College of Computer Science
Northeastern University
COM 3510

Lecture 2
4 October 2001
Computer Networks: Theory, Modeling, and Analysis

Physical layer and error detection protocols

Handouts: Lecture Notes for Lecture 1.

Lecture Outline:

- Physical layer: signals, channels, and encoding.
- Error dectection.
- Introduction to automatic retransmission techniques.

1 Physical layer: Signals, channels, and encoding

Data can be transmitted in two forms, *analog* and *digital*. The transmission on both kinds of signals is in the form of electromagnetic waves. While an analog signal is a continuously varying electromagnetic wave, a digital signal is a sequence of high and low pulses (thus yielding "square waves") that naturally represent digital data. The data that we would like to send using these channels could also be digital (e.g., a set of bits representing text) or analog (e.g., voice, video, images not digitized).

Both analog and digital data can be transmitted as either analog signals or digital signals; in each case appropriate transformations need to be done to convert the data to the right format. Most long-haul telecommunication facilities are adopting digital signaling techniques. In this class, we will focus our attention on the transmission of digital data using digital signals.

We will begin by studying how transmission signals and channels can be characterized. Then, we will discuss the encoding of data into digital signals. Finally, we will consider the inherent capacity limitations of transmitting digital data.

Signals and channels. Any electromagnetic signal sent over a channel can be described as a function $x(t)$ of voltage or current over time. For concreteness, we adopt the convention here that $x(t)$ is the voltage of the signal at time t.

One characterization of communication channels is in the *frequency domain*. Suppose that a *sinusoidal* signal $x(t) = \cos(2\pi f t)$ that oscillates at frequency of f Hz is applied to a channel. The channel output usually consists of a sinusoidal signal of frequency f but of different amplitude and phase:

$$y(t) = A(f)\cos(2\pi f t + \phi(f)) = A(f)\cos(2\pi f(t - \tau(f))) \tag{1}$$

Thus a channel is characterized by the functions $A(f)$, the *amplitude response*, and $\phi(f)$, the *phase shift*. Alternatively, the channel can be specified by the functions $A(f)$ and the *delay* $\tau(f)$.

Equation 1 only characterizes how a channel affects a sinusoidal signal. How about other signals? Most signals of interest can be represented as a superposition of sinusoidal waves. Thus, for example a signal $x(t)$ may be expressed as

$$x(t) = \sum_f x_f \cos(2\pi f t) \tag{2}$$

where the summation is over a discrete collection of frequencies. Since the signal frequencies can range over the entire positive real numbers, the superposition usually needs to expressed as an integration, not a discrete sum. For mathematical convenience, in expressing the superposition as an integration, sinusoidal waves are written as $e^{i2\pi ft}$, where $i = \sqrt{-1}$. (Recall that $e^{ix} = \cos x + i \sin x$.) Thus, any signal $x(t)$ can be written as

$$x(t) = \int_{-\infty}^{\infty} X(f) e^{i2\pi ft} df, \tag{3}$$

where

$$X(f) = \int_{-\infty}^{\infty} x(t) e^{-i2\pi ft} dt.$$

The usage of negative frequencies is for mathematical convenience and is beyond the scope of this class. We can interpret Equation 3 as saying that $x(t)$ is a superposition of sinusoidal waves with frequencies ranging from $-\infty$ to ∞, where the contribution of the sinusoidal wave of frequency f is given by $X(f)$.

When a signal equal to a linear combination of sinusoidal signals is applied at the input of the channel, then a identically-weighted linear combination of the output sinusoidal signals given by Equation 1 is obtained. Thus, for example, if the signal given by Equation 2 is applied to a channel with amplitude-response $A(\cdot)$ and phase shift $\phi(\cdot)$, then the output signal is given by

$$y(t) = \sum_f A(f) \cos(2\pi ft + \phi(f)).$$

A square wave, viewed as a combination of sinusoidal waves, has components in an infinite range of frequencies and suffers from significant attenuation when sent directly on most communication channels. Consequently, the different symbols represented within a square wave input are difficult to extract from the corresponding output. Using signaling methods that incur zero intersymbol interference, however, digital signals can be transmitted such that the output signal can be sampled appropriately to yield the digital data.

Encoding. A natural "waveform representation" of any sequence of bits is a "square wave" in which we assign a high voltage (say, 5V) to bit 1 and a low voltage (say, -5V) to bit 0. Such an encoding is called *Non-Return to Zero* encoding (NRZ) and is illustrated in Figure 1.

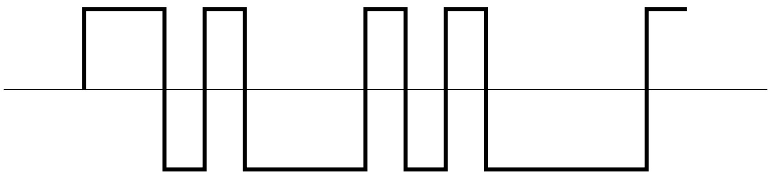

Figure 1: NRZ encoding for the string 110100010100001.

Other encoding formats include *differential encoding*, in which a 1 is encoded by changing the polarity (the sign of the voltage) while a 0 is encoded by maintaining the polarity of the signal. A third form of encoding is *Manchester encoding*, a 0 is transmitted as a +1 voltage or $T/2$ seconds followed by a -1 voltage for $T/2$ seconds (for a chosen time period T); and a 1 is transmitted as a -1 voltage for $T/2$ seconds followed by a +1 for $T/2$ seconds.

One aspect that encoding techniques need to address is that of bit synchronization between the sender and the receiver. The receiver needs to be able to read off the bits encoded by the sender in the signal in the right format. In particular, the receiver must be able to identify the start and end of the portion of the signal that encodes a particular bit. If the clocks at the two ends were exactly synchronized and the clock rates were truly identical, then bit synchronization is easy to achieve. Perfect clock synchronization is difficult to achieve, as a result of which a long continuous transmission of a high voltage signal (or a low voltage signal) may be difficult to decode. Problem 3 of Problem Set 1 discusses the bit synchronization problem in more detail.

Channel capacity and sampling. The capacity of a channel depends on the range of frequencies that it "admits". A channel does not admit signals of frequency f, if $A(f)$ is smaller than a relevant threshold. *Low-pass* channels essentially eliminate high-frequency components, while passing low-frequency components with little attenuation. *Band-pass* channels admit signals with frequencies in a range $[f_1, f_2]$. We refer to $f_2 - f_1$ as the *frequency spectrum* or *bandwidth* of the channel. Note that the usage of the term "bandwidth" is different than the usage in the traditional networking literature. We will soon see that the two are related (see Shannon's channel capacity theorem below).

We now consider the question: at what rate can we send a given sequence of 0s and 1s on a particular communication channel? A partial answer to this question is provided by the *Nyquist sampling theorem*, an approximate statement of which is given below.

Theorem 1 (Nyquist Sampling Theorem) *Any signal $s(t)$ that is a superposition of sinusoidal waves such that $S(f) = 0$ for $|f| > W$ (such signals are referred to as* low-pass *signals) is completely determined by its values at times $j/2W$, where j is an integer in the range $[-\infty, \infty]$.*

The gist of the Nyquist sampling theorem is the following. We can send new information on a channel with bandwidth W at a maximum rate of $2W$ "items" every second. Similarly, in order to read back information on a channel, it is sufficient to sample it at the rate of $2W$ samples every second. It is important to note that the sampling theorem does not indicate how many bits can be sent in a single sample. It only bounds the sampling rate. The amount of information that can be sent in a single channel, and hence the capacity of a channel is determined by the *Shannon channel capacity theorem.*

Theorem 2 (Shannon Channel Capacity Theorem) *The capacity C of a channel with bandwidth (frequency spectrum) W, signal power S, and noise power N is given by*

$$C = W \log_2(1 + \frac{S}{N}).$$

Proof Sketch: By the Nyquist sampling theorem, it follows that we can sample at the rate of $2W$ samples per second. If we can send k bits with every sample, then the channel capacity is $2Wk$. We now determine k.

Let r denote the range of signal level (voltage) that is available to send data. We can quantify noise by the "distortion" in the signal level being transmitted. Noise with distortion ε transforms signal level v to a level in the range $(v + \varepsilon/2, v - \varepsilon/2)$. It can be seen that if the available signal level range is r and the distortion achieved by noise is ε, then at most $\lceil r/\varepsilon \rceil$ and at least $\lfloor r/\varepsilon \rfloor$ different "items" can be sent on any sample. In number of bits, this means that at most $k = \log_2(r/\varepsilon + 1)$ bits can be sent per sample. Since the power of signal or noise is directly proportional to the square of the voltage (level), it follows that $k \approx (\log_2(S/N + 1))/2$. This completes the proof of the theorem. ■

Note: (0) We have stated the Nyquist theorem for low-pass channels only. There is a version of the theorem that holds for band-pass channels as well. The Shannon capacity theorem holds for all communication channels. (1) The capacity of a channel is what is referred to as *bandwidth* in the networking literature; (2) Usually, the signal to noise ratio is measured in *decibels* and is expressed as $10\log_{10} S/N$ decibels; (3) Example: For a typical telephone line, the bandwidth is $3000Hz$, and S/N is 1000; This yields a channel capacity of $3000 \times \log_2 1000 \approx 30Kbps$.

2 Error Detection

The physical layer provides the abstraction of a *continuous bit stream* to the data link layer. In order to use the abstraction in a reliable manner, the data link layer provides three functionalities.

- It frames the packets and adds idle fill so that the packets can be distinguished from one another and from the transmission when the channel is idle.
- It adds error detection mechanisms to detect errors introduced by the physical channel. Also implemented are retransmissions strategies when errors or losses occur.
- For multiple-access links, the data link layer adds contention resolution protocols.

We begin our study of the data link layer by considering error detection and retransmission strategies. Since we would like to allow an arbitrary bit string to be transmitted, we cannot detect errors by examining the packet itself. So we need to add extra bits. Perhaps, the simplest strategy is that of a single parity check.

Single parity check: In this scheme, we append a single bit at the end of the packet which is the parity of the packet bit. A good example is the ASCII code, where each of the $128 = 2^7$ characters is represented using 8 bits, the eighth bit being the parity. The salient features of the single parity check code are:

- It can detect single bit errors, for a string of *any* length.
- Any odd number of errors are dectected.
- Even number not detected; so about half (not quite accurate) the errors detected, assuming errors occur uniformly at random.
- Several bits could be mapped to a single sample; if the sample has errors, then it is quite likely (half chance) that even number of errors occur.

Horizontal and vertical parity check codes: Horizonal and vertical parity check codes arrange the string in a 2-dimensional array (in row-major order) and add one parity bit on each row and one parity bit on each column. Table 1 illustrates the horizontal and vertical parity check code. As indicated in Table 1, while the code can detect 2- and 3- bit errors, it may miss 4- bit errors.

Internet checksum: Several Internet procotols (e.g., IP) use a checksum to detect errors. Given a sequence of bits broken down into L k-bit words $b_0, b_1, \ldots, b_{L-1}$, the checksum is simply the negative of the sum

$$b_0 + b_1 + \ldots + b_{L-1} \bmod 2^k - 1.$$

If the checksum is added as the word b_L to the sequence, then the receiver needs to check whether the following condition is true:

$$b_0 + b_1 + \ldots + b_{L-1} + b_L = 0 \bmod 2^k - 1.$$

$$
\begin{array}{cccccc|c}
1 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & \underline{1} & 0 & 1 & \underline{1} & 0 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 & 1 & 1 \\
1 & \underline{1} & 0 & 1 & \underline{0} & 0 & 1 \\
\hline
1 & 0 & 1 & 0 & 1 & 1 & 0
\end{array}
$$

Table 1: Illustration of horizontal and vertical parity checks. If the four underlined bits get corrupted, then the code will not detect the error.

Parity check codes: Parity check codes are a natural generalization of the horizontal and vertical parity checks. A parity check code starts with the given string generate parity checks for various subsets of the bits. Let K be the length of the data string, and L be the number of parity check bits added. Thus, a parity check code transforms a K-bit string to a $(K + L)$-bit *code word*. Since each string is mapped to exactly one code word, there are exactly 2^K code words, each of length $K + L$.

The error detection is simple. If the received string is a code word, it is accepted; otherwise it is rejected. Thus, errors that map a code word to another are not detected. All other errors are detected.

How do we calculate the effectiveness of a code? There are 3 measures:

1. Minimum *distance* of the code, which is the smallest number of errors that can convert one code word to another. For single parity check code, it is 2, while for horizontal/vertical code, it is 4.

2. The *burst dectecting capability* of the code, which is the largest integer B such that the code can detect all bursts of length B or less. The *burst* of errors is the number of bit positions between the first error and the last error in the transmitted string (both inclusive). Note that it is *not* required for each bit position between the first and last error positions to have an error. The burst capability of the single parity code is 1, while that of the horizontal/vertical code is one plus the length of the row.

3. The probability that a completely random string is accepted as error free. For a parity check code with L parity check bits, this is 2^{-L}.

Cyclic redundancy check codes (CRC): The most common error detection mechanism used in DLC protocols is the cyclic redundancy check code. It is also a kind of a parity check code, as Problem 5 of Problem Set 1 formally establishes. The basic idea behind this code is as follows. Suppose M is the K-bit string that the sender wants to send. Suppose ℓ is the length of the parity check bits to be added. The bits that are added are given by the remainder obtained when $M \cdot 2^\ell$ is divided by a predetermined $\ell + 1$-bit pattern P. Thus, if T is the string that is transmitted, we have $T = 2^\ell M + R$, where $2^\ell M = P \cdot Q + R$ (Q being the quotient and R being the remainder). We note that all the arithmetic is modulo 2.

We claim that T is divisible by P. This is true since $2^\ell M + R = P \cdot Q + R + R = P \cdot Q$. Thus, the receiver need only check whether the received string is divisible by P.

Usually, the CRC code is expressed in terms of polynomials. Note that the message M can be written as the $(K-1)$-degree polynomial $M(D) = m_{K-1}D^{K-1} + m_{K-2}D^{K-2} + \ldots + m_0D^0$, where m_i is the ith bit (from right to left) of M. Similarly, we can express the pattern P and the CRC code (or the remainder) R as the polynomials $P(D)$ and $R(D)$ of degree L and $L-1$, respectively. In terms of the polynomials, we obtain the code word T sent represents the $(K+L-1)$-degree polynomial $T(D) = D^{\ell}M(D) + R(D)$. Any error can also be represented by a $(K+L-1)$-degree polynomial. The received string is given by the polynomial $T(D)+E(D)$. Since the received string is accepted if and only if it is divisible by $P(D)$, we obtain that an error goes undetected if and only if $E(D)$ is divisible by $P(D)$, which is unlikely, if $P(D)$ is chosen wisely.

An appropriate selection of the polynomial $P(D)$ (usually referred to as the *generator polynomial*) yields several desirable properties:

- All single bit errors are detected by making p_ℓ and p_0 to be 1.
- All double errors are detected if $P(D)$ is chosen to be a *primitive* polynomial, assuming $K \leq 2^{L-1}$. The main property of a primitive polynomial that is of relevance here is the following: If $P(D)$ is a primitive polynomial of degree L, then the smallest n for which D^n+1 is divisible by $P(D)$ is $2^L - 1$.
- All odd errors are detected if $P(D)$ is divisible by $D+1$.
- Burst errors of length less than L are detected.

One reason for the popularity of the CRC code is its efficient implementation in hardware via linear feedback registers. The text by Bertsekas and Gallager has more information on this aspect. An example of a commonly used generator polynomial is CRC-16, which is $D^{16} + D^{15} + D^2 + 1$.

3 Retransmission strategies

The Automatic Repeat reQuest (ARQ) protocols are strategies for retransmission of frames in the presence of errors or losses in the physical channel. ARQ protocols use the error detection schemes of Section 2 to detect errors and then request the transmitting DLC module to retransmit the erroneous frame, that is, retransmit the old packet in a new frame. The design of an ARQ protocol must be checked for *correctness* and *efficiency*. For correctness, we need to prove that each packet is released to the network layer by the receiving DLC *once and only once*, without error. For efficiency, we need to measure the overhead incurred in transmissions.

In order to reason about an ARQ protocol, we need to make certain assumptions about the underlying phsyical channel. These are as follows:

- We assume that error detection works correctly (using CRC codes, e.g.).
- Each transmited frame may be delayed by an arbitrarary and variable time
- Some frames might be lost and may never arrive.
- Frames that arrive are assumed to do so in order, with or without errors.
- There exists $q > 0$ such that any given frame is delivered correctly with probability at least q, independent of the other frames.