**Data Communications and Networking for Today’s Enterprise**

Trends in technology enable the provision of increasing traffic capacity and the support of a wide range of services. Four technology trends are particularly notable:

1. The trend toward faster and cheaper, in both computing and communications, continues. In terms of computing, this means more powerful computers and clusters of computers capable of supporting more demanding applications, such

as multimedia applications. In terms of communications, the increasing use of optical fiber and high-speed wireless has brought transmission prices down and greatly increased capacity. For example, for long-distance telecommunication and data network links, dense wavelength division multiplexing (DWDM) .

2. Today’s networks are more “intelligent” than ever. Two areas of intelligence are noteworthy. First, today’s networks can offer differing levels of quality of service (QoS), which include specifications for maximum delay, minimum throughput, and so on to ensure high-quality support for applications and services. Second, today’s networks provide a variety of customizable services in the areas of network management and security.

3.The Internet, the Web, and associated applications have emerged as dominant features for both business and personal network landscapes. The migration to “everything over IP” continues and has created many opportunities and challenges for information and communications technology (ICT) managers. In addition to exploiting the Internet and the Web to reach customers, suppliers, and partners enterprises have formed intranets and extranets2 to isolate proprietary information to keep it free from unwanted access.

**A Communication Model**

This section introduces a simple model of communications, illustrated by the block diagram in Figure 1.3a.

The fundamental purpose of a communications system is the exchange of data between two parties. Figure 1.3b presents one particular example, which is communication between a workstation and a server over a public telephone network.

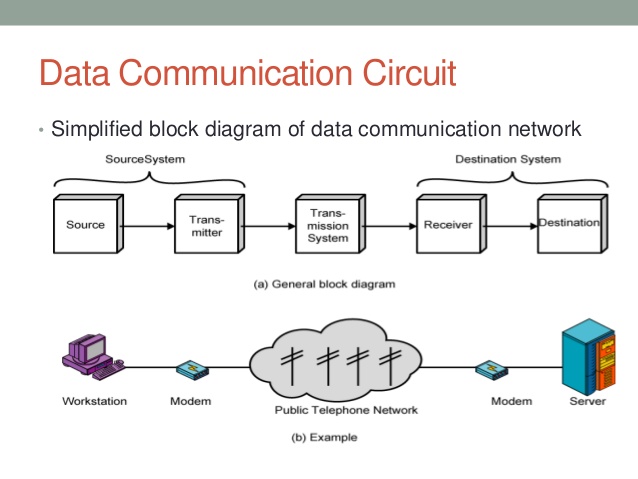
Another example is the exchange of voice signals between two telephones over the same network. The following are key elements of the model:

• **Source**: This device generates the data to be transmitted; examples are telephones and personal computers.

• **Transmitter**: Usually, the data generated by a source system are not transmitted directly in the form in which they were generated. Rather, a transmitter transforms and encodes the information in such a way as to produce electromagnetic signals that can be transmitted across some sort of transmission system.

**For example**, a modem takes a digital bit stream from an attached device such as a personal computer and transforms that bit stream into an analog signal that can be handled by the telephone network.

•**Transmission** system: This can be a single transmission line or a complex network connecting and destination source.

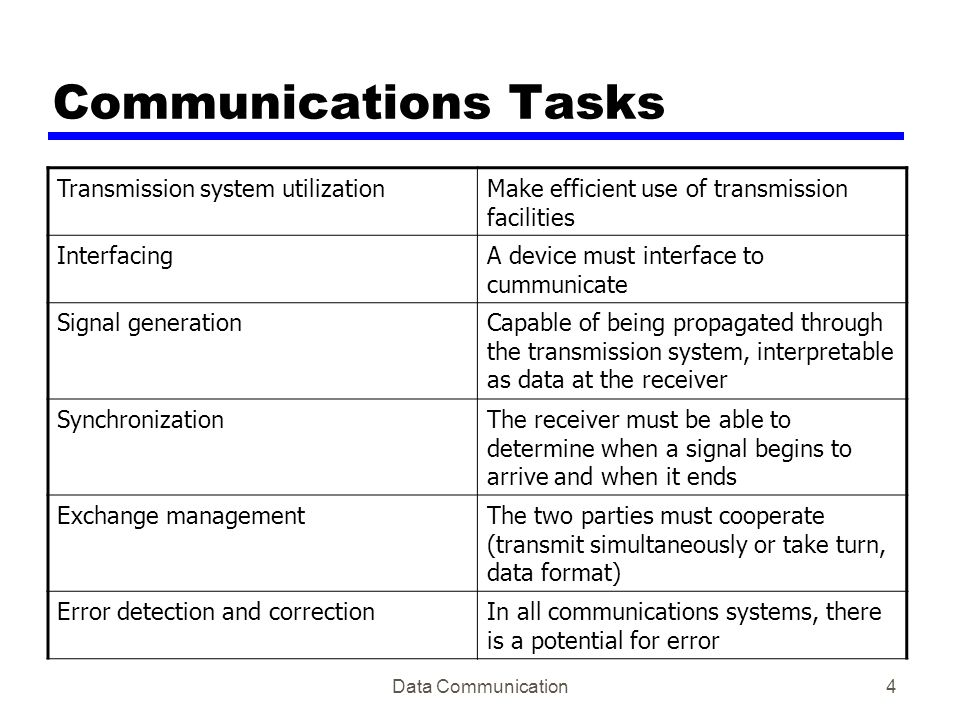


• **Receiver**: The receiver accepts the signal from the transmission system and converts it into a form that can be handled by the destination device.

**For example**, a modem will accept an analog signal coming from a network or transmission line and convert it into a digital bit stream.

• **Destination**: Takes the incoming data from the receiver.

**transmission system utilization:** It refers to the need to make efficient use of transmission facilities that are typically shared among a number of communicating devices. Various techniques (referred to as multiplexing) are used to allocate the total capacity of a transmission medium among a number of users.



To communicate, a device must **interface** with the transmission system. All forms of communication discussed in this book depend on the use of electromagnetic signals propagated over a transmission medium. Thus, once an interface is established, signal generation is required for communication. The properties of the signal, such as form and intensity, must be such that the signal is

(1) capable of being propagated through the transmission system, and

(2) interpretable as data at the receiver.

Not only must the signals be generated to conform to the requirements of the transmission system and receiver, but also there must be some form of **synchronization** between transmitter and receiver. The receiver must be able to determine when a signal begins to arrive and when it ends. It must also know the duration of each signal element.

Beyond the basic matter of deciding on the nature and timing of signals, there is a variety of requirements for communication between two parties that might be collected under the term **exchange management**. If data are to be exchanged in both directions over a period of time, the two parties must cooperate. For example, for two parties to engage in a telephone conversation, one party must dial the number of the other, causing signals to be generated that result in the ringing of the called phone. The called party completes a connection by lifting the receiver. For data processing devices, more will be needed than simply establishing a connection; certain conventions must be decided on. These conventions might include whether both devices may transmit simultaneously or must take turns, the amount of data to be sent at one time, the format of the data, and what to do if certain contingencies such as an error arise.

The next two items might have been included under exchange management, but they seem important enough to list separately. In all communications systems, there is a potential for error; transmitted signals are distorted to some extent before reaching their destination. **Error detection** **and Correction** are required in circumstances where errors cannot be tolerated. This is usually the case with data processing systems. For example, in transferring a file from one computer to another, it is simply not acceptable for the contents of the file to be accidentally altered. **Flow Control** is required to assure that the source does not overwhelm the destination by sending data faster than they can be processed and absorbed.

Next are the related but distinct concepts of **Addressing and Routing**. When more than two devices share a transmission facility, a source system must indicate the identity of the intended destination. The transmission system must assure that the destination system, and only that system, receives the data. Further, the transmission system may itself be a network through which various paths may be taken. A specific route through this network must be chosen.

**Recovery** is a concept distinct from that of error correction. Recovery techniques are needed in situations in which an information exchange, such as a database transaction or file transfer, is interrupted due to a fault somewhere in the system. The objective is either to be able to resume activity at the point of interruption or at least to restore the state of the systems involved to the condition prior to the beginning of the exchange.

**Message formatting** has to do with an agreement between two parties as to the form of the data to be exchanged or transmitted, such as the binary code for characters.

Frequently, it is important to provide some measure of **security** in a data communications system. The sender of data may wish to be assured that only the intended receiver actually receives the data. And the receiver of data may wish to be assured that the received data have not been altered in transit and that the data actually come from the purported sender. Finally, a data communications facility is a complex system that cannot create or run itself.

**Network** **management** capabilities are needed to configure the system, monitor its status, react to failures and overloads, and plan intelligently for future growth.

Thus, we have gone from the simple idea of data communication between source and destination to a rather formidable list of data communications tasks. In this book, we elaborate this list of tasks to describe and encompass the entire set of activities that can be classified under data and computer communication.

**THE INTERNET:**

Origins of the Internet the Internet evolved from the **ARPANET**, which was developed in 1969 by the **Advanced Research Projects Agency (ARPA) of the U.S. Department of Defense.**

It was the first operational packet-switching network. ARPANET began operations in four locations. Today the number of hosts is in the hundreds of millions, the number of users in the billions, and the number of countries participating nearing 200. The number of connections to the Internet continues to grow exponentially.

The network was so successful that ARPA applied the same packet-switching technology to tactical radio communication (packet radio) and to satellite communication (SATNET). Because the three networks operated in very different communication environments, the appropriate values for certain parameters, such as maximum packet size, were different in each case. Faced with the dilemma of integrating these networks, Vint Cerf and Bob Kahn of ARPA developed methods and protocols for *internetworking*—that is, communicating across arbitrary, multiple, packet-switched networks. They published a very influential paper in May 1974 [CERF74] outlining their approach to a Transmission Control Protocol. The proposal was refined and details filled in by the ARPANET community, with major contributions from participants from European networks eventually leading to the TCP (Transmission Control Protocol) and IP (Internet Protocol) protocols, which, in turn, formed the basis for what eventually became the TCP/IP protocol suite. This provided the foundation for the Internet.

**The Need for a Protocol Architecture:**

When computers, terminals, and/or other data processing devices exchange data, the procedures involved can be quite complex. Consider, for example, the transfer of a file between two computers. There must be a data path between the two computers, either directly or via a communication network. But more is needed.

Typical tasks to be performed:

1. The source system must either activate the direct data communication path or inform the communication network of the identity of the desired destination system.

2. The source system must ascertain that the destination system is prepared to receive data.

3. The file transfer application on the source system must ascertain that the file management program on the destination system is prepared to accept and store the file for this particular user.

4. If the file formats used on the two systems are different, one or the other system must perform a format translation function.

It is clear that there must be a high degree of cooperation between the two computer systems. Instead of implementing the logic for this as a single module, the task is broken up into subtasks, each of which is implemented separately. In a protocol architecture, the modules are arranged in a vertical stack. Each layer in the stack performs a related subset of the functions required to communicate with another system. It relies on the next lower layer to perform more primitive functions and to conceal the details of those functions. It provides services to the next higher layer. Ideally, layers should be defined so that changes in one layer do not require changes in other layers.

Of course, it takes two to communicate, so the same set of layered functions must exist in two systems. Communication is achieved by having the corresponding, or peer, layers in two systems communicate. The **peer layers** communicate by means of formatted blocks of data that obey a set of rules or conventions known as a **protocol**.

The key features of a protocol are as follows:

• **Syntax**: Concerns the format of the data blocks.

• **Semantics**: Includes control information for coordination and error handling.

• **Timing**: Includes speed matching and sequencing.

Appendix 2A provides a specific example of a protocol, the **Internet** standard Trivial File Transfer Protocol (TFTP).

**The TCP/IP Protocol Architecture:**

TCP/IP is a result of protocol research and development conducted on the experimental packet-switched network, **ARPANET**, funded by the **Defense Advanced Research Projects Agency (DARPA),** and is generally referred to as the TCP/IP protocol suite. This protocol suite consists of a large collection of protocols that have been issued as Internet standards by the **Internet Activities Board** **(IAB)**. Appendix C provides a discussion of Internet standards.

**The TCP/IP Layers:**

In general terms, computer communications can be said to involve three agents: applications, computers, and networks. Examples of applications include file transfer and electronic mail. The applications that we are concerned with here are distributed applications that involve the exchange of data between two computer systems. These applications, and others, execute on computers that can often support multiple simultaneous applications. Computers are connected to networks, and the data to be exchanged are transferred by the network from one computer to another. Thus, the transfer of data from one application to another involves first getting the data to the computer in which the application resides and then getting the data to the intended application within the computer. With these concepts in mind, we can organize the communication task into five relatively independent layers (Figure 2.3):

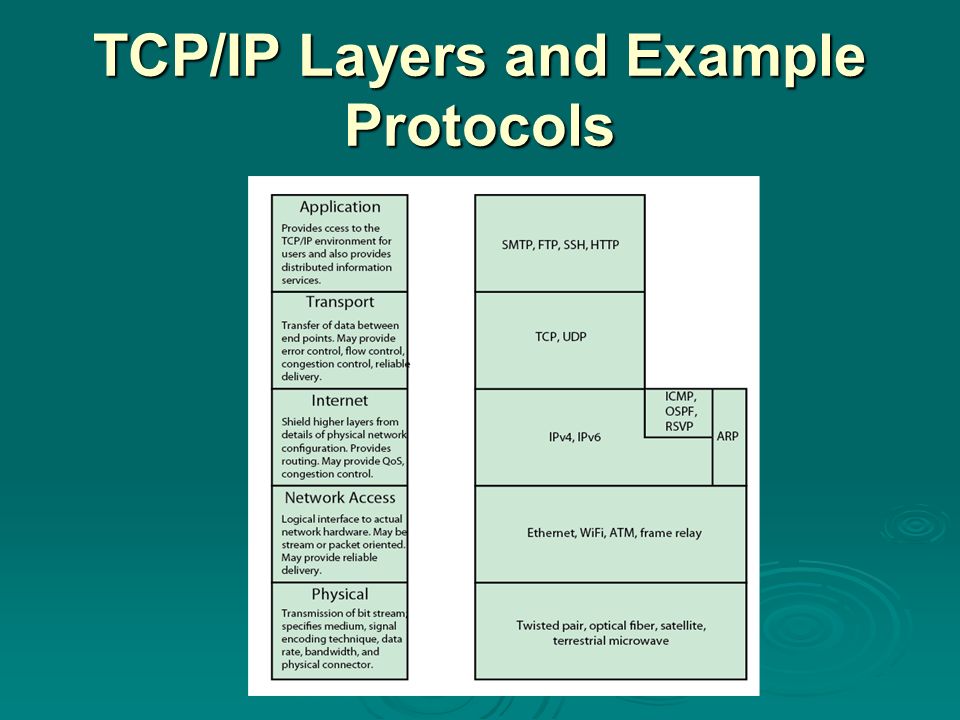
• Physical layer

• Network access/data link layer

• Internet layer

• Host-to-host, or transport layer

• Application layer the physical layer covers the physical interface between a data transmission device (e.g., workstation, computer) and a transmission medium or network. This layer is concerned with specifying the characteristics of the transmission medium, the nature of the signals, the data rate, and related matters.



**The network access/data link layer** is discussed in Section 2.2. This layer is concerned with access to and routing data across a network for two end systems attached to the same network. In those cases where two devices are attached to different networks, procedures are needed to allow data to traverse multiple interconnected networks. This is the function of the **internet** **layer**. The **Internet** **Protocol** **(IP)** is used at this layer to provide the routing function across multiple networks. This protocol is implemented not only in the end systems but also in **routers**. A router is a processor that connects two networks and whose primary function is to relay data from one network to the other on its route from the source to the destination end system.

**The host-to-host layer, or transport layer,** may provide reliable end-to-end service, as discussed in Section 2.2, or merely an end-to-end delivery service without reliability mechanisms. The **Transmission Control Protocol (TCP)** is the most commonly used protocol to provide this functionality.

Finally, the *application layer* contains the logic needed to support the various user applications. For each different type of application, such as file transfer, a separate module is needed that is peculiar to that application.

**Standardization Within a Protocol Architecture:**

**Standards and Protocol Layers**

A protocol architecture, such as the TCP/IP architecture or OSI, provides a framework for standardization. Within the model, one or more protocol standards can be developed at each layer. The model defines in general terms the functions to be performed at that layer and facilitates the standards-making process in two ways:

• Because the functions of each layer are well defined, standards can be developed independently and simultaneously for each layer. This speeds up the standards-making process.

• Because the boundaries between layers are well defined, changes in standards in one layer need not affect already existing software in another layer. This makes it easier to introduce new standards.

Figure 2.9 illustrates the use of a protocol architecture as such a framework. The overall communications function is decomposed into a number of distinct layers. That is, the overall function is broken up into a number of modules, making the interfaces between modules as simple as possible. In addition, the design

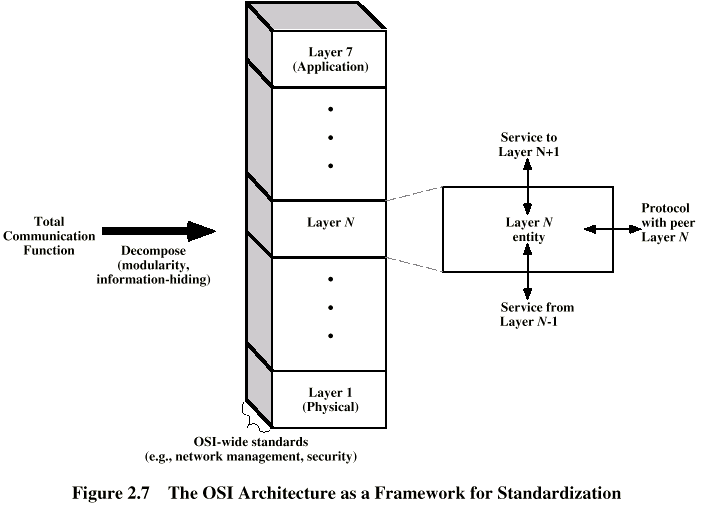
****

FIGURE 2.9

principle of information hiding is used: Lower layers are concerned with greater levels of detail; upper layers are independent of these details. Each layer provides services to the next higher layer and implements a protocol to the peer layer in other systems.

Figure 2.9 also shows more specifically the nature of the standardization required at each layer. Three elements are key:

• **Protocol specification**: Two entities at the same layer in different systems cooperate and interact by means of a protocol. Because two different open systems are involved, the protocol must be specified precisely. This includes the format of the protocol data units exchanged, the semantics of all fields, and the allowable sequence of PDUs.

**• Service definition**: In addition to the protocol or protocols that operate at a given layer, standards are needed for the services that each layer provides to the next higher layer. Typically, the definition of services is equivalent to a functional description that defines what services are provided, but not how the services are to be provided.

**• Addressing**: Each layer provides services to entities at the next higher layer. These entities are referenced by means of a port, or service access point (SAP). Thus, a network service access point (NSAP) indicates a transport entity that is a user of the network service.

**Traditional Internet-Based Applications**

A number of applications have been standardized to operate on top of TCP. We mention three of the most common here.

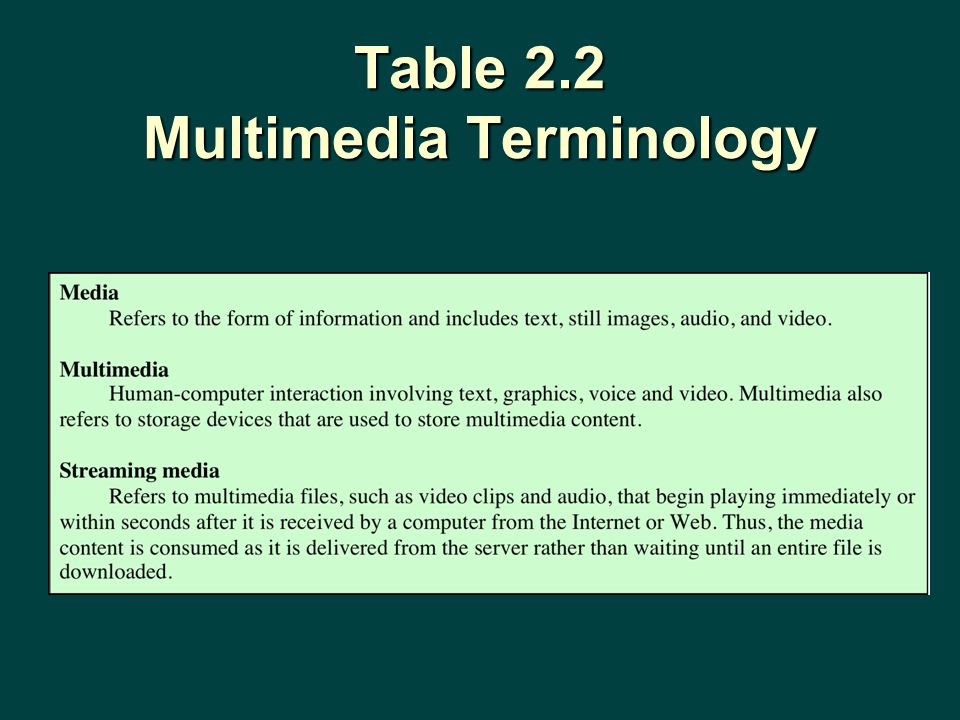
**The Simple Mail Transfer Protocol (SMTP)** provides a basic electronic mail transport facility. It provides a mechanism for transferring messages among separate hosts. Features of SMTP include mailing lists, return receipts, and forwarding. SMTP does not specify the way in which messages are to be created; some local editing or native electronic mail facility is required. Once a message is created, SMTP accepts the message and makes use of TCP to send it to an SMTP module on another host. The target SMTP module will make use of a local electronic mail package to store the incoming message in a user’s mailbox.

**The File Transfer Protocol (FTP**) is used to send files from one system to another under user command. Both text and binary files are accommodated, and the protocol provides features for controlling user access. When a user wishes to engage in file transfer, FTP sets up a TCP connection to the target system for the exchange of control messages. This connection allows user ID and password to be transmitted and allows the user to specify the file and file actions desired. Once a file transfer is approved, a second TCP connection is set up for the data transfer. The file is transferred over the data connection, without the overhead of any headers or control information at the application level. When the transfer is complete, the control connection is used to signal the completion and to accept new file transfer commands.

**SSH (Secure Shell)** provides a secure remote logon capability, which enables a user at a terminal or personal computer to log on to a remote computer and function as if directly connected to that computer. SSH also supports file transfer between the local host and a remote server. SSH enables the user and the remote server to authenticate each other; it also encrypts all traffic in both directions. SSH traffic is carried on a TCP connection.

**Multimedia**

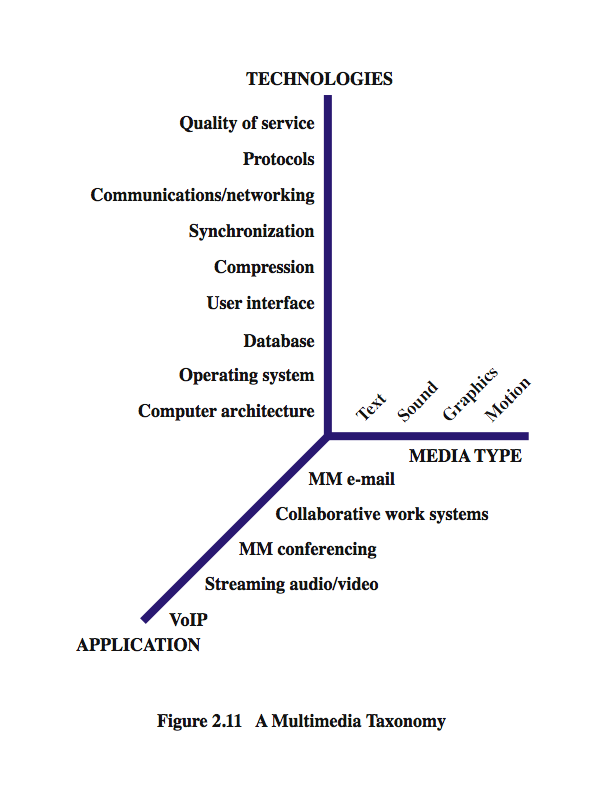
With the increasing availability of broadband access to the Internet has come an increased interest in Web-based and Internet-based **multimedia** **applications**. The terms multimedia and multimedia applications are used rather loosely in the literature and in commercial publications, and no single definition of the term multimedia has been agreed. For our purposes, the definitions in Table 2.2 provide a starting point.

****

One way to organize the concepts associated with multimedia is to look at a taxonomy that captures a number of dimensions of this field. Figure 2.11 looks at multimedia from the perspective of three different dimensions: type of media, applications, and the technology required to support the applications.

**Media Types**:

Typically, the term ***multimedia*** refers to four distinct types of media: text, audio, graphics, and video.

****

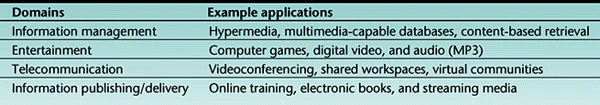
From a communications perspective, the term **text** is self-explanatory, referring to information that can be entered via a keyboard and is directly readable and printable. Text messaging, instant messaging, and text (non-html) e-mail are common examples, as are chat rooms and message boards. However, the term often is used in the broader sense of data that can be stored in files and databases and that does not fit into the other three categories. For example, an organization’s database may contain files of numerical data, in which the data are stored in a more compact form than printable characters.

The term **audio** generally encompasses two different ranges of sound. Voice, or speech, refers to sounds that are produced by the human speech mechanism. Generally, a modest bandwidth (under 4 kHz) is required to transmit voice. Telephony and related applications (e.g., voice mail, audio teleconferencing, and telemarketing) are the most common traditional applications of voice communications technology. A broader frequency spectrum is needed to support music applications, including the download of music files.

The **image** service supports the communication of individual pictures, charts, or drawings. Image-based applications include facsimile, computer-aided design (CAD), publishing, and medical imaging. Images can be represented in a vector graphics format, such as is used in drawing programs and PDF files. In a raster graphics format, an image is represented as a two-dimensional array of spots, called pixels.2 The compressed JPG format is derived from a raster graphics format. The video service carries sequences of pictures in time. In essence, video makes use of a sequence of raster-scan images.

**Multimedia Applications**

The Internet, until recently, has been dominated by information retrieval applications, e-mail, and file transfer, plus Web interfaces that emphasized text and images. Increasingly, the Internet is being used for multimedia applications that involve massive amounts of data for visualization and support of real-time interactivity. Streaming audio and video are perhaps the best known of such applications. An example of an interactive application is a virtual training environment involving distributed simulations and real-time user interaction [VIN98]. Some other examples are shown in Table 2.3.

****

[GONZ00] lists the following multimedia application domains:

• **Information** **systems**: These applications present information using multimedia. Examples include information kiosks, electronic books that include audio and video, and multimedia expert systems.

• **Communication systems**: These applications support collaborative work, such as videoconferencing.

**• Entertainment systems**: These applications include computer and network games and other forms of audiovisual entertainment.

• **Business systems**: These applications include business-oriented multimedia presentation, video brochures, and online shopping.

• **Educational systems**: These applications include electronic books with a multimedia component, simulation and modeling applets, and other teaching support systems.

One point worth noting is highlighted in Figure 2.11. Although traditionally the term multimedia has connoted the simultaneous use of multiple media types (e.g., video annotation of a text document), it has also come to refer to applications that require real-time processing or communication of video or audio alone. Thus, voice over IP (VoIP), streaming audio, and streaming video are considered multimedia applications even though each involves a single media type.

**Multimedia Technologies**

Figure 2.11 lists some of the technologies that are relevant to the support of multimedia applications. As can be seen, a wide range of technologies is involved. The lowest four items on the list are beyond the scope of this book. The other items represent only a partial list of communications and networking technologies for multimedia. These technologies and others are explored throughout the book. Here, we give a brief comment on each area.

• **Compression**: Digitized video, and to a much lesser extent audio, can generate an enormous amount of traffic on a network. A streaming application, which is delivered to many users, magnifies the traffic. Accordingly, standards have been developed for producing significant savings through compression. The most notable standards are JPG for still images and MPG for video.

• **Communications/networking**: This broad category refers to the transmission and networking technologies (e.g., SONET, ATM) that can support high volume multimedia traffic.

• **Protocols**: A number of protocols are instrumental in supporting multimedia traffic. One example is the Real-time Transport Protocol (RTP), which is designed to support **inelastic traffic** (traffic that does not easily adapt, if at all, to changes in delay and throughput across an internet). RTP uses buffering and discarding strategies to assure that real-time traffic is received by the end user in a smooth continuous stream. Another example is the Session Initiation Protocol (SIP), an application-level control protocol for setting up, modifying, and terminating real-time sessions between participants over an IP data network.

• **Quality of service (QoS):** The Internet and its underlying local area and wide area networks must include a QoS capability to provide differing levels of service to different types of application traffic. A QoS capability can deal with priority, delay constraints, delay variability constraints, and other similar requirements.

All of these matters are explored subsequently in this text.