CHAPTER 3

Digital Transmission Fundamentals

Modern communication networks based on digital transmission systems have the potential to carry all types of information and hence to support many types of applications. We saw in Chapter 1 that the design of the early network architectures was tailored to very specific applications that led to the development of corresponding transmission systems. Telegraphy was developed specifically for the transfer of text messages. Morse and Baudot pioneered the use of binary representations for the transfer of text and developed digital transmission systems for the transfer of the resulting binary information. Later telephony was developed for the transfer of voice information. Initially the voice information was transmitted using analog transmission systems. The invention of pulse code modulation (PCM) enabled voice to be transmitted over digital transmission networks. In the same way that the Morse and Baudot codes standardized the transfer of text, PCM standardized the transfer of voice in terms of 0s and 1s. We are currently undergoing another major transition from analog to digital transmission technology, namely, the transition from analog television systems to entirely digital television systems. When this transition is complete, all major forms of information will be represented in digital form. This change will open the way for the deployment of digital transmission networks that can transfer the information for all the major types of information services.

In this chapter we present the fundamental concepts concerning digital transmission. These concepts form the basis for the design of the digital transmission systems that constitute the physical layer of modern network architectures. The chapter is organized into the following sections:

1. *Digital representation of information.* We consider different types of information and their representation in digital form. Text, image, voice, audio, and video are used as examples.
2. *Why digital transmission?* We explain the advantages of digital transmission over analog transmission. We present the key parameters that determine the transmission

capacity of a physical medium. We also indicate where various media are used in digital transmission systems. This section is a summary of the following three sections.[1]

3. *Digital representation of analog signals.* We explain how PCM is used to convert an analog signal into a binary information stream. Voice and audio signals are used as examples.
4. *Characterization of communication channels.* We discuss communication channels in terms of their ability to transmit pulse and sinusoidal signals. We introduce the concept of bandwidth as a measure of a channel's ability to transmit pulse information.
5. *Fundamental limits of digital transmission.* We discuss binary and multilevel digital transmission systems, and we develop fundamental limits on the bit rate that can be obtained over a channel.
6. *Line coding.* We introduce various signal formats for transmitting binary information and discuss the criteria for selecting an appropriate line code.
7. *Modems and digital modulation.* We discuss digital transmission systems that use sinusoidal signals, and we explain existing telephone modem standards.
8. *Properties of transmission media.* We discuss copper wire, radio, and optical fiber systems and their role in access and backbone digital networks. Examples from various physical layer standards are provided.
9. *Error detection and correction.* We present coding techniques that can be used to detect and correct errors that may occur during digital transmission. These coding techniques form the basis for protocols that provide reliable transfer of information. Protocols that use error detection are found in the physical, data link, network, and transport layers.

3.1 DIGITAL REPRESENTATION OF INFORMATION

Applications that run over networks involve the transfer of information of various types. Some applications involve the transfer of blocks of text characters, e-mail, for example. Other applications involve the transfer of a stream of information, such as telephony. In the case of text, the information is already in digital form. In the case of voice, the information is analog in nature and must be converted into digital form. This section focuses on the number of bits required to represent various types of information, for example, text, speech, audio, data, images, and video. In the case of block-oriented information, we are interested in the number of bits required to represent a block. In the case of stream-oriented information, we are interested in the *bit rate* (number of bits/second) required to represent the information.

It is useful to identify which layers in the OSI reference model we are dealing with in this section. In general, the information associated with an application is generated above the application layer. The blocks or streams of information generated by the application must be handled by all the lower layers in the protocol stack. Ultimately,

[1]This arrangement allows Sections 3.3 to 3.5 to be skipped in courses that are under tight time constraints.

the physical layer must carry out the transfer of all the bits generated by the application and the layers below. In a sense, the application generates flows of information that need to be carried across the network; the digital transmission systems at the physical layer provide the pipes that actually carry the information flows across the network. The purpose of this section, then, is to introduce the important examples of information types, such as text, voice, audio, and video, so that we can relate their requirements to the bit rates provided by transmission systems.

3.1.1 Block-Oriented Information

Information can be grouped into two broad categories: information that occurs naturally in the form of a single *block* and *stream information* that is produced continuously and that needs to be transmitted as it is produced. Table 3.1 gives examples of *block-oriented information*, which include data files, black-and-white documents, and pictures.

The most common examples of block information are files that contain text, numerical, or graphical information. We routinely deal with these types of information when we send e-mail and when we retrieve documents. These blocks of information can range from a few bytes to several hundred kilobytes and occasionally several megabytes. The normal form in which these files occur can contain a fair amount of statistical redundancy. For example, in English text certain characters and patterns such as *e* and *the*, occur very frequently. *Data compression* utilities such as compress, zip, and other variations exploit these redundancies to encode the original information into files that require fewer bits to transfer and less disk storage space.[2] Some modem standards also apply these data compression schemes to the information prior to transmission. The *compression ratio* is defined as the ratio of the number of bits in the original file to the number of bits in the compressed file. Typically the compression ratio for these types of information is two or more, thus providing an apparent doubling or more of the transmission speed or storage capacity.

TABLE 3.1 Block-oriented information.

Information type	Data compression technique	Format	Uncompressed	Compressed (compression ratio)	Applications
Text files	Compress, zip, and variations	ASCII	kbytes to Mbytes	(2–6)	Disk storage, file transfer
Scanned black-and-white documents	CCITT Group 3 facsimile standard	A4 page at 200 × 100 pixels/inch and options	256 kbytes	15–54 kbytes (1-D) 5–35 kbytes (2-D) (5–50)	Facsimile transmission, document storage
Color images	JPEG	8 × 10 inch photo scanned at 400 pixels/inch	38.4 Mbytes	1.2–8 Mbytes (5–30)	Image storage or transmission

[2]The details of the data compression techniques discussed in this section are found in Chapter 12.

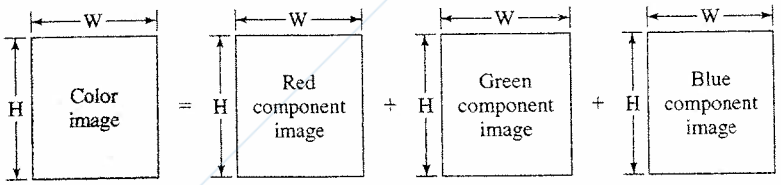
Certain applications require that a block of information be delivered within a certain maximum **delay**. The time to deliver a block of L bits of information over a transmission system of R bits/second consists of the propagation delay and the block transmission time: delay $= t_{prop} + L/R$. The propagation delay $t_{prop} = d/v$ where d is the distance that the information has to travel and v is the speed of light in the transmission medium. Clearly, the designer cannot change the speed of light, but the distance that the information has to travel can be controlled through the placement of the file servers. The time to transmit the file can be reduced by increasing the transmission bit rate R. In the following discussion we consider several examples where delay and bit rate are traded off.

A facsimile document system scans a black-and-white document into an array of dots that are either white or black. A *pixel* is defined as a single dot in a digitized image. The CCITT Group 3 facsimile standards provide for resolutions of 200, 300, or 400 dots per horizontal inch and 100, 200, or 400 vertical dots per inch. For example, a standard A4 page at 200 × 100 pixels/inch (slightly bigger than 8.5 × 11 inches) produces 256 kilobytes prior to compression. At a speed of 28.8 kbps, such an uncompressed page would require more than 1 minute to transmit. Existing fax compression algorithms can reduce this transmission time typically by a factor of 8 to 16.

An individual color image produces a huge number of bits. For example, an 8 × 10-inch picture scanned at a resolution of 400 × 400 pixels per square inch yields $400 \times 400 \times 8 \times 10 = 12.8$ million pixels; see Table 3.1. A color image is decomposed into red, green, and blue subimages as shown in Figure 3.1. Normally eight bits are used to represent each of the red, green, and blue color components, resulting in a total of 12.8 megapixels × 3 bytes/pixel = 38.4 megabytes. At a speed of 28.8 kbps, this image would require about 3 hours to transmit! Clearly, data compression methods are required to reduce these transmission times.

The *Graphics Interchange Format (GIF)* takes image data, in binary form, and applies lossless data compression. *Lossless data compression* schemes produce a compressed file from which the original data can be recovered *exactly*. (Facsimile and file compression utilities also use lossless data compression.) However, lossless data compression schemes are limited in the compression rates they can achieve. For this reason, GIF is used mainly for simple images such as line drawings and images containing simple geometrical shapes.

Lossy data compression schemes produce a compressed file from which only an *approximation* to the original information can be recovered. Much higher compression ratios are possible. In the case of images, lossy compression is acceptable as long as there is little or no visible degradation in image quality. The *Joint Photographic Experts*



Total bits before compression = 3 × H × W pixels × B bits/pixel = 3 HWB

FIGURE 3.1 The three components of a color image.

TABLE 3.2 Properties of audio and video stream information.

Information type	Compression technique	Format	Uncompressed	Compressed	Applications
Voice	PCM	4 kHz voice	64 kbps	n/a	Digital telephony
Voice	ADPCM (+ silence detection)	4 kHz voice	64 kbps	16–32 kbps	Digital telephony, voice mail
Voice	Residual-excited linear prediction	4 kHz voice	64 kbps	8–16 kbps	Digital cellular telephony
Audio	MPEG audio MP3 compression	16–24 kHz audio	512–748 kbps	32–384 kbps	MPEG audio
Video	H.261 coding	176 × 144 or 352 × 288 frames at 10–30 frames/second	2–36.5 Mbps	64 kbps–1.544 Mbps	Video conferencing
Video	MPEG-2	720 × 480 frames at 30 frames/second	249 Mbps	2–6 Mbps	Full-motion broadcast video, DVD
Video	MPEG-2	1920 × 1080 frames at 30 frames/second	1.6 Gbps	19–38 Mbps	High-definition television

Group (JPEG) standard provides a lossy compression algorithm that can be adjusted to balance image quality versus file size.[3] The compression ratio that is achieved for a given image depends on the degree of detail and busyness of the content. Images that contain a few large, smooth objects are highly compressible, as, for example, in a chart containing a few large circles with uniform color in each picture. Images that contain large numbers of small objects, for example, a picture of the fans in the stands in a stadium, will be much less compressible. As an example, JPEG can typically produce a high-quality reproduction with a compression ratio of about 15. Combined with the fastest telephone modems, say 56 kbps, this compression ratio reduces the transmission time of the image in Table 3.1 to several minutes. Clearly in the case of images, we either make do with lower resolution and/or lower quality images or we procure higher speed communications.

3.1.2 Stream Information

Information such as voice, music, or video is produced in a steady stream. Table 3.2 lists the properties of this type of information. In the case of a voice or music signal, the sound, which consists of variations in air pressure, is converted into a voltage that

[3] JPEG is discussed in Chapter 12.

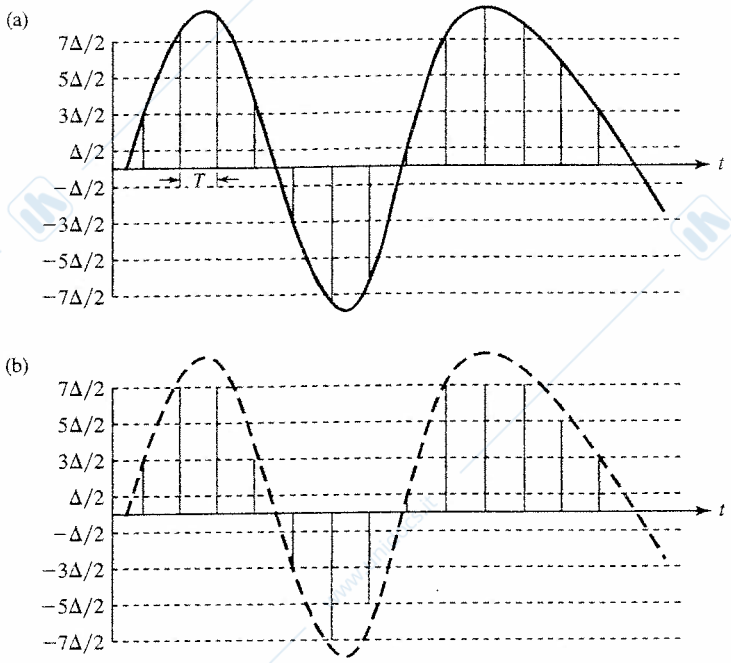


FIGURE 3.2 Sampling of a speech signal: (a) original waveform and the sample values; (b) original waveform and the quantized values.

varies continuously with time assuming values over a continuous range. We refer to these signals as *analog* signals.

The first step in digitizing an analog signal is to obtain sample values of the signal every T seconds as shown in Figure 3.2a. Clearly, the value of T between samples depends on how fast the signal varies with time. The **bandwidth of a signal** is a measure of how fast the signal varies. Bandwidth is measured in cycles/second or Hertz. A basic result from signal theory is that if a signal has bandwidth W then the minimum sampling rate is $2W$ samples/second. For example, for a **pulse code modulation (PCM)** telephone-quality voice, the signal has a bandwidth of 4 kHz and so the signal is sampled at a rate of 8000 samples/second, that is, $T = 1/8000 = 125$ microseconds as shown in Figure 3.2a.[4]

The second step in digitizing a signal involves *quantizing* each of the sample values. Figure 3.2b shows the operation of a quantizer: In this example, each of the signal samples is approximated by one of eight levels. Each level can then be represented by a three-bit number. Clearly, the accuracy of the reproduced signal increases as the number of bits used to represent each sample is increased. In the case of telephone systems, the PCM voice samples are represented by 8 bits in resolution, resulting in a bit rate for PCM of 8000 samples/second $\times$ 8 bits/sample = 64 kbps.

[4]PCM is discussed in Section 3.3.

Many applications involve information that is produced continuously and that needs to be transferred with small delay. For example, communications between people is real-time and requires a maximum delay of about 250 ms to ensure interactivity close to that of normal conversation. Suppose that an information source produces information at a rate of R_s bps. Suppose that the digital transmission system transfers information at a rate of R bps. To attain real-time communications it is then necessary that the transmission rate R be greater than or equal to the rate R_s at which the source produces information. If the source produces information faster than the transmission system can transfer it, then a backlog of information will build up at the input to the transmission system. For example, in the telephone network the digitized voice signal has a bit rate of $R_s = 64$ kbps and the network provides transmission channels of bit rate $R = 64$ kbps. If the real-time requirement for the transfer of information is removed, then the binary encoded stream produced by a signal such as audio or video can be stored and sent as a block. In this sense, the distinction between block and stream information depends on the requirements of the situation.

The high cost of transmission in certain situations, for example, cellular radio systems, has led to the development of more complex algorithms for reducing the bit rate while maintaining the quality of a voice signal that one encounters in conventional networks. Differential PCM (DPCM) encodes the difference between successive samples of the voice signal. Adaptive DPCM (ADPCM) adapts to variations in voice signal level, that is, the loudness of the signal. Linear predictive methods adapt to the type of sound, for example, *ee* versus *ss*. These systems can reduce the bit rate of telephone quality voice to the range 8 to 32 kbps. Despite the fact that they are "lossy," these schemes achieve compression and high quality due to the imperceptibility of the approximation errors.

Music signals vary much more rapidly, that is, have higher bandwidth than voice signals. Thus for example, audio compact disk (CD) systems assume a bandwidth of 22 kHz and sample the music signals at 44 kilosamples/second and at a resolution of 16 bits/sample. For a stereo music system, this sampling results in a bit rate of 44,000 samples/second $\times$ 16 bits/sample $\times$ 2 channels = 1.4 Mbps. One hour of music will then produce 317 Mbytes of information. More complex compression techniques can be used to reduce the bit rate of the digitized signal. For example, the subband coding technique used in the MPEG audio standard, for example, MP3, can reduce the bit rate by a factor of 14 to about 100 kbps.

Video signals ("moving pictures" or "flicks") can be viewed as a succession of pictures that is fast enough to give the human eye the appearance of continuous motion. If there is very little motion, such as a close-up view of a face in a videoconference, then the system needs to transmit only the differences between successive pictures. Typical videoconferencing systems operate with frames of 176 $\times$ 144 pixels at 10 to 30 frames/second as shown in Figure 3.3a. The color of each pixel is initially represented by 24 bits, that is, 8 bits per color component. When compressed, these videoconferencing signals produce bit rates in the range of several hundred kilobits/second as shown in Table 3.2.

Broadcast television requires greater resolution (720 $\times$ 480 pixels/frame) than videoconferencing requires, as shown in Figure 3.3b, and can contain a high degree of motion. The MPEG-2 coding system can achieve a reduction from the uncompressed

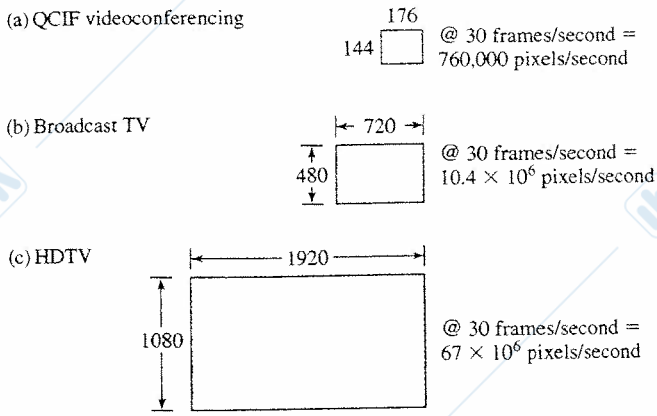


FIGURE 3.3 Video image pixel rates.

bit rate of 249 Mbps to the range of 2 to 6 Mbps. Current DVD movies are encoded in the range of 4 Mbps.[5] The Advanced Television Systems Committee (ATSC) U.S. standard for high-definition television applies the MPEG-2 coding system in a system that operates with more detailed 1920×1080 pixel frames at 30 frames/second as shown in Figure 3.3c. The 16:9 aspect ratio of the frame gives a more theaterlike experience; ordinary television has a 4:3 aspect ratio. The uncompressed bit rate is 1.6 gigabits/second. The MPEG-2 coding can reduce this to 19 to 38 Mbps, which can be supported by digital transmission over terrestrial broadcast and cable television systems.[6]

We have seen in this section that the information generated by various applications can span a broad range of bit rates. Even a specific information type, for example, voice or video, can be represented over a wide range of bit rates and qualities. There is a cost associated with the signal processing required to compress a signal, and in general, this cost increases as the compression ratio increases. On the other hand, there is also a cost associated with the bit rate of the transmission system. The choice of bit rate to represent an information signal and bit rate to transmit the signal depends on the relative value of these two costs. For example, in cellular telephony the cost of transmission is expensive and high-performance voice compression is used. In the case of file transfer in a high-bandwidth local area network, the cost of transmission is cheap, so compression is seldom used. However, file transfer using a long-distance telephone line is expensive, so compression is highly desirable for large files.

[5]Some DVD players can display the variation of the bit rate while playing a movie. Try the display menu in a DVD player and check for the variation in bit rate for different types of movies, for example, cartoons versus sport scenes.

[6]MPEG and MP3 are discussed in Chapter 12.

3.2 WHY DIGITAL COMMUNICATIONS?[7]

A transmission system makes use of a physical **transmission medium** or **channel** that allows the propagation of energy in the form of pulses or variations in voltage, current, or light intensity as shown in Figure 3.4. Copper wire pairs, coaxial cable, optical fiber, infrared, and radio are all examples of transmission media. In analog communications the objective is to transmit a waveform, which is a function that varies continuously with time, as shown in Figure 3.5a. For example, the electrical signal coming out of a microphone corresponds to the variation in air pressure corresponding to sound. This function of time must be reproduced exactly at the output of the analog communication system. In practice, communications channels cannot achieve perfect reproduction, so some degree of distortion is unavoidable.

In digital transmission the objective is to transmit a given symbol that is selected from some finite set of possibilities. For example, in binary digital transmission the objective is to transmit either a 0 or a 1. This can be done, for instance, by transmitting positive voltage for a certain period of time to convey a 1 or a negative voltage to convey a 0, as shown in Figure 3.5b. The task of the receiver is to determine the input symbol. The positive or negative pulses that were transmitted for the given symbols can undergo a great degree of distortion. Where signaling uses positive or negative voltages, the system will operate correctly as long as the receiver can determine whether the original voltage was positive or negative. For example, the waveform in Figure 3.5b corresponds to binary sequence 1, 0, 1. Despite significant distortion, the original binary sequence can still be discerned from the received signal shown in the figure.

3.2.1 Comparison of Analog and Digital Transmission

The cost advantages of digital transmission over analog transmission become apparent when transmitting over a long distance. Consider, for example, a system that involves transmission over a pair of copper wires. As the length of the pair of wires increases, the signal at the output is attenuated and the original shape of the signal is increasingly distorted. In addition, interference from extraneous sources, such as radiation from radio signals, car ignitions, and power lines, as well as noise inherent in electronic systems result in the addition of random noise to the transmitted signal. To transmit over long distances, it is necessary to introduce **repeaters** periodically to compensate for the attenuation and distortion of the signal, as shown in Figure 3.6. Such signal reconditioning is fundamentally different for analog and digital transmission.

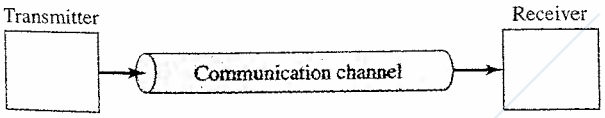


FIGURE 3.4 General transmission system.

[7]This section summarizes the main results of Sections 3.3, 3.4 and 3.5, allowing these three sections to be skipped if necessary.

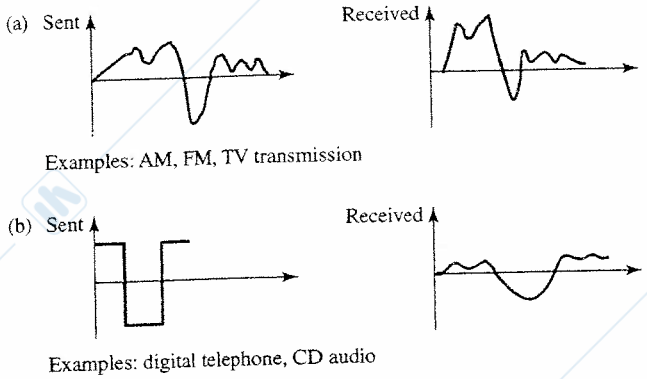


FIGURE 3.5 (a) Analog transmission requires an accurate replica of the original signal whereas (b) digital transmission reproduces discrete levels.

In an analog communication system, the task of the repeater is to regenerate a signal that resembles as closely as possible the signal at the input of the repeater segment. Figure 3.7 shows the basic functions carried out by the repeater. The input to the repeater is an attenuated and distorted version of the original transmitted signal plus the random noise added in the segment. At the transmitter the original signal is much higher in power than the ambient noise. If the signal is attenuated too much then the noise level can become comparable to the desired signal. The function of the repeater is to boost the signal power before this occurs. First the repeater deals with the attenuation by amplifying the received signal. To do so the repeater multiplies the signal by a factor that is the reciprocal of the attenuation a. The resulting signal is still distorted by the channel.

The repeater next uses a device called an **equalizer** in an attempt to eliminate the distortion. The source of the distortion in the signal shape has two primary causes. The first cause is that different frequency components of the signal are attenuated differently.[8] In general, high-frequency components are attenuated more than low-frequency components. The equalizer compensates for this situation by amplifying different frequency components by different amounts. The second cause is that different frequency components of a signal are delayed by different amounts as they propagate through the channel.

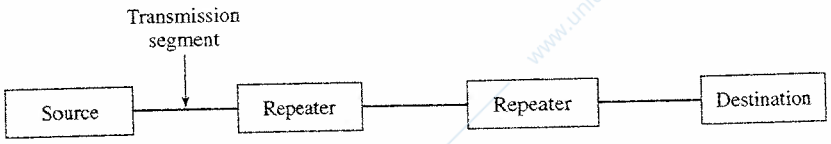


FIGURE 3.6 Typical long-distance link.

[8]Periodic signals can be represented as a sum of sinusoidal signals using Fourier series. Each sinusoidal signal has a distinct frequency. We refer to the sinusoidal signals as the "frequency components" of the original signal. (Fourier series are reviewed in Appendix 3B.)

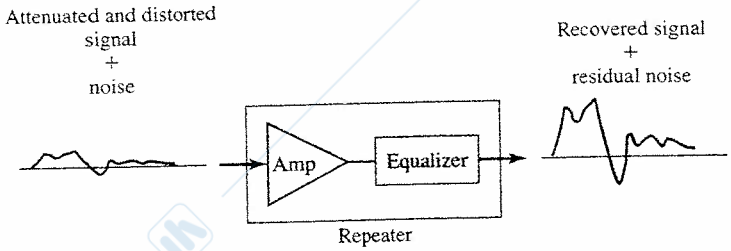


FIGURE 3.7 An analog repeater.

The equalizer attempts to provide differential delays to realign the frequency components. In practice it is very difficult to carry out the two functions of the equalizer. For the sake of argument, suppose that the equalization is perfect. The output of the repeater then consists of the original signal plus the noise.

In the case of analog signals, the repeater is limited in what it can do to deal with noise. If it is known that the original signal does not have components outside a certain frequency band, then the repeater can remove noise components that are outside the signal band. However, the noise within the signal band cannot be reduced and consequently the signal that is finally recovered by the repeater will contain some noise. The repeater then proceeds to send the recovered signal over the next transmission segment.

The effect on signal quality after multiple analog repeaters is similar to that in repeated recordings using analog audiocassette tapes or VCR tapes. The first time a signal is recorded, a certain amount of noise, which is audible as hiss, is introduced. Each additional recording adds more noise. After a large number of recordings, the signal quality degrades considerably.[9] A similar effect occurs in the transmission of analog signals over multiple repeater segments.

Next consider the same copper wire transmission system for digital communications. Suppose that a string of 0s and 1s is conveyed by a sequence of positive and negative voltages. As the length of the pair of wires increases, the pulses are increasingly distorted and more noise is added. A **digital regenerator** is required as shown in Figure 3.8. The sole objective of the regenerator is to restore with high probability the original binary stream. The regenerator also uses an equalizer to compensate for the

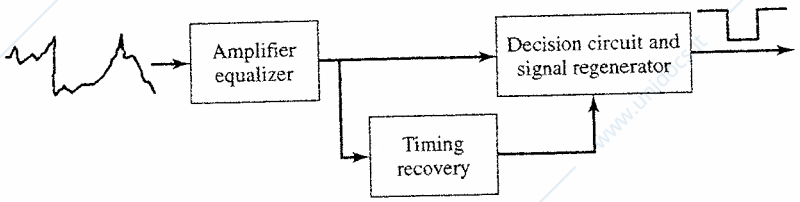


FIGURE 3.8 A digital regenerator.

[9]The recent introduction of digital recording techniques in consumer products almost makes this example obsolete! Another example involves noting the degradation in image quality as a photocopy of a photocopy is made.

distortion introduced by the channel. However, the regenerator does not need to completely recover the original shape of the transmitted signal. It only needs to determine whether the original pulse was positive or negative. To do so, a digital regenerator is organized in the manner shown in Figure 3.8.

A timing recovery circuit keeps track of the intervals that define each pulse by noting the transition instants between pulses. The decision circuit then samples the signal at the midpoint of each interval to determine the polarity of the pulse. In a properly designed system, in the absence of noise, the original symbol would be recovered every time, and consequently the binary stream would be regenerated exactly over any number of regenerators and hence over arbitrarily long distances. Unfortunately, noise is unavoidable in electronic systems, which implies that errors will occur from time to time. An error occurs when the noise signal is sufficiently large to change the polarity of the original signal at the sampling point. Digital transmission systems are designed for very low bit error rates, for example, 10^{-7}, and in optical transmission systems even 10^{-12}, which corresponds to one error in every trillion bits!

The impact on signal quality in multiple digital regenerators is similar to the digital recording of music where the signal is stored as a file of binary information. We can copy the file digitally any number of times with extremely small probabilities of errors being introduced in the process. In effect, the quality of the sound is unaffected by the number of times the file is copied.

The preceding discussion shows that digital transmission has superior performance over analog transmission. Digital regenerators eliminate the accumulation of noise that takes place in analog systems and provide for long-distance transmission that is nearly independent of distance. Digital transmission systems can operate with lower signal levels or with greater distances between regenerators than analog systems can. This factor translates into lower overall system cost and was the original motivation for the introduction of digital transmission.

Another advantage of digital transmission over analog transmission is in monitoring the quality of a transmission channel while the channel is in service. In digital transmission systems, certain predetermined patterns can be imposed on the transmitted information. By checking these patterns it is possible to determine the error rate in the overall channel. Nonintrusive monitoring is much more difficult in analog transmissions systems.

Over time, other benefits of digital transmission have become more prominent. Networks based on digital transmission can multiplex and switch *any* type of information that can be represented in digital form. Thus digital networks are suitable for handling many types of services. Digital transmission also allows networks to exploit the advances in digital computer technology to increase not only the volume of information that can be transmitted but also the types of processing that can be carried out within the network, that is, error correction, data encryption, and the various types of network protocol processing that are the subject of this book.

3.2.2 Basic Properties of Digital Transmission Systems

The purpose of a **digital transmission** system is to transfer a sequence of 0s and 1s from a transmitter (on the left end) to a receiver (on the right) as shown in Figure 3.9.

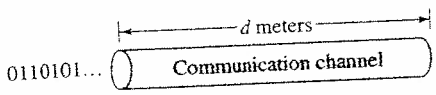


FIGURE 3.9 A digital transmission system.

We are particularly interested in the **bit rate** or transmission speed as measured in bits/second. The bit rate R can be viewed as the cross-section of the information pipe that connects the transmitter to the receiver. As the value of R increases, the volume of information that can flow across the pipe per second increases.

The transmission system uses pulses or sinusoids to transmit binary information over a physical transmission medium. A fundamental question in digital transmission is *how fast* can bits be transmitted *reliably* over a given medium. This capability is clearly affected by several factors including:

- The amount of *energy* put into transmitting each signal.
- The *distance* that the signal has to traverse (because the energy is dissipated and dispersed as it travels along the medium).
- The amount of *noise* that the receiver needs to contend with.
- The *bandwidth* of the transmission channel, which we explain below.

A transmission channel can be characterized by its effect on input sinusoidal signals (tones) of various frequencies. A sinusoid of a given frequency f Hertz is applied at the input, and the sinusoid at the output of the channel is measured. The ability of the channel to transfer a tone of the frequency f is given by the **amplitude-response function** $A(f)$, which is defined as the ratio of the amplitude of the output tone divided by the amplitude of the input tone. Figure 3.10 shows the typical amplitude-response functions of a low-pass channel and its idealized counterpart. As indicated by the figure, the low-pass channel passes sinusoidal signals up to some frequency w and blocks sinusoids of higher frequencies. The **bandwidth of a channel** is defined as the range of frequencies that is passed by a channel.

Consider next what happens when an arbitrary signal is applied to a channel. Recall that the bandwidth of a *signal* W_s is defined as the range of frequencies contained in the signal. On the other hand, the bandwidth of a *channel* W_c is the range of input frequencies *passed* by the channel. Clearly, if the bandwidth of the input signal is

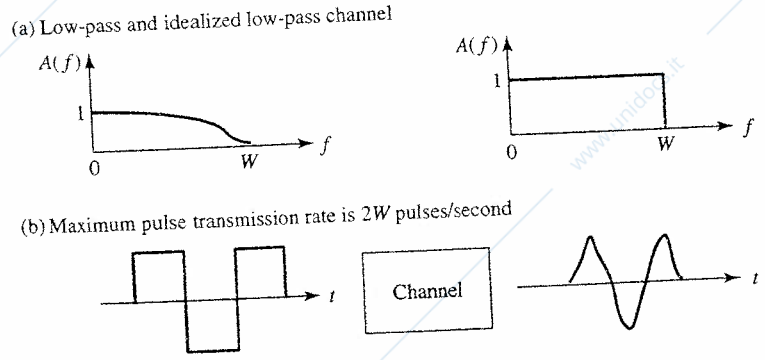


FIGURE 3.10 Typical amplitude-response functions.

larger than the bandwidth of the channel, then the output of the channel will not contain all of the frequencies of the input signal. Therefore the bandwidth of a channel limits the bandwidth of the signals that can pass through the channel. Now let's see what this implies for the transmission of digital signals through a transmission channel. Figure 3.10b shows a typical digital signal before it is input into a channel. The polarity of each pulse corresponds to one bit of information. As we increase the signaling speed, the pulses become narrower and so the signal varies more quickly. Thus higher signaling speed translates into higher signal bandwidth. However we just found that the bandwidth of a channel limits the bandwidth of the input signal that can be passed. Therefore we conclude that the bandwidth of a channel places a limit on the rate at which we can send pulses through the channel.

A major result for digital transmission pertains to the maximum rate at which pulses can be transmitted over a channel. If a channel has bandwidth W, then the narrowest pulse that can be transmitted over the channel has duration $\tau = 1/2W$ seconds. *Thus the maximum rate at which pulses can be transmitted through the channel is given by*: $r_{max} = 2W$ pulses/second.[10]

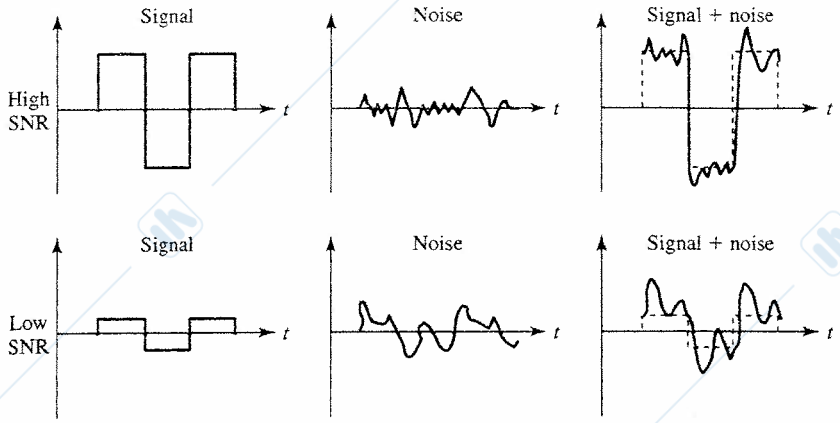
We can transmit binary information by sending a pulse with amplitude $+A$ to send a 1 bit and $-A$ to send a 0 bit. Each pulse transmits one bit of information, so this system then has a bit rate of $2W$ pulses/second $\times$ 1 bit/pulse $= 2W$ bps. We can increase the bit rate by sending pulses with more levels. For example, if pulses can take on amplitudes from the set $\{-A, -A/3, +A/3, +A\}$ to transmit the pairs of bits {00, 01, 10, 11}, then each pulse conveys two bits of information and the bit rate is $4W$ bps. Thus in general, *if we use* ***multilevel transmission*** *with* $\mathrm{M} = 2^{\mathrm{m}}$ *amplitude levels, we can transmit at a bit rate*

$$R = 2W \text{ pulses/second} \times m \text{ bits/pulse} = 2Wm \text{ bits/second} \qquad (3.1)$$

In the absence of noise, the bit rate can be increased without limit by increasing the number of signal levels M. However, noise is an impairment encountered in all communication channels. Noise consists of extraneous signals that are added to the desired signal at the input to the receiver. Figure 3.11 gives two examples where the desired signal is a square wave and where noise is added to the signal. In the first example the amplitude of the noise is less than that of the desired signal, and so the desired signal is discernable even after the noise has been added. In the second example the noise amplitude is greater than that of the desired signal, which is now more difficult to discern. The **signal-to-noise ratio (SNR)**, defined in Figure 3.11, measures the relative amplitudes of the desired signal and the noise. The SNR is usually stated in decibels (dB).

Returning to multilevel transmission, suppose we increase the number of levels while keeping the maximum signal levels $\pm A$ fixed. Each increase in the number of signal levels requires a reduction in the spacing between levels. At some point these reductions will imply significant increases in the probability of detection errors as the noise will be more likely to convert the transmitted signal level into other signal levels. *Thus the presence of noise limits the reliability with which the receiver can correctly determine the information that was transmitted.*

[10]The term *baud rate* is also used to denote the signaling rate in pulses/second.



$$\text{SNR} = \frac{\text{Average signal power}}{\text{Average noise power}}$$

$$\text{SNR (dB)} = 10 \log_{10} \text{SNR}$$

FIGURE 3.11 Signal-to-noise ratio.

The **channel capacity** of a transmission system is the maximum rate at which bits can be transferred reliably. We have seen above that the bit rate and reliability of a transmission system are affected by the channel bandwidth, the signal energy or power, and the noise power. Shannon derived an expression for channel capacity of an ideal low-pass channel. He also showed that reliable communication is not possible at rates above this capacity. The *Shannon channel capacity* is given by the following formula:

$$C = W \log_2(1 + \text{SNR}) \text{ bits/second} \tag{3.2}$$

The above example shows that with an SNR of 40 dB, which is slightly over the maximum possible in a telephone line, Shannon's formula gives a channel capacity of 45.2 kbps. Until 1998 telephone modems achieved speeds below 40 kbps. The V.90 modems that were introduced in 1998 operate at a rate of 56 kbps, well in excess of the Shannon bound! How can this be? The explanation is given in Section 3.5.[11]

Table 3.3 shows the bit rates that are provided by current digital transmission systems over various media. Twisted pairs of copper wires present a wide set of options in telephone access networks and in Ethernet LANs. In traditional phone networks the bandwidth is limited to 4 kHz and bit rates in the 40 kbps range are possible. The same twisted pair of wires, however, can provide much higher bandwidth and in ADSL is used to achieve up to several megabits/second in spans of several kilometers. Ethernet uses twisted pair to achieve up to 100 Mbps over very short distances. Wireless links also provide some options in access networks and LANs. For example, IEEE 802.11 wireless LANs can achieve several Mbps over short distances. Optical fiber transmission provides the high bandwidths required in LANs and backbone networks and can deliver gigabits/second over tens of kilometers. Dense wavelength division

[11] For a detailed explanation refer to [Avanoglu 1998].

TABLE 3.3 Bit rates of digital transmission systems.

Digital transmission system	Bit rate	Observations
Telephone twisted pair	33.6–56 kbps	4 kHz telephone channel
Ethernet over twisted pair	10 Mbps	100 meters over unshielded twisted pair
Fast Ethernet over twisted pair	100 Mbps	100 meters using several arrangements of unshielded twisted pair
Cable modem coaxial cable	500 kbps to 4 Mbps	Shared CATV return channel
ADSL over twisted pair	64–640 kbps inbound 1.536–6.144 Mbps outbound	Uses higher frequency band and coexists with conventional analog telephone signal, which occupies 0–4 kHz band
Radio LAN in 2.4 GHz band	2–54 Mbps	IEEE 802.11 wireless LAN
Digital radio in 28 GHz band	1.5–45 Mbps	5 km multipoint radio link
Optical fiber transmission system	2.5–10 Gbps	Transmission using one wavelength
Optical fiber transmission system	1600 Gbps and higher	Multiple simultaneous wavelengths using wavelength division multiplexing

multiplexing systems will provide huge bandwidths by combining several hundred Gbps optical signals in a single fiber and will profoundly affect network design. Section 3.8 discusses the properties of specific transmission media in more detail.

SHANNON CHANNEL CAPACITY OF TELEPHONE CHANNEL

Consider a telephone channel with $W = 3.4$ kHz and SNR $= 10{,}000$. The channel capacity is then

$$C = 3400 \log_2(1 + 10000) = 45{,}200 \text{ bits/second}$$

The following identities are useful here: $\log_2 x = \ln x / \ln 2 = \log_{10} x / \log_{10} 2$. We note that the SNR is usually stated in dB. Thus if SNR = 10,000, then in dB the SNR is

$$10 \log_{10} \text{ SNR dB} = 10 \log_{10} 10000 = 40 \text{ dB}$$

The above result gives a bound to the achievable bit rate over ordinary analog telephone lines when limited to a bandwidth of 3.4 kHz.[12]

3.3 DIGITAL REPRESENTATION OF ANALOG SIGNALS

We now consider the digital representation of analog signals. Figure 3.12 shows a 3-second interval of a typical speech waveform for the following utterance: The speech

[12] See Section 3.8.1 for a discussion of the bandwidth of twisted pair cables used in telephone networks.

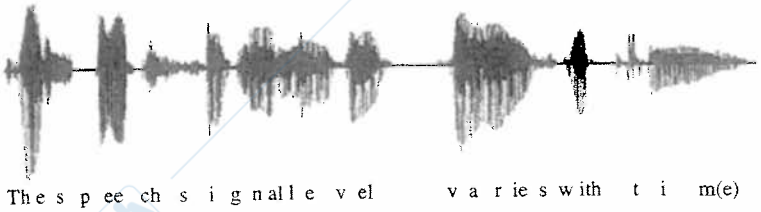


FIGURE 3.12 Example of speech waveform: "The speech signal level varies with time."

signal level varies with time. It can be seen that the signal amplitude can assume any value from an interval that is defined by some maximum value and minimum value. This property characterizes **analog signals**. The *exact* representation of a value in an interval requires an infinite number of bits. For this reason, analog waveforms cannot be represented exactly in practice. In addition to speech and audio signals, other examples of analog information include image and video information. Image information consists of the variation of intensity over a plane. Video and motion pictures involve the variation of intensity over space and time. All of these signals can assume a continuum of values over time and/or space and consequently require infinite precision in their representation. All of these signals are also important in human communications and are increasingly being incorporated into a variety of multimedia applications. In this section we will consider the digitization of voice and audio signals. Image and video signals are considered in Chapter 12.

The digitization of analog signals such as voice and audio involves two steps: (1) measuring samples of the analog waveform at evenly spaced instants of time, say T seconds, and (2) representing each sample value using a finite number of bits, say m bits. The bit rate of the digitized signal is then m/T bits/second. In the next section, we introduce the notion of *bandwidth* of a signal, which is a measure of the rate at which a signal varies with time. Intuitively, a signal that has a higher bandwidth will vary faster and hence will need to be sampled more frequently. In the subsequent section we introduce the *quantizer*, which is a device that produces an approximation of a sample value using m bits.

3.3.1 Bandwidth of Analog Signals

Many signals that are found in nature are periodic and can be represented as the sum of sinusoidal signals. For example, many speech sounds consist of the sum of a sinusoidal wave at some fundamental frequency and its harmonics. These analog signals have the form:

$$x(t) = \sum a_k \cos(2\pi k f_0 t + \phi_k). \tag{3.3}$$

For example, Figure 3.13 shows the periodic voice waveform for the sound "ae" as in cat.

As another example, consider a digital signal that could occur if we were transmitting binary information at a rate of 8 kilobits/second. Suppose that a binary 1 is

FIGURE 3.13 Sample waveform of "ae" sound as in cat.

transmitted by sending a rectangular pulse of amplitude 1 and of duration 0.125 milliseconds, and a 0 by sending a pulse of amplitude −1. Figure 3.14a shows the periodic signal that results if we repeatedly send the octet 10101010, and Figure 3.14b shows the signal that results when we send 11110000. Note that the signals result in square waves that repeat at rates of 4 kHz and 1 kHz, respectively. By using Fourier series analysis (see Appendix 3B), we can show that the first signal is given by

$$x_1(t) = (4/\pi)\{\sin(2\pi(4000)t) + (1/3)\sin(2\pi(12000)t) + (1/5)\sin(2\pi(20000)t + \cdots\} \tag{3.4}$$

and has frequency components at the odd multiples (harmonics) of 4 kHz. Similarly, we can show that the second signal has harmonics at odd multiples of 1 kHz and is given by

$$x_2(t) = (4/\pi)\{\sin(2\pi(1000)t) + (1/3)\sin(2\pi(3000)t) + (1/5)\sin(2\pi(5000)t + \cdots\} \tag{3.5}$$

Figure 3.15a and Figure 3.15b show the "spectrum" for the signals $x_1(t)$ and $x_2(t)$, respectively. The **spectrum** gives the magnitude of the amplitudes of the sinusoidal components of a signal. It can be seen that the first signal has significant components over a much broader range of frequencies than the second signal; that is, $x_1(t)$ has a larger bandwidth than $x_2(t)$. Indeed the bandwidth is an indicator of how fast a signal varies with time. Signals that vary quickly have a larger bandwidth than signals that vary slowly. For example, in Figure 3.15 $x_1(t)$ varies four times faster than $x_2(t)$ and thus has the larger bandwidth.

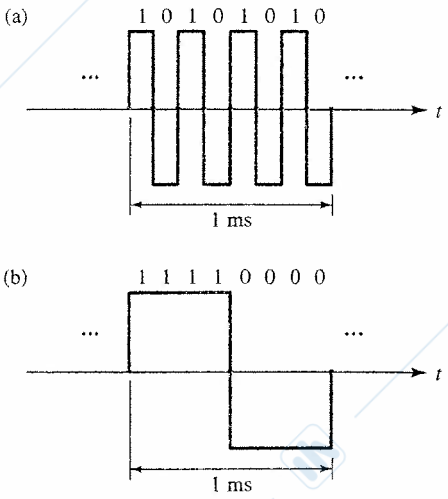


FIGURE 3.14 Signals corresponding to repeated octet patterns.

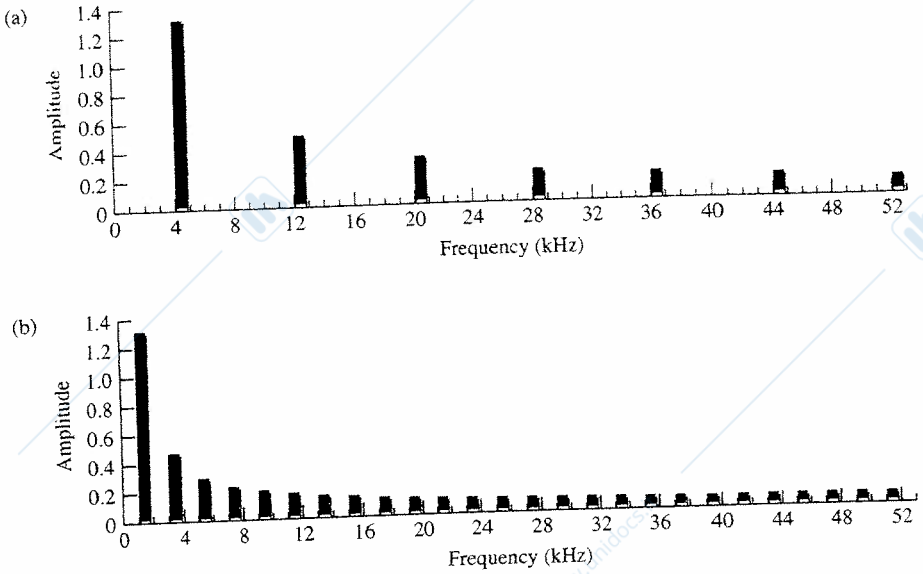


FIGURE 3.15 (a) Frequency components for pattern 10101010; (b) frequency components for pattern 11110000.

Not all signals are periodic. For example, Figure 3.16 shows the sample waveform for the word "speech." It can be seen that the character of the waveform varies according to the sound. For example the "s" sound at the beginning of the utterance has a noise-like waveform, while the "ee" has a periodic structure. The speech waveform varies in structure over time with periods of well-defined high-amplitude periodic structure alternating with periods of low-amplitude noiselike structure. The long-term average spectrum of the voice signal is the average of the spectra over a long time interval and ends up as a smooth function of frequency looking like that shown in Figure 3.17.[13]

We define the bandwidth of an analog signal as the range of frequencies at which the signal contains nonnegligible power, that is, for periodic signals nonnegligible a_k. There are many ways of precisely defining the bandwidth of a signal. For example, the 99 percent bandwidth is defined as the frequency range required to contain 99 percent of the power of the original signal. Usually the appropriate choice of bandwidth of a signal

[13] For examples of long-term spectrum measurements of speech, see [Jayant and Noll 1984, p. 40].

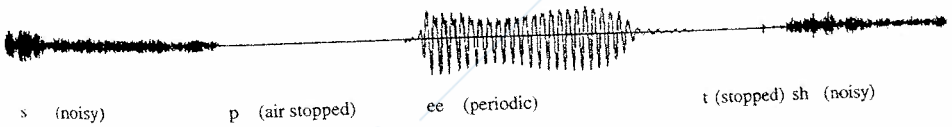


FIGURE 3.16 Waveform for the word *speech*.

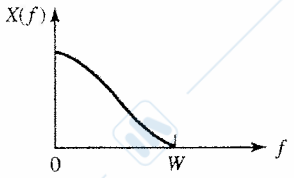


FIGURE 3.17 Spectrum of analog bandwidth W Hz.

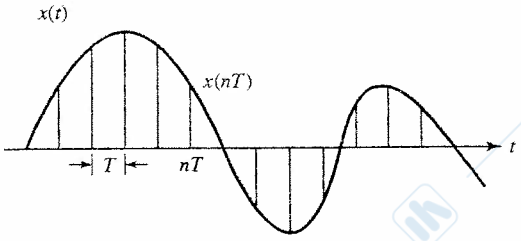


FIGURE 3.18 Sampling of an analog signal.

depends on the application. For example, the human ear can detect signals in the range 20 Hz to 20 kHz. In telephone communications frequencies from 200 Hz to 3.5 kHz are sufficient for speech communications. However, this range of frequencies is clearly inadequate for music that contains significant information content at frequencies higher than 3.5 kHz.

3.3.2 Sampling of an Analog Signal

Suppose we have an analog waveform $x(t)$ that has a spectrum with bandwidth W Hz as shown in Figure 3.17. To convert the signal to digital form, we begin by taking instantaneous samples of the signal amplitude every T seconds to obtain $x(nT)$ for integer values n (see Figures 3.18 and 3.19). Because the signal varies continuously in time, we obtain a sequence of real numbers that for now we assume have an infinite level of precision. Intuitively, we know that if the samples are taken frequently enough relative to the rate at which the signal varies, we can recover a good approximation of the signal from the samples, for example, by drawing a straight line between the sample points. Thus the sampling process replaces the continuous function of time by a sequence of real-valued numbers. The very surprising result is that we can recover

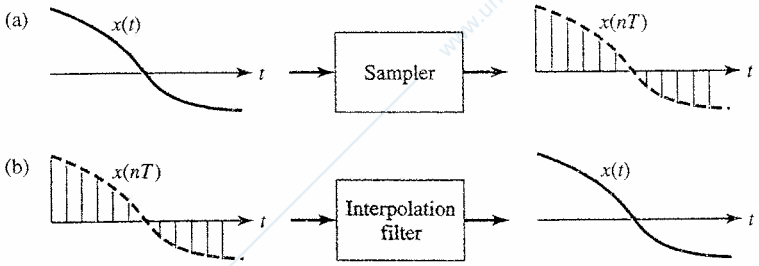


FIGURE 3.19 (a) Sampling of signal $x(t)$; (b) recovery of original signal $x(t)$ by interpolation.

the original signal $x(t)$ from the sequence $x(nT)$ precisely as long as the sampling rate is higher than some minimum value!

The sampling theorem is a mathematical result that states that if the sampling rate $1/T$ is greater than $2W$ samples/second, then the signal $x(t)$ can be recovered from its sample values $\{x(nT)\}$. We refer to $2W$ as the **Nyquist sampling rate**. The reconstruction of $x(t)$ is carried out by interpolating the samples $x(nT)$ according to the following formula

$$x(t) = \sum_n x(nT)s(t - nT) \tag{3.6}$$

where the interpolation function $s(t)$ is given by

$$s(t) = \frac{\sin 2\pi Wt}{2\pi Wt} \tag{3.7}$$

The expression in Equation (3.6) involves summing time-shifted versions of $s(t)$ that are weighted by the sample values. The proof of the sampling theorem is discussed in Appendix 3C. From the discussion there, we find that a possible implementation of the interpolation is to input a series of narrow pulses, T seconds apart, of amplitude $x(nT)$ into an interpolation filter as shown in Figure 3.19b.

As an example of a sampling rate calculation consider the voice signal in the telephone system that has a nominal bandwidth of 4 kHz. The Nyquist sampling rate then requires that the voice signal be sampled at a rate of 8000 samples/second. For the high-quality audio signals encountered in CD recordings, the bandwidth is 22 kHz leading to a sampling rate of 44,000 samples/second. Lastly, an analog TV signal has a bandwidth of 4 MHz, leading to a sampling rate of 8,000,000 samples/second.

3.3.3 Digital Transmission of Analog Signals

Figure 3.20 shows the standard arrangement in the handling of the analog information by digital transmission (and storage) systems. The signal produced by an analog source $x(t)$, which we assume is limited to W Hz, is sampled at the Nyquist sampling rate, producing a sequence of samples at a rate of $2W$ samples/second. These samples have infinite precision, so they are next input into a quantizer that approximates the sample value using m bits to produce an approximation within a specified accuracy. The level of accuracy determines the number of bits m that the quantizer uses to specify the approximation. The bit rate out of the quantizer is $2Wm$ bits/second, since samples occur at a rate of $2W$ samples/second and each sample requires m bits. At this point we have obtained a digital representation of the original analog signal within a specified accuracy or quality. This digital representation can be stored or transmitted any number of times without additional distortion so long as no errors are introduced into the digital representation.

The approximation to the original signal $x(t)$ is recovered by the mirror process shown in Figure 3.20. The approximation of the sample values is obtained from the sequence of groups of m bits, and a sequence of narrow pulses with the corresponding

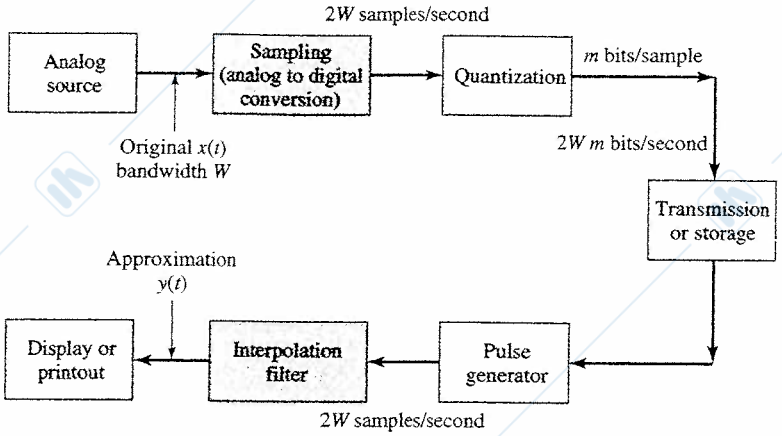


FIGURE 3.20 Digital transmission of analog signal.

amplitudes is generated. Finally an interpolation system is driven by the sequence of narrow pulses to generate an analog signal that approximates the original signal within the prescribed accuracy. Note that this process applies equally to the storage or transmission of analog information.

The accuracy in the approximation in the above process is determined by the **quantizer**. The task of the quantizer is to take sample values $x(nT)$ and produce an approximation $y(nT)$ than can be specified using a fixed number of bits/sample. In general, quantizers have a certain number, say, $M = 2^m$, of approximation values that are used to represent the quantizer inputs. For each input $x(nT)$ the closest approximation point is found, and the index of the approximation point is specified using m bits. The decoder on the receiver side is assumed to have the set of approximation values so the decoder can recover the values from the indices.

The design of a quantizer requires knowledge about the range of values that are assumed by the signal $x(t)$. The set of approximation values is selected to cover this range. For example, suppose $x(t)$ assumes the values in the range $-V$ to V. Then the set of approximation values should be selected to cover only this range. Selecting approximation values outside this range is unnecessary and will lead to inefficient representations. Note that as we increase m, we increase the number of intervals that cover the range $-V$ to V. Consequently, the intervals become smaller and the approximations become more accurate. We next quantify the trade-off between accuracy and the bit rate $2Wm$.

Figure 3.21 shows the simplest type of quantizer, the **uniform quantizer**, in which the range of the amplitudes of the signal is covered by equally spaced approximation values. The range $-V$ to V is divided into 2^m intervals of equal length Δ, and so we have $2V = 2^m\Delta$, and $\Delta = V/2^{m-1}$. When the input $x(nT)$ falls in a given interval, then its approximation value $y(nT)$ is the midpoint of the interval. The output of the quantizer is simply the m bits that specify the interval.

In general, the approximation value is not equal to the original signal value, so an error is introduced in the quantization process. The value of the **quantization error**

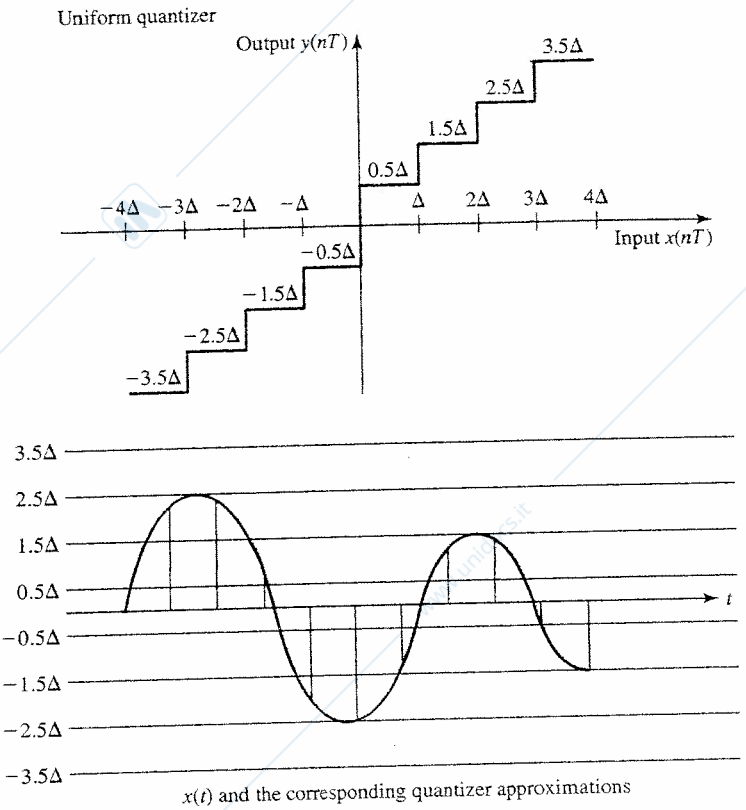


FIGURE 3.21 A uniform quantizer.

$e(nT)$ is given by

$$e(nT) = y(nT) - x(nT) \tag{3.8}$$

Figure 3.22 shows the value of the quantization error as the function of the quantizer input. It can be seen that the error takes on values between $-\Delta/2$ and $\Delta/2$. When the interval length Δ is small, then the quantization error values are small and the quantizer can be viewed as simply adding "noise" to the original signal. For this reason the figure of merit used to assess the quality of the approximation is the **quantizer signal-to-noise ratio (SNR)**:

$$\text{SNR} = \frac{\text{average signal power}}{\text{average noise power}} = \frac{\sigma_x^2}{\sigma_e^2} \tag{3.9}$$

The average power of the error is given by its mean square value, which in Section 3.3.4 is found to be $\sigma_e^2 = \Delta^2/12 = (V/2^{m-1})^2$. The standard deviation σ_x is a measure of the spread of the signal values about the mean, which we are assuming is zero. On the other hand, V is the maximum value that the quantizer assumes can be taken on

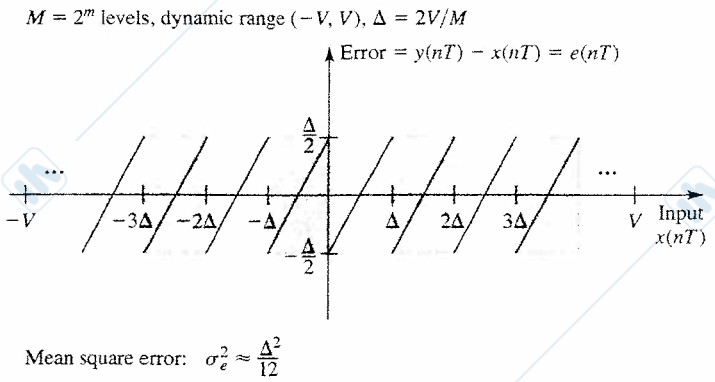


FIGURE 3.22 Quantizer error.

by the signal. Frequently, selecting V so that it corresponds to the maximum value of the signal is too inefficient. Instead V is selected so that the probability that a sample $x(nT)$ exceeds V is negligible. This practice typically leads to ratios of approximately $V/\sigma_x \approx 4$.

SNR is usually stated in decibels. In Section 3.3.4, we show that the SNR for a uniform quantizer is given by

$$\text{SNR dB} = 10\log_{10}\sigma_x^2/\sigma_e^2 = 6m + 10\log_{10}3\sigma_x^2/V^2 \tag{3.10}$$

$$\approx 6m - 7.27 \text{ dB for } V/\sigma_x = 4 \tag{3.11}$$

Equation (3.11) states that each additional bit used in the quantizer will increase the SNR by 6 dB. This result makes intuitive sense, since each additional bit doubles the number of intervals, and so for a given range $-V$ to V, the intervals are reduced in half. The average magnitude of the quantization error is also reduced in half, and the average quantization error power is reduced by a quarter. This result agrees with $10\log_{10}4 = 6$ dB. We have derived this result for the case of uniform quantizers. More general quantizers can be defined in which the intervals are not of the same length. The SNR for these quantizers can be shown to also have the form of the preceding equations where the only difference is in the constant that is added to $6m$ [Jayant and Noll 1984].

We noted before that the human ear is sensitive to frequencies up to 22 kHz. For audio signals such as music, a high-quality representation involves sampling at a much higher rate. The Nyquist sampling rate for $W = 22$ kHz is 44,000 samples/second. The high-quality audio also requires finer granularity in the quantizers. Typically 16 or more bits are used per sample. For a stereo signal we therefore obtain the following bit rate:

$$44{,}000\ \frac{\text{samples}}{\text{second}} \times 16\ \frac{\text{bits}}{\text{sample}} \times 2 \text{ channels} = 1.4 \text{ Mbps} \tag{3.12}$$

PCM

The standard for the digital representation of voice signals in telephone networks is given by the *pulse code modulation (PCM)* format. In PCM the voice signal is filtered to obtain a low-pass signal that is limited to $W = 4$ kHz. The resulting signal is sampled at the Nyquist rate of $2W = 8$ kHz. Each sample is then applied to an $m = 8$ bit quantizer. The standard bit rate for digitized telephone speech signals is therefore 8000 samples/second $\times$ 8 bits/sample = 64 kilobits/second.

The type of quantizers used in telephone systems are *nonuniform quantizers*. A technique called *companding* is used so that the size of the intervals increases with the magnitude of the signal x in logrithmic fashion. The SNR formula for this type of quantization is given by

$$\text{SNR dB} = 6m - 10 \text{ dB for PCM speech}$$

Because $m = 8$, we see that the SNR is 38 dB. Note that an SNR of 1 percent corresponds to 40 dB. In the backbone of modern digital telephone systems, voice signals are carried using the *log-PCM* format, which uses a logarithmic scale to determine the quantization intervals.

We see that high-quality audio signals can require much higher rates than are required for more basic signals such as those of telephony speech. The bit rate for audio is increased even further in modern surround-sound systems. For example, the Digital Audio Compression (AC-3) that is part of the U.S. ATSC high-definition television standard involves five channels (left, right, center, left-surround, right-surround) plus a low-frequency enhancement channel for the 3 Hz to 100 Hz band.

◆ 3.3.4 SNR Performance of Quantizers

We now derive the SNR performance of a uniform quantizer. When the number of levels M is large, then the error values are approximately uniformly distributed in the interval $(-\Delta/2, \Delta/2)$. The power in the error signal is then given by

$$\sigma_e^2 = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} x'^2 \frac{1}{\Delta} dx' = \frac{\Delta^2}{12} \tag{3.13}$$

Let σ_x^2 be the average power of the signal $x(t)$. Then the SNR is given by

$$\text{SNR} = \frac{\sigma_x^2}{\Delta^2/12} \tag{3.14}$$

From the definition of the quantizer, we have that $\Delta = 2V/M$ and that $M = 2^m$; therefore

$$\text{SNR} = \frac{\sigma_x^2}{\Delta^2/12} = \frac{12\sigma_x^2}{4V^2/M^2} = 3\left(\frac{\sigma_x}{V}\right)^2 M^2 = 3\left(\frac{\sigma_x}{V}\right)^2 2^{2m} \tag{3.15}$$

The SNR, stated in decibels, is then

$$\text{SNR dB} = 10\log_{10}\sigma_x^2/\sigma_e^2 = 6m + 10\log_{10}3\sigma_x^2/V^2 \tag{3.16}$$

As indicated before, V is selected so that the probability that a sample $x(nT)$ exceeds V is negligible. If we assume that $V/\sigma_x \approx 4$, then we obtain

$$\text{SNR dB} \approx 6m - 7.27 \text{ dB for } V/\sigma_x = 4 \tag{3.17}$$

3.4 CHARACTERIZATION OF COMMUNICATION CHANNELS

A communication channel is a system consisting of a physical medium and associated electronic and/or optical equipment that can be used for the transmission of information. Commonly used physical media are copper wires, coaxial cable, radio, and optical fiber. Communication channels can be used for the transmission of either digital or analog information. Digital transmission involves the transmission of a sequence of pulses that is determined by a corresponding digital sequence, typically a series of binary 0s and 1s. Analog transmission involves the transmission of waveforms that correspond to some analog signal, for example, audio from a microphone or video from a television camera. Communication channels can be characterized in two principal ways: frequency domain and time domain.

3.4.1 Frequency Domain Characterization

Figure 3.23 shows the approach used in characterizing a channel in the frequency domain. A sinusoidal signal $x(t) = \cos(2\pi ft)$ that oscillates at a frequency of f cycles/second (Hertz) is applied to a channel. The channel output $y(t)$ usually consists of a sinusoidal signal of the same frequency but of different amplitude and phase:[14]

$$y(t) = A(f)\cos(2\pi ft + \varphi(f)) = A(f)\cos(2\pi f(t - \tau(f))). \tag{3.18}$$

The channel is characterized by its effects on the input sinusoidal signal. The first effect involves an attenuation of the sinusoidal signal. This effect is characterized by the **amplitude-response function** $A(f)$, which is the ratio of the output amplitude to the input amplitude of the sinusoids at frequency f. The second effect is a shift in the phase of the output sinusoid relative to the input sinusoid. This is specified by a *phase shift* $\varphi(f)$. In general, both the amplitude response and the phase shift depend on the

[14]The statement applies to channels that are "linear." For such channels the output signal corresponding to a sum of input signals, say, $x_1(t) + x_2(t)$, is equal to the sum of the outputs that would have been obtained for each individual input; that is, $y(t) = y_1(t) + y_2(t)$.

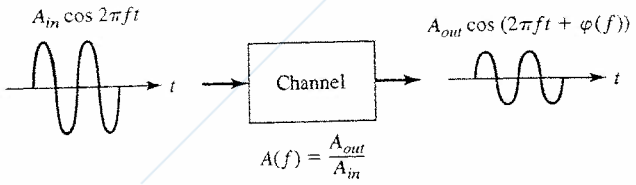


FIGURE 3.23 Channel characterization—frequency domain.

frequency f of the sinusoid. You will see later that for various communication channels the attenuation typically increases with f. Equation (3.18) also shows that the output $y(t)$ can be viewed as the input attenuated by $A(f)$ and delayed by $\tau(f)$.

The frequency-domain characterization of a channel involves varying the frequency f of the input sinusoid to evaluate $A(f)$ and $\varphi(f)$. Figure 3.24 shows the amplitude-response and phase-shift functions for a "low-pass" channel. In this channel very low frequencies are passed, but very high frequencies are essentially eliminated. In addition, frequency components at low frequencies are not phase shifted, but very high frequencies are shifted by 90 degrees.

The **attenuation** of a signal is defined as the reduction or loss in signal power as it is transferred across a system. The attenuation is usually expressed in dB:

$$\text{attenuation} = 10\log_{10}\frac{P_{in}}{P_{out}} \tag{3.19}$$

The power in a sinusoidal signal of amplitude A is $A^2/2$. Therefore, the attenuation in the channel at frequency f in Figure 3.23 is given by $P_{in}/P_{out} = A_{in}^2/A_{out}^2 = 1/A^2(f)$.

The amplitude-response function $A(f)$ can be viewed as specifying a window of frequencies that the channel will pass. The **bandwidth of a channel** W measures

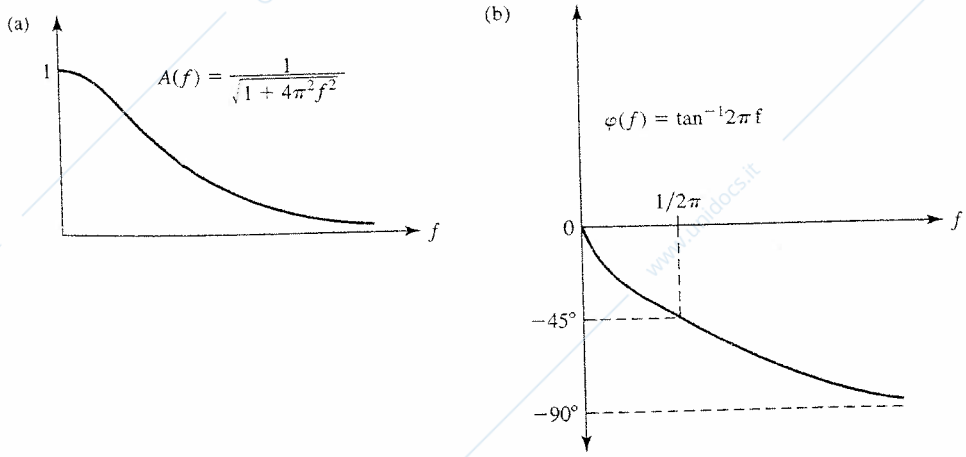


FIGURE 3.24 (a) Amplitude-response function; (b) phase-shift function.

the width of the window of frequencies that are passed by the channel. Figure 3.10a and Figure 3.24a show two typical amplitude-response functions. The low-pass channel passes low-frequency components and increasingly attenuates higher frequency components. In principle this channel has infinite bandwidth, since every frequency is passed to some degree. However, for practical purposes, signals above a specified frequency are considered negligible. We can define such a frequency as the bandwidth and approximate the amplitude-response function by the idealized low-pass function in Figure 3.10a. Another typical amplitude-response function is a "band-pass" channel that passes frequencies in the range f_1 to f_2 instead of low frequencies (see Figure 3.37). The bandwidth for such a channel is $W = f_2 - f_1$.

Communication systems make use of electronic circuits to modify the frequency components of an input signal. When circuits are used in this manner, they are called *filters*. Communication systems usually involve a tandem arrangement of a transmitter filter, a communication channel, and a receiver filter. The overall tandem arrangement can be represented by an overall amplitude-response function $A(f)$ and a phase-shift function $\varphi(f)$. The transmitter and receiver filters are designed to give the overall system the desired amplitude-response and delay properties. For example, devices called loading coils were added to telephone wire pairs to provide a flat amplitude-response function in the frequency range where telephone voice signals occur. Unfortunately, these coils also introduced a much higher attenuation at the higher frequencies, greatly reducing the bandwidth of the overall system.

Let us now consider the impact of communication channels on other signals. The effect of a channel on an arbitrary input signal can also be determined from $A(f)$ and $\varphi(f)$ as follows. As discussed before, many signals can be represented as the sum of sinusoidal signals. Consider, for example,

$$x(t) = \sum a_k \cos(2\pi f_k t) \tag{3.20}$$

For example, periodic functions have the preceding form with $f_k = kf_0$ where f_0 is the fundamental frequency.

Now suppose that a periodic signal $x(t)$ is applied to a channel with a channel characterized by $A(f)$ and $\varphi(f)$. The channel attenuates the sinusoidal component at frequency kf_0 by $A(kf_0)$, and it also phase shifts the component by $\varphi(kf_0)$. The output signal of the channel, which we assume to be linear, will therefore be

$$y(t) = \sum a_k A(kf_0) \cos(2\pi k f_0 t + \varphi(kf_0)) \tag{3.21}$$

This expression shows how the channel distorts the input signal. In general, the amplitude-response function varies with frequency, and so the channel alters the relative weighting of the frequency components. In addition, the different frequency components will be delayed by different amounts, altering the relative alignment between the components. Not surprisingly, then, the shape of the output $y(t)$ generally differs from $x(t)$.

Note that the output signal will have its frequencies restricted to the range where the amplitude-response function is nonzero. Thus the bandwidth of the output signal is

necessarily less than that of the channel. Note also that if $A(f)$ is equal to a constant, say, C, and if $\varphi(f) = 2\pi f t_d$, over the range of frequencies where a signal $x(t)$ has nonnegligible components, then the output $y(t)$ will be equal to the input signal scaled by the factor C and delayed by t_d seconds:

$$y(t) = \sum Ca_k \cos\left(2\pi k f_0 t + 2\pi f t_d\right) = C \sum a_k \cos\left(2\pi k f_0 (t + t_d)\right) = Cx(t + t_d) \tag{3.22}$$

EXAMPLE **Effect of a Channel on the Shape of the Output Signal**

Suppose that binary information is transmitted at a rate of 8 kilobits/second. A binary 1 is transmitted by sending a rectangular pulse of amplitude 1 and of duration 0.125 milliseconds, and a 0 by sending a pulse of amplitude -1. Consider the signal $x_3(t)$ in Figure 3.25 that corresponds to the repetition of the pattern 10000001 over and over again. Using Fourier series, this periodic signal can be expressed as a sum of sinusoids with frequencies at 1000 Hz, 2000 Hz, 3000 Hz and so on:

$$x_3(t) = -0.5 + \left(\frac{4}{\pi}\right)\left\{ \sin\left(\frac{\pi}{4}\right)\cos(2\pi 1000t) + \frac{\sin\left(\frac{2\pi}{4}\right)}{2}\cos(2\pi 2000t) + \frac{\sin\left(\frac{3\pi}{4}\right)}{3}\cos(2\pi 3000t) + \cdots \right\} \tag{3.23}$$

Suppose that the signal is passed through a communication channel that has $A(f) = 1$ and $\varphi(f) = 0$ for f in the range 0 to W and $A(f) = 0$ elsewhere. Figures 3.26a, b, and c show the output (the solid line) of the communication channel for values of W (1.5 kHz, 2.5 kHz, and 4.5 kHz) that pass the frequencies only to the first, second, and fourth harmonic, respectively. As the bandwidth of the channel increases, more of the harmonics are passed and the output of the channel more closely approximates the input. This example shows how the bandwidth of the channel affects the ability to transmit digital information in the form of pulses. Clearly, *as bandwidth is decreased, the precision with which the pulses can be identified is reduced.*

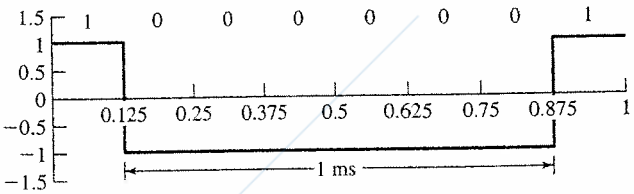


FIGURE 3.25 Signals corresponding to repeated octet patterns.

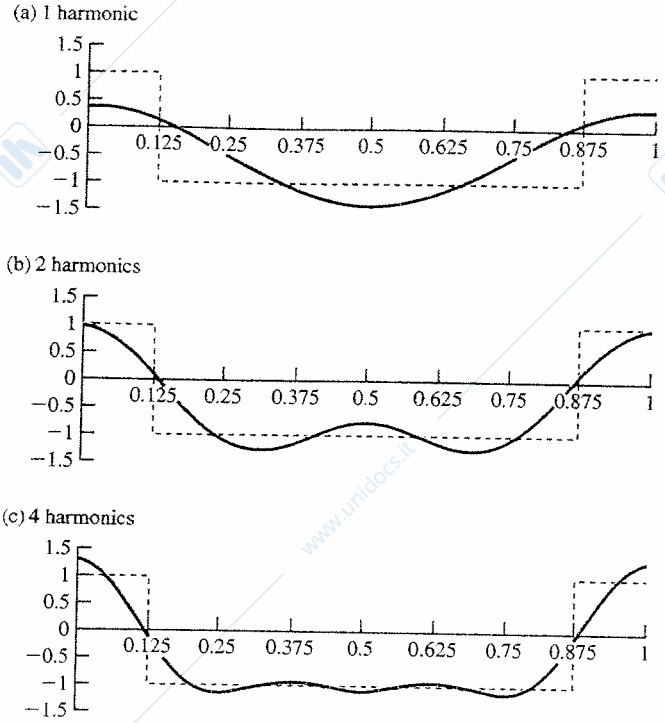


FIGURE 3.26 Output of low-pass communication channels for input signal in Figure 3.25.

3.4.2 Time Domain Characterization

Figure 3.27 considers the time domain characterization of a communication channel. A very narrow pulse is applied to the channel at time $t = 0$. The energy associated with the pulse appears at the output of the channel as a signal $h(t)$ some propagation time later. The propagation speed, of course, cannot exceed the speed of light in the given medium. The signal $h(t)$ is called the **impulse response** of the channel. Invariably the output pulse $h(t)$ is spread out in time. The width of the pulse is an indicator of how quickly the output follows the input and hence of how fast pulses can be transmitted over the channel. In digital transmission we are interested in maximizing

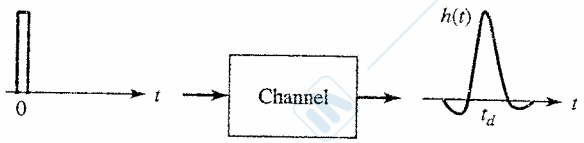


FIGURE 3.27 Channel characterization—time domain.

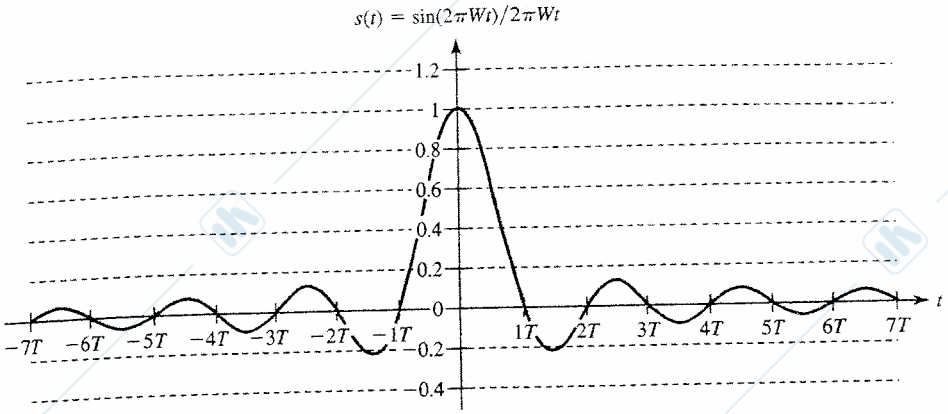


FIGURE 3.28 Signaling pulse with zero intersymbol interference.

the number of pulses transmitted per second in order to maximize the rate at which information can be transmitted. Equivalently, we are interested in minimizing the time T between consecutive input pulses. This minimum spacing is determined by the degree of interference between pulses at the output of the channel.

Suppose that we place transmitter and receiver filters around a communication channel and suppose as well that we are interested in using the frequencies in the range 0 to W Hz. Furthermore, suppose that the filters can be selected so that the overall system is an idealized low-pass channel; that is, $A(f) = 1$, and $\varphi(f) = 2\pi f t_d$. It can be shown that the impulse response of the system is given by

$$h(t) = s(t - t_d) \tag{3.24}$$

which is a delayed version of

$$s(t) = \frac{\sin(2\pi Wt)}{2\pi Wt} \tag{3.25}$$

Figure 3.28 shows $s(t)$. It can be seen that this function is equal to 1 at $t = 0$ and that it has zero crossings at nonzero integer multiples of $T = 1/2W$. Note that the pulse is mostly confined to the interval from $-T$ to T, so it is approximately $2T = 2/2W = 1/W$ seconds wide. Thus we see that *as the bandwidth* W *increases, the width of the pulse* s(t) *decreases, suggesting that pulses can be input into the system more closely spaced, that is, at a higher rate.* The next section shows that the signal $s(t)$ plays an important role in the design of digital transmission systems.

Recall that $h(t)$ is the response to a narrow pulse at time $t = 0$, so we see that our ideal system has the strange property that its output $h(t) = s(t - t_d)$ anticipates the input that will be applied and begins appearing at the output before time $t = 0$. In practice, this idealized filter cannot be realized; however, delayed and slightly modified versions of $s(t)$ are approximated and implemented in real systems.

3.5 FUNDAMENTAL LIMITS IN DIGITAL TRANSMISSION

In this section we consider **baseband transmission**, which is the transmission of digital information over a low-pass communication channel. The quality of a digital transmission system is determined by the bit rate at which information bits can be transmitted reliably. Thus the quality is measured in terms of two parameters: *transmission speed*, or *bit rate*, in bits per second and the *bit error rate*, the fraction of bits that are received in error. We will see that these two parameters are determined by the bandwidth of the communication channel and by the SNR, which we will define formally later in the section.

Figure 3.29 shows the simplest way to transmit a binary information sequence. Every T seconds the transmitter accepts a binary information bit and transmits a pulse with amplitude $+A$ if the information bit is a 1 and with $-A$ if the information bit is a 0. In the examples in Section 3.4, we saw how a channel distorts an input signal and limits the ability to correctly detect the polarity of a pulse. In this section we show how the problem of channel distortion is addressed and how the pulse transmission rate is maximized at the same time. In particular, in Figure 3.29 each pulse at the input results in a pulse at the output. We also show how the pulses that arrive at the receiver can be packed as closely as possible if they are shaped appropriately by the transmitter and receiver filters.

3.5.1 The Nyquist Signaling Rate

Let $p(t)$ be the basic pulse that appears at the receiver after it has been sent over the combined transmitter filter, communication channel, and receiver filter. The first pulse is transmitted, centered at $t = 0$. If the input bit was 1, then $+Ap(t)$ should be received; if the input was 0, then $-Ap(t)$ should be received instead. For simplicity, we assume that the propagation delay is zero. To determine what was sent at the transmitter, the

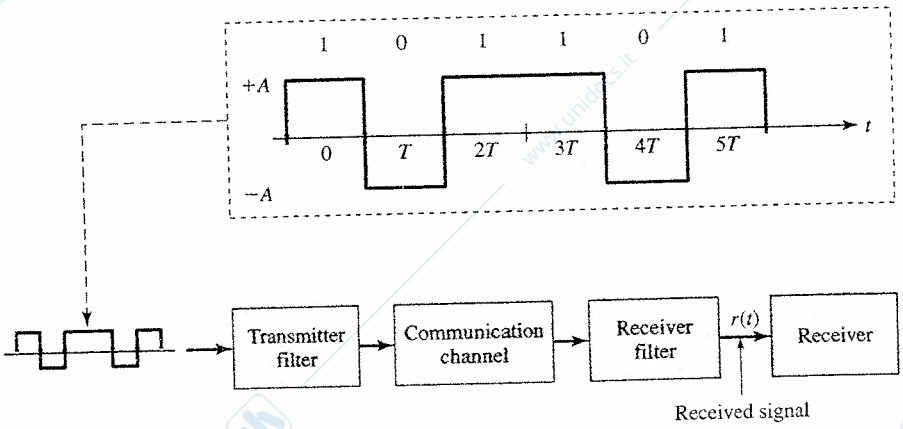


FIGURE 3.29 Digital baseband signal and baseband transmission system.

receiver samples the signal it receives at $t = 0$. If the sample is positive, the receiver decides that a 1 was sent; if the sample is negative, the receiver decides that a 0 was sent.

Every T seconds the transmitter sends an additional information bit by transmitting another pulse with the appropriate polarity. For example, the second bit is sent at time $t = T$ and will be either $+Ap(t - T)$ or $-Ap(t - T)$, depending on the information bit. The receiver samples its signal at $t = T$ to determine the corresponding input. However, the pulses are sent as part of a sequence, and so the total signal $r(t)$ that appears at the receiver is the *sum* of all the inputs:

$$r(t) = \sum_k A_k p(t - kT) \tag{3.26}$$

where A_k is determined by the polarity of the kth signal. According to this expression, when the receiver samples the signal at $t = 0$, it measures

$$r(0) = A_0 p(0) + \sum_{k \neq 0} A_k p(-kT) \tag{3.27}$$

In other words, the receiver must contend with *intersymbol interference* from all the other transmitted pulses. What a mess! Note, however, that all the terms in the summation disappear if we use a pulse that has zero crossings at $t = kT$ for nonzero integer values k. The pulse $s(t)$ introduced in Section 3.4.2 and shown in Figure 3.28 satisfies this property. This pulse is an example of the class of *Nyquist pulses* that have the property of providing *zero intersymbol interference* at the times $t = kT$ at the receiver. Figure 3.30a shows the three pulses corresponding to the sequence 110,

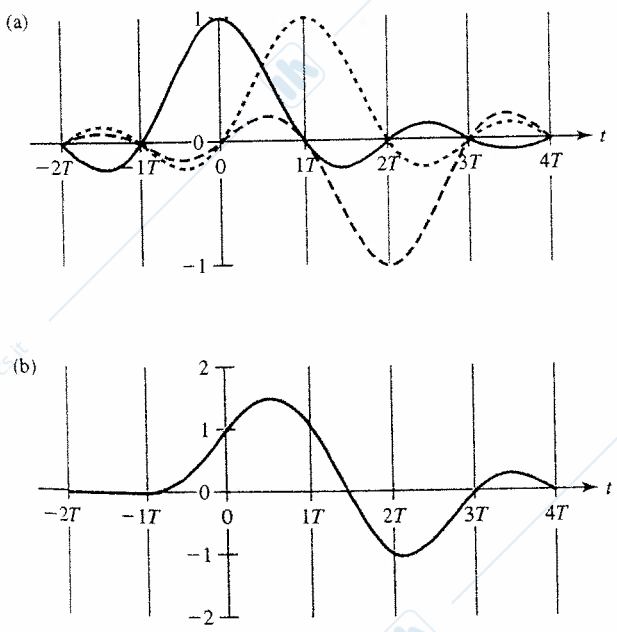


FIGURE 3.30 System response to binary input 110: (a) three separate pulses; (b) combined signal.

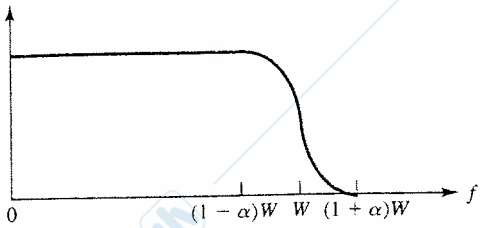


FIGURE 3.31 Raised cosine transfer function.

before they are added. Figure 3.30b shows the signal that results when the three pulses are combined. It can be seen that the combined signal has the correct values at $t = 0$, 1, and 2.

The above transmission system sends a bit every T seconds, where $T = 1/2W$ and W is the bandwidth of the overall system in Figure 3.29. For example, if $W =$ 1 MHz, then pulses would be sent every $T = 1/2{,}000{,}000 = 0.5$ microseconds, which corresponds to a rate of 2,000,000 pulses/second. A bit is sent with every pulse, so the bit rate is 2 Mbits/second. The **Nyquist Signaling Rate** is defined by

$$r_{max} = 2W \text{ pulses/second} \tag{3.28}$$

The Nyquist rate r_{max} *is the maximum signaling rate that is achievable through an ideal low-pass channel with no intersymbol interference.*

We already noted that the ideal low-pass system in Figure 3.10 cannot be implemented in practice. Nyquist also found other pulses that have zero intersymbol interference but that require some additional bandwidth. Figure 3.31 shows the amplitude-response function for one such pulse. Here a transition region with odd symmetry about $f = W$ is introduced. The more gradual roll-off of these systems makes the appropriate transmitter and receiver filters simpler to attain in practice.

The operation of the baseband transmission systems in this section depends critically on having the receiver synchronized precisely to intervals of duration T. Additional processing of the received signal is carried by the receiver to obtain this synchronization. If the receiver loses synchronization, then it will start sampling the signal at time instants that do contain intersymbol interference. The use of pulses corresponding to the system in Figure 3.31 provides some tolerance to small errors in sampling time.

3.5.2 The Shannon Channel Capacity

Up to this point we have been assuming that the input pulses can have only two values, 0 or 1. This restriction can be relaxed, and the pulses can be allowed to assume a greater number of values. Consider **multilevel transmission** where binary information is transmitted in a system that uses one of 2^m distinct levels in each input pulse. The binary information sequence can be broken into groups of m bits. Proceeding as before, each T seconds the transmitter accepts a group of m bits. These m bits determine a unique amplitude of the pulse that is to be input into the system. As long as the signaling rate does not exceed the Nyquist rate $2W$, the interference between pulses will still be zero, and by measuring the output at the right time instant we will be able to determine

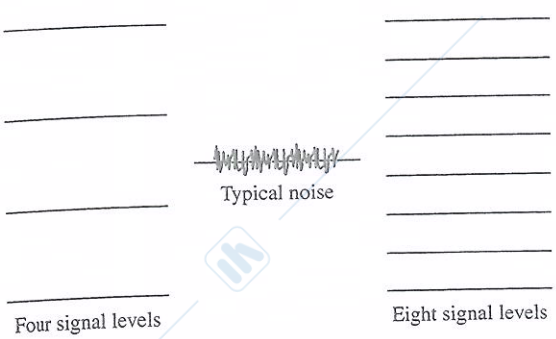


FIGURE 3.32 Effect of noise on transmission errors as number of levels is increased.

the input. Thus if we suppose that we transmit $2W$ pulses/second over a channel that has bandwidth W and if the number of amplitude values is 2^m, then the **bit rate** R of the system is

$$R = 2W \text{ pulses/second} \times m \text{ bits/pulse} = 2Wm \text{ bits/second} \tag{3.29}$$

In principle we can attain arbitrarily high bit rates by increasing the number of levels 2^m. However, we cannot do so in practice because of the limitations on the accuracy with which measurements can be made and also the presence of random noise. The random noise implies that the value of the overall response at time $t = kT$ will be the sum of the input amplitude plus some random noise. This noise can cause the measurement system to make an incorrect decision. *To keep the probability of decision errors small, we must maintain some minimum spacing between amplitude values* as shown in Figure 3.32. Here four signal levels are shown next to the typical noise. In the case of four levels, the noise is not likely to cause errors when it is added to a given signal level. Figure 3.32 also shows a case with eight signal levels. It can be seen that if the spacing between levels becomes too small, then the noise signals can cause the receiver to make the wrong decision.

We can make the discussion more precise by considering the statistics of the noise signal. Figure 3.33 shows the Gaussian probability density function, which is frequently a good model for the noise amplitudes. The density function gives the relative frequency of occurrence of the noise amplitudes. It can be seen that for the Gaussian density, the amplitudes are centered around zero. The average power of this noise signal is given by σ^2, where σ is the standard deviation of the noise.

Consider how errors occur in multilevel transmission. If we have maximum amplitudes $\pm A$ and M levels, then the separation between adjacent levels is $\delta = 2A/(M-1)$. When an interior signal level is transmitted, an error occurs if the noise causes the received signal to be closer to one of the other signal levels. This situation occurs if the

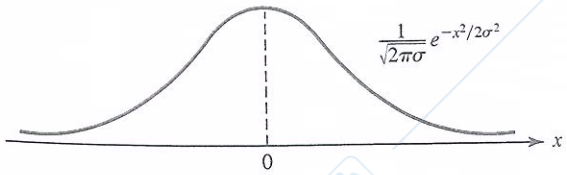


FIGURE 3.33 Gaussian probability density function.

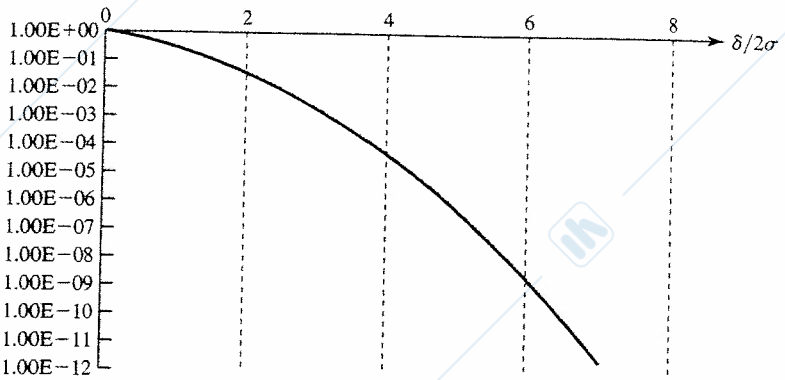


FIGURE 3.34 Probability of error for an interior signal level.

noise amplitude is greater than $\delta/2$ or less than $-\delta/2$. Thus the probability of error for an interior signal level is given by

$$P_e = \int_{-\infty}^{-\delta/2} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}\,dx + \int_{\delta/2}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} e^{-x^2/2\sigma^2}\,dx = 2\int_{\delta/2\sigma}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2}\,dx$$
$$= 2Q\left(\frac{\delta}{2\sigma}\right) \tag{3.30}$$

The expression on the right-hand side is evaluated using tables or analytic aproximations [Leon-Garcia, 1994, Chapter 3]. Figure 3.34 shows how P_e varies with $\delta/2\sigma = A/(M-1)\sigma$. For larger values of separation $\delta/2\sigma$, large decreases in the probability of error are possible with small increases in $\delta/2\sigma$. However, as the number of signal levels M is increased, $\delta/2\sigma$ is decreased, leading to large increases in the probability of error. We conclude that the bit rate cannot be increased to arbitrarily high values by increasing M without incurring significantly higher bit error rates.

We have now seen that two parameters affect the performance of a digital transmission system: bandwidth and SNR. Shannon addressed the question of determining the maximum achievable bit rate at which reliable communication is possible over an ideal channel of bandwidth W and of a given SNR. The phrase *reliable communication* means that it is possible to achieve arbitrarily small error probabilities by using sufficiently complex coding. Shannon derived the channel capacity for such a channel under the condition that the noise has a Gaussian distribution. This **channel capacity** is given by the following formula:

$$C = W \log_2(1 + \text{SNR}) \text{ bits/second} \tag{3.31}$$

Shannon showed that the probability of error can be made arbitrarily small only if the transmission rate R is less than channel capacity C. Therefore, the channel capacity is the maximum possible transmission rate over a system with given bandwidth and SNR.

SHANNON CHANNEL CAPACITY AND THE 56KBPS MODEM

The Shannon channel capacity for a telephone channel gives a maximum possible bit rate of 45.2 kbps at 40 dB SNR. How is it that the V.90 modems achieve rates of 56 kbps? In fact, a look at the fine print shows that the bit rate is 33.6 kbps inbound into the network. The inbound modem signal must undergo an analog-to-digital conversion when it is converted to PCM at the entrance to the telephone network. This step introduces the PCM approximation error or noise. At the maximum allowable signal level, we have a maximum possible SNR of 39 dB, so the 56 kbps is not attainable in the inbound direction, and hence the inbound operation is at 33.6 kbps. In the direction from the Internet server provider (ISP) to the user, the signal from the ISP is already digital and so it does not need to undergo analog-to-digital conversion. Hence the quantization noise is not introduced, a higher SNR is possible, and speeds approaching 56 kbps can be achieved from the network to the user.

3.6 LINE CODING

Line coding is the method used for converting a binary information sequence into a digital signal in a digital communications system. The selection of a line coding technique involves several considerations. In the previous sections, we focused on maximizing the bit rate over channels that have limited bandwidths. Maximizing bit rate is the main concern in digital transmission when bandwidth is at a premium. However, in other situations, such as in LANs, other concerns are also of interest. For example, an important design consideration is the ease with which the bit timing information can be recovered from the digital signal so that the receiving sample clock can maintain its synchronization with respect to the transmitting clock. Many systems do not pass dc and low-frequency components, so another design consideration is that the line code produce a signal that does not have dc and low-frequency content. Also, some line coding methods have built-in error detecting capabilities, and some methods have better immunity to noise and interference. Finally, the complexity and the cost of the line code implementations are always factors in the selection for a given application.

Figure 3.35 shows various line codes that are used in practice. The figure shows the digital signals that are produced by the line codes for the binary sequence 101011100. The simplest scheme is the unipolar **nonreturn-to-zero (NRZ) encoding** in which a binary 1 is transmitted by sending a $+A$ voltage level, and a 0 is transmitted by sending a 0 voltage. If binary 0s and 1s both occur with probability 1/2, then the average transmitted power for this line code is $(1/2)A^2 + (1/2)0^2 = A^2/2$. The **polar NRZ encoding** method that maps a binary 1 to $+A/2$ and binary 0 to $-A/2$ is more efficient than unipolar NRZ in terms of average transmitted power. Its average power is given by $(1/2)(+A/2)^2 + (1/2)(-A/2)^2 = A^2/4$.

The spectrum that results from applying a given line code is of interest. We usually assume that the binary information is equally likely to be 0 or 1 and that bits are

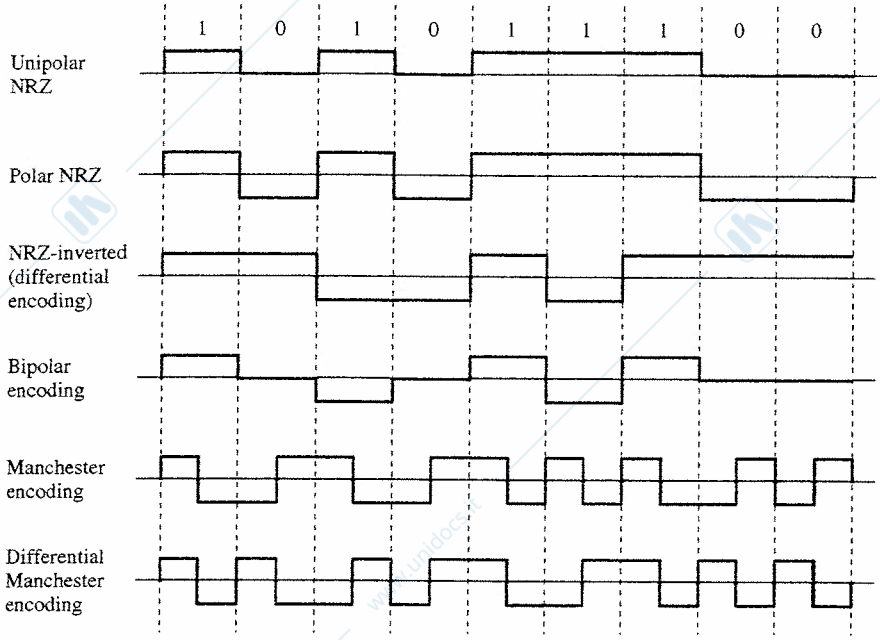


FIGURE 3.35 Line coding methods.

statistically independent of each other, much as if they were produced by a sequence of independent coin flips. The unipolar and the polar NRZ encoding methods have the same frequency components because they produce essentially the same variations in a signal as a function of time. Strings of consecutive 0s and consecutive 1s lead to periods where the signal remains constant. These strings of 0s and 1s occur frequently enough to produce a spectrum that has its components concentrated at the lower frequencies as shown in Figure 3.36.[15] This situation presents a problem when the communications channel does not pass low frequencies. For example, most telephone transmission systems do not pass the frequencies below about 200 Hz.

The **bipolar encoding** method was developed to produce a spectrum that is more amenable to channels that do not pass low frequencies. In this method binary 0s are mapped into 0 voltage, thus making no contribution to the digital signals; consecutive 1s are alternately mapped into $+A/2$ and $-A/2$. Thus a string of consecutive 1s will produce a square wave with the frequency $1/2T$ Hz. As a result, the spectrum for the bipolar code has its frequency content centered around the frequency $1/2T$ Hz and has small content at low frequencies as shown in Figure 3.36.

Timing recovery is an important consideration in the selection of a line code. The timing-recovery circuit in the receiver monitors the transitions at the edge of the bit

[15]The formulas for the spectra produced by the line codes in Figure 3.36 can be found in [Smith 1985, pp. 198–203].

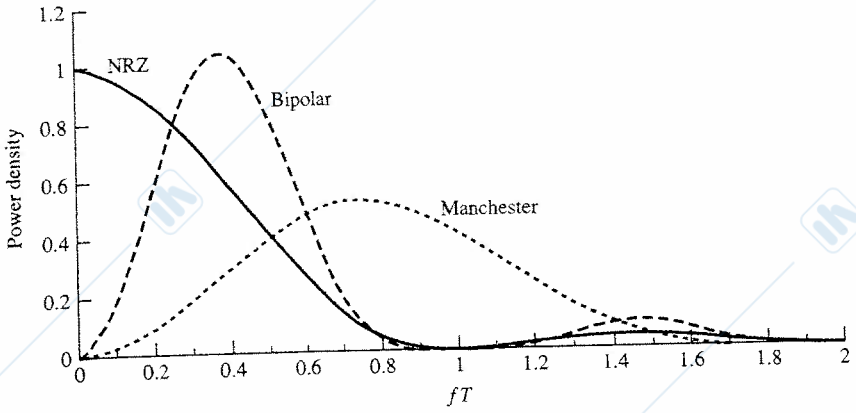


FIGURE 3.36 Spectra for different line codes.

intervals to determine the boundary between bits, and generates a clock signal that determines when the received signal is sampled. Long strings of 0s and 1s in the binary and the polar binary encodings can cause the timing circuit to lose synchronization because of the absence of transitions. In the bipolar encoding long strings of 1s result in a square wave that has strong timing content; however, long strings of 0s still pose a problem. To address this problem, the bipolar line codes used in telephone transmission systems place a limit on the maximum number of 0s that may be encoded into the digital signal. Whenever a string of N consecutive 0s occurs, the string is encoded into a special binary sequence that contains 0s and 1s. To alert the receiver that a substitution has been made, the sequence is encoded so that the mapping in the bipolar line code is violated; that is, two consecutive 1s do not alternate in polarity.

A problem with polar coding is that a systematic error in polarity can cause all 0s to be detected as 1s and all 1s as 0s.[16] The problem can be avoided by mapping the binary information into *transitions* at the beginning of each interval. A binary 1 is transmitted by enforcing a transition at the beginning of a bit time, and a 0 by having no transition. The signal level within the actual bit time remains constant. Figure 3.35 shows an example of how **differential encoding**, or **NRZ inverted**, carries out this mapping. Starting at a given level, the sequence of bits determines the subsequent transitions at the beginning of each interval. Note that differential encoding will lead to the same spectrum as binary and polar encoding. However, errors in differential encoding tend to occur in pairs. An error in one bit time will provide the wrong reference for the next time, thus leading to an additional error in the next bit.

[16] This polarity inversion occurs when the polar-encoded stream is fed into a phase modulation system such as the one discussed in Section 3.7.

EXAMPLE **Ethernet and Token-Ring Line Coding**

Bipolar coding has been used in long-distance transmission where bandwidth efficiency is important. In LANs, where the distances are short, bandwidth efficiency is much less important than cost per station. The Manchester encodings shown in Figure 3.35 are used in Ethernet and token-ring LAN standards. In **Manchester encoding** a binary 1 is denoted by a transition from $A/2$ to $-A/2$ in the middle of the bit time interval, and a binary 0 by a transition from $-A/2$ to $A/2$. The Manchester encoding is said to be self-clocking: The presence of a transition in the middle of every bit interval makes timing recovery particularly easy and also results in small content at low frequencies. However, the pulse rate is essentially double that of binary encoding, and this factor results in a spectrum with significantly larger bandwidth as shown in Figure 3.36. **Differential Manchester encoding**, which is used in token-ring networks, retains the transition in the middle of every bit time, but the binary sequence is mapped into the presence or absence of transitions in the beginning of the bit intervals. In this type of encoding, a binary 0 is marked by a transition at the beginning of an interval, whereas a 1 is marked by the absence of a transition.

Note that the Manchester encoding can be viewed as the transmission of two pulses for each binary bit. A binary 1 is mapped into the binary pair of 10, and the corresponding polar encoding for these two bits is transmitted; A binary 0 is mapped into 01. The Manchester code is an example of a mBnB code (where m is 1 and n is 2) in which m information bits are mapped into $n > m$ encoded bits. The encoded bits are selected so that they provide enough pulses for timing recovery and limit the number of pulses of the same level.

Optical transmission systems use the intensity of a light pulse and hence can only take on a positive value and a zero value. For example, an optical version of a Manchester code uses the above encoding in unipolar format. A 4B5B code is used in the optical fiber transmission system in the Fiber Distributed Data Interface (FDDI) LAN, and an 8B10B line code is used in Gigabit Ethernet.

3.7 MODEMS AND DIGITAL MODULATION

In Section 3.5 we considered digital transmission over channels that are low pass in nature. We now consider band-pass channels that do not pass the lower frequencies and instead pass power in some frequency range from f_1 to f_2, as shown in Figure 3.37. We assume that the bandwidth of the channel is $W = f_2 - f_1$ and discuss the use of modulation to transmit digital information over this type of channel. The basic function of the modulation is to produce a signal that contains the information sequence and that occupies frequencies in the range passed by the channel. A **modem** is a device that carries out this basic function. In this section we first consider the principles of digital modulation, and then we show how these principles are applied in telephone modem standards.

Let f_c be the frequency in the center of the band-pass channel in Figure 3.37; that is, $f_c = (f_1 + f_2)/2$. The sinusoidal signal $\cos(2\pi f_c t)$ has all of its power located

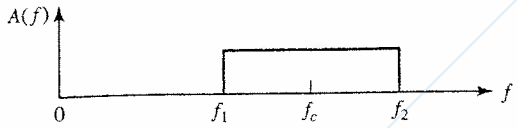


FIGURE 3.37 Bandpass channel passes frequencies in the range f_1 to f_2.

precisely at frequency f_c. The various types of modulation schemes involve imbedding the binary information sequence into the transmitted signal by varying, or modulating, some attribute of the sinusoidal signal. In **amplitude shift keying (ASK)** the sinusoidal signal is turned on and off according to the information sequence as shown in Figure 3.38a. The demodulator for an ASK system needs only to determine the presence or absence of a sinusoid in a given time interval. In **frequency shift keying (FSK)**, shown in Figure 3.38b, the frequency of the sinusoid is varied according to the information. If the information bit is a 0, the sinusoid has frequency $f_1 = f_c - \varepsilon$, and if it is a 1, the sinusoid has a frequency $f_2 = f_c + \varepsilon$. The demodulator for an FSK system must be able to determine which of two possible frequencies is present at a given time. In **phase shift keying (PSK)**, the phase of the sinusoid is altered according to the information sequence. In Figure 3.38c a binary 1 is transmitted by $\cos(2\pi f_c t)$, and a binary 0 is transmitted by $\cos(2\pi f_c t + \pi)$. Because $\cos(2\pi f_c t + \pi) = -\cos(2\pi f_c t)$, we note that this PSK scheme is equivalent to multiplying the sinusoidal signal by $+1$ when the information is a 1 and by -1 when the information bit is a 0. Thus the demodulator for a PSK system must be able to determine the phase of the received sinusoid with respect to some reference phase. In the remainder of this section, we concentrate on phase modulation techniques.

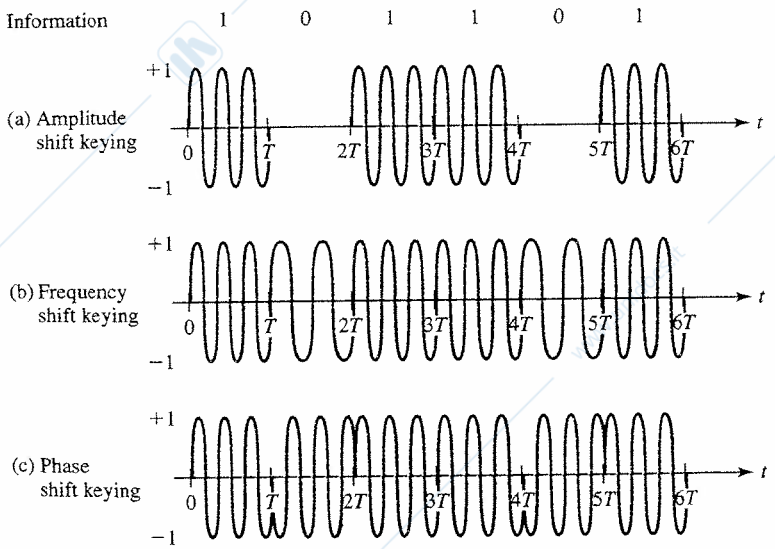


FIGURE 3.38 Amplitude, frequency, and phase modulation techniques.

3.7.1 Binary Phase Modulation

Consider the problem of transmitting a binary information sequence over an ideal band-pass channel using PSK. We are interested in developing modulation techniques that can achieve pulse rates that are comparable to that achieved by Nyquist signaling over low-pass channels. In Figure 3.39c we show the waveform $Y_i(t)$ that results when a binary 1 is transmitted using a cosine wave with amplitude $+A$, and a binary 0 is transmitted using a cosine wave with amplitude $-A$. The corresponding modulator is shown in Figure 3.40a. Every T seconds the modulator accepts a new binary information symbol and adjusts the amplitude A_k accordingly. In effect, as shown in Figure 3.39c, the modulator transmits a T-second segment of the signal as follows:

$+A\cos(2\pi f_c t)$ if the information symbol is a 1.
$-A\cos(2\pi f_c t)$ if the information symbol is a 0.

Note that the modulated signal is no longer a pure sinusoid, since the overall transmitted signal contains glitches between the T-second intervals, but its primary oscillations are still around the center frequency f_c; therefore, we expect that the power of the signal will be centered about f_c and hence located in the range of frequencies that are passed by the band-pass channel.

By monitoring the polarity of the signal over the intervals of T seconds, a receiver can recover the original information sequence. Let us see more precisely how this recovery may be accomplished. As shown in Figure 3.40b suppose we multiply the modulated signal $Y_i(t)$ by $2\cos(2\pi f_c t)$. The resulting signal is $+2A\cos^2(2\pi f_c t)$ if the original information symbol is a 1 or $-2A\cos^2(2\pi f_c t)$ if the original information symbol is 0. Because $2\cos^2(2\pi f_c t) = (1 + \cos(4\pi f_c t))$, we see that the resulting signals are as shown in Figure 3.39d. By smoothing out the oscillatory part with a

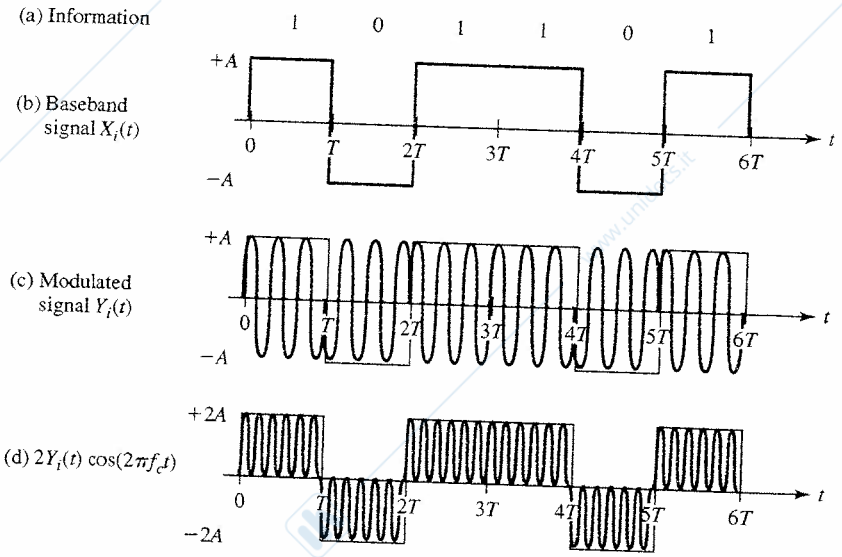


FIGURE 3.39 Modulating a signal.

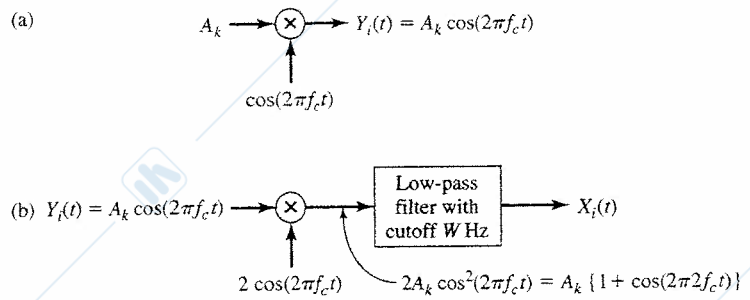


FIGURE 3.40 Modulator and demodulator: (a) Modulate $\cos(2\pi f_c t)$ by multiplying it by A_k for $(k-1)T < t < kT$; (b) Demodulate (recover) A_k by multiplying by $2\cos(2\pi f_c t)$ and low-pass filtering.

so-called low-pass filter, we can easily determine the original baseband signal $X_i(t)$ and the A_k and subsequently the original binary sequence.

3.7.2 QAM and Signal Constellations

When we developed the Nyquist signal result in Section 3.5.1, we found that for a low-pass channel of bandwidth W Hz the maximum signaling rate is $2W$ pulses/second. It can be shown that the system we have just described in the previous section can transmit only W pulses/second over a band-pass channel that has bandwidth W.[17] Consequently, the time per pulse is given by $T = 1/W$. Thus this scheme attains only half the signaling rate of the low-pass case. Next we show how we can recover this factor of 2 by using Quadrature Amplitude Modulation.

Suppose we have an original information stream that is generating symbols at a rate of $2W$ symbols/second. In **Quadrature Amplitude Modulation (QAM)** we split the original information stream into two sequences that consist of the odd and even symbols, say, B_k and A_k, respectively, as shown in Figure 3.41. Each sequence now has the rate W symbols/second. Suppose we take the even sequence A_k and produce a modulated signal by multiplying it by $\cos(2\pi f_c t)$; that is, $Y_i(t) = A_k \cos(2\pi f_c t)$ for a T-second interval. As before, this modulated signal will be located within the band of the band-pass channel. Now suppose that we take the odd sequence B_k and produce another modulated signal by multiplying it by $\sin(2\pi f_c t)$; that is, $Y_q(t) = B_k \sin(2\pi f_c t)$ for a T-second interval. This modulated signal also has its power located within the band of the band-pass channel. We finally obtain a composite modulated signal by adding $Y_i(t)$ and $Y_q(t)$, as shown in Figure 3.41.

$$Y(t) = Y_i(t) + Y_q(t) = A_k \cos(2\pi f_c t) + B_k \sin(2\pi f_c t). \tag{3.32}$$

[17] In the remainder of this section, we continue to use the term *pulse* for the signal that is transmitted in a T-second interval.

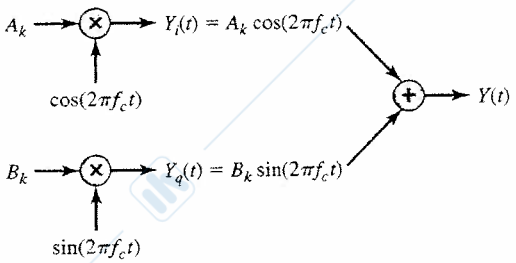


FIGURE 3.41 QAM modulator: Modulate $\cos(2\pi f_c t)$ and $\sin(2\pi f_c t)$ by multiplying them by A_k and B_k, respectively, for $(k-1)T < t < kT$.

This equation shows that we have generated what amounts to a two-dimensional modulation scheme. The first component A_k is called the *in-phase component*; the second component B_k is called the *quadrature-phase component*.

We now transmit the sum of these two modulated signals over the band-pass channel. The composite sinusoidal signal $Y(t)$ will be passed without distortion by the linear band-pass channel. We now need to demonstrate how the original information symbols can be recovered from $Y(t)$. We will see that our ability to do so depends on the following properties of cosines and sines:

$$2\cos^2(2\pi f_c t) = 1 + \cos(4\pi f_c t) \tag{3.33}$$

$$2\sin^2(2\pi f_c t) = 1 - \sin(4\pi f_c t) \tag{3.34}$$

$$2\cos(2\pi f_c t)\sin(2\pi f_c t) = 0 + \sin(4\pi f_c t) \tag{3.35}$$

These properties allow us to recover the original symbols as shown in Figure 3.42. By multiplying $Y(t)$ by $2\cos(2\pi f_c t)$ and then low-pass filtering the resulting signal, we obtain the sequence A_k. Note that the cross-product term $B_k(t)\sin(4\pi f_c t)$ is removed by the low-pass filter. Similarly, the sequence B_k is recovered by multiplying $Y(t)$ by $2\sin(2\pi f_c t)$ and low-pass filtering the output. Thus QAM is a two-dimensional system that achieves an effective signaling rate of $2W$ pulses/second over the band-pass channel of W Hz. This result matches the performance of the Nyquist signaling procedure that we developed in Section 3.5.

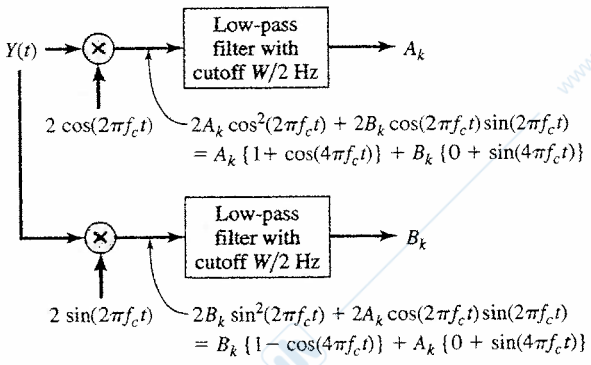
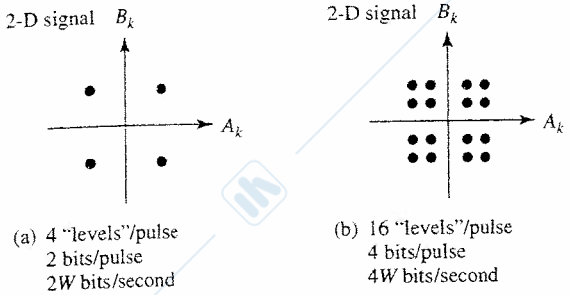


FIGURE 3.42 QAM demodulator A_k.



(a) 4 "levels"/pulse
2 bits/pulse
$2W$ bits/second

(b) 16 "levels"/pulse
4 bits/pulse
$4W$ bits/second

FIGURE 3.43 (a) 4-point and (b) 16-point signal constellations.

The two-dimensional nature of the above signaling scheme can be used to plot the various combinations of levels that are allowed in a given signaling interval of T seconds. (Note: It is important to keep in mind that $T = 1/W$ in this discussion). In the case considered so far, the term multiplying the cosine function can assume the value $+A$ or $-A$; the term multiplying the sine function can also assume the value $+A$ or $-A$. In total, four combinations of these values can occur. These are shown as the four points in the two-dimensional plane in Figure 3.43a. We call the set of signal points a **signal constellation**. At any given T-second interval, only one of the four points in this signal constellation can be in use. It is therefore clear that in every T-second interval we are transmitting two bits of information. As in the case of baseband signaling, we can increase the number of bits that can be transmitted per T-second interval by increasing the number of levels that are used. Figure 3.43b shows a 16-point constellation that results when the terms multiplying the cosine and sine functions are allowed to assume four possible levels. In this case only one of the 16 points in the constellation is in use in any given T-second interval, and hence four bits of information are transmitted at every such interval.

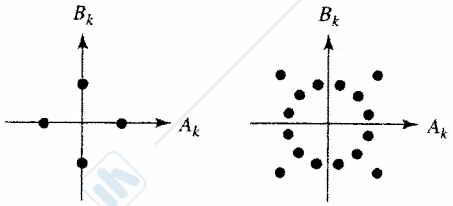
Another way of viewing QAM is as the simultaneous modulation of the amplitude and phase of a carrier signal, since

$$A_k \cos(2\pi f_c t) + B_k \sin(2\pi f_c t) = \left(A_k^2 + B_k^2\right)^{1/2} \cos(2\pi f_c t + \tan^{-1} B_k/A_k) \quad (3.36)$$

Each signal constellation point can then be seen as determining a specific amplitude and phase.

Many signal constellations that are used in practice have nonrectangular arrays of signal points such as those shown in Figure 3.44. To produce this type of signaling, we need to modify the above encoding scheme only slightly. Suppose that the constellation has 2^m points. Each T-second interval, the transmitter accepts m information bits, identifies the constellation point assigned to these bits, and then transmits cosine and sine signals with the amplitudes that correspond to the constellation point.

The presence of noise in transmission systems implies that the pair of recovered values for the cosine and sine components will differ somewhat from the transmitted values. This pair of values will therefore specify a point in the plane that deviates from the transmitted constellation point. The task of the receiver is to take the received pair of values and identify the closest constellation point.



4 "levels"/pulse
2 bits/pulse
$2W$ bits/second

16 "levels"/pulse
4 bits/pulse
$4W$ bits/second

FIGURE 3.44 Other signal constellations.

3.7.3 Telephone Modem Standards

Signal constellation diagrams of the type shown in Figure 3.43 and Figure 3.44 are used in the various signaling standards that have been adopted for use over telephone lines. For the purposes of data communications, most telephone channels have a usable bandwidth in the range $f_1 = 500$ Hz to $f_2 = 2900$ Hz. This implies $W = 2400$ and hence a signaling rate of $1/T = W = 2400$ pulses/second. Table 3.4 lists some of the parameters that specify the ITU V.32bis and V.34bis modem standards that are in current use. Each standard can operate at a number of speeds that depend on the quality of the channel available. Both standards operate at a rate of 2400 pulses/second, and the actual bit rate is determined by which constellation is used. The QAM 4 systems uses four constellation points and hence two bits/pulse, giving a bit rate of 4800 bps.

Trellis modulation systems are more complex in that they combine error-correction coding with the modulation. In trellis modulation the number of constellation points is 2^{m+1}. At every T-second interval, the trellis coding algorithm accepts m bits and generates $m + 1$ bits that specify the constellation point that is to be used. In effect, only 2^m out of the 2^{m+1} possible constellation points are valid during any given interval. This extra degree of redundancy improves the robustness of the modulation scheme with respect to errors. In Table 3.4, the trellis 32 system has 2^5 constellation points out of which 16 are valid at any given time; thus the bit rate is $4 \times 2400 = 9600$ bps. Similarly, the trellis 128 system gives a bit rate of $6 \times 2400 = 14{,}400$ bps.

The V.34bis standard can operate at rates of 2400, 2743, 2800, 3000, 3200, or 3429 pulses/second. The modem precedes communications with an initial phase during which the channel is probed to determine the usable bandwidth in the given telephone

TABLE 3.4 Modem standards.

V.32bis	**Modulation**	**Pulse rate**
14,000 bps	Trellis 128	2400 pulses/second
9600 bps	Trellis 32	2400 pulses/second
4800 bps	QAM 4	2400 pulses/second
V.34bis		
2400–33,600 bps	Trellis 960	2400–3429 pulses/second

connection. The modem then selects a pulse rate. For each of these pulse rates, a number of possible trellis encoding schemes are defined. Each encoding scheme selects a constellation that consists of a subset of points from a superconstellation of 860 points. A range of bit rates are possible, including 2400, 4800, 9600, 14,400, 19,200 and 28,800, 31,200 and 33,600 bps.

It is instructive to consider how close the V.34bis modem comes to the maximum possible transmission rate predicted by Shannon's formula for the traditional telephone channel. Suppose we have a maximum useful bandwidth of 3400 Hz and assume a maximum SNR of 40 dB, which is a bit over the maximum possible in a telephone line. Shannon's formula then gives a maximum possible bit rate of 45,200 bits/second. It is clear then that the V.34bis modem is coming close to achieving the Shannon bound. In 1997 a new class of modems, the ITU-T V.90 standard, was introduced that tout a bit rate of 56,000 bits/second, well in excess of the Shannon bound! As indicated earlier, the 56 kbps speed is attained only under particular conditions that do not correspond to the normal telephone channel.[18]

BEEEEEEP, CHIRP, CHIRP, KTWANG, KTWANG, shhhhh, SHHHHHH

What are the calling and answering V.34 modems up to when they make these noises? They are carrying out the handshaking that is required to set up communication. V.34 handshaking has four phases:

Phase 1: Because the modems don't know anything about each other's characteristics, they begin by communicating using simple, low-speed, 300 bps FSK. They exchange information about the available modulation modes, the standards they support, the type of error correction that is to be used, and whether the connection is cellular.

Phase 2: The modems perform probing of the telephone line by transmitting tones with specified phases and frequencies that are spaced 150 Hz apart and that cover the range from 150 Hz to 3750 Hz. The tones are sent at two distinct signal levels (amplitudes). The probing allows the modems to determine the bandwidth and distortion of the telephone line. The modems then select their carrier frequency and signal level. At the end of this phase, the modems exchange information about their carrier frequency, transmit power, symbol rate, and maximum data rate.

Phase 3: The receiver equalizer starts its adaptation to the channel. The echo canceler is started. The function of the echo canceler is to suppress its modem's transmission signal so that the modem can hear the arriving signal.

Phase 4: The modems exchange information about the specific modem parameters that are to be used, for example, signal constellation, trellis encoding, and other encoding parameters.

The modems are now ready to work for you!

[18] Indeed, we will see later when we discuss transmission medium that much higher bit rates are attainable over twisted-wire pairs.

Modem standards have also been developed for providing error control as well as data compression capabilities. The V.42bis standard specifies the Link Access Procedure for Modems (LAPM) for providing error control. LAPM is based on the HDLC data link control, which is discussed in Chapter 5. V.42bis also specifies the use of the Lempel-Ziv data compression scheme for the compression of information prior to transmission. This scheme can usually provide compression ratios of two or more, thus providing an apparent modem speedup of two or more. The data compression scheme is explained in Chapter 12.

In this section we have considered only telephone modem applications. Digital modulation techniques are also used extensively in other digital transmission systems such as digital cellular telephony and terrestrial and satellite communications. These systems are discussed further in Section 3.8.

3.8 PROPERTIES OF MEDIA AND DIGITAL TRANSMISSION SYSTEMS

For transmission to occur, we must have a *transmission medium* that conveys the energy of a signal from a sender to a receiver. A communication system places transmitter and receiver equipment on either end of a transmission medium to form a communications channel. In previous sections we discussed how communications channels are characterized in general. In this section we discuss the properties of the transmission media that are used in modern communication networks.

We found that the capability of a channel to carry information reliably is determined by several properties. First, the manner in which the medium transfers signals at various frequencies (that is, the amplitude-response function $A(f)$ and the phase-shift function $\varphi(f)$ determines the extent to which the input pulses are distorted. The transmitter and receiver equipment must be designed to remove enough distortion to make reliable detection of the pulses possible. Second, the transmitted signal is attenuated as it propagates through the medium and noise is also introduced in the medium and in the receiver. These phenomena determine the SNR at the receiver and hence the probability of bit errors. In discussing specific transmission media, we are therefore interested in the following characteristics:

- The amplitude-response function $A(f)$ and the phase-shift function $\varphi(f)$ of the medium and the associated bandwidth as a function of distance.
- The susceptibility of the medium to noise and interference from other sources.

A typical communications system transmits information by modulating a sinusoidal signal of frequency f_0 that is inserted into a guided medium or radiated through an antenna. The sinusoidal variations of the modulated signal propagate in a medium at a speed of v meters/second, where

$$v = \frac{c}{\sqrt{\varepsilon}} \tag{3.37}$$

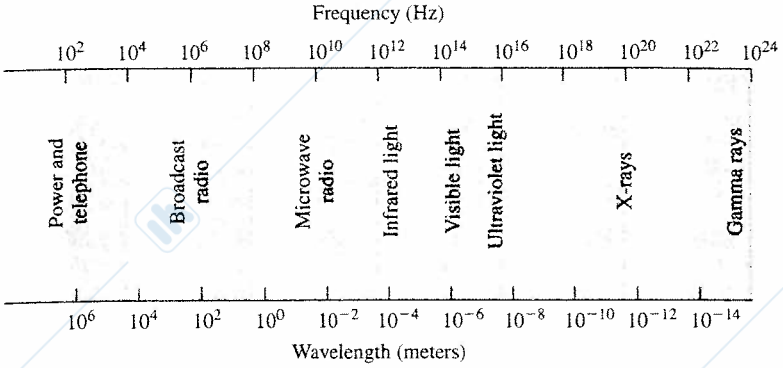


FIGURE 3.45 Electromagnetic spectrum.

and where $c = 3 \times 10^8$ meters/second is the speed of light in a vacuum and ε is the dielectric constant of the medium. In free space, $\varepsilon = 1$, and $\varepsilon \geq 1$ otherwise. The **wavelength** λ of the signal is given by the distance in space spanned by one period of the sinusoid:

$$\lambda = v/f_0 \text{ meters} \tag{3.38}$$

As shown in Figure 3.45, a 100 MHz carrier signal, which corresponds to FM broadcast radio, has a wavelength of 3 meters, whereas a 3 GHz carrier signal has a wavelength of 10 cm. Infrared light covers the range from 10^{12} to 10^{14} Hz, and the light used in optical fiber occupies the range 10^{14} to 10^{15} Hz.

The speed of light c is a maximum limit for propagation speed in free space and cannot be exceeded. Thus if a pulse of energy enters a communications channel of distance d at time $t = 0$, then none of the energy can appear at the output before time $t = d/c$ as shown in Figure 3.46. Note that in copper wire signals propagate at a speed of 2.3×10^8 meters/second, and in optical fiber systems the speed of light v is approximately 2×10^8 meters/second.

The two basic types of media are wired media, in which the signal energy is contained and guided within a solid medium, and wireless media, in which the signal energy propagates in the form of unguided electromagnetic signals. Copper pair wires, coaxial cable, and optical fiber are examples of wired media. Radio and infrared light are examples of wireless media.

Wired and wireless media differ in a fundamental way. In its most basic form, wired media provide communications from point to point. By interconnecting wires at

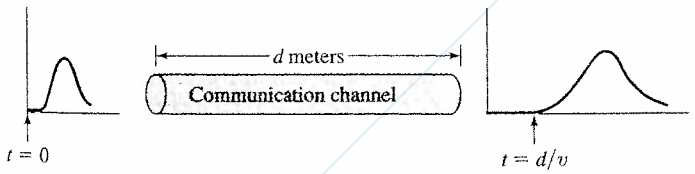


FIGURE 3.46 Propagation delay of a pulse over the communication channel, where v is the speed of light in the medium.

various repeater or switching points, wired media lead to well-defined *discrete* network topologies. Since the energy is confined within the medium, additional transmission capacity can be procured by adding more wires. Unguided media, on the other hand, can achieve only limited directionality and can be transmitted, as in the case of broadcast radio, in all directions making the medium broadcast in nature. This condition leads to a network topology that is *continuous* in nature. In addition, all users within receiving range of each other must share the frequency band that is available and can thus interfere with each other. The radio spectrum is finite, and so, unlike wired media, it is not possible to procure additional capacity. A given frequency band can be reused only in a sufficiently distant geographical area. To maximize its utility, the radio spectrum is closely regulated by government agencies.

Another difference between wired and wireless media is that wired media require establishing a right-of-way through the land that is traversed by the cable. This process is complicated, costly, and time-consuming. On the other hand, systems that use wireless media do not require the right-of-way and can be deployed by procuring only the sites where the antennas are located. Wireless systems can therefore be deployed more quickly and at lower cost.

Finally, we note that for wired media the attenuation has an exponential dependence on distance; that is, the attenuation at a given frequency is of the form 10^{kd} where the constant k depends on the specific frequency and d is that distance. The attenuation for wired media in dB is then

$$\text{attenuation for wired media} = kd \text{ dB} \tag{3.39}$$

that is, the attenuation in dB increases linearly with the distance. For wireless media the attenuation is proportional to d^n where n is the *path loss exponent*. For free space $n = 2$, and for environments where obstructions are present $n > 2$. The attenuation for wireless media in dB is then

$$\text{attenuation for wireless media is proportional to } n \log_{10} d \text{ dB} \tag{3.40}$$

and so the attenuation in dB only increases logarithmically with the distance. Thus in general the signal level in wireless systems can be maintained over much longer distances than in wired systems.

3.8.1 Twisted Pair

The simplest guided transmission medium consists of two parallel insulated conducting (e.g., copper) wires. The signal is transmitted through one wire while a ground reference is transmitted through the other. This two-wire system is susceptible to crosstalk and noise. *Crosstalk* refers to the picking up of electrical signals from other adjacent wires. Because the wires are unshielded, there is also a tendency to pick up noise, or *interference*, from other electromagnetic sources such as broadcast radio. The receiver detects the information signal by the voltage difference between the ground reference signal and the information signal. If either one is greatly altered by interference or crosstalk, then the chance of error is increased. For this reason parallel two-wire lines are limited to short distances.

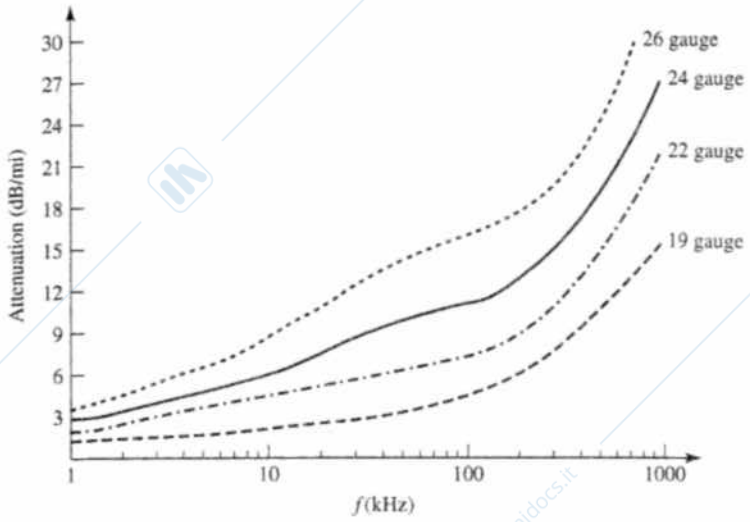


FIGURE 3.47 Attenuation versus frequency for twisted pair [after Smith 1985].

A **twisted pair** consists of two wires that are twisted together to reduce the susceptibility to interference. The close proximity of the wires means that any interference will be picked up by both and so the difference between the pair of wires should be largely unaffected by the interference. Twisting also helps to reduce (but not eliminate) the crosstalk interference when multiple pairs are placed within one cable. Much of the wire in the telephone system is twisted-pair wire. For example, it is used between the customer and the central office, also called the *subscriber loop*, and often between central offices, called the *trunk plant*. Because multiple pairs are bundled together within one telephone cable, the amount of crosstalk is still significant, especially at higher frequencies.

A twisted pair can pass a relatively wide range of frequencies. The attenuation for twisted pair, measured in dB/mile, can range from 1 to 4 dB/mile at 1 kHz to 10 to 20 dB/mile at 500 kHz, depending on the gauge (diameter) of the wire as shown in Figure 3.47. Since the attenuation/km is higher for higher frequencies, the bandwidth of twisted pair decreases with distance. Table 3.5 shows practical limits on data rates that

TABLE 3.5 Data rates of 24-gauge twisted pair.

Standard	Data rate	Distance
T-1	1.544 Mbps	18,000 feet, 5.5 km
DS2	6.312 Mbps	12,000 feet, 3.7 km
1/4 STS-1	12.960 Mbps	4500 feet, 1.4 km
1/2 STS-1	25.920 Mbps	3000 feet, 0.9 km
STS-1	51.840 Mbps	1000 feet, 300 m

can be achieved in a unidirectional link over a 24-gauge (0.016-inch-diameter wire) twisted pair for various distances.

The first digital transmission system used twisted pair and was used in the trunk portion of the telephone network. This T-1 carrier system achieved a transmission rate of 1.544 Mbps and could carry 24 voice channels. The T-1 carrier system used baseband pulse transmission with bipolar encoding. The T-1 carrier system is discussed further in Chapter 4 in the context of the telephone network. Twisted pair in the trunk portion of the telephone network is being replaced by optical fiber. However, twisted pair constitutes the bulk of the access network that connects users to the telephone office, and as such, is crucial to the evolution of future digital networks. In the remainder of this section, we discuss how new systems are being introduced to provide high-speed digital communications in the access network.

Originally, in optimizing for the transmission of speech, the telephone company elected to transmit frequencies within the range of 0 to 4 kHz. Limiting the frequencies at 4 kHz reduced the crosstalk that resulted between different cable pairs at the higher frequencies and provided the desired voice quality. Within the subscriber loop portion, *loading coils* were added to further improve voice transmission within the 3 kHz band by providing a flatter transfer function. This loading occurred in lines longer than 5 kilometers. While improving the quality of the speech signal, the loading coils also increased the attenuation at the higher frequencies and hence reduced the bandwidth of the system. Thus the choice of a 4 kHz bandwidth for the voice channel and the application of loading coils, not the inherent bandwidth of twisted pair, are the factors that limit digital transmission over telephone lines to approximately 40 kbps.

APPLICATION Digital Subscriber Loops

Several digital transmission schemes were developed in the 1970s to provide access to *Integrated Services Digital Network (ISDN)* using the twisted pair (without loading coils) in the subscriber loop network. These schemes provide for two bearer (B) 64 kbps channels and one data (D) 16 kbps channel from the user to the telephone network. These services were never deployed on a wide scale.

To handle the recent demand from consumers for higher speed data transmission, the telephone companies are introducing a new technology called **asymmetric digital subscriber line (ADSL)**. The objective of this technology is to use existing twisted-pair lines to provide the higher bit rates that are possible with unloaded twisted pair. The frequency spectrum is divided into two regions. The lower frequencies are used for conventional analog telephone signals. The region above is used for bidirectional digital transmission. The system is asymmetric in that the user can transmit upstream into the network at speeds ranging from 64 kbps to 640 kbps but can receive information from the network at speeds from 1.536 Mbps to 6.144 Mbps, depending on the distance from the telephone central office. This asymmetry in upstream/downstream transmission rates is said to match the needs of current applications such as upstream requests and downstream page transfers in the World Wide Web application.

The ITU-T G.992.1 standard for ADSL uses the Discrete Multitone (DMT) system that divides the available bandwidth into a large number of narrow subchannels. The binary information is distributed among the subchannels, each of which uses QAM. DMT can adapt to line conditions by avoiding subchannels with poor SNR. ITU-T has also approved standard G.992.2 as a "lite" version that provides access speeds of up to 512 kbps from the user and download speeds of up 1.5 Mbps. The latter is the simpler and less expensive standard because it does not require a "splitter" to separate telephone voice signals from the data signal and can instead be plugged directly into the PC by the user as is customary for most voiceband modems.

APPLICATION **Local Area Networks**

Twisted pair is installed during the construction of most office buildings. The wires that terminate at the wall plate in each office are connected to wiring closets that are placed at various locations in the building. Consequently, twisted pair is a good candidate for use in local area computer networks where the maximum distance between a computer and a network device is in the order of 100 meters. As a transmission medium, however, high-speed transmission over twisted pairs poses serious challenges. Several categories of twisted-pair cable have been defined for use in LANs. Category 3 **unshielded twisted pair (UTP)** corresponds to ordinary voice-grade twisted pair and can be used at speeds up to 16 Mbps. Category 5 UTP is intended for use at speeds up to 100 Mbps. Category 5 twisted pairs are twisted more tightly than are those in category 3, resulting in much better crosstalk immunity and signal quality. *Shielded twisted pair* involves providing a metallic braid or sheath to cover each twisted pair. It provides better performance than UTP but is more expensive and more difficult to use.

10BASE-T Ethernet LAN. The most widely deployed version of Ethernet LAN uses the 10BASE-T physical layer. The designation 10BASE-T denotes *10* Mbps operation using *base*band transmission over *t*wisted-pair wire. The NIC card in each computer is connected to a hub in a star topology as shown in Figure 3.48. Two category 3 UTP cables provide the connection between computer and hub. The transmissions use Manchester line coding, and the cables are limited to a maximum distance of 100 meters.

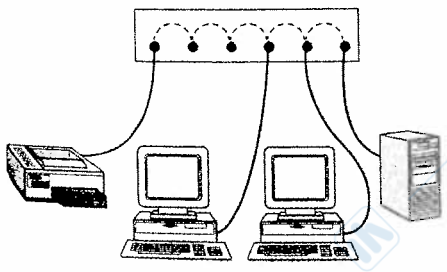


FIGURE 3.48 Ethernet hub.

100BASE-T Ethernet LAN. The 100BASE-T Ethernet LAN is also known as *Fast Ethernet*. As indicated by the designation, 100BASE-T Ethernet operates at a speed of 100 Mbps using twisted-pair wire. The computers are connected to a *hub* or a *switch* in a star topology, and the distance of the twisted pairs is limited to 100 meters. Operating 100 Mbps on UTP is challenging, and so three options for doing so were developed, one for category 3 UTP, one for shielded twisted pair, and one for category 5 UTP. One problem with extending the 10BASE-T transmission format is that Manchester line coding is inefficient in its use of bandwidth. Recall from the section on line coding that Manchester coding pulses vary at twice the information rate, so the use of Manchester coding would have required operation at 200 Mpulses/second. Another problem is that higher pulse rates result in more electromagnetic interference. For this reason, new and more efficient line codes were used in the new standards.

In the 100BASE-T4 format, four category 3 twisted-pair wires are used. At any given time three pairs are used to jointly provide 100 Mbps in a given direction; that is, each pair provides 33 1/3 Mbps. The fourth pair is used for collision detection. The transmission uses ternary signaling in which the transmitted pulses can take on three levels, $+A$, 0, or $-A$. The line code maps a group of eight bits into a corresponding group of six ternary symbols that are transmitted over the three parallel channels over two pulse intervals, or equivalently four bits into three ternary symbols/pulse interval. This mapping is possible because $2^4 = 16 < 3^3 = 27$. The transmitter on each pair sends 25 Mpulses/second, which gives a bit rate of 25 Mp/s $\times$ 4 bits/3 pulses = 33 1/3 Mbps as required. As an option, four category 5 twisted pairs can be used instead of category 3 twisted pairs.

In the 100BASE-TX format, two category 5 twisted pairs are used to connect to the hub. Transmission is full duplex with each pair transmitting in one of the directions at a pulse rate of 125 Mpulses/second. The line code used takes a group of four bits and maps it into five binary pulses, giving a bit rate of 125 Mpulses/second $\times$ 4 bits/5 pulses = 100 Mbps. An option allows two pairs of shielded twisted wire to be used instead of the category 5 pairs.

3.8.2 Coaxial Cable

In **coaxial cable** a solid center conductor is located coaxially within a cylindrical outer conductor. The two conductors are separated by a solid dielectric material, and the outer conductor is covered with a plastic sheath as shown in Figure 3.49. The coaxial arrangement of the two conductors provides much better immunity to interference and crosstalk

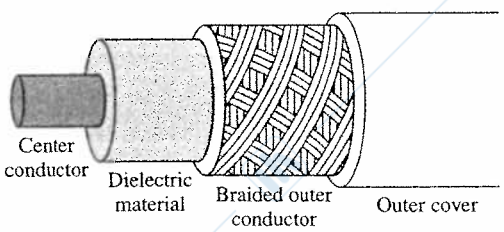


FIGURE 3.49 Coaxial cable.

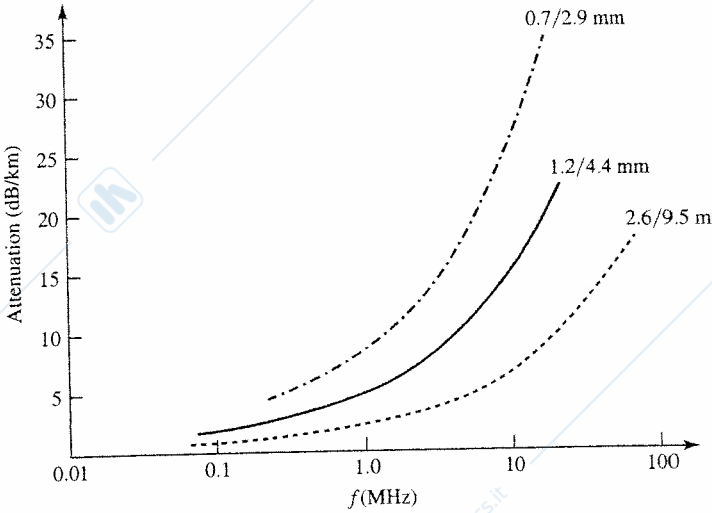


FIGURE 3.50 Attenuation versus frequency for coaxial cable [after Smith 1985].

than twisted pair does. By comparing Figure 3.50 with Figure 3.47, we can see that coaxial cable can provide much higher bandwidths (hundreds of MHz) than twisted pair (a few MHz). For example, existing cable television systems use a bandwidth of 500 MHz.

Coaxial cable was initially deployed in the backbone of analog telephone networks where a single cable could be used to carry in excess of 10,000 simultaneous analog voice circuits. Digital transmission systems using coaxial cable were also deployed in the telephone network in the 1970s. These systems operate in the range of 8.448 Mbps to 564.992 Mbps. However, the deployment of coaxial cable transmission systems in the backbone of the telephone network was discontinued because of the much higher bandwidth and lower cost of optical fiber transmission systems.

APPLICATION **Cable Television Distribution**

The widest use of coaxial cable is for distribution of television signals in cable TV systems. Existing coaxial cable systems use the frequency range from 54 MHz to 500 MHz. A National Television Standards Committee (NTSC) analog television signal occupies a 6 MHz band, and a phase alternation by line (PAL) analog television signal occupies 8 MHz, so 50 to 70 channels can be accommodated.[19] Existing cable television systems are arranged in a tree-and-branch topology as shown in Figure 3.51. The master television signal originates at a head end office, and unidirectional analog amplifiers maintain the signal level. The signal is split along different branches until all subscribers are reached. Because all the information flows from the head end to the subscribers, cable television systems were designed to be unidirectional.

[19]The NTSC and PAL formats are two standards for analog television. The NTSC format is used in North America and Japan, and the PAL format is used in Europe.

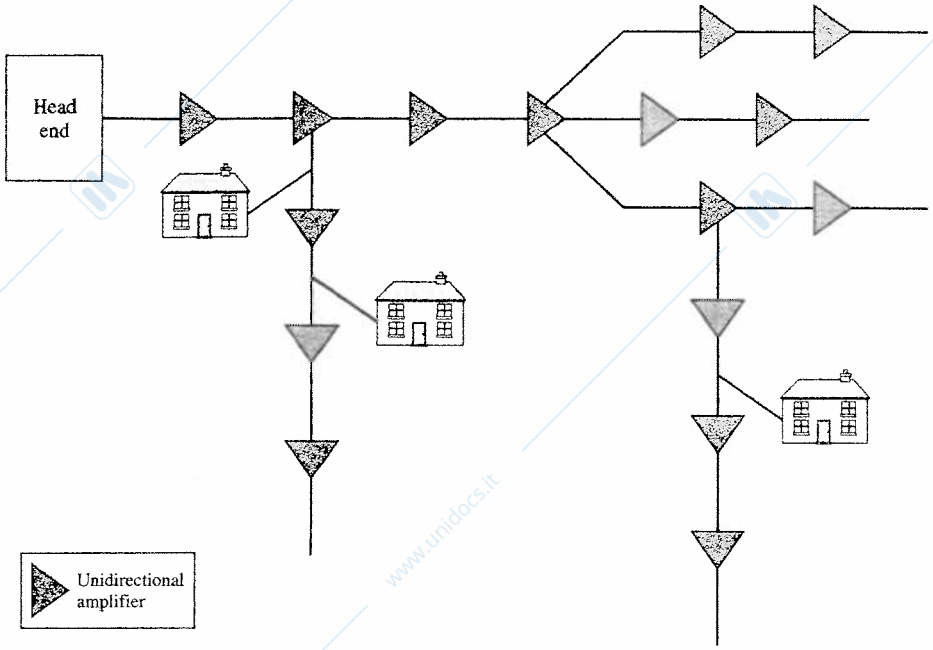


FIGURE 3.51 Tree-and-branch topology of conventional cable TV systems.

APPLICATION **Cable Modem**

The coaxial cable network has a huge bandwidth flowing from the network to the user. For example, a single analog television channel will provide approximately 6 MHz of bandwidth. If QAM modulation is used with a 64-point constellation, then a bit rate of 6 Mpulses/second × 6 bits/pulse = 36 Mbps is possible. However, the coaxial network was not designed to provide communications from the user to the network. Figure 3.52 shows how coaxial cable networks are being modified to provide upstream communications through the introduction of bidirectional split-band amplifiers that allow information to flow in both directions.

Figure 3.53a shows the existing cable spectrum that uses the band from 54 MHz to 500 MHz for the distribution of analog television signals. Figure 3.53b shows the proposed spectrum for hybrid fiber-coaxial systems. The band from 550 MHz to 750 MHz would be used to carry new digital video and data signals as well as downstream telephone signals. In North America channels are 6 MHz wide, so these downstream channels can support bit rates in the range of 36 Mbps. The band from 5 MHz to 42 MHz, which was originally intended for pay-per-view signaling, would be converted for **cable modem** upstream signals as well as for *cable telephony*. This lower band is subject to much worse interference and noise than the downstream channels. Using channels of

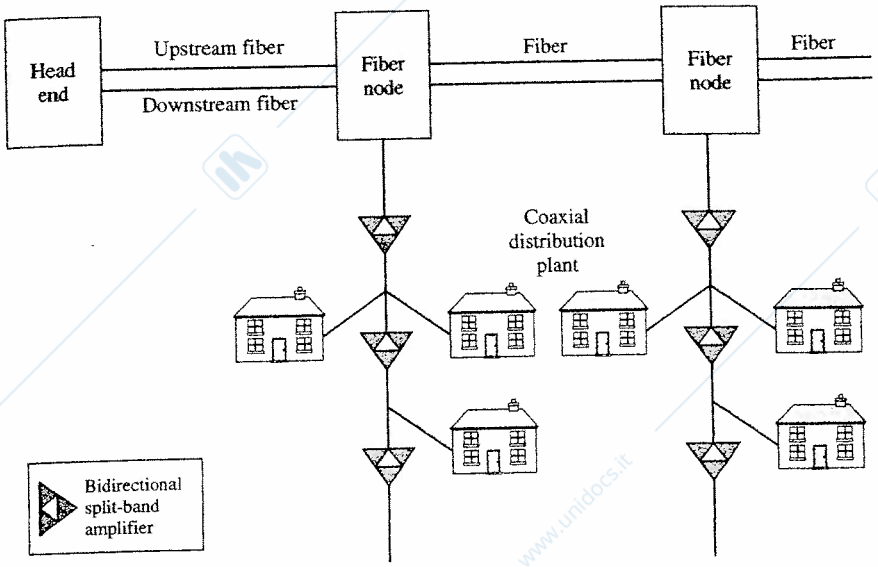


FIGURE 3.52 Topology of hybrid fiber-coaxial systems.

approximately 2 MHz, upstream transmission rates from 500 kbps to 4 Mbps can be provided. As in the case of ADSL, we see that the upstream/downstream transmission rates are asymmetric.

Both the upstream and downstream channels need to be *shared* among subscribers in the feeder line. The arrangement is similar to that of a local area network in that

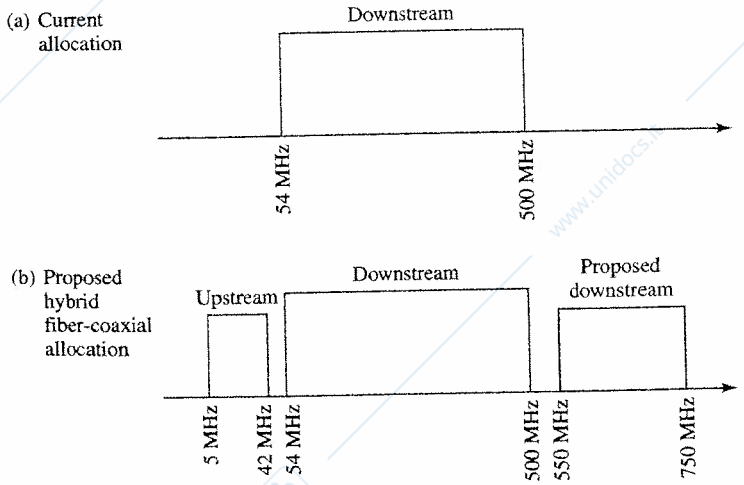


FIGURE 3.53 Frequency allocation in cable TV systems.

the cable modems from the various users must communicate with a *cable modem termination system (CMTS)* at the operator's end of the cable. The cable modems must listen for packets destined to them on an assigned downstream channel. They must also contend to obtain time slots to transmit their information in an assigned channel in the upstream direction.

APPLICATION **Ethernet LAN**

The original design of the Ethernet LAN used coaxial cable for the shared medium (see Figure 1.13a). Coaxial cable was selected because it provided high bandwidth, offered good noise immunity, and led to a cost-effective transceiver design. The original standard specified 10Base5, which uses thick (10 mm) coaxial cable operating at a bit rate of *10* Mbps, using *base*band transmission and with a maximum segment length of 500 meters. The transmission uses Manchester coding. This cabling system required the use of a *transceiver* to attach the NIC card to the coaxial cable. The thick coaxial cable Ethernet was typically deployed along the ceilings in building hallways, and a connection from a workstation in an office would tap onto the cable. Thick coaxial cable is awkward to handle and install. The 10Base2 standard uses thin (5 mm) coaxial cable operating 10 Mbps and with a maximum segment of 185 meters. The cheaper and easier to handle thin coaxial cable makes use of T-shaped BNC connectors. 10Base5 and 10Base2 segments can be combined through the use of a *repeater* that forwards the signals from one segment to the other.

3.8.3 Optical Fiber

The deployment of digital transmission systems using twisted pair and coaxial cable systems established the trend toward digitization of the telephone network during the 1960s and 1970s. These new digital systems provided significant economic advantages over previous analog systems. Optical fiber transmission systems, which were introduced in the 1970s, offered even greater advantages over copper-based digital transmission systems and resulted in a dramatic acceleration of the pace toward digitization of the network. Figure 1.1 of Chapter 1 showed that optical fiber systems represented an acceleration in the long-term rate of improvement in transmission capacity.

The typical T-1 or coaxial transmission system requires repeaters about every 2 km. Optical fiber systems, on the other hand, have maximum regenerator spacings in the order of tens to hundreds and even thousands of kilometers. The introduction of optical fiber systems has therefore resulted in great reductions in the cost of digital transmission. Optical fiber systems have also allowed dramatic reductions in the space required to house the cables. A single fiber strand is much thinner than twisted pair or coaxial cable. Because a single optical fiber can carry much higher transmission rates than copper systems, a single cable of optical fibers can replace many cables of copper wires. In addition, optical fibers do not radiate significant energy and do not pick up

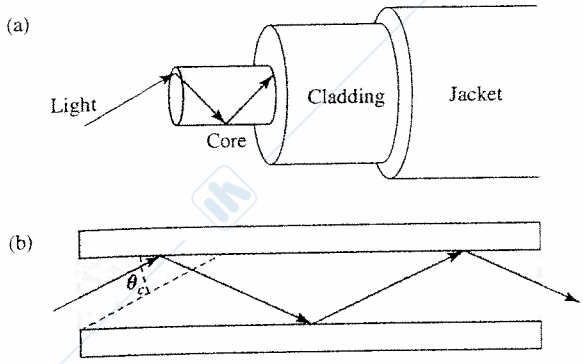


FIGURE 3.54 Transmission of light waves in optical fiber: (a) geometry of optical fiber; (b) reflection in optical fiber.

interference from external sources. Thus compared to electrical transmission, optical fibers are more secure from tapping and are also immune to interference and crosstalk.

Optical fiber consists of a very fine cylinder of glass (core) surrounded by a concentric layer of glass (cladding) as shown in Figure 3.54. The information itself is transmitted through the core in the form of a fluctuating beam of light. The core has a slightly higher optical density (index of refraction) than the cladding. The ratio of the indices of refraction of the two glasses defines a critical angle θ_c. When a ray of light from the core approaches the cladding at an angle less than θ_c, the ray is completely reflected back into the core. In this manner the ray of light is guided within the fiber.

The attenuation in the fiber can be kept low by controlling the impurities that are present in the glass. When it was invented in 1970, optical fiber had a loss of 20 dB/km. Within 10 years systems with a loss of 0.2 dB/km had become available. Figure 3.55 shows that minimum attenuation of optical fiber varies with the wavelength of the

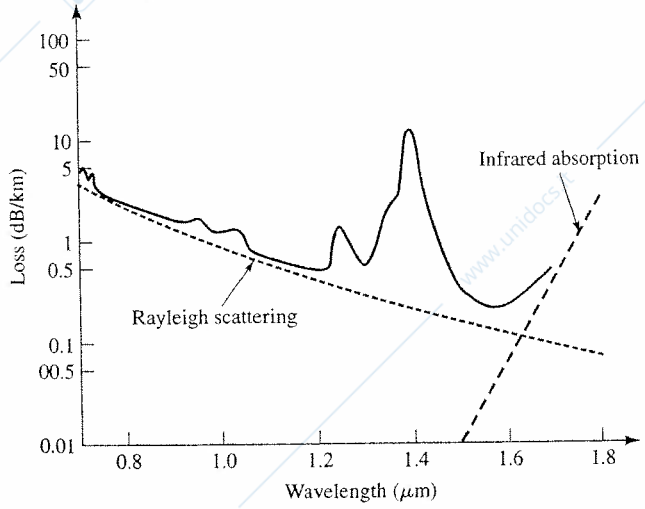


FIGURE 3.55 Attenuation versus wavelength for optical fiber.

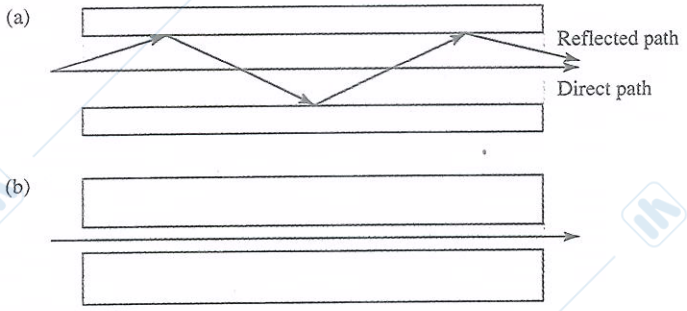


FIGURE 3.56 (a) Multimode optical fiber: multiple rays follow different paths; (b) single-mode optical fiber: only direct path propagates in optical fiber.

signal. It can be seen that the systems that operate at wavelengths of 1300 nanometer (nm) and 1550 nm occupy regions of low attenuation. The attenuation peak in the vicinity of 1400 nm is due to residual water vapor in the glass fiber.[20] Early optical fiber transmission systems operated in the 850 nm region at bit rates in the tens of megabits/second and used relatively inexpensive light emitting diodes (LEDs) as the light source. Second- and third-generation systems use laser sources and operate in the 1300 nm and 1500 nm region achieving gigabits/second bit rates.

A **multimode fiber** has an input ray of light reach the receiver over multiple paths, as shown in Figure 3.56a. Here the first ray arrives in a direct path, and the second ray arrives through a reflected path. The difference in delay between the two paths causes the rays to interfere with each other. The amount of interference depends on the duration of a pulse relative to the delays of the paths. The presence of multiple paths limits the maximum bit rates that are achievable using multimode fiber. By making the core of the fiber much narrower, it is possible to restrict propagation to the single direct path. These **single-mode fibers** can achieve speeds of many gigabits/second over hundreds of kilometers.

Figure 3.57 shows an optical fiber transmission system. The transmitter consists of a light source that can be modulated according to an electrical input signal to produce

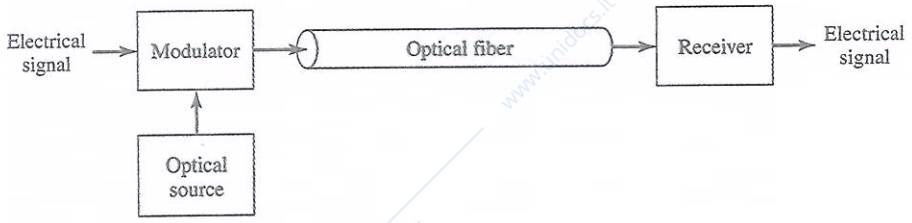


FIGURE 3.57 Optical transmission system.

[20]New optical fiber designs remove the water peak in the 1400 nm region.

a beam of light that is inserted into the fiber. Typically the binary information sequence is mapped into a sequence of on/off light pulses at some particular wavelength. The key attribute of optical communications is that these light pulses can be switched on and off at extremely high rates so extremely high bit rates are possible. An optical detector at the receiver end of the system converts the received optical signal into an electrical signal from which the original information can be detected. Optical fiber communication involves various types of dispersion that are not encountered in electronic systems. See [Keiser 2000] for a discussion on dispersion types.

The width of an optical band is typically stated in terms of nanometers. We obtain an expression for bandwidth in terms of frequency as follows. Let f_1 correspond to the wavelength λ_1 and let f_2 correspond to the wavelength $\lambda_1 + \Delta\lambda$, and suppose that $\Delta\lambda$ is much smaller than λ_1. From Equation (3.38) the bandwidth in Hz is given by

$$B = f_1 - f_2 = \frac{v}{\lambda_1} - \frac{v}{\lambda_1 + \Delta\lambda} = \frac{v}{\lambda_1}\left(1 - \frac{1}{1 + \frac{\Delta\lambda}{\lambda_1}}\right) = \frac{v}{\lambda_1}\left(\frac{\frac{\Delta\lambda}{\lambda_1}}{1 + \frac{\Delta\lambda}{\lambda_1}}\right) \approx \frac{v\Delta\lambda}{\lambda_1^2} \tag{3.41}$$

The region around 1300 nm contains a band with attenuation less than 0.5 dB/km. The region has $\Delta\lambda$ of approximately 100 nm which gives a bandwidth of approximately 12 terahertz. One terahertz is 10^{12} Hz, that is, 1 million MHz! The region around 1550 nm has another band with attenuation as low as 0.2 dB/km [Mukherjee 1997]. This region has a bandwidth of about 15 THz. Clearly, existing optical transmission systems do not come close to utilizing this bandwidth.

Wavelength-division multiplexing (WDM) is an effective approach to exploiting the bandwidth that is available in optical fiber. In WDM multiple wavelengths are used to carry simultaneously several information streams over the same fiber. WDM is a form of multiplexing and is covered in Chapter 4. Early WDM systems handled 16 wavelengths each transmitting 2.5 Gbps for a total of 40 Gbps/fiber. Two basic types of WDM systems are in use. *Coarse WDM (CWDM)* systems are optimized for simplicity and low cost and involve the use of a few wavelengths (4–8) with wide interwavelength spacing. *Dense WDM (DWDM)* systems, on the other hand, maximize the bandwidth carried in a fiber through dense packing of wavelengths. Current DWDM systems can pack 80 to 160 wavelengths, where each wavelength can carry 10 Gbps and in some cases 40 Gbps. The ITU Grid specifies a separation of 0.8 nm between wavelengths and is used for systems that carry 10 Gbps signals per wavelength. Newer systems use a tighter spacing of 0.4 nm and typically carry 2.5 Gbps signals.

The attenuation in the optical fiber typically limits the range of the optical transmitted signal to tens of kilometers. In the absence of optical amplifiers, electronic regenerators must be inserted between spans of optical fiber because current optical processing cannot provide all-optical timing recovery and signal detection. At each regenerator, the optical signal is converted to an electrical signal, timing is recovered electronically, the original data is detected electronically, and the resulting recovered data sequence is used to drive a laser that pumps a regenerated optical signal along the next span as shown in Figure 3.58a. Optical-to-electronic conversion is expensive

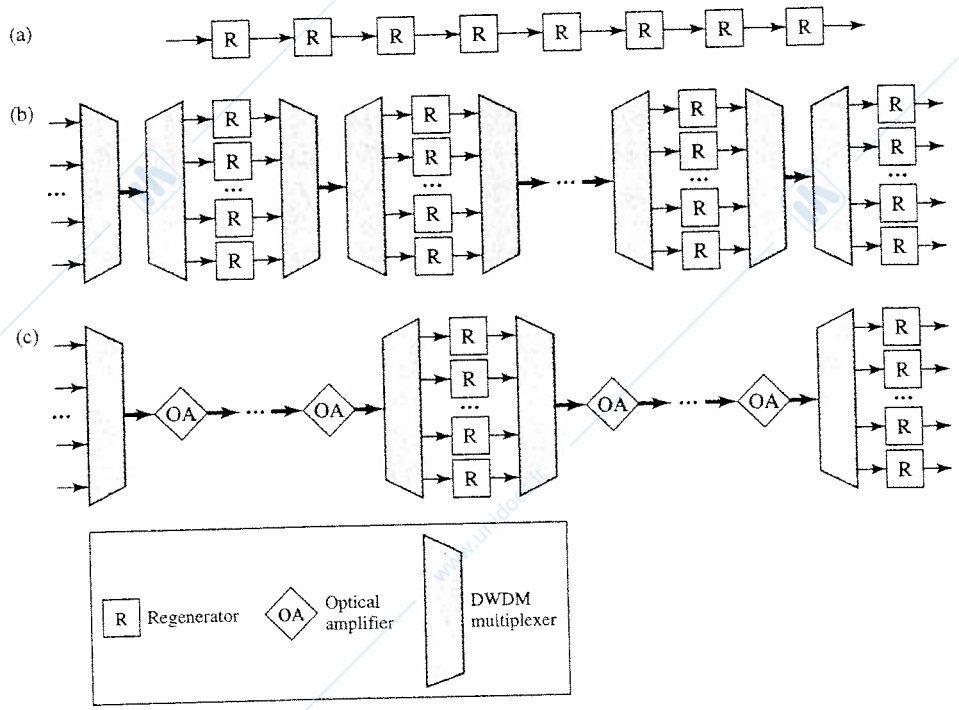


FIGURE 3.58 Optical transmission systems: (a) single signal/fiber with 1 regenerator/span; (b) DWDM composite signal/fiber with 1 regenerator/wavelength-span; (c) DWDM composite signal with optical amplifiers.

because of the cost of lasers and high-speed electronics. The cost of regeneration becomes acute when DWDM is introduced. As shown in Figure 3.58b, at each regeneration point the composite DWDM signal must be split into its separate optical signals and each individual signal must be electronically regenerated individually.

A major advance took place with the invention of **optical amplifiers** that can amplify the band of wavelengths that is occupied by a DWDM signal. An Erbium-Doped Fiber Amplifier (EDFA) is a device that can boost the intensity of optical signals that are carried within the 1530 to 1620 nm band in an optical fiber. An EDFA involves taking a weak arriving (DWDM) optical signal and combining it with a locally generated higher power optical signal in a length of fiber that has been doped with the erbium. The erbium atoms are excited by this action and they generate photons at the same phase and direction as the arriving signal. In effect the DWDM signal is amplified. The EDFA devices must be designed so that they provide nearly equal amplification for all the wavelengths in its band. The availability of EDFA optical amplifiers reduces the need for regenerators as shown in Figure 3.58c. There is a limit on how many EDFA devices can be cascaded in series and so eventually regenerators need to be inserted. Nevertheless, the distance between regenerators and the associated optical-to-electronic conversion can be increased to hundreds and even thousands of kilometers.

In Chapter 4 we examine the impact of these new optical technologies on transport networks.

APPLICATION **Access and Backbone Networks**

Optical fiber transmission systems are widely deployed in the backbone of networks. In Chapter 4 we present the digital multiplexing hierarchy that has been developed for electrical and optical digital transmission systems. Current optical fiber transmission systems provide transmission rates from 45 Mbps to 10 Gbps using single wavelength transmission and 40 Gbps to 1600 Gbps using WDM. Optical fiber systems are very cost-effective in the backbone of networks because the cost is spread over a large number of users. Optical fiber transmissions systems provide the facilities for long-distance telephone communications and data communications. Regeneratorless optical fiber transmission systems are also used to interconnect telephone offices in metropolitan areas.

The cost of installing optical fiber in the subscriber portion of the network remains higher than the cost of using the existing installed base of twisted pair and coaxial cable. Fiber-to-the-home proposals that would provide huge bandwidths to the user remain too expensive. Fiber-to-the-curb proposals attempt to reduce this cost by installing fiber to a point that is sufficiently close to the subscriber. Twisted pair or coaxial cable can then connect the subscriber to the curb at high data rates.

APPLICATION **Local Area Networks**

Optical fiber is used as the physical layer of several LAN standards. The 10BASE-FP Ethernet physical layer standard uses optical fiber operating with an 850 nm source. The transmission system uses Manchester coding and intensity-light modulation and allows distances up to 2 km. The Fiber Distributed Data Interface (FDDI) ring-topology LAN uses optical fiber transmission at a speed of 100 Mbps, using LED light sources at 1300 nm with repeater spacings of up to 2 km. The binary information is encoded using a 4B5B code followed by NRZ-inverted line coding. The 100BASE-FX Fast Ethernet physical layer standard uses two fibers, one for send and one for receive. The maximum distance is limited to 100 meters. The transmission format is the same as that of FDDI with slight modifications.

Optical fiber is the preferred medium for Gigabit Ethernet. As has become the practice in the development of physical layer standards, the 1000BASE-X standards are based on the preexisting fiber channel standard. The pulse transmission rate is 1.25 gigapulses/second, and an 8B10B code is used to provide the 1 Gbps transmission rate. There are two variations of the 1000BASE-X standard. The 1000BASE-SX uses a "shortwave" light source, nominally 850 nm, and multimode fiber. The distance limit is 550 meters. The 1000BASE-LX uses a "longwave" light source, nominally at 1300 nm, single mode or multimode fiber. For multimode fiber the distance limit is 550 meters, and for single-mode fiber the distance is 5 km.

APPLICATION **10 Gigabit Ethernet Networks**

A 10 Gbps Ethernet standard has been specified for use in LANs as well as in wide area networks. To support the installed base of Gigabit Ethernet, two multimode fiber interfaces have been defined. An 850 nm interface supports LAN deployments of up to 65 meters. A 1310 nm interface supports applications up to 300 meters. For wide area networks, single-mode fiber interfaces at 1310 nm and 1550 nm have been defined that provide reaches of 10 km and 40 km respectively.

3.8.4 Radio Transmission

Radio encompasses the electromagnetic spectrum in the range of 3 kHz to 300 GHz. In radio communications the signal is transmitted into the air or space, using an antenna that radiates energy at some carrier frequency. For example, in QAM modulation the information sequence determines a point in the signal constellation that specifies the amplitude and phase of the sinusoidal wave that is transmitted. Depending on the frequency and the antenna, this energy can propagate in either a unidirectional or omnidirectional fashion. In the unidirectional case a properly aligned antenna receives the modulated signal, and an associated receiver in the direction of the transmission recovers the original information. In the omnidirectional case any receiver with an antenna in the area of coverage can pick up the signal.

Radio communication systems are subject to a variety of transmission impairments. We indicated earlier that the attenuation in radio links varies logarithmically with the distance. Attenuation for radio systems also increases with rainfall. Radio systems are subject to multipath fading and interference. Multipath fading refers to the interference that results at a receiver when two or more versions of the same signal arrive at slightly different times. If the arriving signals differ in polarity, then they will cancel each other. Multipath fading can result in wide fluctuations in the amplitude and phase of the received signal. Interference refers to energy that appears at the receiver from sources other than the transmitter. Interference can be generated by other users of the same frequency band or by equipment that inadvertently transmits energy outside its band and into the bands of adjacent channels. Interference can seriously affect the performance of radio systems, and for this reason regulatory bodies apply strict requirements on the emission properties of electronic equipment.

Figure 3.59 gives the range of various frequency bands and their applications. The frequency bands are classified according to wavelengths. Thus the low frequency (LF) band spans the range 30 kHz to 300 kHz, which corresponds to a wavelength of 1 km to 10 km, whereas the extremely high frequency (EHF) band occupies the range from 30 to 300 GHz corresponding to wavelengths of 1 millimeter to 1 centimeter. Note that the progression of frequency bands in the logarithmic frequency scale have increasingly larger bandwidths, for example, the "band" from 10^{11} to 10^{12} Hz has a bandwidth of 0.9×10^{12} Hz, whereas the band from 10^5 to 10^6 Hz has a bandwidth of 0.9×10^6 Hz.

The propagation properties of radio waves vary with the frequency. Radio waves at the VLF, LF, and MF bands follow the surface of the earth in the form of ground

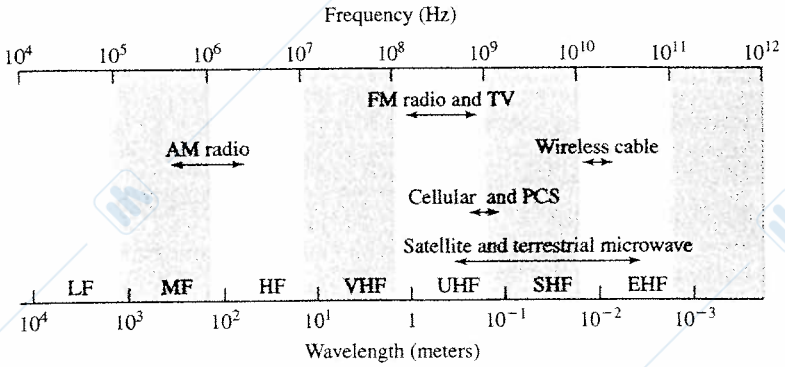


FIGURE 3.59 Radio spectra.

waves. VLF waves can be detected at distances up to about 1000 km, and MF waves, for example, AM radio, at much shorter distances. Radio waves in the HF band are reflected by the ionosphere and can be used for long-distance communications. These waves are detectable only within certain specific distances from the transmitter. Finally, radio waves in the VHF band and higher are not reflected back by the ionosphere and are detectable only within line-of-sight.

In general, radio frequencies below 1 GHz are more suitable for omnidirectional applications, such as those shown in Table 3.6. For example, paging systems ("beepers") are an omnidirectional application that provides one-way communications. A high-power transmission system is used to reach simple, low-power pocket-size receivers in some geographic area. The purpose of the system is to alert the owner that someone wishes to communicate with him or her. The system may consist of a single high-powered antenna, or a network of interconnected antennas, or be a nationwide satellite-based transmission system. These systems deliver the calling party's telephone number and short text messages. Paging systems have operated in a number of frequency bands. Most systems currently use the 930 to 932 MHz band.

Cordless telephones are an example of an omnidirectional application that provides two-way communications. Here a simple base station connects to a telephone outlet and relays signaling and voice information to a cordless phone. This technology allows the user to move around in an area of a few tens of meters while talking on the phone. The first generation of cordless phones used analog radio technology and subsequent generations have used digital technology.

TABLE 3.6 Examples of omnidirectional systems.

System	Description	Distance
Paging	Short message	10s of kilometers
Cordless telephone	Analog/digital voice	10s of meters
Cellular telephone	Analog/digital voice and data	kilometers
Personal communication services	Digital voice and data	100s of meters
Wireless LAN	High-speed data	100 meters

APPLICATION **Cellular Communications**

Analog cellular telephone systems were introduced in 1979 in Japan. This system provided for 600 two-way channels in the 800 MHz band. In Europe the Nordic Mobile Telephone system was developed in 1981 in the 450 MHz band. The U.S. Advanced Mobile Phone System (AMPS) was deployed in 1983 in a frequency band of 50 MHz in the 800 MHz region. This band is divided into 30 kHz channels that can each carry a single FM-modulated analog voice signal.

Analog cellular phones quickly reached their capacity in large metropolitan areas because of the popularity of cellular phone service. Several digital cellular telephone systems based on digital transmission have been introduced. In Europe the Global System for Mobile (GSM) standard was developed to provide for a pan-European digital cellular system in the 900 MHz band. In 1991 Interim Standard IS-54 in the United States allowed for the replacement of a 30 kHz channel with a digital channel that can support three users. This digital channel uses differential QAM modulation in place of the analog FM modulation. A cellular standard based on code division multiple access (CDMA) was also standardized as IS-95. This system, based on direct sequence spread spectrum transmission, can handle more users than earlier systems could. These cellular systems are discussed further in Chapter 6.

In 1995 personal communication services (PCS) licenses were auctioned in the U.S. for spectrum in the 1800/1900 MHz region. PCS is intended to extend digital cellular technology to a broader community of users by using low-power transmitters that cover small areas, "microcells." PCS thus combines aspects of conventional cellular telephone service with aspects of cordless telephones. The first large deployment of PCS is in the Japanese Personal Handiphone system that operates in the 1800/1900 band. This system is now very popular. In Europe the GSM standard has been adapted to the 1800/1900 band.

APPLICATION **Wireless LANs**

Wireless LANs are another application of omnidirectional wireless communications. The objective here is to provide high-speed communications among a number of computers located in relatively close proximity. Most standardization efforts in the United States have focused on the Industrial/Scientific/Medical (ISM) bands, which span 902 to 928 MHz, 2400 to 2483.5 MHz, and 5725 to 5850 MHz, respectively. Unlike other frequency bands, the ISM band is designated for unlicensed operation so each user must cope with the interference from other users. In Europe, the high-performance radio LAN (HIPERPLAN) standard was developed to provide high-speed (20 Mbps) operation in the 5.15 to 5.30 GHz band. In 1996 the Federal Communications Commission (FCC) in the United States announced its intention to make 350 MHz of spectrum in the 5.15 to 5.35 GHz and 5.725 to 5.825 GHz bands available for unlicensed use in LAN applications. More recently, the IEEE 802.11 group has developed standards for wireless LANs. These developments are significant because these systems will provide

high-speed communications to the increasing base of portable computers. This new spectrum allocation will also enable the development of ad hoc digital radio networks in residential and other environments.

APPLICATION **Point-to-Point and Point-to-Multipoint Radio Systems**

Highly directional antennas can be built for microwave frequencies that cover the range from 2 to 40 GHz. For this reason point-to-point wireless systems use microwave frequencies and were a major component of the telecommunication infrastructure introduced several years ago. Digital microwave transmission systems have been deployed to provide long-distance communications. These systems typically use QAM modulation with fairly large signal constellations and can provide transmission rates in excess of 100 Mbps. The logarithmic, rather than linear, attenuation gave microwave radio systems an advantage over coaxial cable systems by requiring regenerator spacings in the tens of kilometers. In addition, microwave systems did not have to deal with right-of-way issues. Microwave transmission systems can also be used to provide inexpensive digital links between buildings.

Microwave frequencies in the 28 GHz band have also been licensed for point-to-multipoint "wireless cable" systems. In these systems microwave radio beams from a telephone central office would send 50 Mbps directional signals to subscribers within a 5 km range. Reflectors would be used to direct these beams so that all subscribers can be reached. These signals could contain digital video and telephone as well as high-speed data. Subscribers would also be provided with transmitters that would allow them to send information upstream into the network. The providers of this service have about 1 GHz in total bandwidth available.

APPLICATION **Satellite Communications**

Early satellite communications systems can be viewed as microwave systems with a single repeater in the sky. A (geostationary) satellite is placed at an altitude of about 36,000 km above the equator where its orbit is stationary relative to the rotation of the earth. A modulated microwave radio signal is beamed to the satellite on an uplink carrier frequency. A transponder in the satellite receives the uplink signal, regenerates it, and beams it down back to earth on a downlink carrier frequency. A satellite typically contains 12 to 20 transponders so it can handle a number of simultaneous transmissions. Each transponder typically handles about 50 Mbps. Satellites operate in the 4/6, 11/14, and 20/30 GHz bands, where the first number indicates the downlink frequency and the second number the uplink frequency.

Geostationary satellite systems have been used to provide point-to-point digital communications to carry telephone traffic between two points. Satellite systems have an advantage over fiber systems in situations where communications needs to be established quickly or where deploying the infrastructure is too costly. Satellite systems are

inherently broadcast in nature, so they are also used to simultaneously beam television, and other signals, to a large number of users. Satellite systems are also used to reach mobile users who roam wide geographical areas.

Constellations of low-earth orbit satellites (LEOS) have also been deployed. These include the Iridium and Teledesic systems. The satellites are not stationary with respect to the earth, but they rotate in such a way that there is continuous coverage of the earth. The component satellites are interconnected by high-speed links forming a network in the sky.

3.8.5 Infrared Light

Infrared light is a communication medium whose properties differ significantly from radio frequencies. Infrared light does not penetrate walls, so an inherent property is that it is easily contained within a room. This factor can be desirable from the point of view of reducing interference and enabling reuse of the frequency band in different rooms. Infrared communications systems operate in the region from 850 nm to 900 nm where receivers with good sensitivity are available. Infrared light systems have a very large potential bandwidth that is not yet exploited by existing systems. A serious problem is that the sun generates radiation in the infrared band, which can be a cause of severe interference. The infrared band is being investigated for use in the development of very high speed wireless LANs.

APPLICATION **IrDA Links**

The Infrared Data Association (IrDA) was formed to promote the development of infrared light communication systems. A number of standards have been developed under its auspices. The IrDA-C standard provides bidirectional communications for cordless devices such as keyboards, mice, joysticks, and handheld computers. This standard operates at a bit rate of 75 kbps at distances of up to 8 meters. The IrDA-D standard provides for data rates from 115 kb/s to 4 Mb/s over a distance of 1 meter. It was designed as a wireless alternative to connecting devices, such as a laptop to a printer.

3.9 ERROR DETECTION AND CORRECTION

In most communication channels a certain level of noise and interference is unavoidable. Even after the design of the digital transmission system has been optimized, bit errors in transmission will occur with some small but nonzero probability. For example, typical bit error rates for systems that use copper wires are in the order of 10^{-6}, that is, one in a million. Modern optical fiber systems have bit error rates of 10^{-9} or less. In contrast, wireless transmission systems can experience error rates as high as 10^{-3} or worse. The acceptability of a given level of bit error rate depends on the particular application. For example, certain types of digital speech transmission are tolerant to fairly high bit error rates. Other types of applications such as electronic funds transfer

require essentially error-free transmission. In this section we introduce **error-control** techniques for improving the error-rate performance that is delivered to an application in situations where the inherent error rate of a digital transmission system is unacceptable.

There are two basic approaches to error control. The first approach involves the detection of errors and an *automatic retransmission request (ARQ)* when errors are detected. This approach presupposes the availability of a return channel over which the retransmission request can be made. For example, ARQ is widely used in computer communication systems that use telephone lines. ARQ is also used to provide reliable data transmission over the Internet. The second approach, **forward error correction (FEC)**, involves the detection of errors followed by further processing of the received information that attempts to correct the errors. FEC is appropriate when a return channel is not available, retransmission requests are not easily accommodated, or a large amount of data is sent and retransmission to correct a few errors is very inefficient. For example, FEC is used in satellite and deep-space communications. Another application is in audio CD recordings where FEC is used to provide tremendous robustness to errors so that clear sound reproduction is possible even in the presence of smudges and scratches on the disk surface. Error detection is the first step in both ARQ and FEC. The difference between ARQ and FEC is that ARQ "wastes" bandwidth by using retransmissions, whereas FEC in general requires additional redundancy in the transmitted information and incurs significant processing complexity in performing the error correction.

In this section we discuss parity check codes, the Internet checksum, and polynomial codes that are used in error detection. We also present methods for assessing the effectiveness of these codes in several error environments. These results are used in Chapter 5, in the discussion of ARQ protocols. An optional section on linear codes gives a more complete introduction to error detection and correction.

3.9.1 Error Detection

First we discuss the idea of error detection in general terms, using the single parity check code as an example throughout the discussion. The basic idea in performing **error detection** is very simple. As illustrated in Figure 3.60, the information produced by an application is encoded so that the stream that is input into the communication channel satisfies a specific *pattern* or condition. The receiver checks the stream coming out of the communication channel to see whether the pattern is satisfied. If it is not, the receiver can be certain that an error has occurred and therefore sets an alarm to alert the user. This certainty stems from the fact that no such pattern would have been transmitted by the encoder.

The simplest code is the **single parity check code** that takes k information bits and appends a single **check bit** to form a **codeword**. The parity check ensures that the total number of 1s in the codeword is even; that is, the codeword has even parity.[21] The check bit in this case is called a *parity bit*. This error-detection code is used in ASCII where characters are represented by seven bits and the eighth bit consists of a parity bit.

[21] Some systems use odd parity by defining the check bit to be the binary complement of Equation (3.42).

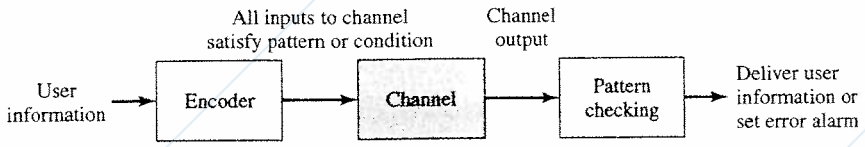


FIGURE 3.60 General error-detection system.

This code is an example of the so-called linear codes because the parity bit is calculated as the modulo 2 sum of the information bits:

$$b_{k+1} = b_1 + b_2 + \cdots + b_k \quad \text{modulo 2} \tag{3.42}$$

where $b_1, b_2, \ldots, b_k$ are the information bits.

Recall that in modulo 2 arithmetic $0+0=0$, $0+1=1$, $1+0=1$, and $1+1=0$. Thus, if the information bits contain an even number of 1s, then the parity bit will be 0; and if they contain an odd number, then the parity bit will be 1. Consequently, the above rule will assign the parity bit a value that will produce a *codeword* that *always contains an even number of 1s.* This pattern defines the single parity check code.

If a codeword undergoes a single error during transmission, then the corresponding binary block at the output of the channel will contain an odd number of 1s and the error will be detected. More generally, if the codeword undergoes an odd number of errors, the corresponding output block will also contain an odd number of 1s. Therefore, the single parity bit allows us to detect all error patterns that introduce an odd number of errors. On the other hand, the single parity bit will fail to detect any error patterns that introduce an even number of errors, since the resulting binary vector will have even parity. Nonetheless, the single parity bit provides a remarkable amount of error-detection capability, since the addition of a single check bit results in making half of all possible error patterns detectable, regardless of the value of k.

Figure 3.61 shows an alternative way of looking at the operation of this example. At the transmitter a checksum is calculated from the information bits and transmitted along with the information. At the receiver, the checksum is recalculated, based on the received information. The received and recalculated checksums are compared, and the error alarm is set if they disagree.

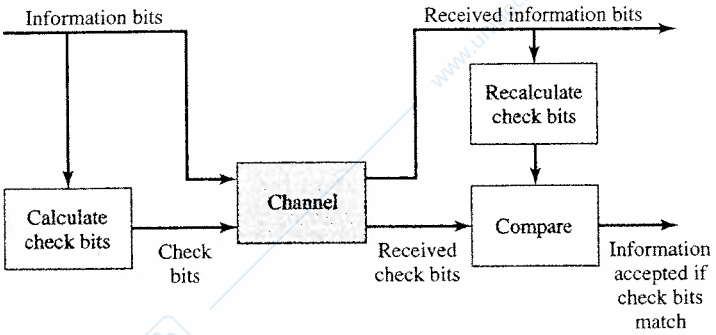


FIGURE 3.61 Error-detection system using check bits.

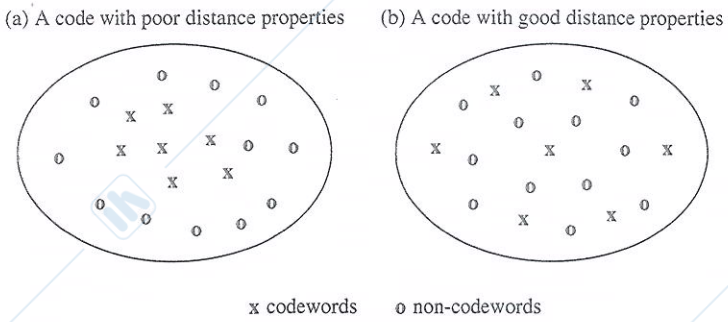


FIGURE 3.62 Distance properties of codes.

This simple example can be used to present two fundamental observations about error detection. The first observation is that error detection requires **redundancy** in that the amount of information that is transmitted is over and above the required minimum. For a single parity check code of length $k + 1$, k bits are information bits, and one bit is the parity bit. Therefore, the fraction $1/(k + 1)$ of the transmitted bits is redundant.

The second fundamental observation is that *every error-detection technique will fail to detect some errors*. In particular, an error-detection technique will always fail to detect transmission errors that convert a valid codeword into another valid codeword. For the single parity check code, an even number of transmission errors will always convert a valid codeword to another valid codeword.

The objective in selecting an error-detection code is to select the codewords that reduce the likelihood of the transmission channel converting one valid codeword into another. To visualize how this is done, suppose we depict the set of all possible binary blocks as the space shown in Figure 3.62, with codewords shown by xs in the space and noncodewords by os. To minimize the probability of error-detection failure, we want the codewords to be selected so that they are spaced as far away from each other as possible. Thus the code in Figure 3.62a is a poor code because the codewords are close to each other. On the other hand, the code in Figure 3.62b is good because the distance between codewords is maximized. The effectiveness of a code clearly depends on the types of errors that are introduced by the channel. We next consider how the effectiveness is evaluated for the example of the single parity check code.

EFFECTIVENESS OF ERROR-DETECTION CODES

The effectiveness of an error-detection code is measured by the probability that the system fails to detect an error. To calculate this probability of error-detection failure, we need to know the probabilities with which various errors occur. These probabilities depend on the particular properties of the given communication channel. We will consider three models of error channels: the random error vector model, the random bit error model, and burst errors.

Suppose we transmit a codeword that has n bits. Define the error vector $\underline{e} = (e_1, e_2, \ldots, e_n)$ where $e_i = 1$ if an error occurs in the ith transmitted bit and $e_i = 0$ otherwise. In one extreme case, the **random error vector model**, all 2^n possible error vectors are equally likely to occur. In this channel model the probability of $\underline{e}$ does not

depend on the number of errors it contains. Thus the error vector $(1, 0, \ldots, 0)$ has the same probability of occurrence as the error vector $(1, 1, \ldots, 1)$. The single parity check code will fail when the error vector has an even number of 1s. Thus for the random error vector channel model, the probability of error detection failure is $1/2$.

Now consider the **random bit error model** where the bit errors occur independently of each other. Satellite communications provide an example of this type of channel. Let p be the probability of an error in a single-bit transmission. The probability of an error vector that has j errors is $p^j(1-p)^{n-j}$, since each of the j errors occurs with probability p and each of the $n-j$ correct transmissions occurs with probability $1-p$. By rewriting this probability we obtain:

$$p[\underline{e}] = (1-p)^{n-w(\underline{e})}p^{w(\underline{e})} = (1-p)^n\left(\frac{p}{1-p}\right)^{w(\underline{e})} \tag{3.43}$$

where the **weight** $w(\underline{e})$ is defined as the number of 1s in $\underline{e}$. For any useful communication channel, the probability of bit error is much smaller than 1, and so $p < 1/2$ and $p/(1-p) < 1$. This implies that for the random bit error channel the probability of $\underline{e}$ decreases with the weight $w(\underline{e})$, that is, as the number of errors (1s) increases. In other words, an error pattern with a given number of bit errors is more likely than an error pattern with a larger number of bit errors. Therefore this channel tends to map a transmitted codeword into binary blocks that are clustered around the codeword.

The single parity check code will fail if the error pattern has an even number of 1s. Therefore, in the random bit error model:

$$\begin{aligned} P[\text{error detection failure}] &= P[\text{undetectable error pattern}] \\ &= P[\text{error patterns with even number of 1s}] \\ &= \binom{n}{2}p^2(1-p)^{n-2} + \binom{n}{4}p^4(1-p)^{n-4} + \cdots \end{aligned} \tag{3.44}$$

where the number of terms in the sum extends up to the maximum possible even number of errors. In the preceding equation we have used the fact that the number of distinct binary n-tuples with j ones and $n-j$ zeros is given by the binomial coefficient

$$\binom{n}{j} = \frac{n!}{j!(n-j!)} \tag{3.45}$$

In any useful communication system, the probability of a single-bit error p is much smaller than 1. We can then use the following approximation: $p^i(1-p)^j \approx p^i(1-p^j) \approx p^i$. For example, if $p = 10^{-3}$ then $p^2(1-p)^{n-2} \approx 10^{-6}$ and $p^4(1-p)^{n-4} \approx 10^{-12}$. Thus the probability of detection failure is determined by the first term in Equation (3.44). For example, suppose $n = 32$ and $p = 10^{-4}$. Then the probability of error-detection failure is 5×10^{-6}, a reduction of nearly two orders of magnitude.

We see then that a wide gap exists in the performance achieved by the two preceding channel models. Many communication channels combine aspects of these two channels in that errors occur in **bursts**. Periods of low error-rate transmission are interspersed with periods in which clusters of errors occur. The periods of low error rate are similar to the random bit error model, and the periods of error bursts are similar to the random

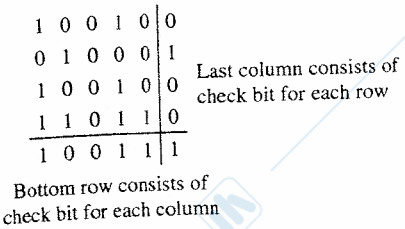


FIGURE 3.63 Two-dimensional parity check code.

error vector model. The probability of error-detection failure for the single parity check code will be between those of the two channel models. In general, measurement studies are required to characterize the statistics of burst occurrence in specific channels.

3.9.2 Two-Dimensional Parity Checks

A simple method to improve the error-detection capability of a single parity check code is to arrange columns that consist of k information bits followed by a check bit at the bottom of each column, as shown in Figure 3.63. The right-most bit in each row is the check bit of the other bits in the row, so in effect the last column is a "check codeword" over the previous m columns. The resulting encoded matrix of bits satisfies the pattern that all rows have even parity and all columns have even parity.

If one, two, or three errors occur anywhere in the matrix of bits during transmission, then at least one row or parity check will fail, as shown in Figure 3.64. However, some patterns with four errors are not detectable, as shown in the figure.

The two-dimensional code was used in early data link controls where each column consisted of seven bits and a parity bit and where an overall check character was added at the end. The two-dimensional parity check code is another example of a linear code. It has the property that error-detecting capabilities can be identified visually, but it does not have particularly good performance. Better codes are discussed in a later (optional) section on linear codes.

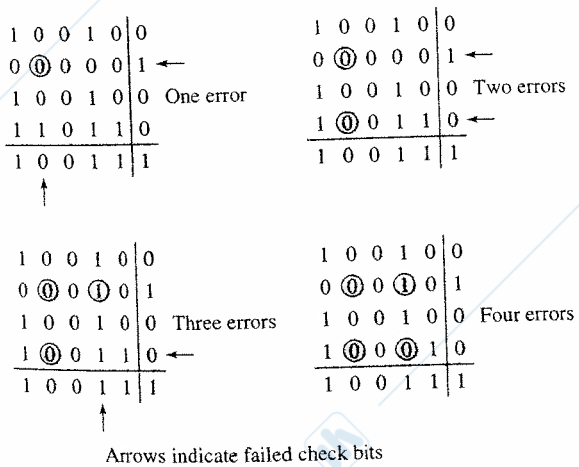


FIGURE 3.64 Detectable and undetectable error patterns for two-dimensional code.

3.9.3 Internet Checksum

Several Internet protocols (e.g., IP, TCP, UDP) use check bits to detect errors. With IP a **checksum** is calculated for the contents of the header and included in a special field. Because the checksum must be recalculated at every router, the algorithm for the checksum was selected for its ease of implementation in software rather than for the strength of its error-detecting capabilities.

The algorithm assumes that the header consists of a certain number, say, L, of 16-bit words, $\mathbf{b}_0, \mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_{L-1}$ plus a checksum $\mathbf{b}_L$. These L words correspond to the "information" in the terminology introduced in the previous sections. The 16-bit checksum $\mathbf{b}_L$ corresponds to the parity bits and is calculated as follows:

1. Each 16-bit word is treated as an integer, and the L words are added modulo $2^{16} - 1$:

$$\mathbf{x} = \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_{L-1} \text{ modulo } 2^{16} - 1 \tag{3.46}$$

2. The checksum then consists of the negative value of $\mathbf{x}$:

$$\mathbf{b}_L = -\mathbf{x} \tag{3.47}$$

3. The checksum $\mathbf{b}_L$ is then inserted in a dedicated field in the header.

The contents of all headers, including the checksum field, must then satisfy the following pattern:

$$\mathbf{0} = \mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_{L-1} + \mathbf{b}_L \text{ modulo } 2^{16} - 1 \tag{3.48}$$

Each router can then check for errors in the header by calculating the preceding equation for each received header.

As an example, suppose we use four-bit words, so we do the checksum using modulo $2^4 - 1 = 15$ arithmetic. The sum of the words 1100 and 1010 is then $12 + 10 = 22$, which in modulo 15 arithmetic is equal to 7. The additive inverse of 7 in modulo 15 arithmetic is 8. Therefore the checksum for 1100 and 1010 is 1000.

Now consider using normal binary addition to do the above calculation. When we add 1100 and 1010 we obtain 10110. However, 10000 corresponds to 16, which in modulo arithmetic is equal to 1. Therefore each time there is a carry in the most significant bit (in this example the fourth bit) we swing the carry bit back to the least significant bit. Thus in modulo $2^4 - 1 = 15$ arithmetic we obtain $1100 + 1010 = 0111$, which is 7 as expected. The 1s complement of 0111 is 1000, which is 8 as before.

The actual Internet algorithm for calculating the checksum is described in terms of 1s complement arithmetic. In this arithmetic, addition of integers corresponds to modulo $2^{16} - 1$ addition, and the negative of the integer corresponding to the 16-bit word $\mathbf{b}$ is found by taking its 1s complement; that is, every 0 is converted to a 1 and vice versa. This process leads to the peculiar situation where there are two representations for $\mathbf{0}$, $(0, 0, \ldots, 0)$ and $(1, 1, \ldots, 1)$, which in turn results in additional redundancy in the context of error detection. Given these properties of 1s complement arithmetic, step 1 above then corresponds to simply adding the 16-bit integers $\mathbf{b}_0 + \mathbf{b}_1 + \mathbf{b}_2 + \cdots + \mathbf{b}_{L-1}$ using regular 32-bit addition. The modulo $2^{16} - 1$ reduction is done by taking the 16 higher-order bits in the sum, shifting them down by 16 positions, and adding them

```
unsigned short cksum(unsigned short *addr, int count)
    {
        /* Compute Internet checksum for "count" bytes
         * beginning at location "addr".
         */
    register long sum = 0;
    while ( count > 1 ) {
        /* This is the inner loop*/
              sum += *addr++;
              count -=2;
        }

        /* Add left-over byte, if any */
    if ( count > 0 )
        sum += *addr;

        /* Fold 32-bit sum to 16 bits */
    while (sum >>16)
        sum = (sum & 0xffff) + (sum >> 16);

    return ~sum;
}
```

FIGURE 3.65 C language function for computing Internet checksum.

back to the sum. Step 2 produces the negative of the resulting sum by taking the 1s complement. Figure 3.65 shows a C function for calculating the Internet checksum adapted from [RFC 1071].

3.9.4 Polynomial Codes

We now introduce the class of **polynomial codes** that are used extensively in error detection and correction. Polynomial codes are readily implemented using shift-register circuits and therefore are the most widely implemented error-control codes. Polynomial codes involve generating check bits in the form of a **cyclic redundancy check** (**CRC**). For this reason they are also known as CRC codes.

In polynomial codes the information symbols, the codewords, and the error vectors are represented by polynomials with binary coefficients. The k information bits $(i_{k-1}, i_{k-2}, \ldots, i_1, i_0)$ are used to form the **information polynomial** of degree $k-1$:

$$i(x) = i_{k-1}x^{k-1} + i_{k-2}x^{k-2} + \cdots + i_1 x + i_0 \tag{3.49}$$

The encoding process takes $i(x)$ and produces a codeword polynomial $b(x)$ that contains the information bits and additional check bits and that satisfies a certain *pattern*. To detect errors, the receiver checks to see whether the pattern is satisfied. Before we explain this process, we need to review polynomial arithmetic.

The polynomial code uses polynomial arithmetic to calculate the codeword corresponding to the information polynomial. Figure 3.66 gives examples of polynomial addition, multiplication, and division using binary coefficients. Note that with binary

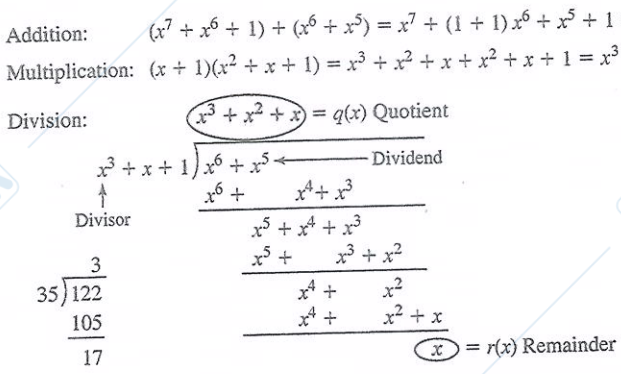


FIGURE 3.66 Polynomial arithmetic.

arithmetic, we have $x^j + x^j = (1 + 1)x^j = 0$. Therefore in the addition example, we have $x^7 + x^6 + 1 + x^6 + x^5 = x^7 + x^6 + x^6 + x^5 + 1 = x^7 + x^5 + 1$. In the multiplication example, we have: $(x+1)(x^2+x+1) = x(x^2+x+1)+1(x^2+x+1) = x^3 + x^2 + x + x^2 + x + 1 = x^3 + 1$.

The division example is a bit more involved and requires reviewing the Euclidean Division algorithm.[22] When we divide a polynomial $p(x)$ by $g(x)$ our goal is to find a quotient $q(x)$ and a remainder $r(x)$ so that $p(x) = q(x)g(x) + r(x)$. If this sounds quite foreign to you, consider the more familiar problem of dividing the integer 122 by 35. As shown in Figure 3.66, the result of longhand division gives a quotient of 3 and a remainder of 17, and sure enough we find that $122 = 3 \times 35 + 17$. Now consider the example in the figure where we divide $x^6 + x^5$ by $x^3 + x + 1$:

1. The first term of the quotient is chosen so that when we multiply the given term by the divisor, $x^3 + x + 1$, the highest power of the resulting polynomial is the same as the highest power of the dividend, $x^6 + x^5$. Clearly the first term of the quotient needs to be x^3.
2. Now multiply the term x^3 by the divisor, which gives $x^3(x^3 + x + 1) = x^6 + x^4 + x^3$, and subtract the result from the dividend. However, in modulo-two arithmetic addition is the same as subtraction, so we *add* $x^6 + x^4 + x^3$ to the dividend $x^6 + x^5$, and obtain the interim remainder polynomial $x^5 + x^4 + x^3$.
3. If the highest power of the interim remainder polynomial is equal or greater than the highest power of the divisor, then the above two steps are repeated using the interim remainder polynomial as the new dividend polynomial, and a new quotient term is computed along with a new interim remainder polynomial. The algorithm stops when the remainder polynomial has lower power than the divisor polynomial.

It is important to note that when the division is completed the remainder $r(x)$ will have a degree smaller than the degree of the divisor polynomial. In the example the

[22] In grade school, the fancy sounding Euclidean Division algorithm was called "longhand division."

Steps:

1. Multiply $i(x)$ by x^{n-k} (puts zeros in $(n-k)$ low-order positions).

2. Divide $x^{n-k}i(x)$ by $g(x)$. Quotient Remainder

$$x^{n-k}i(x) = g(x)\,q(x) + r(x)$$

3. Add remainder $r(x)$ to $x^{n-k}\,i(x)$
(puts check bits in the $n-k$ low-order positions).

$$b(x) = x^{n-k}i(x) + r(x) \leftarrow \text{Transmitted codeword}$$

FIGURE 3.67 Encoding procedure.

divisor polynomial has degree 3, so the division process continues until the remainder term has degree 2 or less.

A polynomial code is specified by its **generator polynomial $g(x)$**. Here we assume that we are dealing with a code in which codewords have n bits, of which k are information bits and $n - k$ are check bits. We refer to this type of code as an (n, k) code. The generator polynomial for such a code has degree $n - k$ and has the form

$$g(x) = x^{n-k} + g_{n-k-1}x^{n-k-1} + \cdots + g_1x + 1 \tag{3.50}$$

where $g_{n-k-1}, g_{n-k-2}, \ldots, g_1$ are binary numbers. An example is shown in Figure 3.68, that corresponds to a (7,4) code with generator polynomial $g(x) = x^3 + x + 1$.

The calculation of the cyclic redundancy check bits is described in Figure 3.67. First the information polynomial is multiplied by x^{n-k}.

$$x^{n-k}i(x) = i_{k-1}x^{n-1} + i_{k-2}x^{n-2} + \cdots + i_1x^{n-k+1} + i_0x^{n-k} \tag{3.51}$$

If you imagine that the k information bits are in the lower k positions in a register of length n, the multiplication by x^{n-k} moves the information bits to the k highest-order positions, since the highest term of $i(x)$ can have degree $k - 1$. This situation is shown in the example in Figure 3.68. The information polynomial is $i(x) = x^3 + x^2$, so the first step yields $x^3i(x) = x^6 + x^5$. After three shifts to the left, the contents of the shift register are (1,1,0,0,0,0,0).

Step 2 involves dividing $x^{n-k}i(x)$ by $g(x)$ to obtain the remainder $r(x)$. The terms involved in division are related by the following expression:

$$x^{n-k}i(x) = g(x)q(x) + r(x) \tag{3.52}$$

The remainder polynomial $r(x)$ provides the CRCs. In the example in Figure 3.68, we have $x^6 + x^5 = g(x)(x^3 + x^2 + x) + x$; that is, $r(x) = x$. In the figure we also show a more compact way of doing the division without explicitly writing the powers of x.

Generator polynomial: $g(x) = x^3 + x + 1$
Information: $(1,1,0,0) \rightarrow i(x) = x^3 + x^2$
Encoding: $x^3 i(x) = x^6 + x^5$

$$
\begin{array}{r}
x^3 + x^2 + x \\
x^3 + x + 1 \overline{)\, x^6 + x^5} \\
\underline{x^6 + \quad x^4 + x^3} \\
x^5 + x^4 + x^3 \\
\underline{x^5 + \quad x^3 + x^2} \\
x^4 + \quad x^2 \\
\underline{x^4 + \quad x^2 + x} \\
x
\end{array}
\qquad
\begin{array}{r}
1110 \\
1011 \overline{)\, 1100000} \\
\underline{1011} \\
1110 \\
\underline{1011} \\
1010 \\
\underline{1011} \\
010
\end{array}
$$

Transmitted codeword:
$b(x) = x^6 + x^5 + x$
$\rightarrow \underline{b} = (1,1,0,0,0,1,0)$

FIGURE 3.68 Example of CRC encoding.

The final step in the encoding procedure obtains the binary codeword $b(x)$ by adding the remainder $r(x)$ from $x^{n-k}i(x)$:

$$b(x) = x^{n-k}i(x) + r(x). \tag{3.53}$$

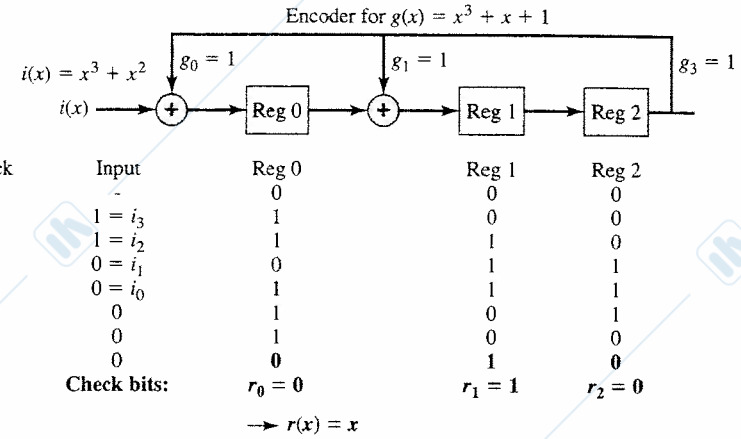
Because the divisor $g(x)$ had degree $n-k$, the remainder $r(x)$ can have maximum degree $n-k-1$ or lower. Therefore $r(x)$ has at most $n-k$ terms. In terms of the previously introduced register of length n, $r(x)$ will occupy the lower $n-k$ positions. Recall that the upper k positions were occupied by the information bits. We thus see that this encoding process introduces a binary polynomial in which the k higher-order terms are the information bits and in which the $n-k$ lower-order terms are the cyclic redundancy check bits. In the example in Figure 3.68, the division of $x^3 i(x)$ by $g(x)$ gives the remainder polynomial $r(x) = x$. The codeword polynomial is then $x^6 + x^5 + x$, which corresponds to the binary codeword (1,1,0,0,0,1,0). Note how the first four positions contain the original four information bits and how the lower three positions contains the CRC bits.

In Figure 3.66 we showed that in normal division dividing 122 by 35 yields a quotient of 3 and a remainder of 17. This result implies that $122 = 3(35) + 17$. Note that by subtracting the remainder 17 from both sides, we obtain $122 - 17 = 3(35)$ so that $122 - 17$ is evenly divisible by 35. Similarly, the codeword polynomial $b(x)$ is divisible by $g(x)$ because

$$b(x) = x^{n-k}i(x) + r(x) = g(x)q(x) + r(x) + r(x) = g(x)q(x) \tag{3.54}$$

where we have used the fact that in modulo 2 arithmetic $r(x) + r(x) = 0$. Equation (3.54) implies that *all codewords are multiples of the generator polynomial* $g(x)$. This is the pattern that must be checked by the receiver. *The receiver can check to see whether the pattern is satisfied by dividing the received polynomial by* $g(x)$. *If the remainder is nonzero, then an error has been detected.*

The Euclidean Division algorithm can be implemented using a feedback shift-register circuit that implements division. The feedback taps in this circuit are determined by the coefficients of the generator polynomial. Figure 3.69 shows the division



Clock	Input	Reg 0	Reg 1	Reg 2
0	-	0	0	0
1	$1 = i_3$	1	0	0
2	$1 = i_2$	1	1	0
3	$0 = i_1$	0	1	1
4	$0 = i_0$	1	1	1
5	0	1	0	1
6	0	1	0	0
7	0	**0**	**1**	**0**
	Check bits:	$r_0 = 0$	$r_1 = 1$	$r_2 = 0$

$\rightarrow r(x) = x$

FIGURE 3.69 Shift-register circuit for generated polynomial.

circuit for the generated polynomial $g(x) = x^3 + x + 1$. The coefficients of the dividend polynomial are fed into the shift register one coefficient at a time, starting with the highest order coefficient. In the example, the dividend polynomial is $x^6 + x^5$ which corresponds to the input sequence 1100000, as shown in the input column. The next three columns in the figure show the states of the registers as the algorithm implements the same division that was carried out in the previous encoding example. The contents of the register correspond to the coefficients of the highest terms of the dividend polynomial at any given step in the division algorithm. The rightmost register, Register 2 in the example, contains the coefficient of the highest power term. Whenever Register 2 contains a "1," a pattern corresponding to $g(x)$ is fed back into the shift-register circuit. This corresponds to the step in the division algorithm where the product of the new quotient term and the divisor $g(x)$ is subtracted from the current dividend. Thus clock step 3 corresponds to the step where the quotient x^3 is calculated in Figure 3.68, and similarly steps 4 and 5 correspond to the quotient terms x^2 and x, respectively. The final remainder is contained in the registers after the 7 input bits have been shifted into the circuit.

The same division circuit that was used by the encoder can be used by the receiver to determine whether the received polynomial is a valid codeword polynomial.

3.9.5 Standardized Polynomial Codes

Table 3.7 gives generator polynomials that have been endorsed in a number of standards. The CRC-12 and CRC-16 polynomials were introduced as part of the IBM bisync protocol for controlling errors in a communication line. The CCITT-16 polynomial is used in the HDLC standard and in XMODEM. The CCITT-32 is used in IEEE 802 LAN standards and in Department of Defense protocols, as well as in the CCITT V.42 modem standard. Finally CRC-8 and CRC-10 have recently been recommended for use in ATM networks. In the problem section we explore properties and implementations of these generator polynomials. In the next section we describe the criteria that have been used in the selection of polynomial codes in international standards.

TABLE 3.7 Standard generator polynomials.

Name	Polynomial	Used in
CRC-8	$x^8 + x^2 + x + 1$	ATM header error check
CRC-10	$x^{10} + x^9 + x^5 + x^4 + x + 1$	ATM AAL CRC
CRC-12	$x^{12} + x^{11} + x^3 + x^2 + x + 1$ $= (x + 1)(x^{11} + x^2 + 1)$	Bisync
CRC-16	$x^{16} + x^{15} + x^2 + 1$ $= (x + 1)(x^{15} + x + 1)$	Bisync
CCITT-16	$x^{16} + x^{12} + x^5 + 1$	HDLC, XMODEM, V.41
CCITT-32	$x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10}$ $+ x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$	IEEE 802, DoD, V.42, AAL5

3.9.6 Error-Detecting Capability of a Polynomial Code

We now determine the set of channel errors that a polynomial code cannot detect. In Figure 3.70 we show an additive error model for the polynomial codes. The channel can be viewed as adding, in modulo 2 arithmetic, an error polynomial, which has 1s where errors occur, to the input codeword to produce the received polynomial $R(x)$:

$$R(x) = b(x) + e(x). \tag{3.55}$$

At the receiver, $R(x)$ is divided by $g(x)$ to obtain the remainder that is defined as the **syndrome polynomial** $s(x)$. If $s(x) = 0$, then $R(x)$ is a valid codeword and is delivered to the user. If $s(x) \neq 0$, then an alarm is set, alerting the user to the detected error. Because

$$R(x) = b(x) + e(x) = g(x)q(x) + e(x) \tag{3.56}$$

we see that if an error polynomial $e(x)$ is divisible by $g(x)$, then the error pattern will be undetectable.

The design of a polynomial code for error detection involves first identifying the error polynomials we want to be able to detect and then synthesizing a generator polynomial $g(x)$ that will not divide the given error polynomials. Figure 3.71 and Figure 3.72 show the conditions required of $g(x)$ to detect various classes of error polynomials.

First consider single errors. The error polynomial is then of the form $e(x) = x^i$. Because $g(x)$ has at least two nonzero terms, it is easily shown that when multiplied by any quotient polynomial the product will also have at least two nonzero terms. Thus single errors cannot be expressed as a multiple of $g(x)$, and hence all single errors are detectable.

An error polynomial that has double errors will have the form $e(x) = x^i + x^j = x^i(1 + x^{j-i})$ where $j > i$. From the discussion for single errors, $g(x)$ cannot divide x^i.

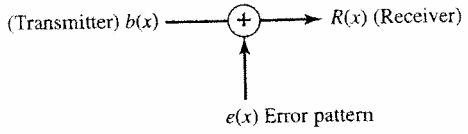


FIGURE 3.70 Additive error model for polynomial codes.

1. Single errors: $e(x) = x^i \quad 0 \le i \le n-1$
If $g(x)$ has more than one term, it cannot divide $e(x)$.

2. Double errors: $e(x) = x^i + x^j \quad 0 \le i \le j \le n-1$
$= x^i(1 + x^{j-i})$

If $g(x)$ is primitive, it will not divide $(1 + x^{j-1})$ for $j - i \le 2^{n-k} - 1$.

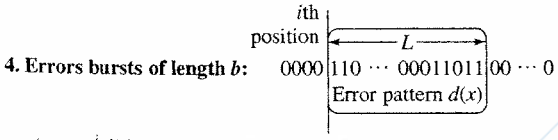
3. Odd number of errors: $e(1) = 1$ if number of errors is odd.
If $g(x)$ has $(x + 1)$ as a factor, then $g(1) = 0$ and all codewords have an even number of 1s.

FIGURE 3.71 Generator polynomials for detecting errors—part 1.

Thus $e(x)$ will be divisible by $g(x)$ only if $g(x)$ divides $(1 + x^{j-i})$. Since i can assume values from 0 to $n - 2$, we are interested in having $1 + x^m$ not be divisible by $g(x)$ for m assuming values from 1 to the maximum possible codeword length for which the polynomial will be used. The class of primitive polynomials has the property that if a polynomial has degree N, then the smallest value of m for which $1 + x^m$ is divisible by the polynomial is $2^N - 1$ [Lin 1983]. Thus if $g(x)$ is selected to be a primitive polynomial with degree $N = n - k$, then it will detect all double errors as long as the total codeword length does not exceed $2^{n-k} - 1$. Several of the generator polynomials used in practice are of the form $g(x) = (1 + x)p(x)$ where $p(x)$ is a primitive polynomial. For example, the CRC-16 polynomial is $g(x) = (1 + x)(x^{15} + x + 1)$ where $p(x) = x^{15} + x + 1$ is a primitive polynomial. Thus this $g(x)$ will detect all double errors as long as the codeword length does not exceed $2^{15} - 1 = 32{,}767$.

Now suppose that we are interested in being able to detect all odd numbers of errors. If we can ensure that all code polynomials have an even number of 1s, then we will achieve this error-detection capability. If we evaluate the codeword polynomial $b(x)$ at $x = 1$, we then obtain the sum of the binary coefficients of $b(x)$. If $b(1) = 0$ for all codeword polynomials, then $x + 1$ must be a factor of all $b(x)$ and hence $g(x)$ must contain $x + 1$ as a factor. For this reason $g(x)$ is usually chosen so that it has $x + 1$ as a factor.

Finally consider the detection of a burst of errors of length L. As shown in Figure 3.72, the error polynomial has the form $x^i d(x)$. If the error burst involves L consecutive bits, then the degree of $d(x)$ is $L - 1$. Reasoning as before, $e(x)$ will be a multiple



$e(x) = x^i d(x)$ where $\deg(d(x)) = L - 1$
$g(x)$ has degree $n - k$;
$g(x)$ cannot divide $d(x)$ if $\deg(g(x)) > \deg(d(x))$

- $L = (n - k)$ or less: all will be detected
- $L = (n - k + 1)$: $\deg(d(x)) = \deg(g(x))$
  i.e. $d(x) = g(x)$ is the only undetectable error pattern,
  fraction of bursts that are undetectable $= 1/2^{L-2}$
- $L > (n - k + 1)$: fraction of bursts that are undetectable $= 1/2^{n-k}$

FIGURE 3.72 Generator polynomials for detecting errors—part 2.

of $g(x)$ only if $d(x)$ is divisible by $g(x)$. Now if the degree of $d(x)$ is less than that of $g(x)$, then it will not be possible to divide $d(x)$ by $g(x)$. We conclude that if $g(x)$ has degree $n-k$, then all bursts of length $n-k$ or less will be detected.

If the burst error has length $L = n-k+1$, that is, degree of $d(x)$ = degree of $g(x)$, then $d(x)$ is divisible by $g(x)$ only if $d(x) = g(x)$. From Figure 3.72 $d(x)$ must have 1 in its lowest-order term and in its highest-order term, so it matches $g(x)$ in these two coefficients. For $d(x)$ to equal $g(x)$, it must also match $g(x)$ in the $n-k-1$ coefficients that are between the lowest- and highest-order terms. Only one of the 2^{n-k-1} such patterns will match $g(x)$. Therefore, the proportion of bursts of length $L = n-k+1$ that is undetectable is $1/2^{n-k-1}$. Finally, it can be shown that in the case of $L > n-k+1$ the fraction of bursts that is undetectable is $1/2^{n-k}$.

◆ 3.9.7 Linear Codes[23]

We now introduce the class of linear codes that are used extensively for error detection and correction. A **binary linear code** is specified by two parameters: k and n. The linear code takes groups of k information bits, $b_1, b_2, \ldots, b_k$, and produces a binary codeword $\underline{b}$ that consists of n bits, $b_1, b_2, \ldots, b_n$. As an example consider the (7,4) linear **Hamming code** in which the first four bits of the codeword $\underline{b}$ consist of the four information bits b_1, b_2, b_3, and b_4 and the three check bits b_5, b_6, and b_7 are given by

$$\begin{aligned} b_5 &= b_1 + b_3 + b_4 \\ b_6 &= b_1 + b_2 + b_4 \\ b_7 &= b_2 + b_3 + b_4 \end{aligned} \tag{3.57}$$

We have arranged the preceding equations so that it is clear which information bits are being checked by which check bits, that is, b_5 checks information bits b_1, b_3, and b_4. These equations allow us to determine the codeword for any block of information bits. For example, if the four information bits are (0, 1, 1, 0), then the codeword is given by $(0, 1, 1, 0, 0+1+0, 0+1+0, 0+1+0, 1+1+0) = (0, 1, 1, 0, 1, 1, 0)$. Table 3.8 shows the set of 16 codewords that are assigned to the 16 possible information blocks.

In general, the $n-k$ check bits of a linear code $b_{k+1}, \ldots, b_n$, are determined by $n-k$ linear equations:[24]

$$\begin{aligned} b_{k+1} &= a_{11}b_1 + a_{12}b_2 + \cdots + a_{1k}b_k \\ b_{k+2} &= a_{21}b_1 + a_{22}b_2 + \cdots + a_{2k}b_k \\ &\vdots \\ b_n &= a_{(n-k)1}b_1 + a_{(n-k)2}b_2 + \cdots + a_{(n-k)k}b_k \end{aligned} \tag{3.58}$$

[23] Section titles preceded by ◆ provide additional details and are not essential for subsequent sections.
[24] We require the set of linear equations to be linearly independent; that is, no equation can be written as a linear combination of the other equations.

TABLE 3.8 Hamming (7,4) code.

Information				Codeword							Weight
b_1	b_2	b_3	b_4	b_1	b_2	b_3	b_4	b_5	b_6	b_7	$w(\underline{b})$
0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	1	1	1	4
0	0	1	0	0	0	1	0	1	0	1	3
0	0	1	1	0	0	1	1	0	1	0	3
0	1	0	0	0	1	0	0	0	1	1	3
0	1	0	1	0	1	0	1	1	0	0	3
0	1	1	0	0	1	1	0	1	1	0	4
0	1	1	1	0	1	1	1	0	0	1	4
1	0	0	0	1	0	0	0	1	1	0	3
1	0	0	1	1	0	0	1	0	0	1	3
1	0	1	0	1	0	1	0	0	1	1	4
1	0	1	1	1	0	1	1	1	0	0	4
1	1	0	0	1	1	0	0	1	0	1	4
1	1	0	1	1	1	0	1	0	1	0	4
1	1	1	0	1	1	1	0	0	0	0	3
1	1	1	1	1	1	1	1	1	1	1	7

The coefficients in the preceding equations are binary numbers, and the addition is modulo 2. We say that b_{k+j} checks the information bit b_i if a_{ji} is 1. Therefore b_{k+j} is given by the modulo 2 sum of the information bits that it checks, and thus the redundancy in general linear codes is determined by parity check sums on subsets of the information bits. Note that when all of the information bits are 0, then all of the check bits will be 0. Thus the n-tuple $\underline{0}$ with all zeros is always one of the codewords of a linear code. Many linear codes can be defined by selecting different coefficients $[a_{ji}]$. Coding books such as those listed at the back of the chapter contain catalogs of good codes that can be selected for various applications.

In linear codes the redundancy is provided by the $n - k$ check bits. Thus if the transmission channel has a bit rate of R bits/seconds, then k of every n transmitted bits are information bits, so the rate at which user information flows through the channel is $R_{\text{info}} = (k/n)R$ bits/second.

Linear codes provide a very simple method for detecting errors. Before considering the general case, we illustrate the method using the Hamming code (Table 3.8). Suppose that in Equation (3.58) we add b_5 to both sides of the first equation, b_6 to both sides of the second equation, and b_7 to both sides of the third equation. We then obtain

$$\begin{aligned} 0 = b_5 + b_5 &= b_1 + b_3 + b_4 + b_5 \\ 0 = b_6 + b_6 &= b_1 + b_2 + b_4 + b_6 \\ 0 = b_7 + b_7 &= + b_2 + b_3 + b_4 + b_7 \end{aligned} \tag{3.59}$$

where we have used the fact that in modulo 2 arithmetic any number plus itself is always zero. The preceding equations state the conditions that must be satisfied by every codeword. Thus if $\underline{r} = (r_1, r_2, r_3, r_4, r_5, r_6, r_7)$ is the output of the transmission channel, then $\underline{r}$ is a codeword only if its components satisfy these equations. If we

write the equations in matrix form, we obtain a more compact representation for the conditions that all codewords must meet.

$$\begin{bmatrix}0\\0\\0\end{bmatrix} = \begin{bmatrix}1011100\\1101010\\0111001\end{bmatrix}\begin{bmatrix}b_1\\b_2\\b_3\\b_4\\b_5\\b_6\\b_7\end{bmatrix} = \mathbf{H}\underline{b}^t = \underline{0} \tag{3.60}$$

The matrix **H** is defined as the 3×7 matrix and $\underline{b}^t$ is the codeword arranged as a column vector. (Above we had previously defined $\underline{b}$ as row vector.)

By following the same procedure we have used in this example, we find that *the condition or pattern that all codewords in a linear code must meet* is given by the matrix equation:

$$\mathbf{H}\underline{b}^t = \underline{0} \tag{3.61}$$

where $\underline{0}$ is a column vector with $n - k$ components all equal to zero, $\underline{b}^t$ is the codeword arranged as a column vector, and **check matrix H** has dimension $n - k$ rows by n columns and is given by

$$\mathbf{H} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} & 10\ldots0 \\ a_{21} & a_{22} & \cdots & a_{2k} & 01\ldots0 \\ & & \vdots & & \ddots \\ a_{(n-k)1} & a_{(n-k)2} & \cdots & a_{(n-k)k} & 00\ldots1 \end{bmatrix} \tag{3.62}$$

Errors introduced in a binary channel can be modeled by the additive process shown in Figure 3.73a. The output of the binary channel can be viewed as the modulo 2 addition of an error bit e to the binary input b. If the error bit equals 0, then the output of the channel will correspond to the input of the channel and no transmission error occurs; if the error bit equals 1, then the output will differ from the input and a transmission error occurs. Now consider the effect of the binary channel on a transmitted codeword. As shown in Figure 3.73b, the channel can be viewed as having a vector input that consists

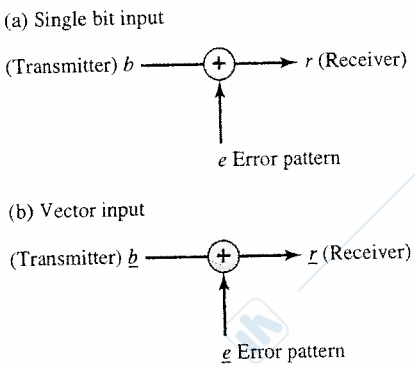


FIGURE 3.73 Additive error channel.

of the n bits that correspond to the codeword. The output of the channel $\underline{r}$ is given by the modulo 2 sum of the codeword $\underline{b}$ and an error vector $\underline{e}$ that has 1s in the components where an error occurs and 0s elsewhere:

$$\underline{r} = \underline{b} + \underline{e} \tag{3.63}$$

The error-detection system that uses a linear code checks the output of a binary channel $\underline{r}$ to see whether $\underline{r}$ is a valid codeword. The system does this by checking to see whether $\underline{r}$ satisfies Equation (3.61). The result of this calculation is an $(n-k) \times 1$ column vector called the **syndrome**:

$$\underline{s} = H\underline{r} \tag{3.64}$$

If $\underline{s} = \underline{0}$, then $\underline{r}$ is a valid codeword; therefore, the system assumes that no errors have occurred and delivers $\underline{r}$ to the user. If $\underline{s} \neq \underline{0}$, then $\underline{r}$ is not a valid codeword and the error-detection system sets an alarm indicating that errors have occurred in transmission. In an ARQ system a retransmission is requested in response to the alarm. In an FEC system the alarm would initiate a processing based on the syndrome that would attempt to identify which bits were in error and then proceed to correct them.

The error-detection system fails when $\underline{s} = 0$ but the output of the channel is not equal to the input of the channel; that is, $\underline{e}$ is nonzero. In terms of Equation (3.64) we have

$$\underline{0} = \underline{s} = H\underline{r} = H(\underline{b} + \underline{e}) = H\underline{b} + H\underline{e} = \underline{0} + H\underline{e} = H\underline{e} \tag{3.65}$$

where the fourth equality results from the linearity property of matrix multiplication and the fifth equality uses Equation (3.61). The equality $H\underline{e} = \underline{0}$ implies that when $\underline{s} = \underline{0}$ the error pattern $\underline{e}$ satisfies Equation (3.61) and hence must be a codeword. This implies that error detection using linear codes fails when the error vector is a codeword that transforms the input codeword $\underline{b}$ into a different codeword $\underline{r} = \underline{b} + \underline{e}$. Thus the set of all undetectable error vectors is the set of all nonzero codewords, and the probability of detection failure is the probability that the error vector equals any of the nonzero codewords.

In Figure 3.74 we show an example of the syndrome calculation using the (7,4) Hamming code for error vectors that contain single, double, and triple errors. We see from the example that if a single error occurred in the jth position, then the syndrome will be equal to the jth column of the $\mathbf{H}$ matrix. Since all the columns of $\mathbf{H}$ are nonzero, it follows that the syndrome will always be nonzero when the error vector contains a single error. Thus all single errors are detectable. The second example shows that if the error vector contains an error in location i and an error in location j, then the syndrome is equal to the sum of the ith and jth columns of $\mathbf{H}$. We note that for the Hamming (7,4) code all columns are distinct. Thus the syndrome will be nonzero, and all error vectors with two errors are detectable. The third example shows an error vector that contains three errors. The syndrome for this particular error vector is zero. In conclusion, we find that this Hamming code can detect all single and double errors but fails to detect some triple errors.

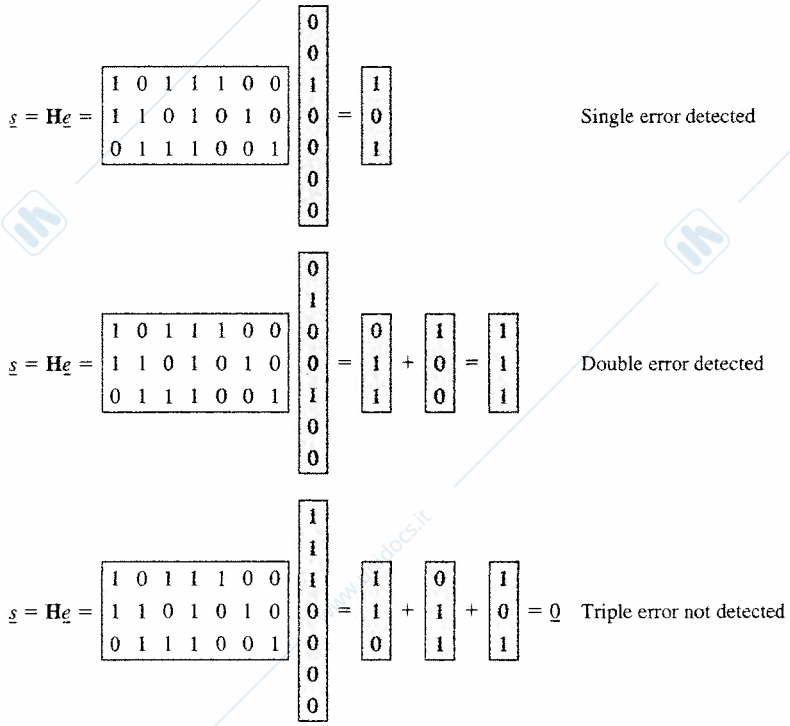


FIGURE 3.74 Syndrome calculation.

The *general class of Hamming codes* can be defined with **H** matrices that satisfy the properties identified in the preceding example.[25] Note that in the Hamming (7,4) code each of the $2^3 - 1$ possible nonzero binary triplets appears once and only once as a column of the **H** matrix. This condition enables the code to detect all single and double errors. Let m be an integer greater than or equal to 2. We can then construct an **H** matrix that has as its columns the $2^m - 1$ possible nonzero binary m-tuples. This **H** matrix corresponds to a linear code with codewords of length $n = 2^m - 1$ and with $n - k = m$ check bits. All codes that have this **H** matrix are called Hamming codes, and they are all *capable of detecting all error vectors that have single and double errors.* In the examples we have been using the $m = 3$ Hamming code. It is interesting to note that the Hamming codes can be implemented using polynomial circuits of the type discussed earlier in this section.

PERFORMANCE OF LINEAR CODES

In Figure 3.62 we showed qualitatively that we can minimize the probability of error-detection failure by spacing codewords apart in the sense that it is unlikely for errors to convert one codeword into another. In this section we show that the error-detection performance of a code is determined by the distances between codewords.

[25] The codes are named after their inventor Richard Hamming, who also pioneered many of the first concepts of linear codes.

The **Hamming distance** $d(\underline{b}_1, \underline{b}_2)$ between the binary vectors $\underline{b}_1$ and $\underline{b}_2$ is defined as the number of components in which they differ. Thus the Hamming distance between two vectors increases as the number of bits in which they differ increases. Consider the modulo 2 sum of two binary n-tuples $\underline{b}_1 + \underline{b}_2$. The components of this sum will equal one when the corresponding components in $\underline{b}_1$ and $\underline{b}_2$ differ, and they will be zero otherwise. Clearly, this result is equal to the number of 1s in $\underline{b}_1 + \underline{b}_2$, so

$$d(\underline{b}_1, \underline{b}_2) = w(\underline{b}_1 + \underline{b}_2) \tag{3.66}$$

where w is the weight function introduced earlier. The extent to which error vectors with few errors are more likely than error vectors with many errors suggests that we should design linear codes that have codewords that are far apart in the sense of Hamming distance.

Define the **minimum distance** d_{min} of a code as follows:

$$d_{min} = \text{distance between two closest distinct codewords} \tag{3.67}$$

For any given linear code, the pair of closest codewords is the most vulnerable to transmission error, so d_{min} can be used as a worst-case type of measure. From Equation (3.65) we have that if $\underline{b}_1$ and $\underline{b}_2$ are codewords, then $\underline{b}_1 + \underline{b}_2$ is also a codeword. To find d_{min}, we need to find the pair of distinct codewords $\underline{b}_1$ and $\underline{b}_2$ that minimize $d(\underline{b}_1, \underline{b}_2)$. By Equation (3.66), this is equivalent to finding the nonzero codeword with the smallest weight. Thus

$$d_{min} = \text{weight of the nonzero codeword with the smallest number of 1s} \tag{3.68}$$

From Table 3.8 above, we see the Hamming (7,4) code has $d_{min} = 3$.

If we start changing the bits in a codeword one at a time until another codeword is obtained, then we will need to change at least d_{min} bits before we obtain another codeword. This situation implies that all error vectors with $d_{min} - 1$ or fewer errors are detectable. We say that *a code is* t*-error detecting* if $d_{min} \geq t + 1$.

Finally, let us consider the probability of error-detection failure for a general linear code. In the case of the random error vector channel model, all 2^n possible error patterns are equally probable. A linear (n, k) code fails to detect only the $2^k - 1$ error vectors that correspond to nonzero codewords. We can state then that *the probability of error-detection failure for the random error vector channel model is* $(2^k - 1)/2^n \approx 1/2^{n-k}$. Furthermore, we can decrease the probability of detection failure by increasing the number of parity bits $n - k$.

Consider now the random bit error channel model. The probability of detection failure is given by

$$\begin{aligned} P[\text{detection failure}] &= P[\underline{e} \text{ is a nonzero codeword}] \\ &= \sum_{\text{nonzero codewords } \underline{b}} (1-p)^{n-w(\underline{b})} p^{w(\underline{b})} \\ &= \sum_{w=d_{min}}^{d_{max}} N_w (1-p)^{n-w} p^w \\ &\approx N_{d_{min}} p^{d_{min}} \qquad \text{for } p \ll 1 \end{aligned} \tag{3.69}$$

The second summation adds the probability of all nonzero codewords. The third summation combines all codewords of the same weight, so N_w is the total number of codewords that have weight w. The approximation results from the fact that the summation is dominated by the leading term when p is very small.

Consider the (7,4) Hamming code as an example once again. For the random error vector model, the probability of error-detection failure is $1/2^3 = 1/8$. On the other hand, for the random bit error channel the probability of error-detection failure is approximately $7p^3$, since $d_{min} = 3$ and seven codewords have this weight. If $p = 10^{-4}$, then the probability of error-detection failure is 7×10^{-12}. Compared to the single parity check code, the Hamming code yields a tremendous improvement in error-detection capability.

3.9.8 Error Correction

In FEC the detection of transmission errors is followed by processing to determine the most likely error locations. Assume that an error has been detected so $\underline{s} \neq \underline{0}$. Equation (3.70) describes how an FEC system attempts to carry out the correction.

$$\underline{s} = H\underline{r} = H(\underline{b} + \underline{e}) = H\underline{b} + H\underline{e} = \underline{0} + H\underline{e} = H\underline{e}. \tag{3.70}$$

The receiver uses Equation (3.70) to calculate the syndrome and then to diagnose the most likely error pattern. If $\mathbf{H}$ were an invertible matrix, then we could readily find the error vector from $\underline{e} = \mathbf{H}^{-1}\underline{s}$. Unfortunately, $\mathbf{H}$ is not invertible. Equation (3.70) consists of $n - k$ equations in n unknowns, $e_1, e_2, \ldots, e_n$. Because we have fewer equations than unknowns, the system is underdetermined and Equation (3.70) has more than one solution. In fact, it can be shown that 2^k binary n-tuples satisfy Equation (3.70). Thus for any given nonzero $\underline{s}$, Equation (3.70) allows us to identify the 2^k possible error vectors that could have produced $\underline{s}$. The error-correction system cannot proceed unless it has information about the probabilities with which different error patterns can occur. The error-correction system uses such information to identify the most likely error pattern from the set of possible error patterns.

We provide a simple example to show how error correction is carried out. Suppose we are using the Hamming (7,4) code. Assume that the received vector is $\underline{r} =$ (0,0,1,0,0,0,1). The syndrome calculation gives $\underline{s} = (1,0,0)^t$. Because the fifth column of $\mathbf{H}$ is (1,0,0), one of the error vectors that gives this syndrome is (0,0,0,0,1,0,0). Note from Equation 3.70 that if we add a codeword to this error vector, we obtain another vector that gives the syndrome $(1,0,0)^t$. The $2^k = 16$ possible error vectors are obtained by adding the 16 codewords to (0,0,0,0,1,0,0) and are listed in Table 3.9. The error-correction system must now select the error vector in this set that is most likely to have been introduced by the channel. Almost all error-correction systems simply select the error vector with the smallest number of 1s. Note that this error vector also corresponds to the most likely error vector for the random bit error channel model. For this example the error-correction system selects $\underline{e}$ = (0,0,0,0,1,0,0) and then outputs the codeword $\underline{r} + \underline{e} =$ (0,0,1,0,1,0,1) from which the user extracts the information bits, 0010. Algorithms have been developed that allow the calculation of the most likely error vector from the syndrome. Alternatively, the calculations can be carried out once, and then a

TABLE 3.9 Error vectors corresponding to syndrome $(\mathbf{1}, \mathbf{0}, \mathbf{0})^t$.

Error vectors							Weight
e_1	e_2	e_3	e_4	e_5	e_6	e_7	$w(\underline{e})$
0	0	0	0	1	0	0	1
0	0	0	1	0	1	1	3
0	0	1	0	0	0	1	2
0	0	1	1	1	1	0	4
0	1	0	0	1	1	1	4
0	1	0	1	0	0	0	2
0	1	1	0	0	1	0	3
0	1	1	1	1	0	1	5
1	0	0	0	0	1	0	2
1	0	0	1	1	0	1	4
1	0	1	0	1	1	1	5
1	0	1	1	0	0	0	3
1	1	0	0	0	0	1	3
1	1	0	1	1	1	0	5
1	1	1	0	1	0	0	4
1	1	1	1	0	1	1	6

table can be set up that contains the error vector that is to be used for correction for each possible syndrome. The error-correction system then carries out a table lookup each time a nonzero syndrome is found.

The error-correction system is forced to select only one error vector out of the 2^k possible error vectors that could have produced the given syndrome. Thus the error-correction system will successfully recover the transmitted codeword only if the error vector is the most likely error vector in the set. When the error vector is one of the other 2^{k-1} possible error vectors, the error-correction system will perform corrections in the wrong locations and actually introduce more errors! In the preceding example, assuming a random bit error channel model, the probability of the most likely error vector is $p(1-p)^6 \approx p$ for the error vector of weight 1; the probability of the other error vectors is approximately $3p^2(1-p)^5$ for the three error vectors of weight 2 and where we have neglected the remainder of the error patterns. Thus when the error-correction system detects an error the system's attempt to correct the error fails with probability

$$\frac{3p^2(1-p)^5}{p(1-p)^6 + 3p^2(1-p)^5} \approx 3p \tag{3.71}$$

Figure 3.75 summarizes the four outcomes that can arise from the error-correction process. We begin with the error vector that is revealed through its syndrome. If $\underline{s} = \underline{0}$, then the received vector $\underline{r}$ is accepted as correct and delivered to the user. Two outcomes lead to $\underline{s} = \underline{0}$: the first corresponds to when no errors occur in transmission and has probability $(1-p)^7 \approx 1 - 7p$; the second corresponds to when the error vector is undetectable and has probability $7p^3$. If $\underline{s} \neq \underline{0}$, the system attempts to perform error correction. This situation occurs with probability $1 - P[\underline{s} = \underline{0}] = 1 - \{1 - 7p + 7p^3\} \approx 7p$. Two further outcomes are possible in the $\underline{s} \neq \underline{0}$ case: the third outcome is when the error vector is correctable and has conditional probability $(1-3p)$; the fourth is when

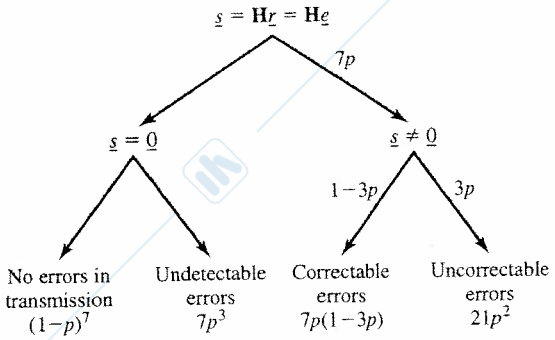


FIGURE 3.75 Summary of error-correction process outcomes.

the error vector is not correctable and has conditional probability $3p$. From the figure we see that the probability that the error-correction system fails to correct an error pattern is $21p^2$. To summarize, the first and third outcomes yield correct user information. The second and fourth outcomes result in the delivery of incorrect information to the user. Through this example we have demonstrated the analysis required to determine the effectiveness of any error-correction system.

EXAMPLE **Performance Improvement of Hamming Code**

Suppose that the (7,4) code is used in a channel that has a bit error rate of $p = 10^{-3}$. The probability that the decoder fails to correct an error pattern is then $21p^2 = 21 \times 10^{-6}$, which is a reduction in bit error rate of two orders of magnitude from the original bit error rate of 10^{-3}.

Now suppose that the code is used in a "relatively" clean optical transmission system, for example, $p = 10^{-12}$, then the probability of incorrect decoding is 2.1×10^{-23}. If the transmission speed is 1 Gbps, then this corresponds to a decoding error occurring roughly every 1.5 million years! In other words, the optical digital transmission system can be made error free as long as the error-producing mechanism can be modeled by independent bit errors.

The minimum distance of a code is useful in specifying its error-correcting capability. In Figure 3.76 we consider a code with $d_{min} = 5$, and we show two codewords that are separated by the minimum distance. If we start by changing the bits in $\underline{b}_1$, one bit at a time until we obtain $\underline{b}_2$, we find four n-tuples between the two codewords. We can imagine drawing a sphere of radius 2 around each codeword. The sphere around $\underline{b}_1$ will contain two of the n-tuples, and the sphere around $\underline{b}_2$ will contain the other n-tuples.

Note that because all pairs of codewords are separated by at least distance d_{min}, we can draw a sphere of radius 2 around every single codeword, and these spheres will all be nonoverlapping. This geometrical view gives us another way of looking at error correction. We can imagine that the error-correction system takes the vector $\underline{r}$ and looks up which sphere it belongs to; the system then generates the codeword that is at the center of the sphere. Note that if the error vector introduced by the channel has

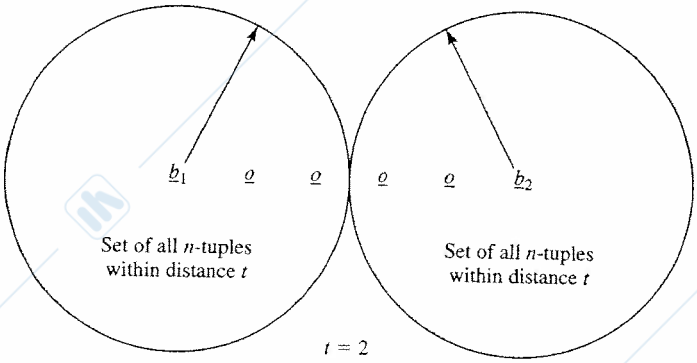


FIGURE 3.76 Partitioning of n-tuples into disjoint spheres: If $d_{\min} = 2t + 1$, nonoverlapping spheres of radius t can be drawn around each codeword.

two or fewer errors, then the error-correction system will always produce the correct codeword. Conversely, if the number of errors is more than two, the error-correction system will produce an incorrect codeword.

The discussion of Figure 3.76 can be generalized as follows. Given a linear code with $d_{min} \geq 2t + 1$, it is possible to draw nonoverlapping spheres of radius t around all the codewords. Hence the error-correction system is guaranteed to operate correctly whenever the number of errors is smaller than t. For this reason we say that *a code is* t*-error correcting* if $d_{min} \geq 2t + 1$.

The Hamming codes introduced above all have $d_{min} = 3$. Consequently, *all Hamming codes are single-error correcting*. The Hamming codes use $m = n - k$ bits of redundancy and are capable of correcting single errors. An interesting question is, if we use $n - k = 2m$ bits in a code of length $n = 2^m - 1$, can we correct all double errors? Similarly, if we use $n - k = 3m$, can we correct triple errors? The answer is yes in some cases and leads to the classes of BCH and Reed-Solomon codes [Lin 1983].

In this book we have presented only linear codes that operate on non-overlapping blocks of information. These block codes include the classes of Hamming codes, BCH codes, and Reed-Solomon codes that have been studied extensively and are in wide use. These codes provide a range of choice in terms of n, k, and d_{min} that allows a system designer to select a code for a given application. *Convolutional codes* are another important class of error-correcting codes. These codes operate on overlapping blocks of information and are also in wide use. [Lin 1983] provides an introduction to convolutional codes.

Finally, we consider the problem of error correction in channels that introduce bursts of errors. The codes discussed up to this point correct error vectors that contain $(d_{min} - 1)/2$ or fewer errors. These codes can be used in channels with burst errors if combined with the following **interleaving** method. The user information is encoded using the given linear code, and the codewords are written as columns in an array as shown in Figure 3.77. The array is transmitted over the communication channels *row by row*. The interleaver depth L is selected so that the errors associated with a burst are distributed over many codewords. The error-correction system will be effective if

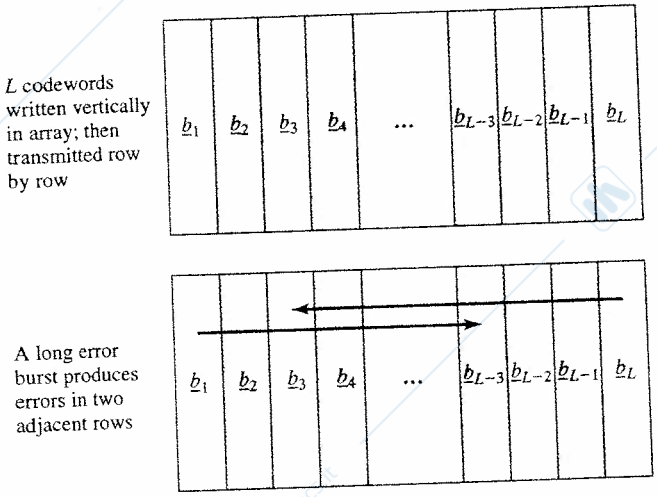


FIGURE 3.77 Interleaving.

the number of errors in each codeword is within its error-correcting capability. For example, if the linear code can correct up to two errors, then interleaving makes it possible to correct any burst of length less than $2L$.

SUMMARY

Binary information, "bits," are at the heart of modern communications. All information can be represented as blocks or streams of bits. Modern communication networks are designed to carry bits and therefore can handle *any* type of information.

We began this chapter with a discussion of the basic properties of common types of information such as text, image, voice, audio, and video. We discussed how analog signals can be converted into sequences of binary information. We also discussed the amount of information that is required to represent them in terms of bits or bits/second.

We described the difference between digital and analog communication and explained why digital communication has prevailed. We then considered the design of digital transmission systems. The characterization of communication channels in terms of their response to sinusoidal signals and to pulse signals was introduced. The notion of bandwidth of a channel was also introduced.

We first considered baseband digital transmission systems. We showed how the bandwidth of a channel determines the maximum rate at which pulses can be transmitted with zero intersymbol interference. This is the Nyquist signaling rate. We then showed the effect of SNR on the reliability of transmissions and developed the notion of channel capacity as the maximum reliable transmission rate that can be achieved over a channel.

Next we explained how modems use sinusoidal signals to transmit binary information over bandpass channels. The notion of a signal constellation was introduced and used to explain the operation of telephone modem standards.

The properties of different types of transmission media were discussed next. We first considered twisted-pair cable, coaxial cable, and optical fiber, which are used in "wired" transmission. We then discussed radio and infrared light, which are used in wireless transmission. Important physical layer standards were used as examples where the various types of media are used.

Finally, we presented coding techniques that are used in error control. Basic error-detection schemes that are used in many network standards were introduced first. An optional section then discussed error-correction schemes that are used when a return channel is not available.

CHECKLIST OF IMPORTANT TERMS

amplitude-response function
amplitude shift keying (ASK)
analog signal
asymmetric digital subscriber line (ADSL)
attenuation
bandwidth of a channel
bandwidth of a signal
baseband transmission
◆ binary linear code
bipolar encoding
bit rate
burst error
cable modem
channel
channel capacity
check bit
◆ check matrix
checksum
coaxial cable
codeword
cyclic redundancy check (CRC)
delay
differential encoding
differential Manchester encoding
digital transmission
equalizer
error control
error detection
forward error correction (FEC)
frequency shift keying (FSK)
generator polynomial
◆ Hamming codes
◆ Hamming distance
impulse response
information polynomial
◆ interleaving
line coding
Manchester encoding
◆ minimum distance
modem
multilevel transmission
multimode fiber
nonreturn-to-zero (NRZ) encoding
NRZ inverted
Nyquist sampling rate
Nyquist signaling rate
optical amplifier
optical fiber
phase shift keying (PSK)
polar NRZ encoding
polynomial code
pulse code modulation (PCM)
quadrature amplitude modulation (QAM)
quantizer
quantizer error
quantizer signal-to-noise ratio
random bit error model
random error vector model
redundancy
regenerator
repeater
signal constellation
signal-to-noise ratio (SNR)
single parity check code
single-mode fiber

spectrum
◆ syndrome
syndrome polynomial
◆ t-error detecting
transmission error
transmission medium
twisted-pair cable
uniform quantizer
unshielded twisted pair (UTP)
wavelength
wavelength-division multiplexing (WDM)
weight

FURTHER READING

Ahamed, S. W. and V. B. Lawrence, *Design and Engineering of Intelligent Communication Systems*, Kluwer Academic Publishers, Boston, 1997. Detailed information about properties of transmission media.

Ayanoglu, E., N. Dagdeviren, G. D. Golden, and J. E. Mazo, "An Equalizer Design Technique for the PCM Modem: A New Modem for the Digital Public Switched Network," *IEEE Transactions on Communications*, Vol. 46, June 1998, pp. 763–774.

Bell Telephone Laboratories, *Transmission Systems for Communications*, 1971. Classic on transmission aspects of telephony.

Bellamy, J., *Digital Telephony*, John Wiley & Sons, Inc., New York, 1991. Excellent coverage of digital telephony.

Dutton, H. J. R. and P. Lenhard, *High-Speed Networking Technology*, Prentice Hall PTR, Upper Saddle River, New Jersey, 1995. Good coverage of physical layer aspects of computer networks.

Glover, I. A. and P. M. Grant, *Digital Communications*, Prentice-Hall, Englewood Cliffs, New Jersey, 1998. Up-to-date introduction to digital transmission systems.

Keiser, G., *Optical Fiber Communications*, McGraw-Hill, New York, 2000.

Leon-Garcia, A., *Probability and Random Processes for Electrical Engineering*, Addison-Wesley, Reading, Massachusetts, 1994.

Lin, S. and D. J. Costello, *Error Control Coding: Fundamentals and Applications*, Prentice Hall, Englewood Cliffs, New Jersey, 1983.

Mukherjee, B., *Optical Communication Networks*, McGraw-Hill, New York, 1997.

Seifert, R., *Gigabit Ethernet*, Addison-Wesley, Reading, Massachusetts, 1998.

Smith, D. R., *Digital Transmission Systems*, Van Nostrand Reinhold Company, New York, 1985.

RFC 1071, R. Braden and D. Dorman, "Computing the Internet Checksum," September 1988.

See our website for additional references available through the Internet.

PROBLEMS

3.1. Suppose the size of an uncompressed text file is 1 megabyte.
(a) How long does it take to download the file over a 32 kilobit/second modem?
(b) How long does it take to download the file over a 1 megabit/second modem?
(c) Suppose data compression is applied to the text file. How much do the transmission times in parts (a) and (b) change?

3.2. A scanner has a resolution of 600 × 600 pixels/square inch. How many bits are produced by an 8-inch × 10-inch image if scanning uses 8 bits/pixel? 24 bits/pixel?

3.3. Suppose a computer monitor has a screen resolution of 1200 × 800 pixels. How many bits are required if each pixel uses 256 colors? 65,536 colors?

3.4. Explain the difference between facsimile, GIF, and JPEG coding. Give an example of an image that is appropriate to each of these three methods.

3.5. A digital transmission system has a bit rate of 45 megabits/second. How many PCM voice calls can be carried by the system?

3.6. Suppose a storage device has a capacity of 1 gigabyte. How many 1-minute songs can the device hold using conventional CD format? using MP3 coding?

3.7. How many high-quality audio channels can be transmitted using an HDTV channel?

3.8. How many HDTV channels can be transmitted simultaneously over the optical fiber transmission systems in Table 3.3?

3.9. Comment on the properties of the sequence of frame images and the associated bit rates in the following examples:
(a) A children's cartoon program.
(b) A music video.
(c) A tennis game; a basketball game.
(d) A documentary on famous paintings.

3.10. Suppose that at a given time of the day, in a city with a population of 1 million, 1 percent of the people are on the phone.
(a) What is the total bit rate generated by all these people if each voice call is encoded using PCM?
(b) What is the total bit rate if all of the telephones are replaced by H.261 videoconferencing terminals?

3.11. Consider an analog repeater system in which the signal has power σ_x^2 and each stage adds noise with power σ_n^2. For simplicity assume that each repeater recovers the original signal without distortion but that the noise accumulates. Find the SNR after n repeater links. Write the expression in decibels: SNR dB $= 10 \log_{10}$SNR.

3.12. Suppose that a link between two telephone offices has 50 repeaters. Suppose that the probability that a repeater fails during a year is 0.01 and that repeaters fail independently of each other.
(a) What is the probability that the link does not fail at all during one year?
(b) Repeat (a) with 10 repeaters; with 1 repeater.

3.13. Suppose that a signal has twice the power as a noise signal that is added to it. Find the SNR in decibels. Repeat if the signal has 10 times the noise power? 2^n times the noise power? 10^k times the noise power?

3.14. A way of visualizing the Nyquist theorem is in terms or periodic sampling of the second hand of a clock that makes one revolution around the clock every 60 seconds. The Nyquist sampling rate here should correspond to two samples per cycle, that is, sampling should be done at least every 30 seconds.
(a) Suppose we begin sampling when the second hand is at 12 o'clock and that we sample the clock every 15 seconds. Draw the sequence of observations that result. Does the second hand appear to move forward?

(b) Now suppose we sample every 30 seconds. Does the second hand appear to move forward or backward? What if we sample every 29 seconds?
(c) Explain why a sinusoid should be sampled at a little more than twice its frequency.
(d) Now suppose that we sample every 45 seconds. What is the sequence of observations of the second hand?
(e) Motion pictures are made by taking a photograph 24 times a second. Use part (c) to explain why car wheels in movies often appear to spin backward while the cars are moving forward!

3.15. "Software radios" are devices that can demodulate and decode any radio signal regardless of format or standard. The basic idea in software radio is to immediately convert the transmitted radio signal into digital form so that digital signal processing software can be used to do the particular required processing. Suppose that a software radio is to demodulate FM radio and television. What sampling rate is required in the A/D conversion? The transmission bandwidth of FM radio is 200 kHz, and the transmission bandwidth of television is 6 MHz.

3.16. An AM radio signal has the form $x(t) = m(t)\cos(2\pi f_c t)$, where $m(t)$ is a low-pass signal with bandwidth W Hz. Suppose that $x(t)$ is sampled at a rate of $2W$ samples/second. Sketch the spectrum of the sampled sequence. Under which conditions can $m(t)$ be recovered from the sampled sequence? *Hint*: See Appendix 3C.

3.17. A black-and-white image consists of a variation in intensity over the plane.
(a) By using an analogy to time signals, explain spatial frequency in the horizontal direction; in the vertical spatial direction. Hint: Consider bands of alternating black and white bands. Do you think there is a Nyquist sampling theorem for images?
(b) Now consider a circle and select a large even number N of equally spaced points around the perimeter of the circle. Draw a line from each point to the center and color alternating regions black and white. What are the spatial frequencies in the vicinity of the center of the circle?

3.18. A high-quality speech signal has a bandwidth of 8 kHz.
(a) Suppose that the speech signal is to be quantized and then transmitted over a 28.8 kbps modem. What is the SNR of the received speech signal?
(b) Suppose that instead a 64 kbps modem is used? What is the SNR of the received speech signal?
(c) What modem speed is needed if we require an SNR of 40 dB?

3.19. An analog television signal is a low-pass signal with a bandwidth of 4 MHz. What bit rate is required if we quantize the signal and require an SNR of 60 dB?

3.20. An audio digitizing utility in a PC samples an input signal at a rate of 44 kHz and 16 bits/sample. How big a file is required to record 20 seconds?

3.21. Suppose that a signal has amplitudes uniformly distributed between $-V$ and V.
(a) What is the SNR for a uniform quantizer that is designed specifically for this source?
(b) Suppose that the quantizer design underestimates the dynamic range by a factor of 2; that is, the actual dynamic range is $-2V$ to $2V$. Plot the quantization error versus signal amplitude for this case. What is the SNR of the quantizer?

3.22. A telephone office line card is designed to handle modem signals of the form $x(t) = A\cos(2\pi f_c t + \phi(t))$. These signals are to be digitized to yield an SNR of 40 dB using a uniform quantizer. Due to variations in the length of lines and other factors, the value of A varies by up to a factor of 100.
(a) How many levels must the quantizer have to produce the desired SNR?
(b) Explain how an adaptive quantizer might be used to address this problem.

3.23. The basic idea in companding is to obtain robustness with respect to variations in signal level by using small quantizer intervals for small signal values and larger intervals for larger signal values. Consider an eight-level quantizer in which the inner four intervals are Δ wide and the outer four intervals are 2Δ wide. Suppose the quantizer covers the range -1 to 1. Find the SNR if the input signal is uniformly distributed between $-V$ and V for $1/2 < V < 1$. Compare to the SNR of a uniform quantizer.

3.24. Suppose that a speech signal is A/D and D/A converted four times in traversing a telephone network that contains analog and digital switches. What is the SNR of the speech signal after the fourth D/A conversion?

3.25. A square periodic signal is represented as the following sum of sinusoids:

$$g(t) = \frac{2}{\pi}\sum_{k=0}^{\infty}\frac{(-1)^k}{2k+1}\cos(2k+1)\pi t$$

(a) Suppose that the signal is applied to an ideal low-pass filter with bandwidth 15 Hz. Plot the output from the low-pass filter and compare to the original signal. Repeat for 5 Hz; for 3 Hz. What happens as W increases?
(b) Suppose that the signal is applied to a bandpass filter that passes frequencies from 5 to 9 Hz. Plot the output from the filter and compare to the original signal.

3.26. Suppose that the 8 kbps periodic signal in Figure 3.25 is transmitted over a system that has an attenuation function equal to 1 for all frequencies and a phase function that is equal to $-90°$ for all frequencies. Plot the signal that comes out of this system. Does it differ in shape from the input signal?

3.27. A 10 kHz baseband channel is used by a digital transmission system. Ideal pulses are sent at the Nyquist rate, and the pulses can take 16 levels. What is the bit rate of the system?

3.28. Suppose a baseband transmission system is constrained to a maximum signal level of ± 1 volt and that the additive noise that appears in the receiver is uniformly distributed between $[-1/15, 1/15]$. How many levels of pulses can this transmission system use before the noise starts introducing errors?

3.29. What is the maximum reliable bit rate possible over a telephone channel with the following parameters:
(a) $W = 2.4$ kHz SNR $= 20$ dB
(b) $W = 2.4$ kHz SNR $= 40$ dB
(c) $W = 3.0$ kHz SNR $= 20$ dB
(d) $W = 3.0$ kHz SNR $= 40$ dB

3.30. Suppose we wish to transmit at a rate of 64 kbps over a 3 kHz telephone channel. What is the minimum SNR required to accomplish this?

3.31. Suppose that a low-pass communications system has a 1 MHz bandwidth. What bit rate is attainable using 8-level pulses? What is the Shannon capacity of this channel if the SNR is 20 dB? 40 dB?

3.32. Most digital transmission systems are "self-clocking" in that they derive the bit synchronization from the signal itself. To do this, the systems use the transitions between positive and negative voltage levels. These transitions help define the boundaries of the bit intervals.

(a) The nonreturn-to-zero (NRZ) signaling method transmits a 0 with a $+1$ voltage of duration T, and a 1 with a -1 voltage of duration T. Plot the signal for the sequence n consecutive 1s followed by n consecutive 0s. Explain why this code has a synchronization problem.

(b) In differential coding the sequence of 0s and 1s induces changes in the polarity of the signal; a binary 0 results in no change in polarity, and a binary 1 results in a change in polarity. Repeat part (a). Does this scheme have a synchronization problem?

(c) The Manchester signaling method transmits a 0 as a $+1$ voltage for $T/2$ seconds followed by a -1 for $T/2$ seconds; a 1 is transmitted as a -1 voltage for $T/2$ seconds followed by a $+1$ for $T/2$ seconds. Repeat part (a) and explain how the synchronization problem has been addressed. What is the cost in bandwidth in going from NRZ to Manchester coding?

3.33. Consider a baseband transmission channel with a bandwidth of 10 MHz. Which bit rates can be supported by the bipolar line code and by the Manchester line code?

3.34. The impulse response in a T-1 copper-wire transmission system has the idealized form where the initial pulse is of amplitude 1 and duration 1 and the afterpulse is of amplitude -0.1 and of duration 10.

(a) Let $\delta(t)$ be the narrow input pulse in Figure 3.27. Suppose we use the following signaling method: Every second, the transmitter accepts an information bit; if the information bit is 0, then $-\delta(t)$ is transmitted, and if the information bit is 1, then $\delta(t)$ is transmitted. Plot the output of the channel for the sequence 1111000. Explain why the system is said to have "dc" or baseline wander.

(b) The T-1 transmission system uses bipolar signaling in the following fashion: If the information bit is a 0, then the input to the system is $0 * \delta(t)$; if the information bit is a 1, then the input is $\delta(t)$ for an even occurrence of a 1 and $-\delta(t)$ for an odd occurrence of a 1. Plot the output of the channel for the sequence 1111000. Explain how this signaling solves the "dc" or baseline wander problem.

3.35. The raised cosine transfer function, shown in Figure 3.31, has a corresponding impulse response given by

$$p(t) = \frac{\sin(\pi t/T)}{\pi t/T} \frac{\cos(\pi \alpha t/T)}{1 - (2\alpha t/T)^2}$$

(a) Plot the response of the information sequence 1010 for $\alpha = \frac{1}{2}; \alpha = \frac{1}{8}$.

(b) Compare this plot to the response, using the pulse in Figure 3.27.

3.36. Suppose a CATV system uses coaxial cable to carry 100 channels, each of 6 MHz bandwidth. Suppose that QAM modulation is used.
(a) What is the bit rate/channel if a four-point constellation is used? eight-point constellation?
(b) Suppose a digital TV signal requires 4 Mbps. How many digital TV signals can each channel handle for the two cases in part (a)?

3.37. Explain how ASK was used in radio telegraphy. Compare the use of ASK to transmit Morse code with the use of ASK to transmit text using binary information.

3.38. Suppose that a modem can transmit eight distinct tones at distinct frequencies. Every T seconds the modem transmits an arbitrary combination of tones (that is, some are present, and some are not present).
(a) What bit rate can be transmitted using this modem?
(b) Is there a relationship between T and the frequency of the signals?

3.39. A phase modulation system transmits the modulated signal $A\cos(2\pi f_c t + \phi)$ where the phase ϕ is determined by the two information bits that are accepted every T-second interval:

$$\text{for } 00,\ \phi = 0;\ \text{for } 01,\ \phi = \pi/2;\ \text{for } 10,\ \phi = \pi;\ \text{for } 11,\ \phi = 3\pi/2.$$

(a) Plot the signal constellation for this modulation scheme.
(b) Explain how an eight-point phase modulation scheme would operate.

3.40. Suppose that the receiver in a QAM system is not perfectly synchronized to the carrier of the received signal; that is, the receiver multiplies the received signal by $2\cos(2\pi f_c t + \phi)$ and by $2\sin(2\pi/f_c t + \phi)$ where ϕ is a small phase error. What is the output of the demodulator?

3.41. In differential phase modulation the binary information determines the *change* in the phase of the carrier signal $\cos(2\pi f_c t)$. For example, if the information bits are 00, the phase change is 0; if 01, it is $\pi/2$; for 10, it is π; and for 11, it is $3\pi/2$.
(a) Plot the modulated waveform that results from the binary sequence 01100011. Compare it to the waveform that would be produced by ordinary phase modulation as described in problem 3.39.
(b) Explain how differential phase modulation can be demodulated.

3.42. A new broadcast service is to transmit digital music using the FM radio band. Stereo audio signals are to be transmitted using a digital modem over the FM band. The specifications for the system are the following: Each audio signal is sampled at a rate of 40 kilosamples/second and quantized using 16 bits; the FM band provides a transmission bandwidth of 200 kiloHertz.
(a) What is the total bit rate produced by each stereo audio signal?
(b) How many points are required in the signal constellation of the digital modem to accommodate the stereo audio signal?

3.43. A twisted-wire pair has an attenuation of 0.7 dB/kilometer at 1 kHz.
(a) How long can a link be if an attenuation of 20 dB can be tolerated?
(b) A twisted pair with loading coils has an attenuation of 0.2 dB/kilometer at 1 kHz. How long can the link be if an attenuation of 20 dB can be tolerated?

3.44. Use Figure 3.47 and Figure 3.50 to explain why the bandwidth of twisted-wire pairs and coaxial cable decreases with distance.

3.45. Calculate the bandwidth of the range of light covering the range from 1200 nm to 1400 nm. How many Hz per person are available if the population of the world is six billion people? Repeat for 1400 nm to 1600 nm. (Note that the speed of light in fiber is approximately 2×10^8 m/sec.)

3.46. Suppose that we wish to delay an optical signal by 1 nanosecond. How long a length of optical fiber is needed to do this? How much is the signal attenuated? Repeat for 1 millisecond.

3.47. Compare the attenuation in a 100 km link for optical fibers operating at 850 nm, 1300 nm, and 1550 nm.

3.48. The power of an optical signal in dBm is defined as $10 \log_{10} P$ where P is in milliwatts.
(a) What is the power in milliwatts of a –30 dBm signal? 6 dBm signal?
(b) What is the power in dBm of 1 microwatt signal?
(c) What is the power of an optical signal if initially it is 2 mW and then undergoes attenuation by 10 dB?

3.49. A 10 dBm optical signal propagates across N identical devices. What is the output signal power if the loss per device is 1 dB? (See Problem 3.48.)

3.50. Suppose that WDM wavelengths in the 1550 nm band are separated by 0.8 nm. What is the frequency separation in Hz? What is an appropriate bit rate for signals carried on these wavelengths? Repeat for 0.4 nm and 0.2 nm.

3.51. Can WDM be used for simultaneous transmission of optical signals in opposite directions?

3.52. Explain how prisms and prismlike devices can be used in WDM systems.

3.53. Compare the transition from analog repeaters to digital regenerators for copper-based transmission systems to the current transition from single-wavelength digital regenerator optical systems to multiwavelength optically amplified systems? What is the same and what is different? What is the next transition for optical transmission systems?

3.54. Suppose a network provides wavelength services to users by establishing end-to-end wavelengths across a network. A wavelength converter is a device that converts an optical signal from one wavelength to another wavelength. Explain the role of wavelength converters in such a network.

3.55. A satellite is stationed approximately 36,000 km above the equator. What is the attenuation due to distance for the microwave radio signal?

3.56. Suppose a transmission channel operates at 3 Mbps and has a bit error rate of 10^{-3}. Bit errors occur at random and independent of each other. Suppose that the following code is used. To transmit a 1, the codeword 111 is sent; to transmit a 0, the codeword 000 is sent. The receiver takes the three received bits and decides which bit was sent by taking the majority vote of the three bits. Find the probability that the receiver makes a decoding error.

3.57. An early code used in radio transmission involved codewords that consist of binary bits and contain the same number of 1s. Thus the two-out-of-five code only transmits blocks of five bits in which two bits are 1 and the others 0.
(a) List the valid codewords.
(b) Suppose that the code is used to transmit blocks of binary bits. How many bits can be transmitted per codeword?
(c) What pattern does the receiver check to detect errors?
(d) What is the minimum number of bit errors that cause a detection failure?

3.58. Find the probability of error-detection failure for the code in problem 3.57 for the following channels:
(a) The random error vector channel.
(b) The random bit error channel.

3.59. Suppose that two check bits are added to a group of $2n$ information bits. The first check bit is the parity check of the first n bits, and the second check bit is the parity check of the second n bits.
(a) Characterize the error patterns that can be detected by this code.
(b) Find the error-detection failure probability in terms of the error-detection probability of the single parity check code.
(c) Does it help to add a third parity check bit that is the sum of all the information bits?

3.60. Let $g(x) = x^3 + x + 1$. Consider the information sequence 1001.
(a) Find the codeword corresponding to the preceding information sequence.
(b) Suppose that the codeword has a transmission error in the first bit. What does the receiver obtain when it does its error checking?

3.61. ATM uses an eight-bit CRC on the information contained in the header. The header has six fields:
First 4 bits: GFC field
Next 8 bits: VPI field
Next 16 bits: VCI field
Next 3 bits: Type field
Next 1 bit: CLP field
Next 8 bits: CRC
(a) The CRC is calculated using the following generator polynomial: $x^8 + x^2 + x + 1$. Find the CRC bits if the GFC, VPI, Type, and CLP fields are all zero and the VCI field is 00000000 00001111. Assume the GFC bits correspond to the highest-order bits in the polynomial.
(b) Can this code detect single errors? Explain why.
(c) Draw the shift register division circuit for this generator polynomial.

3.62. Suppose a header consists of four 16-bit words: (11111111 11111111, 11111111 00000000, 11110000 11110000, 11000000 11000000). Find the Internet checksum for this code.

3.63. Let $g_1(x) = x + 1$ and let $g_2(x) = x^3 + x^2 + 1$. Consider the information bits (1,1,0,1,1,0).
(a) Find the codeword corresponding to these information bits if $g_1(x)$ is used as the generating polynomial.

(b) Find the codeword corresponding to these information bits if $g_2(x)$ is used as the generating polynomial.
(c) Can $g_2(x)$ detect single errors? double errors? triple errors? If not, give an example of an error pattern that cannot be detected.
(d) Find the codeword corresponding to these information bits if $g(x) = g_1(x)g_2(x)$ is used as the generating polynomial. Comment on the error-detecting capabilities of $g(x)$.

3.64. Take any binary polynomial of degree 7 that has an even number of nonzero coefficients. Show by longhand division that the polynomial is divisible by $x + 1$.

3.65. A repetition code is an $(n, 1)$ code in which the $n - 1$ parity bits are repetitions of the information bit. Is the repetition code a linear code? What is the minimum distance of the code?

3.66. A transmitter takes K groups of k information bits and appends a single parity bit to each group. The transmitter then appends a block parity check word in which the jth bit in the check word is the modulo 2 sum of the jth components in the K codewords.
(a) Explain why this code is a $((K + 1)(k + 1), Kk)$ linear code.
(b) Write the codeword as a $(k + 1)$ row by $(K + 1)$ column array in which the first K columns are the codewords and the last column is the block parity check. Use this array to show how the code can detect all single, double, and triple errors. Give an example of a quadruple error that cannot be detected.
(c) Find the minimum distance of the code. Can it correct all single errors? If so, show how the decoding can be done.
(d) Find the probability of error-detection failure for the random bit error channel.

3.67. Consider the $m = 4$ Hamming code.
(a) What is n, and what is k for this code?
(b) Find the parity check matrix for this code.
(c) Give the set of linear equations for computing the check bits in terms of the information bits.
(d) Write a program to find the set of all codewords. Do you notice anything peculiar about the weights of the codewords?
(e) If the information is produced at a rate of 1 Gbps, what is the bit rate of the encoded sequence?
(f) What is the bit error rate improvement if the code is used in a channel with $p = 10^{-4}$? $p = 10^{-10}$?

3.68. Show that an easy way to find the minimum distance is to find the minimum number of columns of $\mathbf{H}$ whose sum gives the zero vector.

3.69. Suppose we take the (7,4) Hamming code and obtain an (8,4) code by adding an overall parity check bit.
(a) Find the $\mathbf{H}$ matrix for this code.
(b) What is the minimum distance?
(c) Does the extra check bit increase the error-correction capability? the error-detection capability?

3.70. A (7,3) linear code has check bits given by

$$b_4 = b_1 + b_2$$
$$b_5 = b_1 + b_3$$
$$b_6 = b_2 + b_3$$
$$b_7 = b_1 + b_2 + b_3$$

(a) Find the **H** matrix.
(b) Find the minimum distance.
(c) Find the set of all codewords. Do you notice anything peculiar about the set of codewords.

3.71. An error-detecting code takes k information bits and generates a codeword with $2k + 1$ encoded bits as follows:
The first k bits consist of the information bits.
The next k bits repeat the information bits.
The next bit is the XOR of the first k bits.
(a) Find the check matrix for this code.
(b) What is the minimum distance of this code?
(c) Suppose the code is used on a channel that introduces independent random bit errors with probability 10^{-3}. Estimate the probability that the code fails to detect an erroneous transmission.

3.72. A (6,3) linear code has check bits given by

$$b_4 = b_1 + b_2$$
$$b_5 = b_1 + b_3$$
$$b_6 = b_2 + b_3$$

(a) Find the check matrix for this code.
(b) What is the minimum distance of this code?
(c) Find the set of all codewords.

3.73. (Appendix 3A). Consider an asynchronous transmission system that transfers N data bits between a start bit and a stop bit. What is the maximum value of N if the receiver clock frequency is within 1 percent of the transmitter clock frequency?

APPENDIX 3A: ASYNCHRONOUS DATA TRANSMISSION

The Recommended Standard (RS) 232, better known as the serial line interface, typically provides a communication channel between a computer and a device such as a modem. RS-232 is an Electronic Industries Association (EIA) standard that specifies the interface between data terminal equipment (DTE) and data communications equipment (DCE) for the purpose of transferring serial data. Typically, DTE represents a computer or a terminal, and DCE represents a modem. CCITT recommended a similar standard called V.24.

RS-232 specifies the connectors, various electrical signals, and transmission procedures. The connectors have 9 or 25 pins, referred to as DB-9 or DB-25, respectively. The D-type connector contains two rows of pins. From the front view of a DB-25 connector,

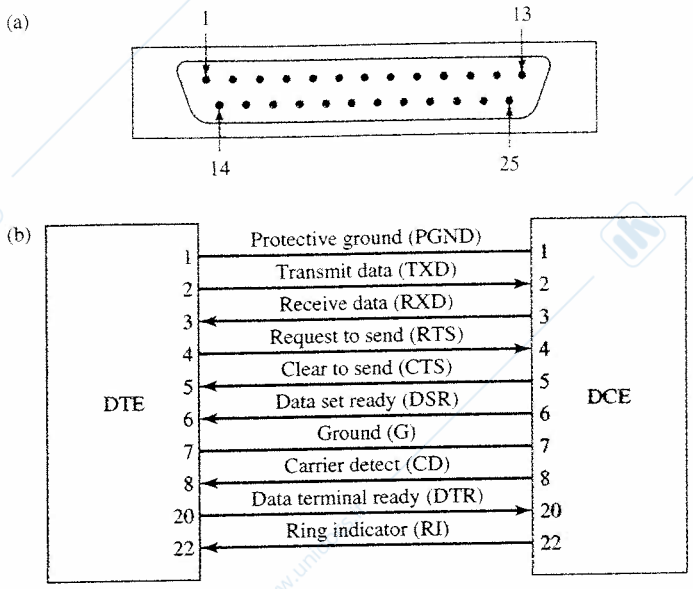


FIGURE 3.78 Commonly used pins in DB-25 connector.

the pins at the top row are numbered from 1 to 13, and the pins at the bottom row are numbered from 14 to 25. Figure 3.78a shows a typical 25-pin connector.

The electrical specification defines the signals associated with connector pins. Polar NRZ with a voltage between +3 to +25 volts is interpreted to be a binary 0, and −3 to −25 volts a binary 1. Figure 3.78b shows the functional description of commonly used signals. DTR is used by the DTE to tell the DCE that the DTE is on. DSR is used by the DCE to tell the DTE that the DCE is also on. When the DCE detects a carrier indicating that the channel is good, the DCE asserts the CD pin. If there is an incoming call, the DCE notifies the DTE via the RI signal. The DTE asserts the RTS pin if the DTE wants to send data. The DCE asserts the CTS pin if the DCE is ready to receive data. Finally, data is transmitted in full-duplex mode, from DTE to DCE on the TXD line and from DCE to DTE on the RXD line.

In RS-232, data transmission is said to be *asynchronous* because the receiver clock is free-running and not synchronized to the transmitter clock. It is easy to see that even if both clocks operate at nearly the same frequencies, the receiver will eventually sample the transmitter bit stream incorrectly due to slippage. The solution is to transmit data bits in short blocks with each block delimited by a start bit at the beginning and a stop bit at the end of a block. The short block ensures that slippage will not occur before the end of the block. Figure 3.79 illustrates the asynchronous transmission process. When the receiver detects the leading edge of the start bit, the receiver begins sampling the data bits after 1.5 periods of the receiver clock to ensure that sampling starts near the middle of the first data bit and slippage will not occur at the subsequent data bits. Typically a block consists of a start bit, a character of seven or eight data bits, and a stop bit. A parity bit can also be optionally added to enable the receiver to check the integrity of the data bits.

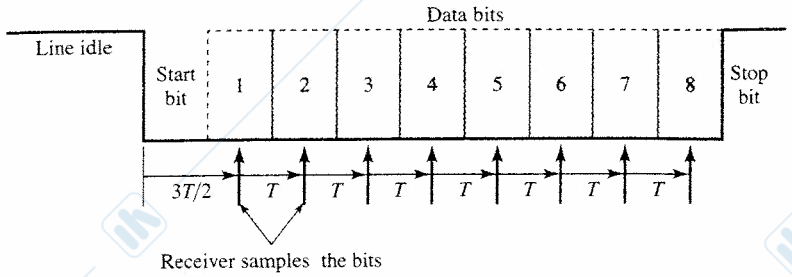


FIGURE 3.79 Framing and synchronization in asynchronous transmission.

Suppose that the transmitter pulse duration is X and the receiver pulse duration is T. If the receiver clock is slower than the transmitter clock and the last sample must occur before the end of the stop bit, then we must have $9.5T < 10X$. If the receiver clock is faster than the transmitter clock and the last sample must occur after the beginning of the stop bit, then we must have $9.5T > 9X$. These two inequalities can be satisfied if $|(T - X)/X| < 5.3$ percent. In other words, the receiver clock frequency must be within 5.3 percent of the transmitter clock frequency.

APPENDIX 3B: FOURIER SERIES

Let $x(t)$ represent a periodic signal with period T. The Fourier series resolves this signal into an infinite sum of sine and cosine terms

$$x(t) = a_0 + 2\sum_{n=1}^{\infty}\left[a_n \cos\left(\frac{2\pi nt}{T}\right) + b_n \sin\left(\frac{2\pi nt}{T}\right)\right] \tag{3B.1}$$

where the coefficients a_n and b_n represent the amplitude of the cosine and sine terms, respectively. The quantity n/T represents the nth harmonic of the fundamental frequency $f_0 = 1/T$.

The coefficient a_0 is given by the time average of the signal over one period

$$a_0 = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\,dt \tag{3B.2}$$

which is simply the time average of $x(t)$ over one period.

The coefficient a_n is obtained by multiplying both sides of Equation (3B.1) by the cosine function $\cos(2\pi nt/T)$ and integrating over the interval $-T/2$ to $T/2$. Using Equations (3B.1) and (3B.2) we obtain

$$a_n = \frac{1}{T}\int_{-T/2}^{T/2} x(t)\cos\left(\frac{2\pi nt}{T}\right)dt, \qquad n = 1, 2, \ldots \tag{3B.3}$$

The coefficient b_n of the sinusoid components is obtained in a similar manner:

$$b_n = \frac{2}{T}\int_{-T/2}^{T/2} x(t)\sin\left(\frac{2\pi nt}{T}\right)dt, \qquad n = 1, 2, \ldots \tag{3B.4}$$

The following trigonometric identity

$$A\cos\mu + B\sin\mu = \sqrt{A^2 + B^2}\cos\left(\mu - \tan^{-1}\frac{B}{A}\right) \tag{3B.5}$$

allows us to rewrite equation (3B.1) as follows:

$$x(t) = a_0 + 2\sum_{n=1}^{\infty}\sqrt{a_n^2 + b_n^2}\cos\left(\frac{2\pi nt}{T} - \tan^{-1}\frac{b_n}{a_n}\right) = a_0 + 2\sum_{n=1}^{\infty}|c_n|\cos\left(\frac{2\pi nt}{T} + \theta_n\right) \tag{3B.6}$$

A periodic function $x(t)$ is said to have a **discrete spectrum** with components at the frequencies, $0, f_0, 2f_0, \ldots$. The magnitude of the discrete spectrum at the frequency component nf_0 is given by

$$|c_n| = \sqrt{a_n^2 + b_n^2} \tag{3B.7}$$

and the phase of the discrete spectrum at nf_0 is given by

$$\theta_n = -\tan^{-1}\frac{b_n}{a_n} \tag{3B.8}$$

APPENDIX 3C: SAMPLING THEOREM

Reliable recovery of analog signals from digital form requires a sampling rate greater than some minimum value. We now explain how the Nyquist sampling theorem comes about. Let $x(nT)$ be the sequence of samples that result from the sampling of the analog signal $x(t)$. Consider a sequence of very narrow pulses $\delta(t - nT)$ that are spaced T seconds apart and whose amplitudes are modulated by the sample values $x(nT)$ as shown in Figure 3.18.

$$y(t) = \sum_n x(nT)\delta(t - nT) \tag{3C.1}$$

Signal theory enables us to show that $y(t)$ has the spectrum in Figure 3.80a, where the spectrum of $x(t)$ is given by its Fourier transform:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}\,dt = \int_{-\infty}^{\infty} x(t)\cos 2\pi ft\,dt + j\int_{-\infty}^{\infty} x(t)\sin 2\pi ft\,dt \tag{3C.2}$$

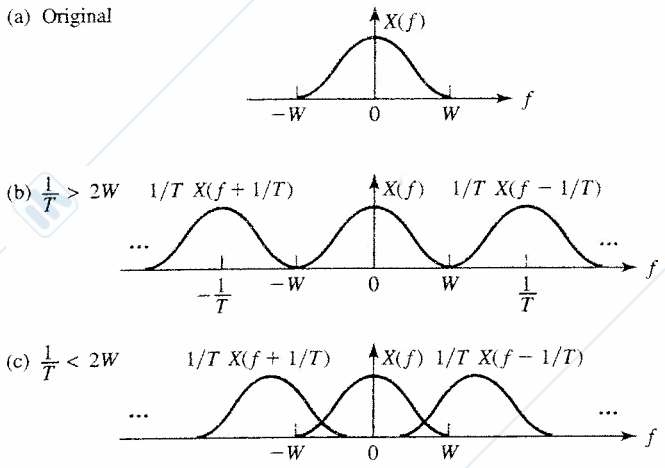


FIGURE 3.80 Spectrum of sampled signal: If the sampling rate is less than $2W$ then the original signal cannot be recovered.

The spectrum $X(f)$ is defined for positive and negative frequencies and is complex valued. The spectrum of the sampled signal is

$$Y(f) = \frac{1}{T}\sum_k X\left(f - \frac{k}{T}\right) \tag{3C.3}$$

The sampling theorem result depends on having these repeated versions of the spectrum be sufficiently apart. If the sampling rate $1/T$ is greater than $2W$, then the translated versions of $X(f)$ will be nonoverlapping. When a signal is applied to an ideal lowpass filter, the spectrum of the output signal consists of the portion of the spectrum of the input signal that falls in the range zero to W. Therefore, if we apply a low-pass filter to $y(t)$ as shown in Figure 3.80b, then we will recover the original exact spectrum $X(f)$ and hence $x(t)$. We conclude that the analog signal $x(t)$ can be recovered exactly from the sequence of its sample values as long as the sampling rate is $2W$ samples/second.

Now consider the case where the sampling rate $1/T$ is less than $2W$. The repeated versions of $X(f)$ now overlap, and we cannot recover $x(t)$ precisely (see Figure 3.80c). If we were to apply $y(t)$ to a low-pass filter in this case, the output would include an *aliasing error* that results from the additional energy that was introduced by the tails of the adjacent signals. In practice, signals are not strictly bandlimited, and so measures must be taken to control aliasing errors. In particular, signals are frequently passed through a low-pass filter prior to sampling to ensure that their energy is confined to the bandwidth that is assumed in the sampling rate.