Chapter 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 in 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 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).

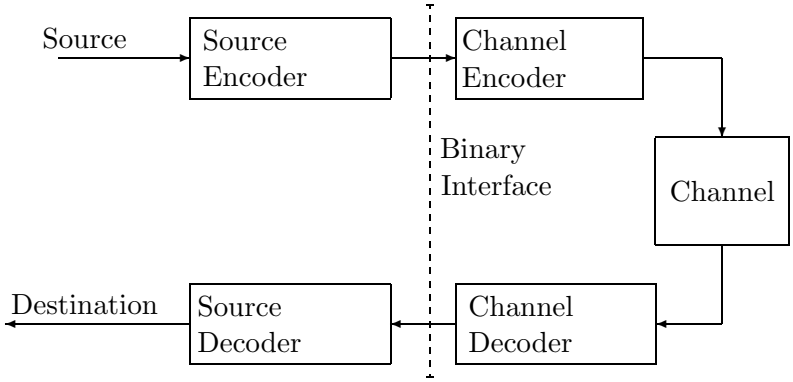

Figure 1.1: Placing a binary interface between source and channel. The source encoder converts the source output to a binary sequence and the channel encoder (often called a modulator) processes the binary sequence for transmission over the channel. The channel decoder (demodulator) recreates the incoming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

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. By 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 already 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] above 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 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

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

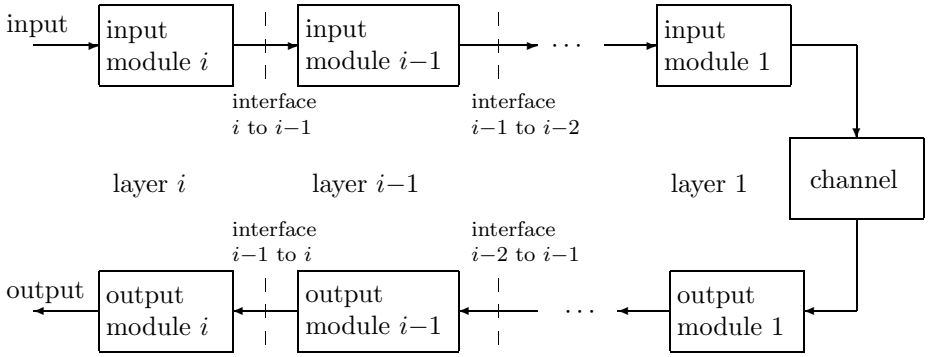

Figure 1.2: Layers and interfaces: The specification of the interface between layers i and $i-1$ should specify how input module i communicates with input module $i-1$, how the corresponding output modules communicate, and, most important, the input/output behavior of the system to the right of interface. 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 multi-layer process can simplify the overall system design.

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 to simply 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 3 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 (treated in Chapter 2) and correspond to the inner layer of Figure 1.3. Quantization is also an important topic in its own right, (treated in Chapter 3). After both of these are understood, waveform sources become quite simple to understand.

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

[3]Notation is nonstandard here. A channel coder (including both coding and modulation) is often referred to (both here and elsewhere) as a modulator.

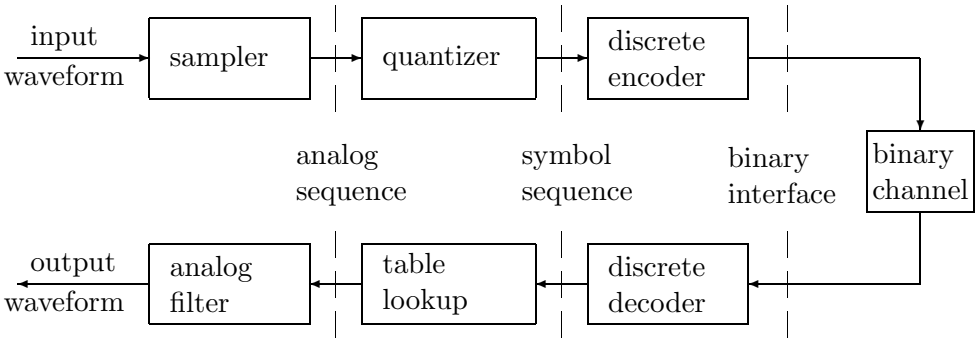

Figure 1.3: Breaking the source coding/decoding layer into 3 layers for a waveform source. The input side of the outermost layer converts the waveform into a sequence of samples and output side converts the recovered samples back to the waveform. The quantizer then converts each sample into one of a 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.

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 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 [1], for example, for a treatment of these control issues.

The following three sections give 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 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 succesive 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 yet, 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 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 kb/s rather than 64 kb/s.

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 n-tuples and then quantize the resulting n-tuples. As 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 both for sources and channels.

1.3 Communication channels

We next 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 such 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), then 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 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.

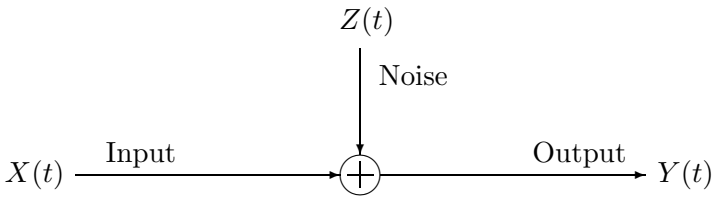

Figure 1.4: An additive white Gaussian noise (AWGN) channel.

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

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

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

i.e.,

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

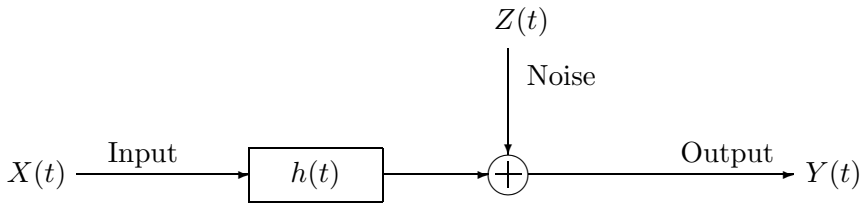

Figure 1.5: Linear Gaussian channel model.

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 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 $Y(t) = \int_{-\infty}^{\infty} X(t-u)H(u,t)du + Z(t)$. 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 $\{u_1, u_2, \dots, \}$ denote the incoming binary sequence, where each u_n is ± 1 (rather than the traditional 0/1). Let $p(t)$ be a given elementary waveform such as a rectangular pulse or a $\frac{sin(\omega t)}{\omega t}$ function. Assuming that the binary digits enter at R bits per second (bps), the sequence $u_1, u_2, \dots$ is mapped into the waveform $\sum_n u_n p(t - \frac{n}{R})$.

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, multi-level 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 $u_1, u_2, \dots$ of these values is then mapped into the waveform $\sum_n u_n p(t - \frac{mn}{R})$. 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 $\sum_n u_n p(t - \frac{mn}{R})$.

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. The detector attempts to choose which possible input sequence is most likely to have given rise to the given received waveform. Chapter 7 develops the background in random processes necessary to understand this problem, and Chapter 8 uses the geometric signal-space approach to analyze and understand the detection problem.

1.3.2 Error correction

Frequently the error probability incurred with simple modulation and demodulation techniques is too high. One possible solution is to separate the channel encoder into two layers, first an error-correcting code, and then a simple modulator.

As a very simple example, the bit rate into the channel encoder could be reduced by a factor of 3, and then each binary input could be repeated 3 times before entering the modulator. If at most one of the 3 binary digits coming out of the demodulator were incorrect, it could be corrected by majority rule at the decoder, thus reducing the error probability of the system at a considerable cost in data rate.

The scheme above (repetition encoding followed by majority-rule decoding) is a very simple example of error-correction coding. Unfortunately, with this scheme, small error probabilities are achieved only at the cost of very small transmission rates.

What Shannon showed was the very unintuitive fact that more sophisticated coding schemes can achieve arbitrarily low error probability at any data rate above a value known as the *channel capacity*. The channel capacity is a function of the probabilistic description of the output conditional on each possible input. Conversely, it is not possible to achieve low error probability at rates above the channel capacity. A brief proof of this *channel coding theorem* is given in Chapter 8, but readers should refer to texts on Information Theory such as [7] or [4]) for detailed coverage.

The channel capacity for a bandlimited additive white Gaussian noise channel is perhaps the most famous result in information theory. If the input power is limited to P, the bandwidth limited to W, and the noise power per unit bandwidth is N_0, then the capacity (in bits per second) is

$$C = \mathsf{W} \log_2 \left(1 + \frac{P}{N_0 \mathsf{W}}\right).$$

Only in the past few years have channel coding schemes been developed that can closely approach this channel capacity.

Early uses of error-correcting codes were usually part of a two-layer system similar to that above, where a digital error-correcting encoder is followed by a modulator. At the receiver, the waveform is first demodulated into a noisy version of the encoded sequence, and then this noisy version is decoded by the error-correcting decoder. Current practice frequently achieves better performance by combining error correction coding and modulation together in coded modulation schemes. Whether the error correction and traditional modulation are separate layers or combined, the combination is generally referred to as a modulator and a device that does this modulation on data in one direction and demodulation in the other direction is referred to as a modem.

The subject of error correction has grown over the last 50 years to the point where complex and lengthy textbooks are dedicated to this single topic (see, for example, [15] and [6].) This text provides only an introduction to error-correcting codes.

The final topic of the text is channel encoding and decoding for wireless channels. Considerable attention is paid here to modeling physical wireless media. Wireless channels are subject not only to additive noise but also random fluctuations in the strength of multiple paths between transmitter and receiver. The interaction of these paths causes fading, and we study how this affects coding, signal selection, modulation, and detection. Wireless communication is also used to discuss issues such as channel measurement, and how these measurements can be used at input and output. Finally there is a brief case study of CDMA (code division multiple access), which ties together many of the topics in the text.

1.4 Digital interface

The interface between the source coding layer and the channel coding layer is a sequence of bits. However, this simple characterization does not tell the whole story. The major complicating factors are as follows:

- Unequal rates: The rate at which bits leave the source encoder is often not perfectly matched to the rate at which bits enter the channel encoder.
- Errors: Source decoders are usually designed to decode an exact replica of the encoded sequence, but the channel decoder makes occasional errors.
- Networks: Encoded source outputs are often sent over networks, traveling serially over several channels; each channel in the network typically also carries the output from a number of different source encoders.

The first two factors above appear both in point-to-point communication systems and in networks. They are often treated in an ad hoc way in point-to-point systems, whereas they must be treated in a standardized way in networks. The third factor, of course, must also be treated in a standardized way in networks.

The usual approach to these problems in networks is to convert the superficially simple binary interface above into multiple layers as illustrated in Figure 1.6

How the layers in Figure 1.6 operate and work together is a central topic in the study of networks and is treated in detail in network texts such as [1]. These topics are not considered in detail here, except for the very brief introduction to follow and a few comments as needed later.

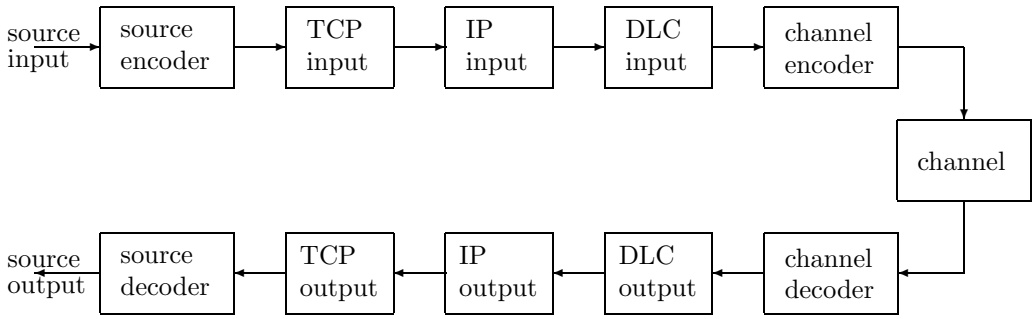

Figure 1.6: The replacement of the binary interface in Figure 1.6 with 3 layers in an oversimplified view of the internet: There is a TCP (transport control protocol) module associated with each source/destination pair; this is responsible for end-to-end error recovery and for slowing down the source when the network becomes congested. There is an IP (internet protocol) module associated with each node in the network; these modules work together to route data through the network and to reduce congestion. Finally there is a DLC (data link control) module associated with each channel; this accomplishes rate matching and error recovery on the channel. In network terminology, the channel, with its encoder and decoder, is called the *physical layer*.

1.4.1 Network aspects of the digital interface

The output of the source encoder is usually segmented into packets (and in many cases, such as email and data files, is already segmented in this way). Each of the network layers then adds some overhead to these packets, adding a header in the case of TCP (transmission control protocol) and IP (internet protocol) and adding both a header and trailer in the case of DLC (data link control). Thus what enters the channel encoder is a sequence of frames, where each frame has the structure illustrated in Figure 1.7.

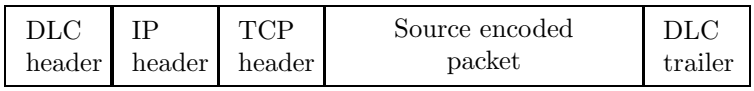

Figure 1.7: The structure of a data frame using the layers of Figure 1.6

.

These data frames, interspersed as needed by idle-fill, are strung together and the resulting bit stream enters the channel encoder at its synchronous bit rate. The header and trailer supplied by the DLC must contain the information needed for the receiving DLC to parse the received bit stream into frames and eliminate the idle-fill.

The DLC also provides protection against decoding errors made by the channel decoder. Typically this is done by using a set of 16 or 32 parity checks in the frame trailer. Each parity check specifies whether a given subset of bits in the frame contains an even or odd number of 1's. Thus if errors occur in transmission, it is highly likely that at least one of these parity checks will fail in the receiving DLC. This type of DLC is used on channels that permit transmission in both directions. Thus when an erroneous frame is detected, it is rejected and a frame in the opposite

direction requests a retransmission of the erroneous frame. Thus the DLC header must contain information about frames traveling in both directions. For details about such protocols, see, for example, [1].

An obvious question at this point is why error correction is typically done both at the physical layer and at the DLC layer. Also, why is feedback (*i.e.*, error detection and retransmission) used at the DLC layer and not at the physical layer? A partial answer is that if the error correction is omitted at one of the layers, the error probability is increased. At the same time, combining both procedures (with the same overall overhead) and using feedback at the physical layer can result in much smaller error probabilities. The two layer approach is typically used in practice because of standardization issues, but in very difficult communication situations, the combined approach can be preferable. From a tutorial standpoint, however, it is preferable to acquire a good understanding of channel encoding and decoding using transmission in only one direction before considering the added complications of feedback.

When the receiving DLC accepts a frame, it strips off the DLC header and trailer and the resulting packet enters the IP layer. In the IP layer, the address in the IP header is inspected to determine whether the packet is at its destination or must be forwarded through another channel. Thus the IP layer handles routing decisions, and also sometimes the decision to drop a packet if the queues at that node are too long.

When the packet finally reaches its destination, the IP layer strips off the IP header and passes the resulting packet with its TCP header to the TCP layer. The TCP module then goes through another error recovery phase[4] much like that in the DLC module and passes the accepted packets, without the TCP header, on to the destination decoder. The TCP and IP layers are also jointly responsible for congestion control, which ultimately requires the ability to either reduce the rate from sources as required or to simply drop sources that cannot be handled (witness dropped cell-phone calls).

In terms of sources and channels, these extra layers simply provide a sharper understanding of the digital interface between source and channel. That is, source encoding still maps the source output into a sequence of bits, and from the source viewpoint, all these layers can simply be viewed as a channel to send that bit sequence reliably to the destination.

In a similar way, the input to a channel is a sequence of bits at the channel's synchronous input rate. The output is the same sequence, somewhat delayed and with occasional errors.

Thus both source and channel have digital interfaces, and the fact that these are slightly different because of the layering is in fact an advantage. The source encoding can focus solely on minimizing the output bit rate (perhaps with distortion and delay constraints) but can ignore the physical channel or channels to be used in transmission. Similarly the channel encoding can ignore the source and focus solely on maximizing the transmission bit rate (perhaps with delay and error rate constraints).

[4]Even after all these layered attempts to prevent errors, occasional errors are inevitable. Some are caught by human intervention, many don't make any real difference, and a final few have consequences. C'est la vie. The purpose of communication engineers and network engineers is not to eliminate all errors, which is not possible, but rather to reduce their probability as much as practically possible.

1.5 Supplementary reading

An excellent text that treats much of the material here with more detailed coverage but less depth is Proakis [21]. Another good general text is Wilson [34]. The classic work that introduced the signal space point of view in digital communication is Wozencraft and Jacobs [35]. Good undergraduate treatments are provided in [22], [12], and [23].

Readers who lack the necessary background in probability should consult [2] or [24]. More advanced treatments of probability are given in [8] and [25]. Feller [5] still remains the classic text on probability for the serious student.

Further material on Information theory can be found, for example, in [7] and [4]. The original work by Shannon [27] is fascinating and surprisingly accessible.

The field of channel coding and decoding has become an important but specialized part of most communication systems. We introduce coding and decoding in Chapter 8, but a separate treatment is required to develop the subject in depth. At M.I.T., the text here is used for the first of a two term sequence and the second term uses a polished set of notes by D. Forney [6] available on the web. Alternatively, [15] is a good choice among many texts on coding and decoding.

Wireless communication is probably the major research topic in current digital communication work. Chapter 9 provides a substantial introduction to this topic, but a number of texts develop wireless communcation in much greater depth. Tse and Viswanath [32] and Goldsmith [9] are recommended and [33] is a good reference for spread spectrum techniques.