6

Data Communications

In the beginning of this chapter we clarify some key terms that we need to describe a certain data communications principle or a system. Then we introduce the concept of data communication protocols, trying to convey a concrete feel for the layered data communication protocol stacks and the reason why we define data communication architectures with the help of the protocol layers. Then we describe various data communications systems used in access networks and for local- and wide-area data communications. In the latter half of this chapter, we concentrate on the Internet and describe its structure, operation, and services.

6.1 Principles of Data Communications

The first data communications system was the telegraph. It was invented more than 100 years ago. The letters to be transmitted were converted into a code called Morse code. The codes were transmitted as pulses along a wire or as radio-frequency bursts in the case of wireless telegraph. Then the development of data communications slowed, but during the last few decades data communications have expanded rapidly as computers have become tools for everyone in both business and residential environments.

6.1.1 Computer Communications

Modern computers manipulate bits, binary symbols, of electrical energy. When a computer communicates with another computer it sends these bits along a cable between them. This is relatively easy if the computers are within the same room or a building. If the distance is longer, a telecommunications network is required that provides an end-to-end communications channel. Data communications can be accomplished by means of many various alternatives, some of which we discuss in the following sections.

6.1.2 Serial and Parallel Data Communications

In a transmission network only one channel is usually allocated for one end-to-end connection in each direction. Let us use as an example source of data a simple *American Standard Code for Information Interchange* (ASCII) terminal. We press keys on the keyboard and each keystroke generates a 7-bit binary word (8 bits with parity) corresponding to the letter or number of the key pressed. For example character *a* corresponds to the binary sequence 1000011 (the first bit on the left) [1]. If we have only one channel available, we have to send bits of this word in turn (first bit on the left) to the channel; such a case represents *serial* data transmission (see Figure 6.1).

When serial transmission is used between a computer and its peripheral device, a parallel clock signal may be used for timing. In serial transmission

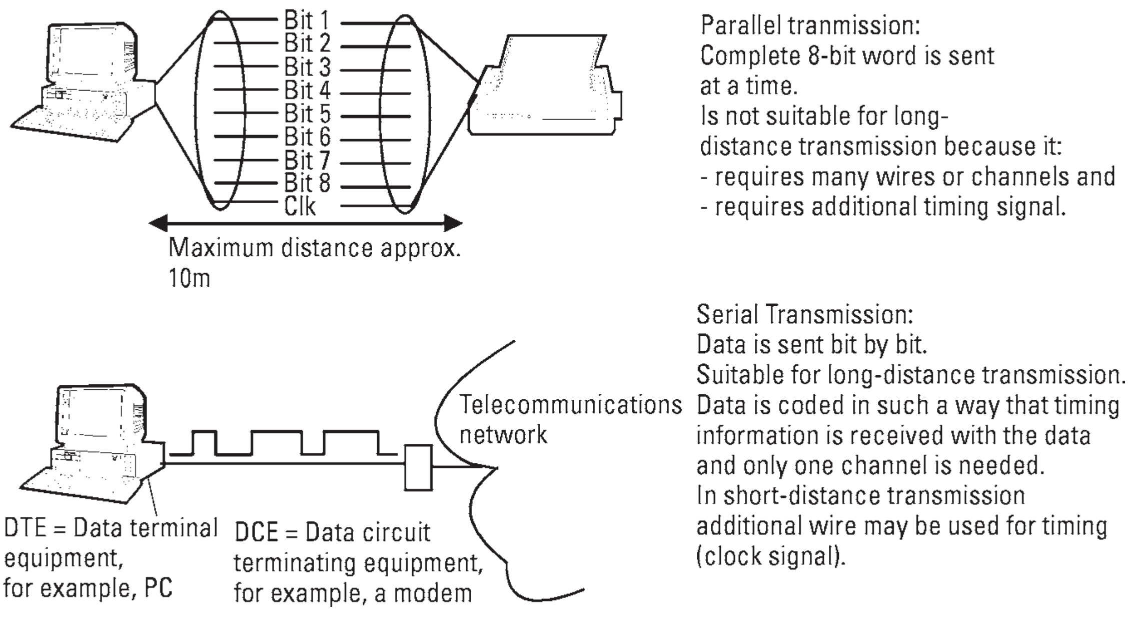

Figure 6.1 Serial and parallel transmission.

over longer distances we want to manage with one channel and we have to use a line code to insert timing information into the data stream. This synchronization information enables the receiver to determine when it has to detect each individual received bit. How we implement this depends on whether we use an asynchronous or synchronous transmission mode, as described in Section 6.1.3.

If a computer needs to communicate with, for example, a printer in the same room, parallel communication is often used. A special cable with several wires is provided between the computer and the printer and all 8 bits of a data word, corresponding to one character, are transferred at the same time in parallel over the cable. Parallel data transmission is much quicker than serial, but we can typically use it only over short distances. The maximum is usually of the order of 10m.

Communicating terminal devices in data communications are called *data terminal equipment* (DTE) and the equipment that terminates the transmission channel that goes through the network is called *data circuit-terminating equipment* (DCE). A modem that we use for data transmission over a telephone network is a typical example of DCE. Many different interface specifications exist for DTE and DCE, and the most common standards are defined by the ITU-T and the *Electronic Industries Association* (EIA). One of the most common data interfaces is ITU-T's V.24/V.28, which corresponds to EIA Standard RS-232-C.

6.1.3 Asynchronous and Synchronous Data Transmission

Over longer distances we use serial transmission either in an asynchronous or synchronous transmission mode. Serial transmission over long distance requires that the timing information for the receiver be transmitted together with the data so that a separate clock signal is not required.

In asynchronous transmission only a small number of bits are transmitted at a time, usually 8 bits that correspond to one ASCII character. In the beginning of each block of 8 bits of data, a *start bit* is sent to indicate to the receiver that it should prepare to receive 8 bits of data (see Figure 6.2). For synchronization the receiver has to know the data rate, which has to be set in advance, so that when it detects the start bit it is able to receive the few following bits. After these bits a *stop bit* is sent that terminates the 8-bit data block. The next block of data is synchronized independently with the help of a new start bit preceding the data bits.

In asynchronous transmission, a simple error-detecting scheme called *parity* can be used. We may use even or odd parity error checking. If even

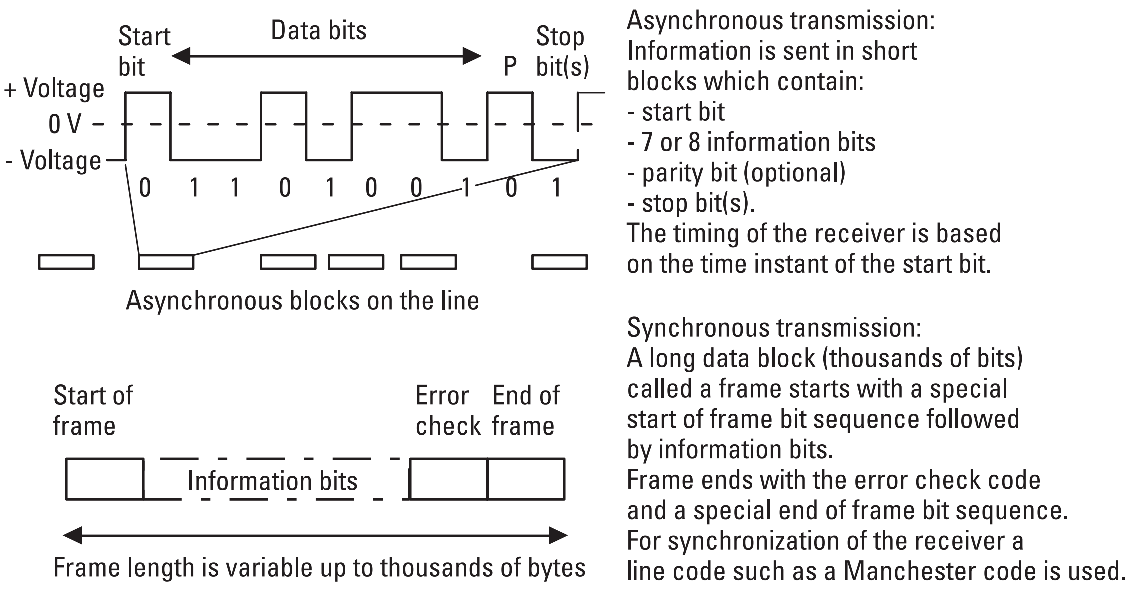

Figure 6.2 Asynchronous and synchronous transmission.

parity is used, the total number of "1" bits in the block, including data bits and the parity bit, is set to be even with the help of the parity bit. In the case of odd parity, the parity bit is set to "1" or "0" so that the total number of "1" bits in the block is odd. To detect possible transmission errors, the receiver determines whether the received number of "1" bits is even or odd depending on the parity agreed. We will see later that this parity check method is a simple example of a data link layer protocol.

Asynchronous transmission is used for the transmission of ASCII characters in conventional terminal–mainframe computer communications. For larger information blocks it is used in some file transfer protocols such as KERMIT and X-LINK. In these protocols special "start of block" characters are sent at the beginning. Then information follows as asynchronous words and at the end special "end of block" characters are sent.

Synchronous transmission is a more modern principle for transmitting a large amount of information in a frame (see Figure 6.2). Each frame starts with a special start-of-frame bit sequence and the frame may contain more than 1,000 bytes of information. Each frame also contains error control words and an end-of-frame sequence. The receiver uses the error control section of the frame to detect if errors have occurred in transmission. The most common detection method for error detection is a *cyclic redundancy check* (CRC). It is much more reliable than the parity check method discussed previously. In the case of errors the transmitter retransmits the frame in error. In the most common protocols the receiver sends an acknowledgment to the transmitter in the other transmission direction for received error-free frame or frames. If errors have occurred, the frame is not acknowledged in a predefined period of time and the transmitter sends it again.

In asynchronous transmission the start bit provided the required timing information for each byte of data. Most synchronous transmission methods are so-called "bit-oriented" protocols in which data blocks are not divided into separate bytes because many types of information, such as graphics, is not presented as a set of bytes. Unique start-of-frame and end-of-frame sequences or flags are used to provide frame synchronization. These flags should be unique and actual data must not include similar data sequences. One common method used to avoid frame misalignment is to use bit stuffing or zero insertion, as shown in Figure 6.3. Consider a flag (01111110) used in the popular *high-level data link control* (HDLC) protocol. After the start-of-frame flag the sequence of six subsequent 1's is not allowed in the data section of the frame. To avoid that, a 0 is inserted in the end of each sequence of five subsequent 1's. In the receiver each 0 following five subsequent 1's is discarded. If binary 1 follows five subsequent 1's, the frame is declared to be finished (end-of-frame flag) [1].

Synchronous transmission requires that the bit timing information be inserted into the data stream itself with the help of line coding because frames are very long. As an example, many LANs use the Manchester line code that we described in Chapter 4.

The principles we have discussed above are contained in physical layer and data link layer definitions in the data communication architecture

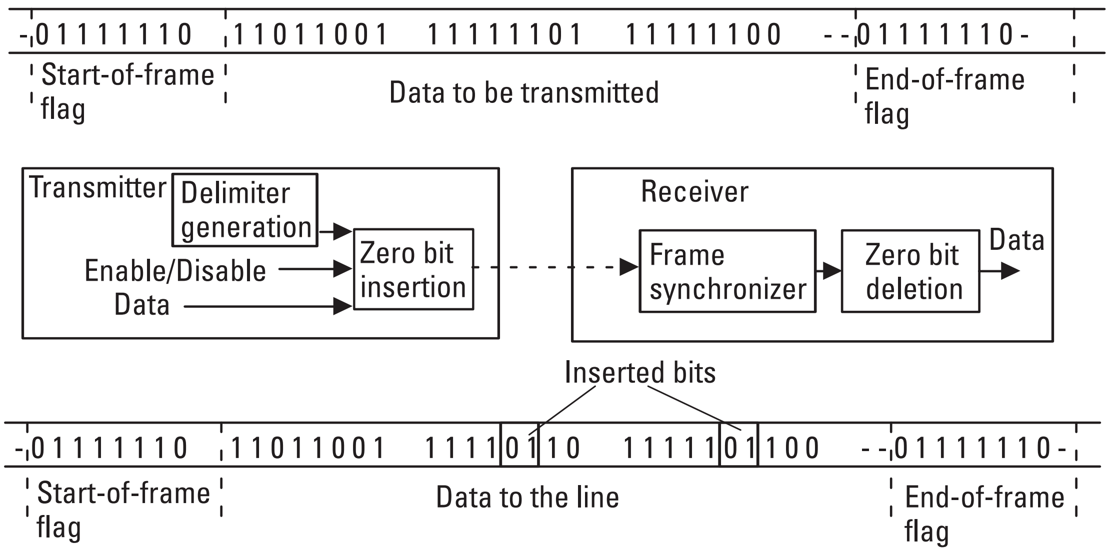

Figure 6.3 Bit stuffing or zero insertion.

described in Section 6.3. As we will see later, these two protocol layers deal with aspects of how data communications over a physical connection between two machines are arranged.

Connection from a computer to another through the data communications network requires a switching function, which routes data frames or packets from the source host to the destination host. We introduce next the basic alternatives for routing and, as we will see in Section 6.3, these functions are implemented into the data link layer or network layer in the protocol hierarchy.

6.2 Circuit and Packet Switching

We can divide data connections through a telecommunications network into different categories based on the principle of how the communications circuit is built between the communicating devices. Data communications through the telecommunications network may use three basic different types of circuits:

1. *Leased or dedicated:* The cost of a leased line is fixed per month and depends on the capacity and length of the connection.
2. *Circuit switched or dial-up:* The cost of switched service depends on the time the service is used, the data rate, and the distance.
3. *Packet switched:* The cost is often fixed and depends on the interface data rate. In some packet-switched networks cost may depend on the amount of transferred data. Agreements with the service provider may specify other parameters that influence the cost, such as the maximum data rate or average data rate.

For corporate data networks, the leased-line solution is often attractive when the LANs of offices in a region need to be interconnected. The network operator provides a permanent circuit and the monthly cost is fixed and depends only on the agreed-on data rate. Over long distances, however, leased lines become expensive and switched service is often preferred. In such a service, several corporate networks share transmission capacity and the cost of the backbone of the telecommunications network operator.

Within the switched category there are two subcategories, circuit- and packet-switched networks as shown in Figure 6.4, both of which are used for data transmission. Figure 6.4 also shows some sample networks and what switching principles they use.

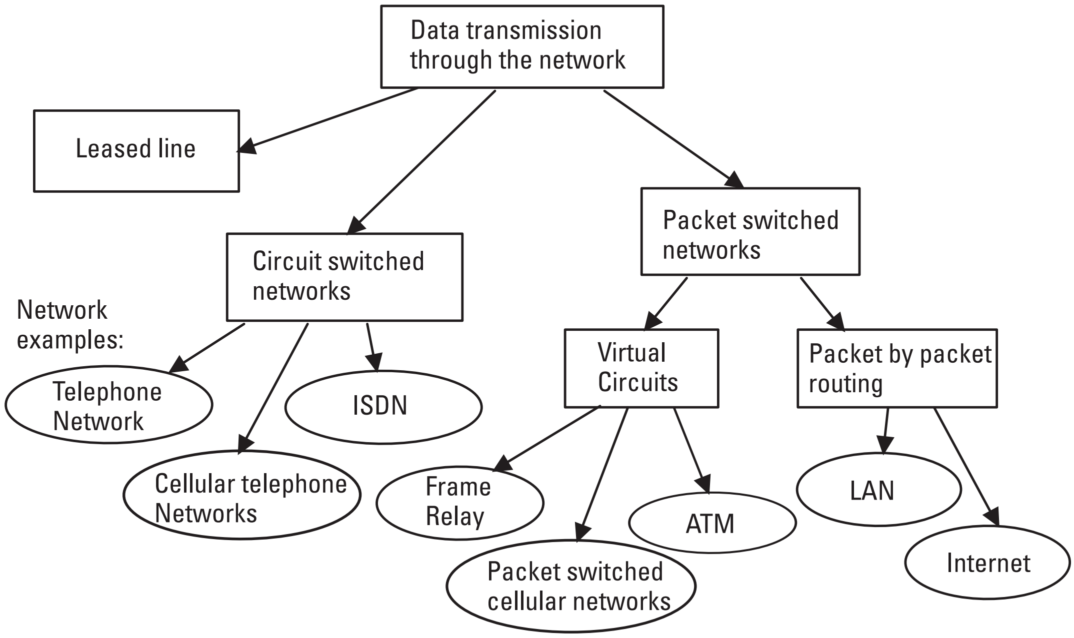

Figure 6.4 Leased lines and circuit- and packet-switched networks.

6.2.1 Circuit Switching

Circuit-switched networks provide fixed bandwidth and very short and fixed delay. It is the primary technology for voice telephone, video telephone and video conferencing. The disadvantage is that it is inflexible for data communications where the demand for transmission data rate is far from constant but varies extensively over short time scales.

Some older generation data networks used the circuit switching principle. In the beginning a circuit-switched connection is dialed up by the data source. The routing is based on the destination subscriber number given when the circuit is established. The connection is released after the communication is over (see Figure 6.5). During a conversation, the data capacity of the connection is fixed and it is reserved only for this conversation regardless of whether the data capacity is used or not. At the end of the call, the circuit is released. ISDN as well as the telephone network use the circuit-switching principle.

6.2.2 Packet Switching

Packet-switched networks are specially designed for data communication. The source data are split into packets containing route or destination identifications. The packets are routed toward the destination by packet-switching nodes on the path through the network. The major drawback of the packet-switched technology is that it usually cannot provide a service for applications that require constant and low delay. There are two basic types of packet-switched networks as illustrated in Figure 6.5: virtual circuits and datagram transmission.

In the case of virtual circuits, the virtual connection is established at the beginning of each conversation or it is permanently set up and every packet belonging to a certain connection is transmitted via the same established route. The main difference between circuit-switched physical circuits and virtual circuits is that many users share the capacity of the transmission lines and channels between network nodes if virtual instead of physical circuits are used. At a certain moment active users may use all the available capacity if other users are not transmitting anything. The complete address information is not needed in the packets when the connection is established. Only a short connection identifier is included in each packet to define the virtual circuit to which the packet belongs. The operation of switched virtual circuits is explained in more detail in Section 6.2.4.

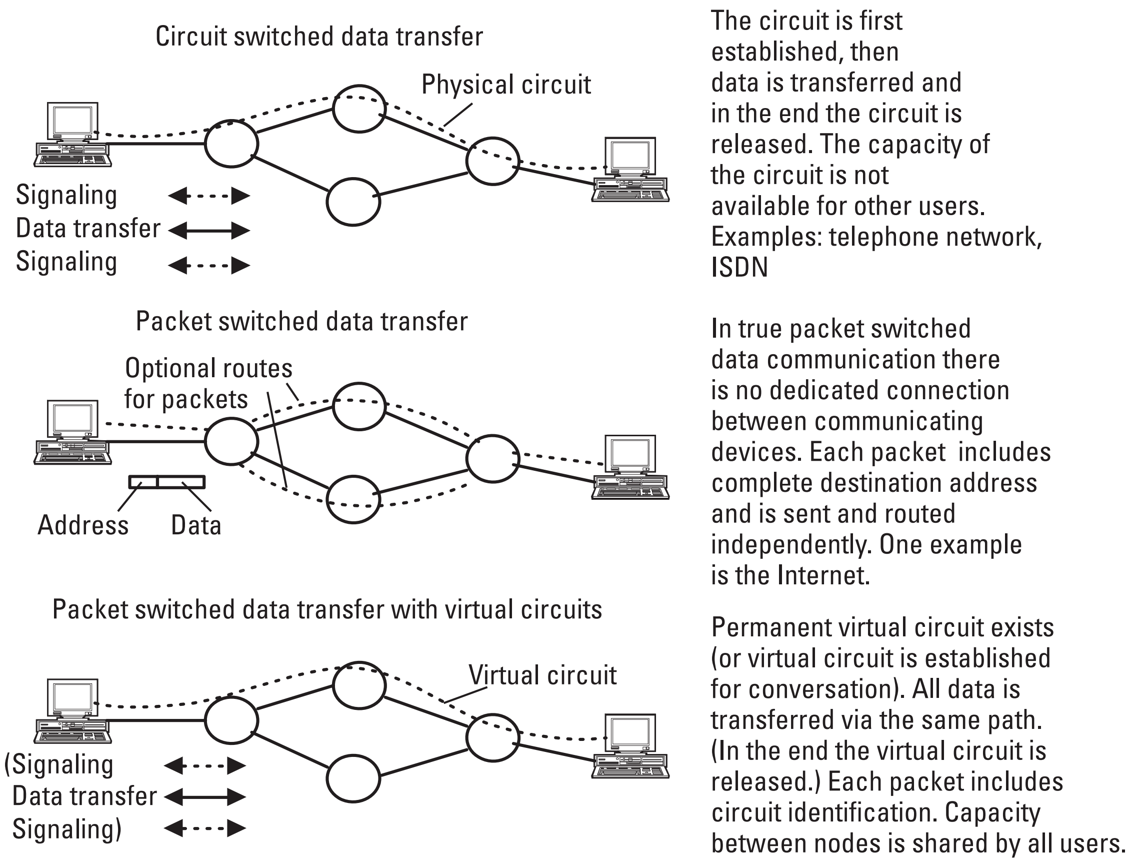

Figure 6.5 Circuit- and packet-switched data transfer.