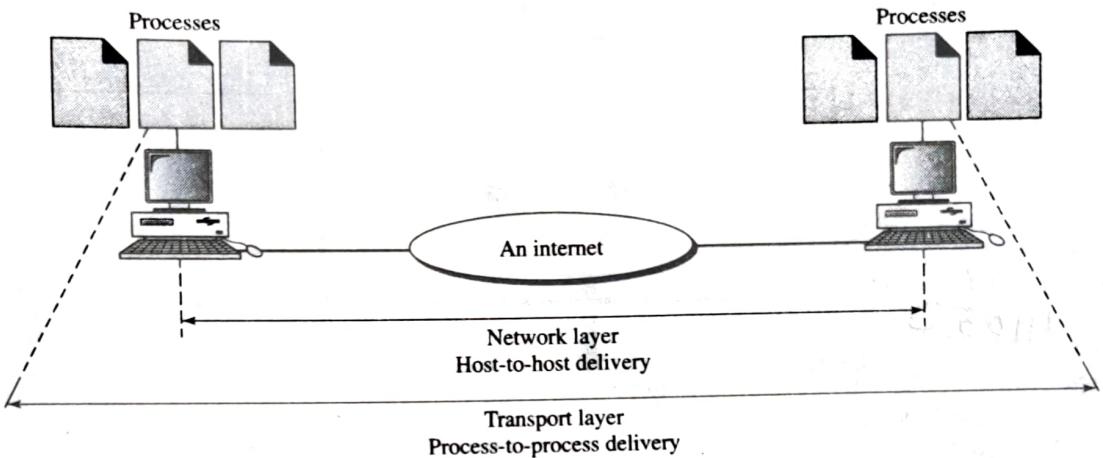
The transport layer is responsible for the delivery of a message from one process to another.

Other 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 must therefore 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.
- ❑ **Segmentation and reassembly.** A message is divided into transmittable segments, with 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 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 process-to-process 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 2.11 illustrates process-to-process delivery by the transport layer.

Figure 2.11 Reliable process-to-process delivery of a message



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 among communicating systems.

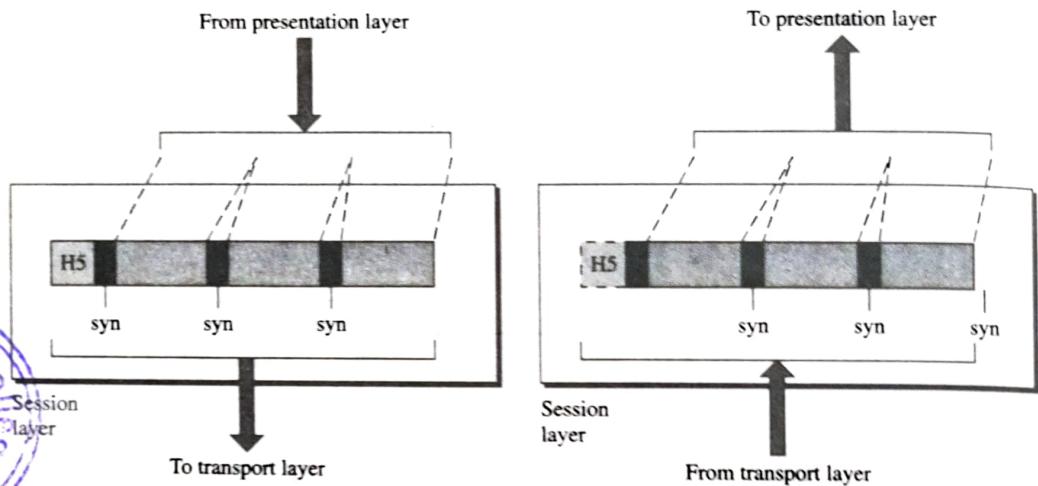
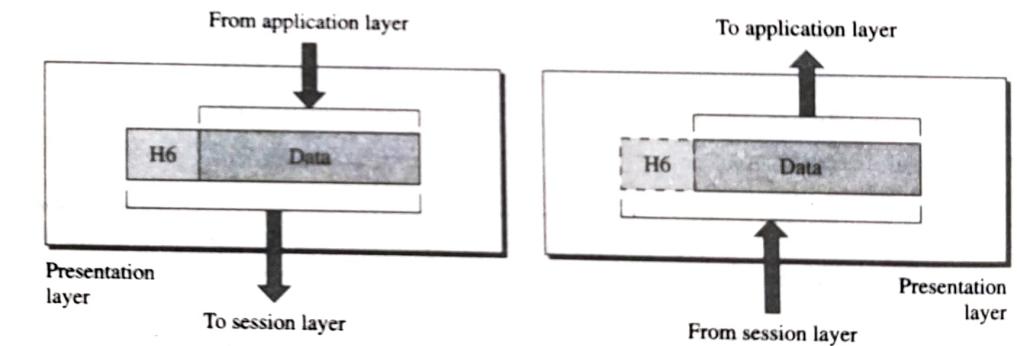
The session layer is responsible for dialog control and synchronization.

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 in either half-duplex (one way at a time) or full-duplex (two ways at a time) mode.
- ❑ **Synchronization.** The session layer allows a process to add checkpoints, or **synchronization points**, to 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, the only pages that need to be resent after system recovery are pages 501 to 523. Pages previous to 501 need not be resent. Figure 2.12 illustrates the relationship of the session layer to the transport and presentation layers.

Presentation Layer

The **presentation layer** is concerned with the syntax and semantics of the information exchanged between two systems. Figure 2.13 shows the relationship between the presentation layer and the application and session layers.

Figure 2.12 Session layer**Figure 2.13 Presentation layer**

The presentation layer is responsible for translation, compression, and encryption.

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 must 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 ensure privacy. Encryption means that the sender transforms the original information to

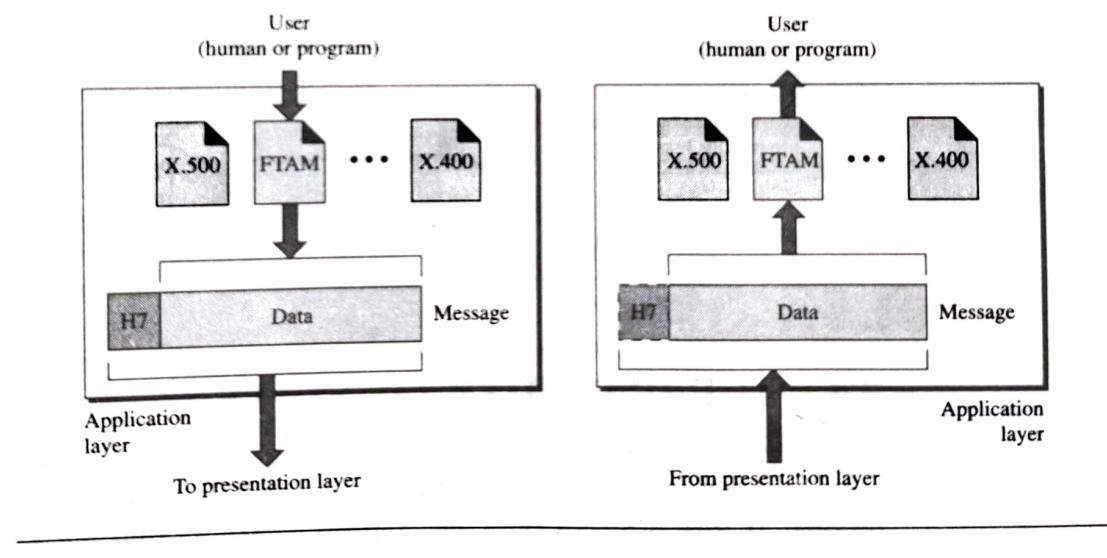
- another form and sends the resulting message out over the network. Decryption reverses the original process to transform the message back to its original form.
- ❑ **Compression.** Data compression reduces the number of bits contained in the information. Data compression becomes particularly important 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 2.14 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 employs X.400 to send an e-mail message.

Figure 2.14 Application layer



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

Specific services provided by the application layer include the following:

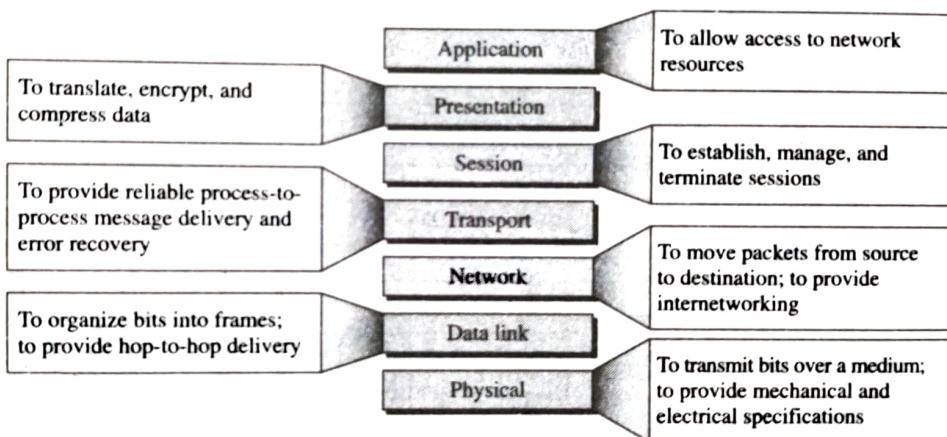
- ❑ **Network virtual terminal.** A network virtual terminal is a software version of a physical terminal, and it allows a user to log on to a remote host. To do so, the application creates a software emulation of a terminal at the remote host. The user's computer talks to the software terminal which, in turn, talks to the host, and vice versa. The remote host believes it is communicating with one of its own terminals and allows the user to log on.

- File transfer, access, and management.** 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.
- 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 Layers

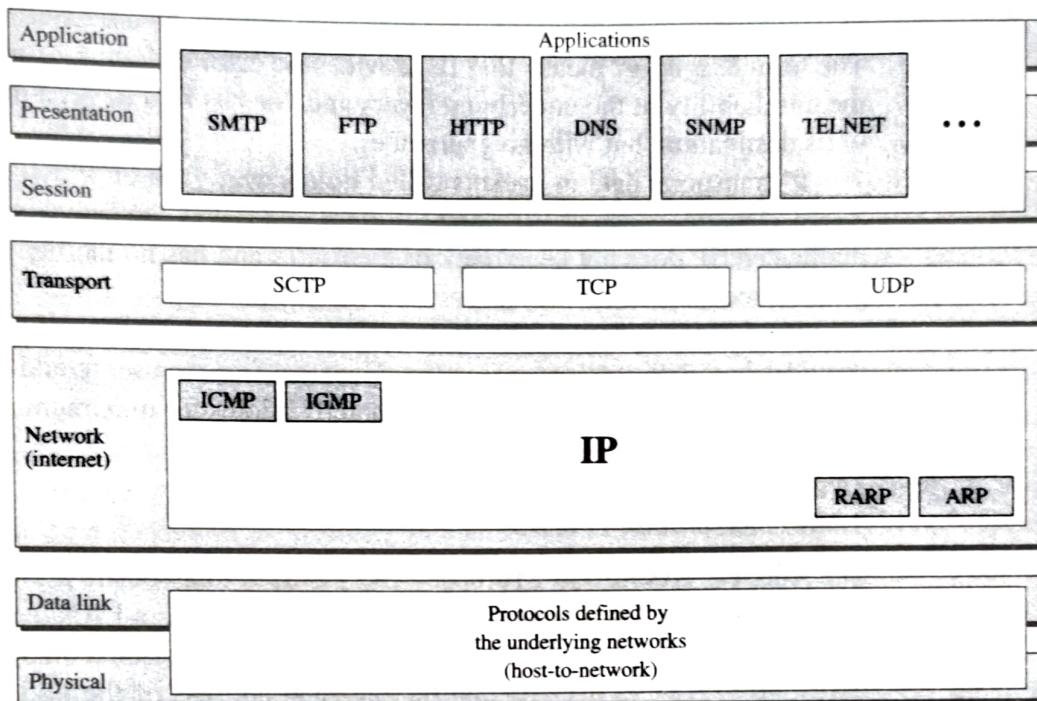
Figure 2.15 shows a summary of duties for each layer.

Figure 2.15 Summary of layers



2.4 TCP/IP PROTOCOL SUITE

The **TCP/IP protocol suite** was developed prior to the OSI model. Therefore, the layers in the TCP/IP protocol suite do not exactly match those in the OSI model. The original TCP/IP protocol suite was defined as having four layers: host-to-network, internet, transport, and application. However, when TCP/IP is compared to OSI, we can say that the host-to-network layer is equivalent to the combination of the physical and data link layers. The internet layer is equivalent to the network layer, and the application layer is roughly doing the job of the session, presentation, and application layers with the transport layer in TCP/IP taking care of part of the duties of the session layer. So in this book, we assume that the TCP/IP protocol suite is made of five layers: physical, data link, network, transport, and application. The first four layers provide physical standards, network interfaces, internetworking, and transport functions that correspond to the first four layers of the OSI model. The three topmost layers in the OSI model, however, are represented in TCP/IP by a single layer called the *application layer* (see Figure 2.16).

Figure 2.16 TCP/IP and OSI model

TCP/IP is a hierarchical protocol made up of interactive modules, each of which provides a specific functionality; however, the modules are not necessarily interdependent. Whereas the OSI model specifies which functions belong to each of its layers, the layers of the TCP/IP protocol suite contain relatively independent protocols that can be mixed and matched depending on the needs of the system. The term *hierarchical* means that each upper-level protocol is supported by one or more lower-level protocols.

At the transport layer, TCP/IP defines three protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Stream Control Transmission Protocol (SCTP). At the network layer, the main protocol defined by TCP/IP is the Internetworking Protocol (IP); there are also some other protocols that support data movement in this layer.

Physical and Data Link Layers

At the physical and data link layers, TCP/IP does not define any specific protocol. It supports all the standard and proprietary protocols. A network in a TCP/IP internetwork can be a local-area network or a wide-area network.

Network Layer

At the network layer (or, more accurately, the internetwork layer), TCP/IP supports the Internetworking Protocol. IP, in turn, uses four supporting protocols: ARP, RARP, ICMP, and IGMP. Each of these protocols is described in greater detail in later chapters.

Internetworking Protocol (IP)

The Internetworking Protocol (IP) is the transmission mechanism used by the TCP/IP protocols. It is an unreliable and connectionless protocol—a **best-effort delivery** service. The term *best effort* means that IP provides no error checking or tracking. IP assumes the unreliability of the underlying layers and does its best to get a transmission through to its destination, but with no guarantees.

IP transports data in packets called *datagrams*, each of which is transported separately. Datagrams can travel along different routes and can arrive out of sequence or be duplicated. IP does not keep track of the routes and has no facility for reordering datagrams once they arrive at their destination.

The limited functionality of IP should not be considered a weakness, however. IP provides bare-bones transmission functions that free the user to add only those facilities necessary for a given application and thereby allows for maximum efficiency. IP is discussed in Chapter 20.

Address Resolution Protocol

The **Address Resolution Protocol (ARP)** is used to associate a logical address with a physical address. On a typical physical network, such as a LAN, each device on a link is identified by a physical or station address, usually imprinted on the network interface card (NIC). ARP is used to find the physical address of the node when its Internet address is known. ARP is discussed in Chapter 21.

Reverse Address Resolution Protocol

The **Reverse Address Resolution Protocol (RARP)** allows a host to discover its Internet address when it knows only its physical address. It is used when a computer is connected to a network for the first time or when a diskless computer is booted. We discuss RARP in Chapter 21.

Internet Control Message Protocol

The **Internet Control Message Protocol (ICMP)** is a mechanism used by hosts and gateways to send notification of datagram problems back to the sender. ICMP sends query and error reporting messages. We discuss ICMP in Chapter 21.

Internet Group Message Protocol

The **Internet Group Message Protocol (IGMP)** is used to facilitate the simultaneous transmission of a message to a group of recipients. We discuss IGMP in Chapter 22.

Transport Layer

Traditionally the transport layer was represented in TCP/IP by two protocols: TCP and UDP. IP is a **host-to-host protocol**, meaning that it can deliver a packet from one physical device to another. UDP and TCP are **transport level protocols** responsible for delivery of a message from a process (running program) to another process. A new **transport layer protocol**, SCTP, has been devised to meet the needs of some newer applications.

User Datagram Protocol

The **User Datagram Protocol (UDP)** is the simpler of the two standard TCP/IP transport protocols. It is a process-to-process protocol that adds only port addresses, checksum error control, and length information to the data from the upper layer. UDP is discussed in Chapter 23.

Transmission Control Protocol

The **Transmission Control Protocol (TCP)** provides full transport-layer services to applications. TCP is a reliable stream transport protocol. The term *stream*, in this context, means connection-oriented: A connection must be established between both ends of a transmission before either can transmit data.

At the sending end of each transmission, TCP divides a stream of data into smaller units called *segments*. Each segment includes a sequence number for reordering after receipt, together with an acknowledgment number for the segments received. Segments are carried across the internet inside of IP datagrams. At the receiving end, TCP collects each datagram as it comes in and reorders the transmission based on sequence numbers. TCP is discussed in Chapter 23.

Stream Control Transmission Protocol

The **Stream Control Transmission Protocol (SCTP)** provides support for newer applications such as voice over the Internet. It is a transport layer protocol that combines the best features of UDP and TCP. We discuss SCTP in Chapter 23.

Application Layer

The *application layer* in TCP/IP is equivalent to the combined session, presentation, and application layers in the OSI model. Many protocols are defined at this layer. We cover many of the standard protocols in later chapters.

2.5 ADDRESSING

Four levels of addresses are used in an internet employing the TCP/IP protocols: **physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses** (see Figure 2.17).

Figure 2.17 Addresses in TCP/IP

