1 Introduction to digital communication

Communication has been one of the deepest needs of the human race throughout recorded history. It is essential to forming social unions, to educating the young, and to expressing a myriad of emotions and needs. Good communication is central to a civilized society.

The various communication disciplines in engineering have the purpose of providing technological aids to human communication. One could view the smoke signals and drum rolls of primitive societies as being technological aids to communication, but communication technology as we view it today became important with telegraphy, then telephony, then video, then computer communication, and today the amazing mixture of all of these in inexpensive, small portable devices.

Initially these technologies were developed as separate networks and were viewed as having little in common. As these networks grew, however, the fact that all parts of a given network had to work together, coupled with the fact that different components were developed at different times using different design methodologies, caused an increased focus on the underlying principles and architectural understanding required for continued system evolution.

This need for basic principles was probably best understood at American Telephone and Telegraph (AT&T), where Bell Laboratories was created as the research and development arm of AT&T. The Math Center at Bell Labs became the predominant center for communication research in the world, and held that position until quite recently. The central core of the principles of communication technology were developed at that center.

Perhaps the greatest contribution from the Math Center was the creation of Information Theory [27] by Claude Shannon (Shannon, 1948). For perhaps the first 25 years of its existence, Information Theory was regarded as a beautiful theory but not as a central guide to the architecture and design of communication systems. After that time, however, both the device technology and the engineering understanding of the theory were sufficient to enable system development to follow information theoretic principles.

A number of information theoretic ideas and how they affect communication system design will be explained carefully in subsequent chapters. One pair of ideas, however, is central to almost every topic. The first is to view all communication sources, e.g., speech waveforms, image waveforms, and text files, as being representable by binary sequences. The second is to design communication systems that first convert the

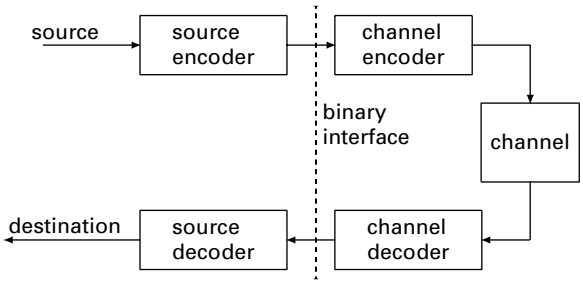

Figure 1.1. Placing a binary interface between source and channel. The source encoder converts the source output to a binary sequence and the channel encoder (often called a modulator) processes the binary sequence for transmission over the channel. The channel decoder (demodulator) recreates the incoming binary sequence (hopefully reliably), and the source decoder recreates the source output.

source output into a binary sequence and then convert that binary sequence into a form suitable for transmission over particular physical media such as cable, twisted wire pair, optical fiber, or electromagnetic radiation through space.

Digital communication systems, by definition, are communication systems that use such a digital[1] sequence as an interface between the source and the channel input (and similarly between the channel output and final destination) (see Figure 1.1).

The idea of converting an analog source output to a binary sequence was quite revolutionary in 1948, and the notion that this should be done before channel processing was even more revolutionary. Today, with digital cameras, digital video, digital voice, etc., the idea of digitizing any kind of source is commonplace even among the most technophobic. The notion of a binary interface before channel transmission is almost as commonplace. For example, we all refer to the speed of our Internet connection in bits per second.

There are a number of reasons why communication systems now usually contain a binary interface between source and channel (i.e., why digital communication systems are now standard). These will be explained with the necessary qualifications later, but briefly they are as follows.

- Digital hardware has become so cheap, reliable, and miniaturized that digital interfaces are eminently practical.
- A standardized binary interface between source and channel simplifies implementation and understanding, since source coding/decoding can be done independently of the channel, and, similarly, channel coding/decoding can be done independently of the source.

[1] A digital sequence is a sequence made up of elements from a finite alphabet (e.g. the binary digits $\{0, 1\}$, the decimal digits $\{0, 1, \ldots, 9\}$, or the letters of the English alphabet). The binary digits are almost universally used for digital communication and storage, so we only distinguish digital from binary in those few places where the difference is significant.

- A standardized binary interface between source and channel simplifies networking, which now reduces to sending binary sequences through the network.
- One of the most important of Shannon's information theoretic results is that if a source can be transmitted over a channel in any way at all, it can be transmitted using a binary interface between source and channel. This is known as the *source/channel separation theorem*.

In the remainder of this chapter, the problems of source coding and decoding and channel coding and decoding are briefly introduced. First, however, the notion of layering in a communication system is introduced. One particularly important example of layering was introduced in Figure 1.1, where source coding and decoding are viewed as one layer and channel coding and decoding are viewed as another layer.

1.1 Standardized interfaces and layering

Large communication systems such as the Public Switched Telephone Network (PSTN) and the Internet have incredible complexity, made up of an enormous variety of equipment made by different manufacturers at different times following different design principles. Such complex networks need to be based on some simple architectural principles in order to be understood, managed, and maintained. Two such fundamental architectural principles are *standardized interfaces* and *layering*.

A standardized interface allows the user or equipment on one side of the interface to ignore all details about the other side of the interface except for certain specified interface characteristics. For example, the binary interface[2] in Figure 1.1 allows the source coding/decoding to be done independently of the channel coding/decoding.

The idea of layering in communication systems is to break up communication functions into a string of separate layers, as illustrated in Figure 1.2.

Each layer consists of an input module at the input end of a communcation system and a "peer" output module at the other end. The input module at layer i processes the information received from layer $i+1$ and sends the processed information on to layer $i-1$. The peer output module at layer i works in the opposite direction, processing the received information from layer $i-1$ and sending it on to layer i.

As an example, an input module might receive a voice waveform from the next higher layer and convert the waveform into a binary data sequence that is passed on to the next lower layer. The output peer module would receive a binary sequence from the next lower layer at the output and convert it back to a speech waveform.

As another example, a *modem* consists of an input module (a modulator) and an output module (a demodulator). The modulator receives a binary sequence from the next higher input layer and generates a corresponding modulated waveform for transmission over a channel. The peer module is the remote demodulator at the other end of the channel. It receives a more or less faithful replica of the transmitted

[2] The use of a binary sequence at the interface is not quite enough to specify it, as will be discussed later.

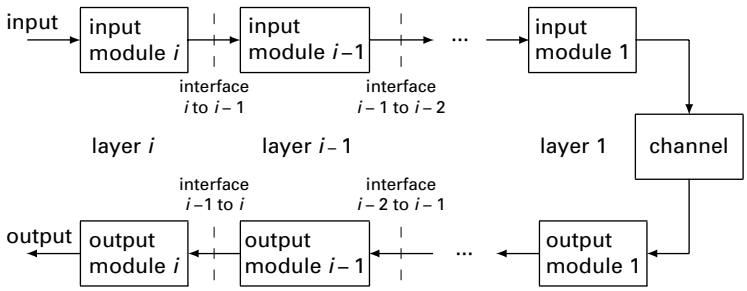

Figure 1.2. Layers and interfaces. The specification of the interface between layers i and $i-1$ should specify how input module i communicates with input module $i-1$, how the corresponding output modules communicate, and, most important, the input/output behavior of the system to the right of the interface. The designer of layer $i-1$ uses the input/output behavior of the layers to the right of $i-1$ to produce the required input/output performance to the right of layer i. Later examples will show how this multilayer process can simplify the overall system design.

waveform and reconstructs a typically faithful replica of the binary sequence. Similarly, the local demodulator is the peer to a remote modulator (often collocated with the remote demodulator above). Thus a modem is an input module for communication in one direction and an output module for independent communication in the opposite direction. Later chapters consider modems in much greater depth, including how noise affects the channel waveform and how that affects the reliability of the recovered binary sequence at the output. For now, however, it is enough simply to view the modulator as converting a binary sequence to a waveform, with the peer demodulator converting the waveform back to the binary sequence.

As another example, the source coding/decoding layer for a waveform source can be split into three layers, as shown in Figure 1.3. One of the advantages of this layering is that discrete sources are an important topic in their own right (discussed in Chapter 2) and correspond to the inner layer of Figure 1.3. Quantization is also an important topic in its own right (discussed in Chapter 3). After both of these are understood, waveform sources become quite simple to understand.

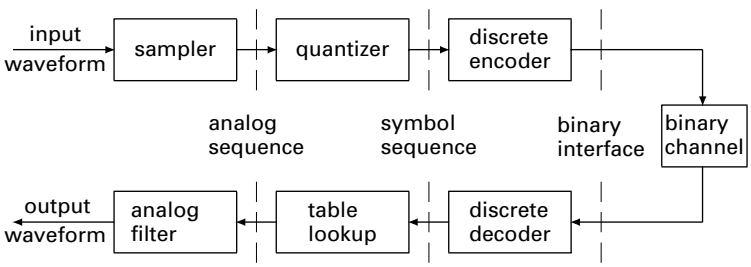

Figure 1.3. Breaking the source coding/decoding layer into three layers for a waveform source. The input side of the outermost layer converts the waveform into a sequence of samples and the output side converts the recovered samples back to the waveform. The quantizer then converts each sample into one of a finite set of symbols, and the peer module recreates the sample (with some distortion). Finally the inner layer encodes the sequence of symbols into binary digits.

The channel coding/decoding layer can also be split into several layers, but there are a number of ways to do this which will be discussed later. For example, binary error-correction coding/decoding can be used as an outer layer with modulation and demodulation as an inner layer, but it will be seen later that there are a number of advantages in combining these layers into what is called coded modulation.[3] Even here, however, layering is important, but the layers are defined differently for different purposes.

It should be emphasized that layering is much more than simply breaking a system into components. The input and peer output in each layer encapsulate all the lower layers, and all these lower layers can be viewed in aggregate as a communication channel. Similarly, the higher layers can be viewed in aggregate as a simple source and destination.

The above discussion of layering implicitly assumed a point-to-point communication system with one source, one channel, and one destination. Network situations can be considerably more complex. With broadcasting, an input module at one layer may have multiple peer output modules. Similarly, in multiaccess communication a multiplicity of input modules have a single peer output module. It is also possible in network situations for a single module at one level to interface with multiple modules at the next lower layer or the next higher layer. The use of layering is at least as important for networks as it is for point-to-point communications systems. The physical layer for networks is essentially the channel encoding/decoding layer discussed here, but textbooks on networks rarely discuss these physical layer issues in depth. The network control issues at other layers are largely separable from the physical layer communication issues stressed here. The reader is referred to Bertsekas and Gallager (1992), for example, for a treatment of these control issues.

The following three sections provide a fuller discussion of the components of Figure 1.1, i.e. of the fundamental two layers (source coding/decoding and channel coding/decoding) of a point-to-point digital communication system, and finally of the interface between them.

1.2 Communication sources

The source might be discrete, i.e. it might produce a sequence of discrete symbols, such as letters from the English or Chinese alphabet, binary symbols from a computer file, etc. Alternatively, the source might produce an analog waveform, such as a voice signal from a microphone, the output of a sensor, a video waveform, etc. Or, it might be a sequence of images such as X-rays, photographs, etc.

Whatever the nature of the source, the output from the source will be modeled as a sample function of a random process. It is not obvious why the inputs to communication

[3] Terminology is nonstandard here. A channel coder (including both coding and modulation) is often referred to (both here and elsewhere) as a modulator. It is also often referred to as a modem, although a modem is really a device that contains both modulator for communication in one direction and demodulator for communication in the other.

systems should be modeled as random, and in fact this was not appreciated before Shannon developed information theory in 1948.

The study of communication before 1948 (and much of it well after 1948) was based on Fourier analysis; basically one studied the effect of passing sine waves through various kinds of systems and components and viewed the source signal as a superposition of sine waves. Our study of channels will begin with this kind of analysis (often called Nyquist theory) to develop basic results about sampling, intersymbol interference, and bandwidth.

Shannon's view, however, was that if the recipient knows that a sine wave of a given frequency is to be communicated, why not simply regenerate it at the output rather than send it over a long distance? Or, if the recipient knows that a sine wave of unknown frequency is to be communicated, why not simply send the frequency rather than the entire waveform?

The essence of Shannon's viewpoint is that the set of possible source outputs, rather than any particular output, is of primary interest. The reason is that the communication system must be designed to communicate whichever one of these possible source outputs actually occurs. The objective of the communication system then is to transform each possible source output into a transmitted signal in such a way that these possible transmitted signals can be best distinguished at the channel output. A probability measure is needed on this set of possible source outputs to distinguish the typical from the atypical. This point of view drives the discussion of all components of communication systems throughout this text.

1.2.1 Source coding

The source encoder in Figure 1.1 has the function of converting the input from its original form into a sequence of bits. As discussed before, the major reasons for this almost universal conversion to a bit sequence are as follows: inexpensive digital hardware, standardized interfaces, layering, and the source/channel separation theorem.

The simplest source coding techniques apply to discrete sources and simply involve representing each successive source symbol by a sequence of binary digits. For example, letters from the 27-symbol English alphabet (including a SPACE symbol) may be encoded into 5-bit blocks. Since there are 32 distinct 5-bit blocks, each letter may be mapped into a distinct 5-bit block with a few blocks left over for control or other symbols. Similarly, upper-case letters, lower-case letters, and a great many special symbols may be converted into 8-bit blocks ("bytes") using the standard ASCII code.

Chapter 2 treats coding for discrete sources and generalizes the above techniques in many ways. For example, the input symbols might first be segmented into m-tuples, which are then mapped into blocks of binary digits. More generally, the blocks of binary digits can be generalized into variable-length sequences of binary digits. We shall find that any given discrete source, characterized by its alphabet and probabilistic description, has a quantity called *entropy* associated with it. Shannon showed that this source entropy is equal to the minimum number of binary digits per source symbol

required to map the source output into binary digits in such a way that the source symbols may be retrieved from the encoded sequence.

Some discrete sources generate finite segments of symbols, such as email messages, that are statistically unrelated to other finite segments that might be generated at other times. Other discrete sources, such as the output from a digital sensor, generate a virtually unending sequence of symbols with a given statistical characterization. The simpler models of Chapter 2 will correspond to the latter type of source, but the discussion of universal source coding in Section 2.9 is sufficiently general to cover both types of sources and virtually any other kind of source.

The most straightforward approach to analog source coding is called analog to digital (A/D) conversion. The source waveform is first sampled at a sufficiently high rate (called the "Nyquist rate"). Each sample is then quantized sufficiently finely for adequate reproduction. For example, in standard voice telephony, the voice waveform is sampled 8000 times per second; each sample is then quantized into one of 256 levels and represented by an 8-bit byte. This yields a source coding bit rate of 64 kilobits per second (kbps).

Beyond the basic objective of conversion to bits, the source encoder often has the further objective of doing this as efficiently as possible – i.e. transmitting as few bits as possible, subject to the need to reconstruct the input adequately at the output. In this case source encoding is often called data compression. For example, modern speech coders can encode telephone-quality speech at bit rates of the order of 6–16 kbps rather than 64 kbps.

The problems of sampling and quantization are largely separable. Chapter 3 develops the basic principles of quantization. As with discrete source coding, it is possible to quantize each sample separately, but it is frequently preferable to segment the samples into blocks of n and then quantize the resulting n-tuples. As will be shown later, it is also often preferable to view the quantizer output as a discrete source output and then to use the principles of Chapter 2 to encode the quantized symbols. This is another example of layering.

Sampling is one of the topics in Chapter 4. The purpose of sampling is to convert the analog source into a sequence of real-valued numbers, i.e. into a discrete-time, analog-amplitude source. There are many other ways, beyond sampling, of converting an analog source to a discrete-time source. A general approach, which includes sampling as a special case, is to expand the source waveform into an orthonormal expansion and use the coefficients of that expansion to represent the source output. The theory of orthonormal expansions is a major topic of Chapter 4. It forms the basis for the signal space approach to channel encoding/decoding. Thus Chapter 4 provides us with the basis for dealing with waveforms for both sources and channels.

1.3 Communication channels

Next we discuss the channel and channel coding in a generic digital communication system.

In general, a channel is viewed as that part of the communication system between source and destination that is given and not under the control of the designer. Thus, to a source-code designer, the channel might be a digital channel with binary input and output; to a telephone-line modem designer, it might be a 4 kHz voice channel; to a cable modem designer, it might be a physical coaxial cable of up to a certain length, with certain bandwidth restrictions.

When the channel is taken to be the physical medium, the amplifiers, antennas, lasers, etc. that couple the encoded waveform to the physical medium might be regarded as part of the channel or as as part of the channel encoder. It is more common to view these coupling devices as part of the channel, since their design is quite separable from that of the rest of the channel encoder. This, of course, is another example of layering.

Channel encoding and decoding when the channel is the physical medium (either with or without amplifiers, antennas, lasers, etc.) is usually called *(digital) modulation* and *demodulation*, respectively. The terminology comes from the days of analog communication where modulation referred to the process of combining a lowpass signal waveform with a high-frequency sinusoid, thus placing the signal waveform in a frequency band appropriate for transmission and regulatory requirements. The analog signal waveform could modulate the amplitude, frequency, or phase, for example, of the sinusoid, but, in any case, the original waveform (in the absence of noise) could be retrieved by demodulation.

As digital communication has increasingly replaced analog communication, the modulation/demodulation terminology has remained, but now refers to the entire process of digital encoding and decoding. In most cases, the binary sequence is first converted to a baseband waveform and the resulting baseband waveform is converted to bandpass by the same type of procedure used for analog modulation. As will be seen, the challenging part of this problem is the conversion of binary data to baseband waveforms. Nonetheless, this entire process will be referred to as modulation and demodulation, and the conversion of baseband to passband and back will be referred to as frequency conversion.

As in the study of any type of system, a channel is usually viewed in terms of its possible inputs, its possible outputs, and a description of how the input affects the output. This description is usually probabilistic. If a channel were simply a linear time-invariant system (e.g. a filter), it could be completely characterized by its impulse response or frequency response. However, the channels here (and channels in practice) always have an extra ingredient – noise.

Suppose that there were no noise and a single input voltage level could be communicated exactly. Then, representing that voltage level by its infinite binary expansion, it would be possible in principle to transmit an infinite number of binary digits by transmitting a single real number. This is ridiculous in practice, of course, precisely because noise limits the number of bits that can be reliably distinguished. Again, it was Shannon, in 1948, who realized that noise provides the fundamental limitation to performance in communication systems.

The most common channel model involves a waveform input $X(t)$, an added noise waveform $Z(t)$, and a waveform output $Y(t) = X(t) + Z(t)$ that is the sum of the input

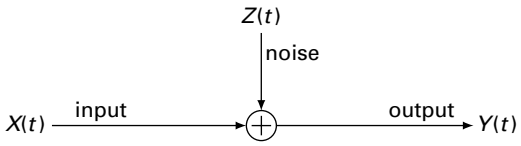

Figure 1.4. Additive white Gaussian noise (AWGN) channel.

and the noise, as shown in Figure 1.4. Each of these waveforms are viewed as random processes. Random processes are studied in Chapter 7, but for now they can be viewed intuitively as waveforms selected in some probabilitistic way. The noise $Z(t)$ is often modeled as white Gaussian noise (also to be studied and explained later). The input is usually constrained in power and bandwidth.

Observe that for any channel with input $X(t)$ and output $Y(t)$, the noise could be defined to be $Z(t) = Y(t) - X(t)$. Thus there must be something more to an additive-noise channel model than what is expressed in Figure 1.4. The additional required ingredient for noise to be called additive is that its probabilistic characterization does not depend on the input.

In a somewhat more general model, called a *linear Gaussian channel*, the input waveform $X(t)$ is first filtered in a linear filter with impulse response $h(t)$, and then independent white Gaussian noise $Z(t)$ is added, as shown in Figure 1.5, so that the channel output is given by

$$Y(t) = X(t) * h(t) + Z(t),$$

where "$*$" denotes convolution. Note that Y at time t is a function of X over a range of times, i.e.

$$Y(t) = \int_{-\infty}^{\infty} X(t-\tau)h(\tau)\mathrm{d}\tau + Z(t).$$

The linear Gaussian channel is often a good model for wireline communication and for line-of-sight wireless communication. When engineers, journals, or texts fail to describe the channel of interest, this model is a good bet.

The linear Gaussian channel is a rather poor model for non-line-of-sight mobile communication. Here, multiple paths usually exist from source to destination. Mobility of the source, destination, or reflecting bodies can cause these paths to change in time in a way best modeled as random. A better model for mobile communication is to

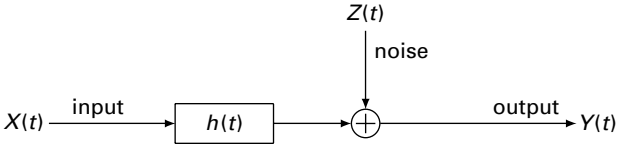

Figure 1.5. Linear Gaussian channel model.

replace the time-invariant filter $h(t)$ in Figure 1.5 by a randomly time varying linear filter, $H(t, \tau)$, that represents the multiple paths as they change in time. Here the output is given by

$$Y(t) = \int_{-\infty}^{\infty} X(t-u)H(u,t)\mathrm{d}u + Z(t).$$

These randomly varying channels will be studied in Chapter 9.

1.3.1 Channel encoding (modulation)

The channel encoder box in Figure 1.1 has the function of mapping the binary sequence at the source/channel interface into a channel waveform. A particularly simple approach to this is called binary pulse amplitude modulation (2-PAM). Let $\{u_1, u_2, \ldots, \}$ denote the incoming binary sequence, and let each $u_n = \pm 1$ (rather than the traditional 0/1). Let $p(t)$ be a given elementary waveform such as a rectangular pulse or a $\sin(\omega t)/\omega t$ function. Assuming that the binary digits enter at R bps, the sequence $u_1, u_2, \ldots$ is mapped into the waveform $\sum_n u_n p(t - n/R)$.

Even with this trivially simple modulation scheme, there are a number of interesting questions, such as how to choose the elementary waveform $p(t)$ so as to satisfy frequency constraints and reliably detect the binary digits from the received waveform in the presence of noise and intersymbol interference.

Chapter 6 develops the principles of modulation and demodulation. The simple 2-PAM scheme is generalized in many ways. For example, multilevel modulation first segments the incoming bits into m-tuples. There are $M = 2^m$ distinct m-tuples, and in M-PAM, each m-tuple is mapped into a different numerical value (such as $\pm 1, \pm 3, \pm 5, \pm 7$ for $M = 8$). The sequence $u_1, u_2, \ldots$ of these values is then mapped into the waveform $\sum_n u_n p(t - mn/R)$. Note that the rate at which pulses are sent is now m times smaller than before, but there are 2^m different values to be distinguished at the receiver for each elementary pulse.

The modulated waveform can also be a complex baseband waveform (which is then modulated up to an appropriate passband as a real waveform). In a scheme called quadrature amplitude modulation (QAM), the bit sequence is again segmented into m-tuples, but now there is a mapping from binary m-tuples to a set of $M = 2^m$ complex numbers. The sequence $u_1, u_2, \ldots$ of outputs from this mapping is then converted to the complex waveform $\sum_n u_n p(t - mn/R)$.

Finally, instead of using a fixed signal pulse $p(t)$ multiplied by a selection from M real or complex values, it is possible to choose M different signal pulses, $p_1(t), \ldots, p_M(t)$. This includes frequency shift keying, pulse position modulation, phase modulation, and a host of other strategies.

It is easy to think of many ways to map a sequence of binary digits into a waveform. We shall find that there is a simple geometric "signal-space" approach, based on the results of Chapter 4, for looking at these various combinations in an integrated way.

Because of the noise on the channel, the received waveform is different from the transmitted waveform. A major function of the demodulator is that of detection.