

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.

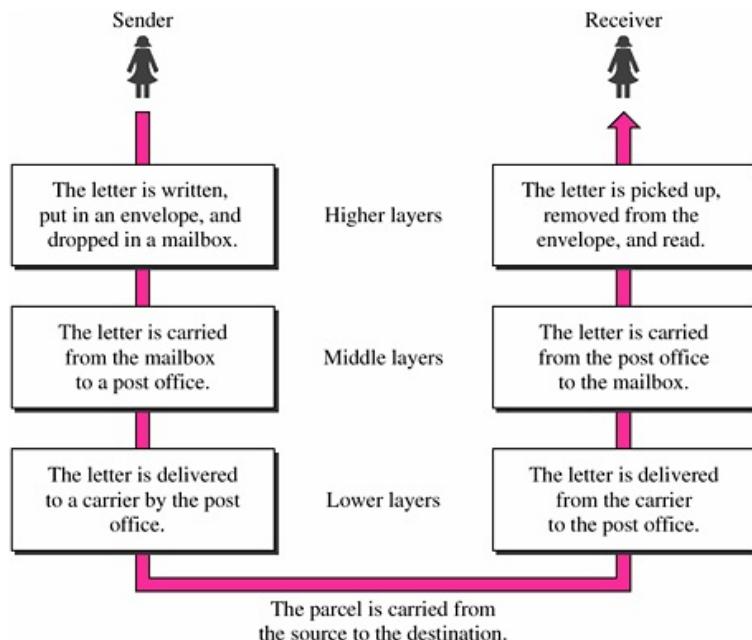
Compare this to a service provided by a computer network. For example, the task of sending an e-mail from one point in the world to another can be broken into several tasks, each performed by a separate software package. Each software package uses the services of another software package. At the lowest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



2.1.1. 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.

2.1.1.1. 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.

2.1.1.2. 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.

2.1.1.3. 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.

2.1.2. Hierarchy

According to our analysis, there are three different activities at the sender site and another three activities at the receiver site. The task of transporting the letter between the sender and the receiver is done by the carrier. Something that 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.

2.1.2.1. 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.

The layered model that dominated data communications and networking literature before 1990 was the **Open Systems Interconnection (OSI) model**. Everyone believed that the OSI model would become the ultimate standard for data communications, but this did not happen. The TCP/IP protocol suite became the dominant commercial architecture because it was used and tested extensively in the Internet; the OSI model was never fully implemented.

In this chapter, first we briefly discuss the OSI model, and then we concentrate on TCP/IP as a protocol suite.

2.2. THE OSI MODEL

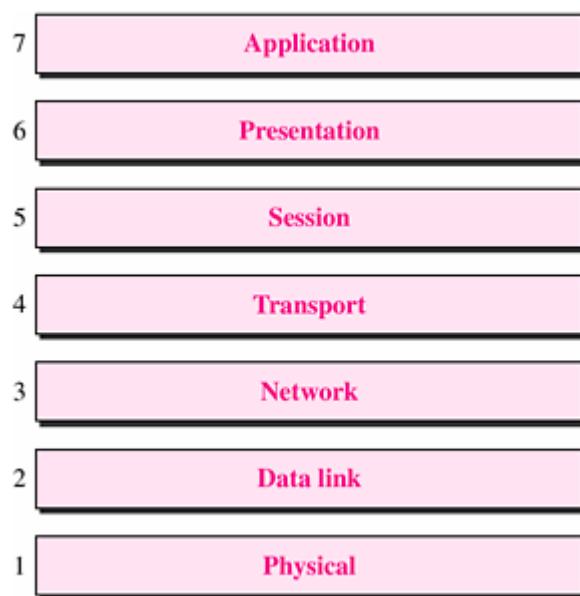
Established in 1947, the International Standards Organization (ISO) is a multinational body dedicated to worldwide agreement

on international standards. An ISO standard that covers all aspects of network communications is the Open Systems Interconnection model. It was first introduced in the late 1970s. An **open system** is a set of protocols that allows any two different systems to communicate regardless of their underlying architecture. The purpose of the OSI model is to show how to facilitate 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.

ISO is the organization. OSI is the model.

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 (see [Figure 2.2](#)). An understanding of the fundamentals of the OSI model provides a solid basis for exploring data communications.

Figure 2.2 Seven layers of the OSI model



2.2.1. 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.

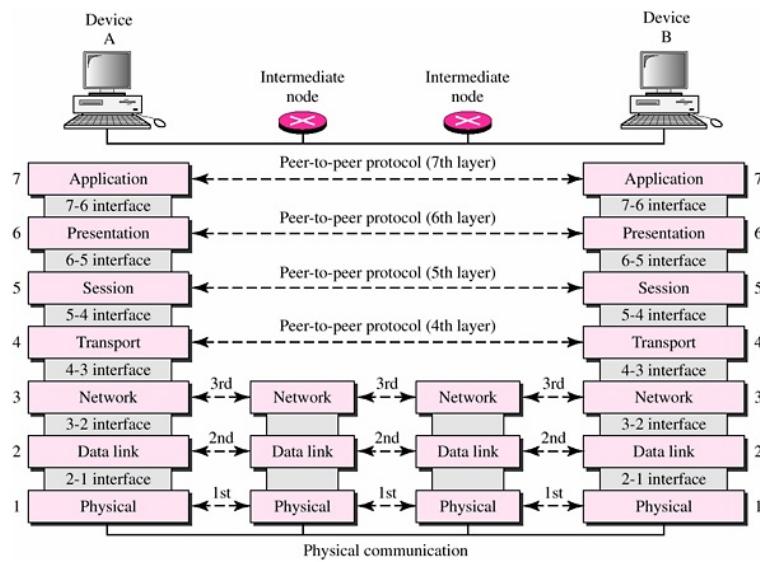
In developing the model, the designers distilled the process of transmitting data to its most fundamental elements. They identified which networking functions had related uses and collected those functions into discrete groups that became the layers. Each layer defines a family of functions distinct 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.

2.2.2. 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 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.

[Figure 2.3](#) The interaction between layers in the OSI model



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.

2.2.2.1. 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.

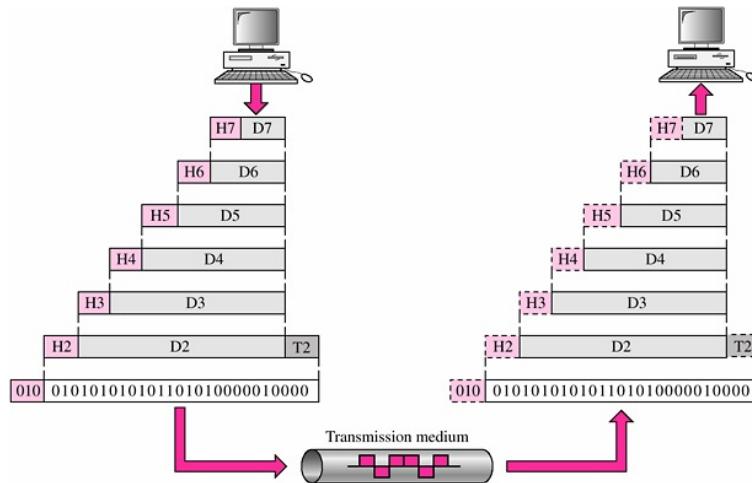
2.2.2.2. Organization of the Layers

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, links the two subgroups and ensures that what the lower layers have transmitted is in a form that the upper layers can use. 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 2.4](#), which gives an overall view of the OSI layers, D7 means the data unit at layer 7, D6 means the data unit at layer 6,

and so on. The process starts at layer 7 (the application layer), then moves from layer to layer in descending, sequential order. At each layer, a **header**, or possibly a **trailer**, can be added to the data unit. Commonly, the trailer is added only at layer 2. 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.

Figure 2.4 An exchange using the OSI model



Upon reaching its destination, the signal passes into layer 1 and is transformed back into digital form. The data units then move back up through the OSI layers. As each block of data reaches the next higher layer, the headers and trailers attached to it at the corresponding sending layer are removed, and actions appropriate to that layer are taken. By the time it reaches layer 7, the message is again in a form appropriate to the application and is made available to the recipient.

2.2.3. Encapsulation

Figure 2.3 reveals another aspect of data communications in the OSI model: encapsulation. A packet (header and data) at level 7 is encapsulated in a packet at level 6. The whole packet at level 6 is encapsulated in a packet at level 5, and so on.

In other words, the data portion of a packet at level $N - 1$ carries the whole packet (data and header and maybe trailer) from level N . The concept is called **encapsulation**; level $N - 1$ is not aware of which part of the encapsulated packet is data and which part is the header or trailer. For level $N - 1$, the whole packet coming from level N is treated as one integral unit.

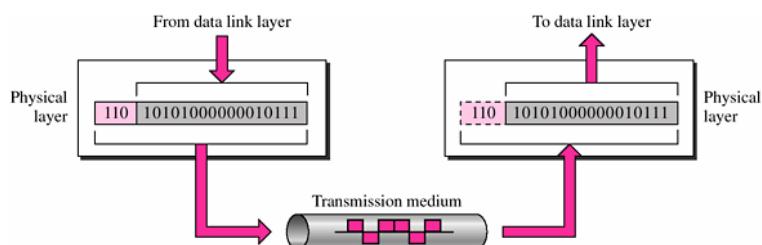
2.3. LAYERS IN THE OSI MODEL

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

2.3.1. 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.

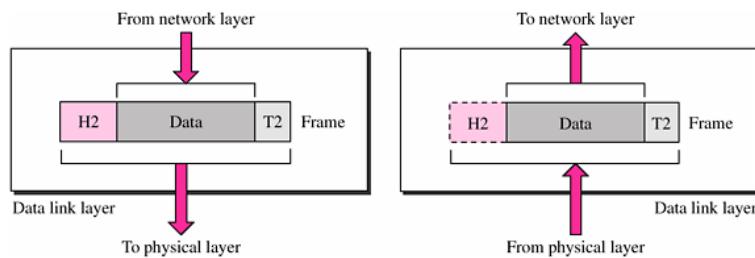
The physical layer is also concerned with the following:

- **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.

2.3.2. 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



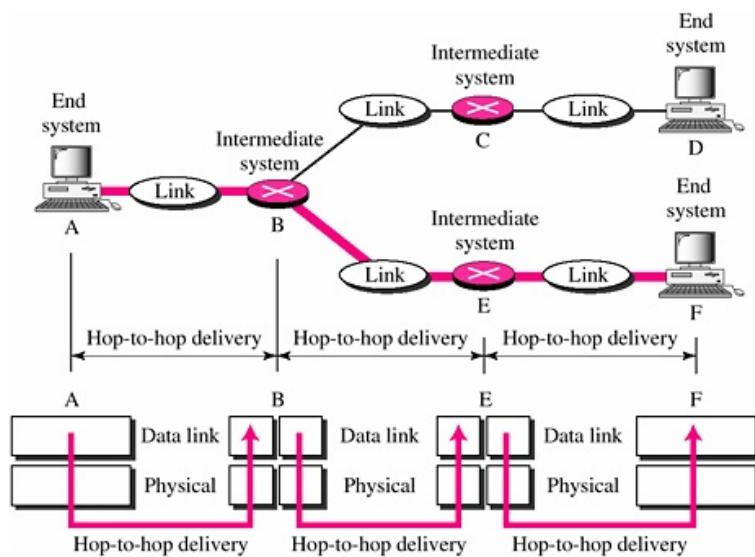
The data link layer is responsible for moving frames from one hop (node) to the next.

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

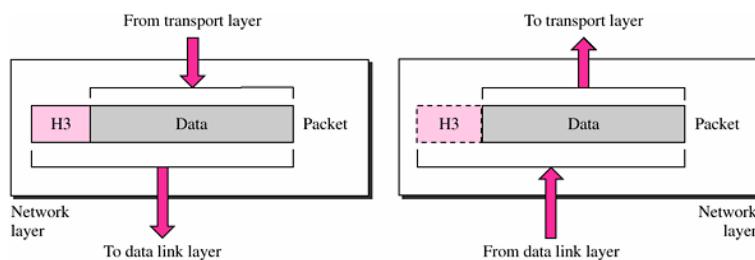
link layer at B sends a new frame to the data link layer at E. Finally, the data link layer at E sends a new frame to the data link layer at F. Note that the frames that are exchanged between the three nodes have different values in the headers. The frame from A to B has B as the destination address 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.

2.3.3. 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



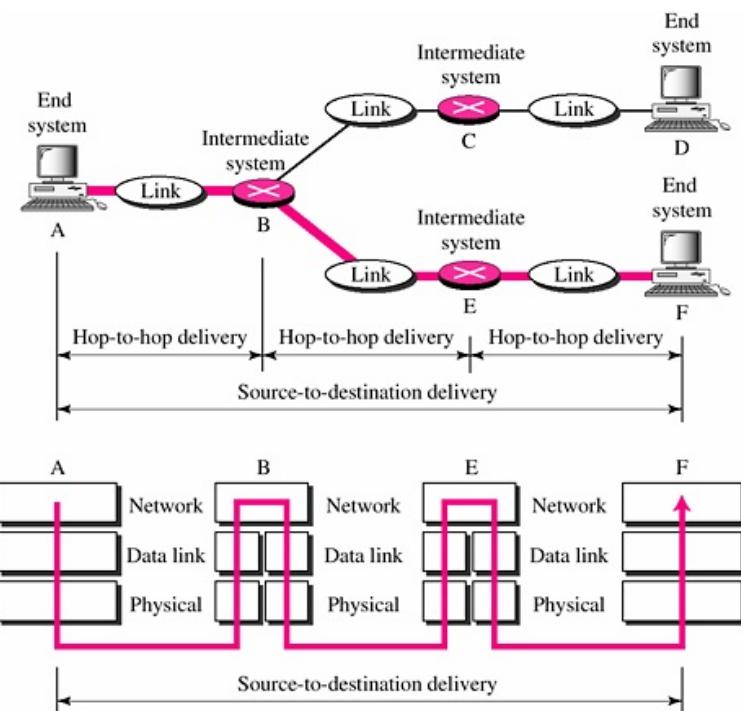
The network layer is responsible for the delivery of individual packets from the source host to the destination host.

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*) 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

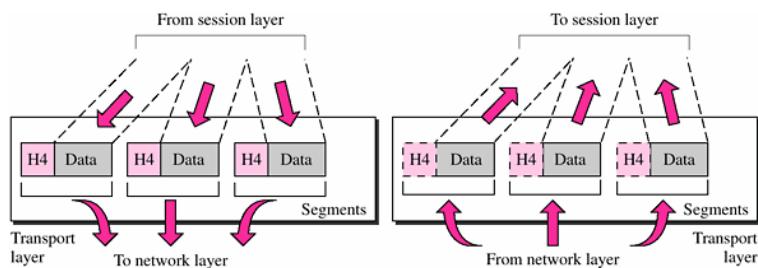


As the figure shows, now we need a source-to-destination delivery. The network layer at A sends the packet to the network layer at B. When the packet arrives at router B, the router makes a decision based on the final destination (F) of the packet. As we will see in later chapters, router B uses its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

2.3.4. 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



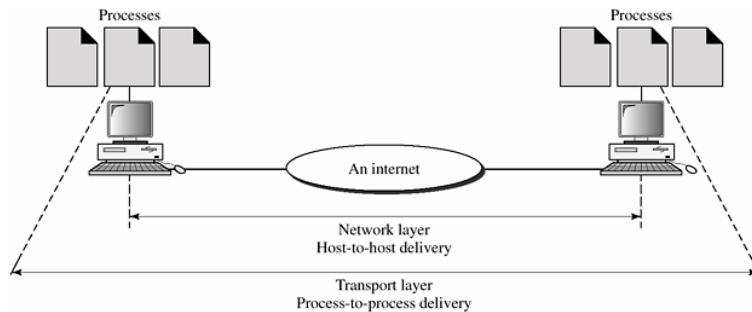
The **transport layer** is responsible for the delivery of a message from one process to another.

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



2.3.5. 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:

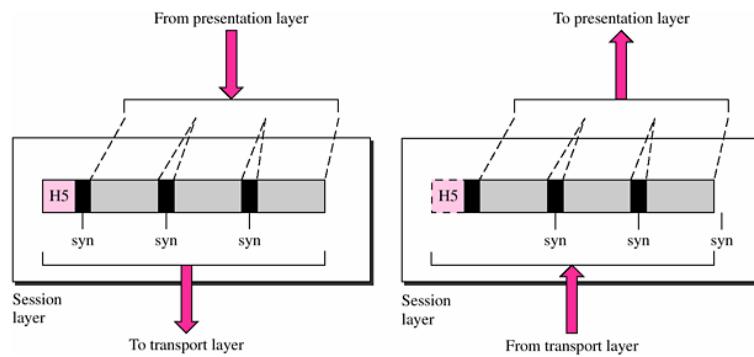
- **Dialog control.** The session layer allows two systems to enter into a dialog. It allows the communication between two processes to take place in either half-duplex (one way at a time) or full-duplex (two ways at a time) mode.
- **Synchronization.** The session layer allows a process to add checkpoints, 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.

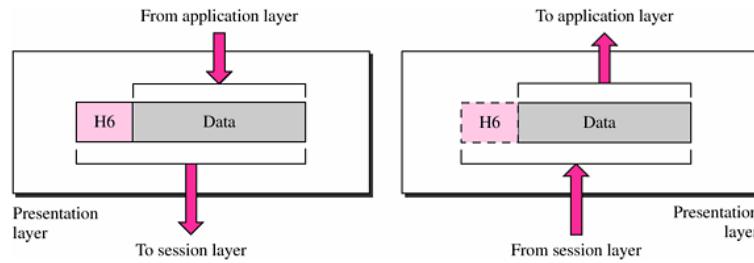
2.3.6. Presentation Layer

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:

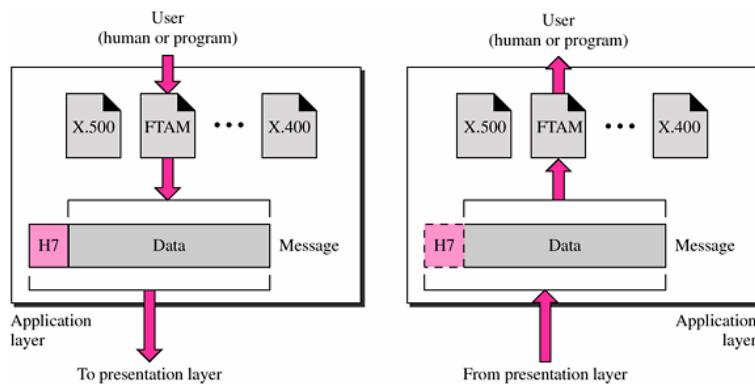
- **Translation.** The processes (running programs) in two systems are usually exchanging information in the form of character strings, numbers, and so on. The information must be changed to bit streams before being transmitted. Because different computers use different encoding systems, the presentation 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.

2.3.7. 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.500 (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.

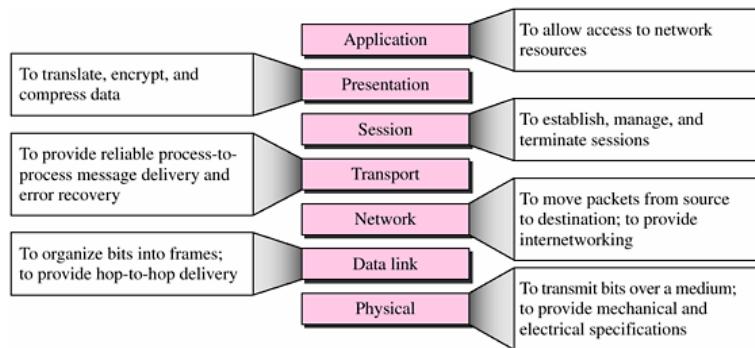
Specific services provided by the application layer include the following:

- **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.

2.3.8. Summary of Layers

Figure 2.15 shows a summary of duties for each layer.

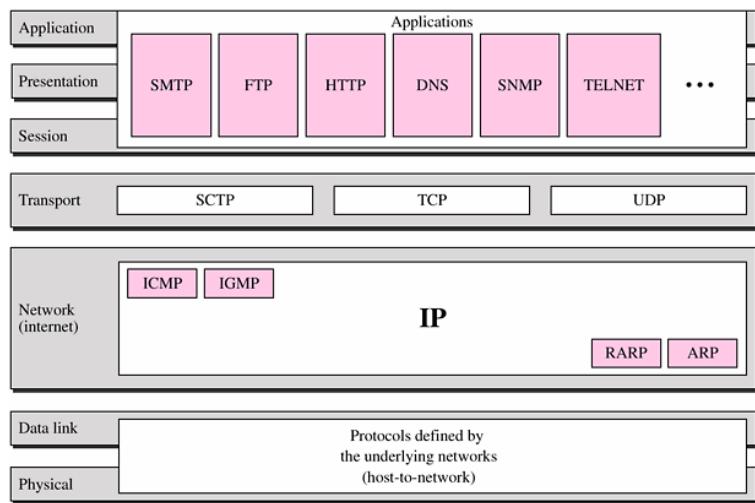
Figure 2.15 Summary of layers



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.

2.4.1. 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.

2.4.2. 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.

2.4.2.1. *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](#).

2.4.2.2. *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](#).

2.4.2.3. *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](#).

2.4.2.4. *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](#).

2.4.2.5. *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](#).

2.4.3. 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.

2.4.3.1. 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](#).

2.4.3.2. 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](#).

2.4.3.3. 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](#).

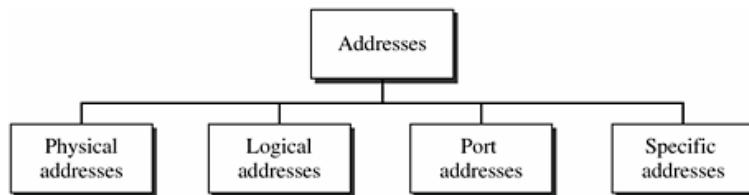
2.4.3.4. Application Layer

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

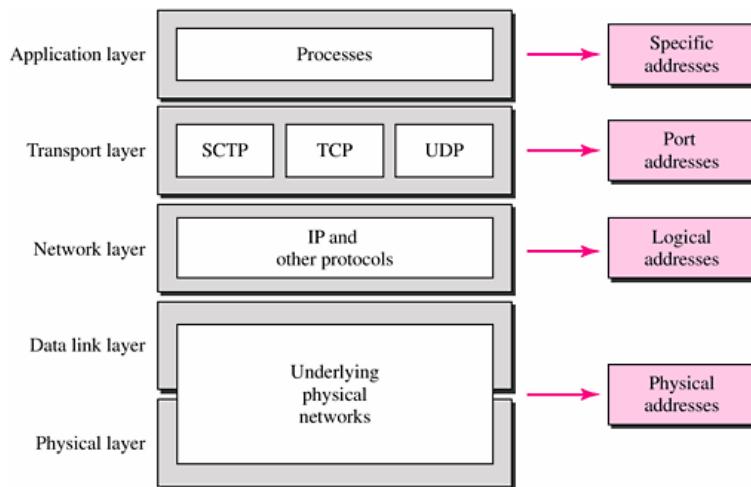
Four levels of addresses are used in an internet employing the TCP/IP protocols: **physical (link) addresses**, **logical (IP) addresses**, **port addresses**, and specific addresses (see [Figure 2.17](#)).

Figure 2.17 Addresses in TCP/IP



Each address is related to a specific layer in the TCP/IP architecture, as shown in Figure 2.18.

Figure 2.18 Relationship of layers and addresses in TCP/IP



2.5.1. Physical Addresses

The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN. It is included in the frame used by the data link layer. It is the lowest-level address.

The physical addresses have authority over the network (LAN or WAN). The size and format of these addresses vary depending on the network. For example, Ethernet uses a 6-byte (48-bit) physical address that is imprinted on the network interface card (NIC). LocalTalk (Apple), however, has a 1-byte dynamic address that changes each time the station comes up.

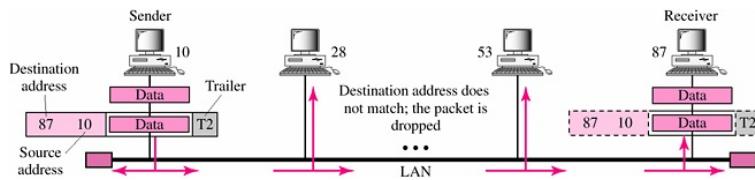
Example 2.1

In Figure 2.19 a node with physical address 10 sends a frame to a node with physical address 87. The two nodes are connected by a link (bus topology LAN). At the data link layer, 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. As the figure shows, the computer with physical address 10 is the sender, and the computer with physical address 87 is the receiver. The data link layer at the sender receives data from an upper layer. It encapsulates the data in a frame, adding a header and a trailer. The header, among other pieces of information, carries the receiver and the sender physical (link) addresses. Note that in most data link protocols, the destination address, 87 in this case, comes before the source address (10 in this case).

We have shown a bus topology for an isolated LAN. In a bus topology, the frame is propagated in both directions (left and right). The frame propagated to the left dies when it reaches the end of the cable if the cable end is terminated appropriately. The frame propagated to the right is sent to every station on the network. Each station with a physical addresses other than 87 drops the frame because the destination address in the frame does not match its own physical address. The intended

destination computer, however, finds a match between the destination address in the frame and its own physical address. The frame is checked, the header and trailer are dropped, and the data part is decapsulated and delivered to the upper layer.

Figure 2.19 Physical addresses



Example 2.2

As we will see in [Chapter 13](#), most local-area networks use a 48-bit (6-byte) physical address written as 12 hexadecimal digits; every byte (2 hexadecimal digits) is separated by a colon, as shown below:

07:01:02:01:2C:4B

A 6-byte (12 hexadecimal digits) physical address

2.5.2. Logical Addresses

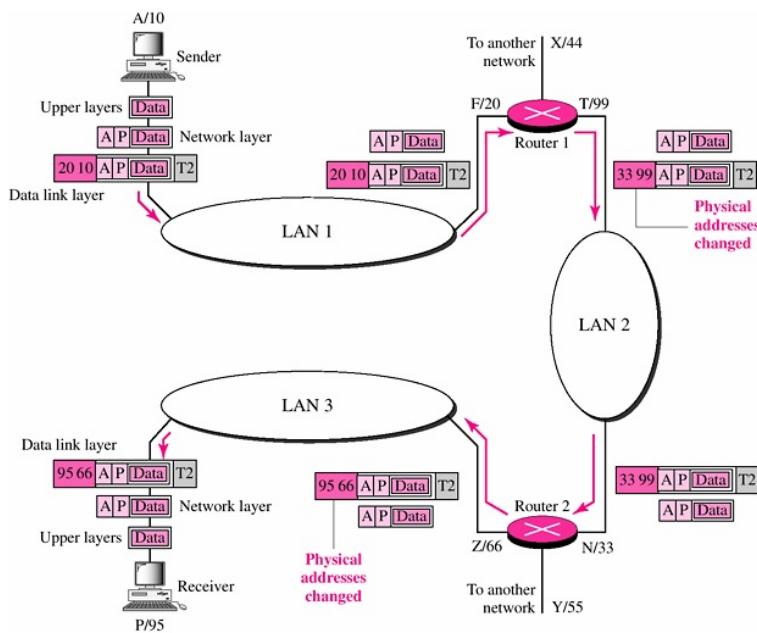
Logical addresses are necessary for universal communications that are independent of underlying physical networks. Physical addresses are not adequate in an internetwork environment where different networks can have different address formats. A universal addressing system is needed in which each host can be identified uniquely, regardless of the underlying physical network.

The logical addresses are designed for this purpose. A logical address in the Internet is currently a 32-bit address that can uniquely define a host connected to the Internet. No two publicly addressed and visible hosts on the Internet can have the same IP address.

Example 2.3

[Figure 2.20](#) shows a part of an internet with two routers connecting three LANs. Each device (computer or router) has a pair of addresses (logical and physical) for each connection. In this case, each computer is connected to only one link and therefore has only one pair of addresses. Each router, however, is connected to three networks (only two are shown in the figure). So each router has three pairs of addresses, one for each connection. Although it may obvious that each router must have a separate physical address for each connection, it may not be obvious why it needs a logical address for each connection. We discuss these issues in [Chapter 22](#) when we discuss routing.

Figure 2.20 IP addresses



The computer with logical address A and physical address 10 needs to send a packet to the computer with logical address P and physical address 95. We use letters to show the logical addresses and numbers for physical addresses, but note that both are actually numbers, as we will see later in the chapter.

The sender encapsulates its data in a packet at the network layer and adds two logical addresses (A and P). Note that in most protocols, the logical source address comes before the logical destination address (contrary to the order of physical addresses). The network layer, however, needs to find the physical address of the next hop before the packet can be delivered. The network layer consults its routing table (see [Chapter 22](#)) and finds the logical address of the next hop (router 1) to be F. The ARP discussed previously finds the physical address of router 1 that corresponds to the logical address of 20. Now the network layer passes this address to the data link layer, which in turn, encapsulates the packet with physical destination address 20 and physical source address 10.

The frame is received by every device on LAN 1, but is discarded by all except router 1, which finds that the destination physical address in the frame matches with its own physical address. The router decapsulates the packet from the frame to read the logical destination address P. Since the logical destination address does not match the router's logical address, the router knows that the packet needs to be forwarded. The router consults its routing table and ARP to find the physical destination address of the next hop (router 2), creates a new frame, encapsulates the packet, and sends it to router 2.

Note the physical addresses in the frame. The source physical address changes from 10 to 99. The destination physical address changes from 20 (router 1 physical address) to 33 (router 2 physical address). The logical source and destination addresses must remain the same; otherwise the packet will be lost.

At router 2 we have a similar scenario. The physical addresses are changed, and a new frame is sent to the destination computer. When the frame reaches the destination, the packet is decapsulated. The destination logical address P matches the logical address of the computer. The data are decapsulated from the packet and delivered to the upper layer. Note that although physical addresses will change from hop to hop, logical addresses remain the same from the source to destination. There are some exceptions to this rule that we discover later in the book.

The physical addresses will change from hop to hop, but the logical addresses usually remain the same.

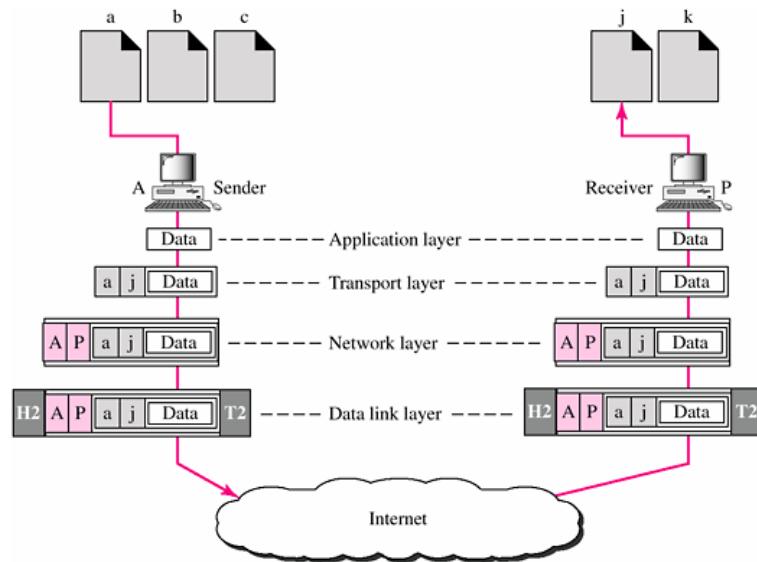
2.5.3. Port Addresses

The IP address and the physical address are necessary for a quantity of data to travel from a source to the destination host. However, arrival at the destination host is not the final objective of data communications on the Internet. A system that sends nothing but data from one computer to another is not complete. Today, computers are devices that can run multiple processes at the same time. The end objective of Internet communication is a process communicating with another process. For example, computer A can communicate with computer C by using TELNET. At the same time, computer A communicates with computer B by using the File Transfer Protocol (FTP). For these processes to receive data simultaneously, we need a method to label the different processes. In other words, they need addresses. In the TCP/IP architecture, the label assigned to a process is called a port address. A port address in TCP/IP is 16 bits in length.

Example 2.4

Figure 2.21 shows two computers communicating via the Internet. The sending computer is running three processes at this time with port addresses a, b, and c. The receiving computer is running two processes at this time with port addresses j and k. Process a in the sending computer needs to communicate with process j in the receiving computer. Note that although both computers are using the same application, FTP, for example, the port addresses are different because one is a client program and the other is a server program, as we will see in Chapter 23. To show that data from process a need to be delivered to process j, and not k, the transport layer encapsulates data from the application layer in a packet and adds two port addresses (a and j), source and destination. The packet from the transport layer is then encapsulated in another packet at the network layer with logical source and destination addresses (A and P). Finally, this packet is encapsulated in a frame with the physical source and destination addresses of the next hop. We have not shown the physical addresses because they change from hop to hop inside the cloud designated as the Internet. Note that although physical addresses change from hop to hop, logical and port addresses remain the same from the source to destination. There are some exceptions to this rule that we discuss later in the book.

Figure 2.21 Port addresses



The physical addresses change from hop to hop, but the logical and port addresses usually remain the same.

Example 2.5

As we will see in Chapter 23, a port address is a 16-bit address represented by one decimal number as shown.

753

A 16-bit port address represented as one single number

2.5.4. Specific Addresses

Some applications have user-friendly addresses that are designed for that specific address. Examples include the e-mail address (for example, forouzan@fhda.edu) and the Universal Resource Locator (URL) (for example, www.mhhe.com). The first defines the recipient of an e-mail (see [Chapter 26](#)); the second is used to find a document on the World Wide Web (see [Chapter 27](#)). These addresses, however, get changed to the corresponding port and logical addresses by the sending computer, as we will see in [Chapter 25](#).

2.6. RECOMMENDED READING

For more details about subjects discussed in this chapter, we recommend the following books and sites. The items enclosed in brackets, [...] refer to the reference list at the end of the text.

2.6.1. Books

Network models are discussed in [Section 1.3](#) of [Tan03], Chapter 2 of [For06], Chapter 2 of [Sta04], Sections [2.2](#) and [2.3](#) of [GW04], [Section 1.3](#) of [PD03], and [Section 1.7](#) of [KR05]. A good discussion about addresses can be found in [Section 1.7](#) of [Ste94].

2.6.2. Sites

The following site is related to topics discussed in this chapter.

- www.osi.org/ Information about OSI.

2.6.3. RFCs

The following site lists all RFCs, including those related to IP and port addresses.

- www.ietf.org/rfc.html

2.7. KEY TERMS

access control

Address Resolution Protocol (ARP)

application layer best-effort delivery bits connection control data link layer encoding error error control flow control frame header hop-to-hop delivery host-to-host protocol interface

Internet Control Message Protocol (ICMP)

Internet Group Message Protocol (IGMP)

logical addressing mail service network layer node-to-node delivery open system

Open Systems Interconnection (OSI) model peer-to-peer process physical addressing physical layer port address presentation layer process-to-process delivery

Reverse Address Resolution Protocol (RARP)

routing segmentation session layer source-to-destination delivery

Stream Control Transmission Protocol (SCTP)

synchronization point

TCP/IP protocol suite trailer

Transmission Control Protocol (TCP)

transmission rate transport layer transport level protocols

User Datagram Protocol (UDP)

2.8. SUMMARY

- The International Standards Organization created a model called the Open Systems Interconnection, which allows diverse systems to communicate.
- The seven-layer OSI model provides guidelines for the development of universally compatible networking protocols.
- The physical, data link, and network layers are the network support layers.
- The session, presentation, and application layers are the user support layers.
- The transport layer links the network support layers and the user support layers.
- The physical layer coordinates the functions required to transmit a bit stream over a physical medium.
- The data link layer is responsible for delivering data units from one station to the next without errors.
- The network layer is responsible for the source-to-destination delivery of a packet across multiple network links.
- The transport layer is responsible for the process-to-process delivery of the entire message.
- The session layer establishes, maintains, and synchronizes the interactions between communicating devices.
- The presentation layer ensures interoperability between communicating devices through transformation of data into a mutually agreed upon format.
- The application layer enables the users to access the network.
- TCP/IP is a five-layer hierarchical protocol suite developed before the OSI model.
- The TCP/IP application layer is equivalent to the combined session, presentation, and application layers of the OSI model.
- Four levels of addresses are used in an internet following the TCP/IP protocols: physical (link) addresses, logical (IP) addresses, port addresses, and specific addresses.
- The physical address, also known as the link address, is the address of a node as defined by its LAN or WAN.
- The IP address uniquely defines a host on the Internet.
- The port address identifies a process on a host.
- A specific address is a user-friendly address.

2.9. PRACTICE SET

2.9.1. Review Questions

1. List the layers of the Internet model.
2. Which layers in the Internet model are the network support layers?
3. Which layer in the Internet model is the user support layer?
4. What is the difference between network layer delivery and transport layer delivery?
5. What is a peer-to-peer process?
6. How does information get passed from one layer to the next in the Internet model?
7. What are headers and trailers, and how do they get added and removed?
8. What are the concerns of the physical layer in the Internet model?
9. What are the responsibilities of the data link layer in the Internet model?
10. What are the responsibilities of the network layer in the Internet model?
11. What are the responsibilities of the transport layer in the Internet model?
12. What is the difference between a port address, a logical address, and a physical address?
13. Name some services provided by the application layer in the Internet model.
14. How do the layers of the Internet model correlate to the layers of the OSI model?

2.9.2. Exercises

15. How are OSI and ISO related to each other?
16. Match the following to one or more layers of the OSI model:
 - a. Route determination
 - b. Flow control
 - c. Interface to transmission media
 - d. Provides access for the end user
17. Match the following to one or more layers of the OSI model:
 - a. Reliable process-to-process message delivery
 - b. Route selection
 - c. Defines frames
 - d. Provides user services such as e-mail and file transfer
 - e. Transmission of bit stream across physical medium
18. Match the following to one or more layers of the OSI model:
 - a. Communicates directly with user's application program

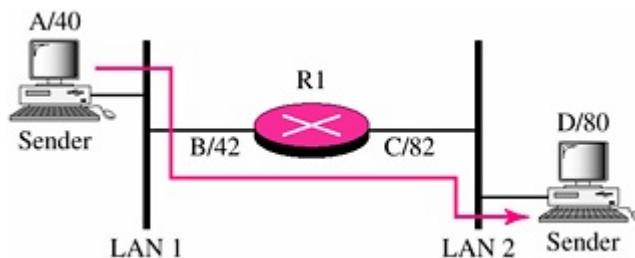
- b. Error correction and retransmission
- c. Mechanical, electrical, and functional interface
- d. Responsibility for carrying frames between adjacent nodes

19. Match the following to one or more layers of the OSI model:

- a. Format and code conversion services
- b. Establishes, manages, and terminates sessions
- c. Ensures reliable transmission of data
- d. Log-in and log-out procedures
- e. Provides independence from differences in data representation

20. In [Figure 2.22](#), computer A sends a message to computer D via LAN1, router R1, and LAN2. Show the contents of the packets and frames at the network and data link layer for each hop interface.

Figure 2.22 Exercise 20



21. In [Figure 2.22](#), assume that the communication is between a process running at computer A with port address *i* and a process running at computer D with port address *j*. Show the contents of packets and frames at the network, data link, and transport layer for each hop.

22. Suppose a computer sends a frame to another computer on a bus topology LAN. The physical destination address of the frame is corrupted during the transmission. What happens to the frame? How can the sender be informed about the situation?

23. Suppose a computer sends a packet at the network layer to another computer somewhere in the Internet. The logical destination address of the packet is corrupted. What happens to the packet? How can the source computer be informed of the situation?

24. Suppose a computer sends a packet at the transport layer to another computer somewhere in the Internet. There is no process with the destination port address running at the destination computer. What will happen?

25. If the data link layer can detect errors between hops, why do you think we need another checking mechanism at the transport layer?

2.9.3. Research Activities

- 26. Give some advantages and disadvantages of combining the session, presentation, and application layer in the OSI model into one single application layer in the Internet model.
- 27. Dialog control and synchronization are two responsibilities of the session layer in the OSI model. Which layer do you think is responsible for these duties in the Internet model? Explain your answer.

-
28. Translation, encryption, and compression are some of the duties of the presentation layer in the OSI model. Which layer do you think is responsible for these duties in the Internet model? Explain your answer.
29. There are several transport layer models proposed in the OSI model. Find all of them. Explain the differences between them.
30. There are several network layer models proposed in the OSI model. Find all of them. Explain the differences between them.