MIT 6.02 **DRAFT** *Lecture Notes*
Fall 2010 (Last update: September, 2010)
Comments, questions or bug reports?
Please contact 6.02-staff@mit.edu

LECTURE 2
Wires and Models

This lecture discusses how to model the channel in a communication channel; we'll focus on the simplest kind of channel, a transmitter and receiver connected by a **wire**. The model is simple and applies more generally to a variety of point-to-point communication channels, including the infrared (IR) channel in the 6.02 lab. The model also partially captures the salient features of a wireless channel.

Why do we care about modeling a wire? The reason is that there is no such thing as a perfect communication channel, as we saw the last time—it is physically impossible for a wire, or any other channel, to transport an arbitrary signal from channel input to channel output without any distortion. Our goal in 6.02 is to understand how to build digital communication networks, and the first step toward that goal is to design a fast and reliable wired communication channel, where a receiver at one end of a wire is able to recover the information sent by a transmitter at the other end of the wire. If we knew nothing about what the wire did to the transmitted signl, we'd have little hope of building a efficient and reliable receiver. Fortunately, despite the wide diversity of ways to make wires (where we are using the term "wire" to broadly indicate any physical system that can carry a signal from the transmitter location to a receiver location), most wires exhibit common characteristics that allow us to develop a **model**. We can then use that model to undo the impact of wire non-idealities, and recover a reasonably accurate representation of the transmitted signal using only the signal at the receiver.

The big ideas in this lecture are:

1. Understanding the relationship between bits, voltage samples, the number of samples per bit, the sampling rate, and the bit rate.
2. Inter-symbol interference (ISI) and eye diagrams.
3. Modeling a wire: causality, linearity, and time-invariance.

The ideas of causality, linearity, and time-invariance enable us to engineer digital communication channels, but they are in fact more widely applicable in many areas of electrical engineering and computer science. They are worth understanding because you will see them time and again in many different contexts.

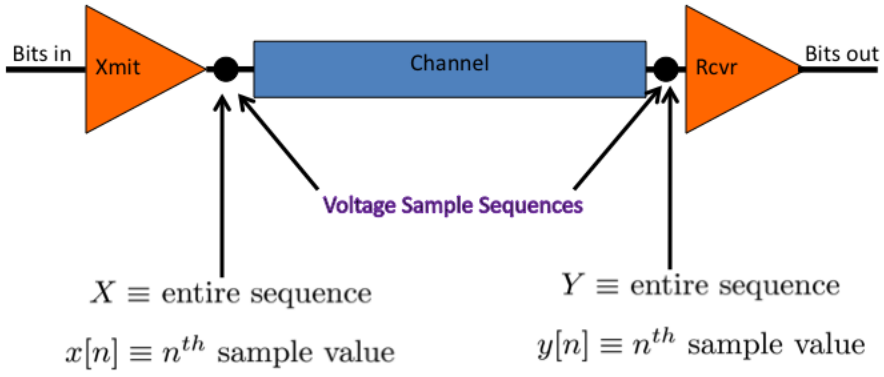

Figure 2-1: Abstract representation of a communication channel.

2.1 How Wires Behave

We start by first describing the problem setup and defining some useful terms.

2.1.1 Setup

Figure 2-1 shows the setup of a communication channnel. The transmitter (xmit) gets digital bits, 1's or 0's, converts them in to a sequence of voltage samples, and sends the samples to the input of a channel (e.g. a wire). The number of voltage samples used to represent each bit is termed the **samples per bit**. The transmitter sends one sample to the channel every τ seconds, where $1/\tau$ is the **sampling frequency**. We use the term **bit period** to refer to the duration of a bit; the bit period is equal to the number of samples per bit multiplied by τ. The **bit rate** of the channel, measured in bits per second, is the rate at which one can convey information over the channel; it is equal to the reciprocal of the bit period and is also equal to the sampling frequency divided by the samples per bit.

The receiver collects voltage samples from the output of the wire or channel, typically at the same sampling frequency as the transmitter, and then converts these samples back in to bits. In 6.02 you will see several different schemes for converting received samples to bits, but it is helpful to have a specific scheme in mind. One simple conversion scheme is for the receiver to select a single candidate from each contiguous set of samples-per-bit samples, and then to convert these bit detection samples to bits by comparing to a **threshold** voltage. A bit would be assigned the value '1' if the associated bit detection sample exceeded the threshold, and would be assigned '0' otherwise.

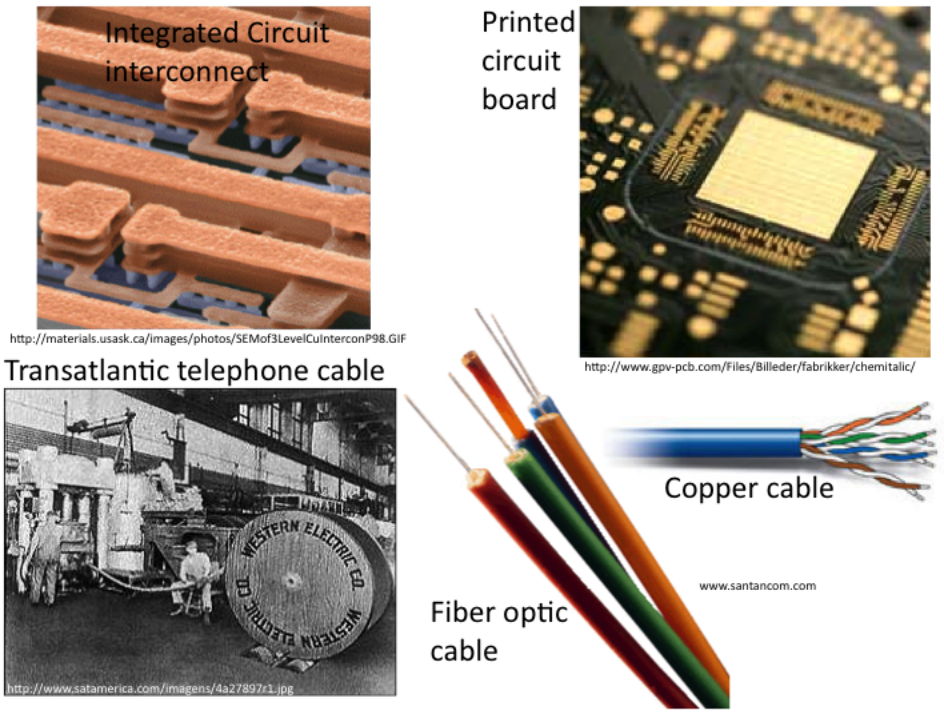

Figure 2-2: Many kinds of wired communication channels.

■ 2.1.2 What Happens to Voltage Samples in a Wire?

There are many kinds of wires used in communication channels, ranging from submicron-wide microns-long copper wires used inside integrated circuits, to millimeters-wide and centimeters-long wires on printed circuit boards, to possibly miles-long coaxial cables, to even longer (and now obsolete) transatlantic telephone cables, to fiber optic cables used in modern wired communication networks. Figure 2-2 shows a few of these examples.

Even though there is an enormous variety in the size and technology of wires, they all exhibit similar types of behavior in response to inputs. Consider, for example, what a receiver at one end of a wire might see when a transmitter, at other end of the wire, sends voltage samples that are set to zero volts for one bit period, then set to one volt for one bit period, then returned to zero volts for one bit period.

1. **A non-zero time to rise and fall.** Ideally, the voltage samples at the receiver end of a wire should be identical to the voltage samples at the transmitter end of the wire. Instead, one finds that when there is a nearly instantaneous transition from zero volts to one volt at the transmitter end of the wire, voltage at the receiver end takes much longer to rise from zero volts to one volt. Similarly, if the transmitter end of the wire transitions from one volt to zero volts nearly instantly, the receiver end of the wire will take much longer to fall from one volt to zero volts. If a wire's receiver voltages rise and fall quickly, we refer to the wire or channel as **fast**; but the receiver voltages take a long time to rise and fall, we say the wire or channel is **slow**. For example, integrated circuit wires and fiber optic cables are fast, rise and fall times are are in the tens to hundreds of picoseconds; household telephone wires are much slower,

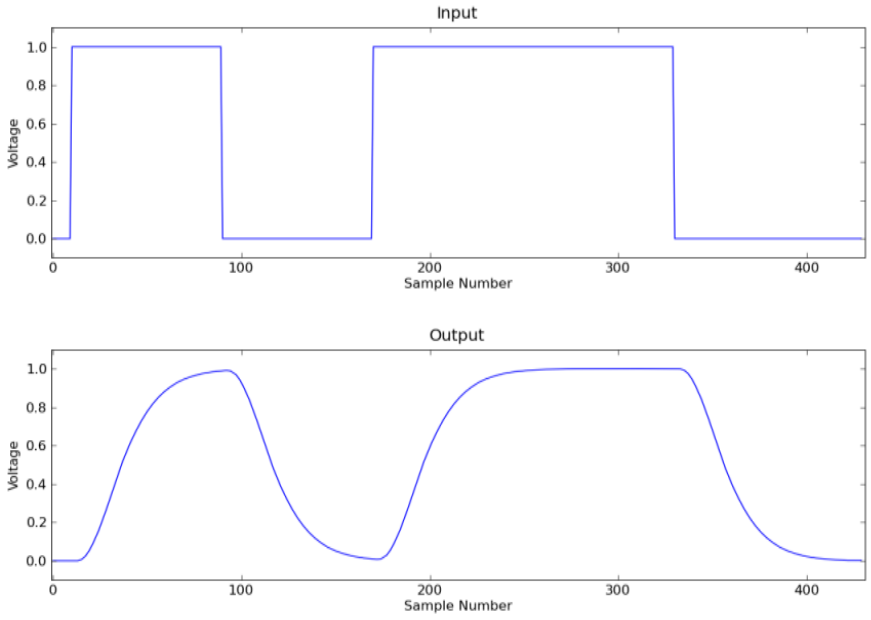

Figure 2-3: Signals sent over a wire to the receiver take non-zero time to rise and fall to their final correct values.

and have rise and fall times in the tens to hundreds of nanoseconds or longer. Our IR transmission system is quite slow, as it has rise and fall times on the order of tens of microseconds. Examples of typical rising and falling transitions at the transmitter and receiver ends of a wire are shown in Figure 2-3. It is important to note that if the time between transitions at transmitter's end of the wire is shorter than rise and fall time at the reciever end of the wire, a receiver will struggle to infer the value of the transmitted bits using the voltage samples from the wire's output.

2. **A non-zero delay.** The speed of electrical signals in a copper wire, or the speed or photons moving in an optical fiber, are both bounded by the speed of light in vacuum, though they typically travel much more slowly. The speed of light is a fundamental limit, and sets a lower bound on the time between the occurance of a transition at the transmitter end of a wire and the beginning of the response to that transition at the receiver's end of the wire (a lower bound that is unlikely to change).[1] An engineer designing a communication channel must consider that wires will have delays, and often must develop ways of recovering data without really knowing what the wire delay might be.

3. **"Ringing".** In some cases, voltage samples at the receiver end of a wire will oscillate before settling to a steady value. In copper wires, this can be due to a "sloshing" back and forth of the energy stored in electric and magnetic fields, or it can be the result of

[1]Short of radical changes in fundamental physics, which we shouldn't hold our breath for!

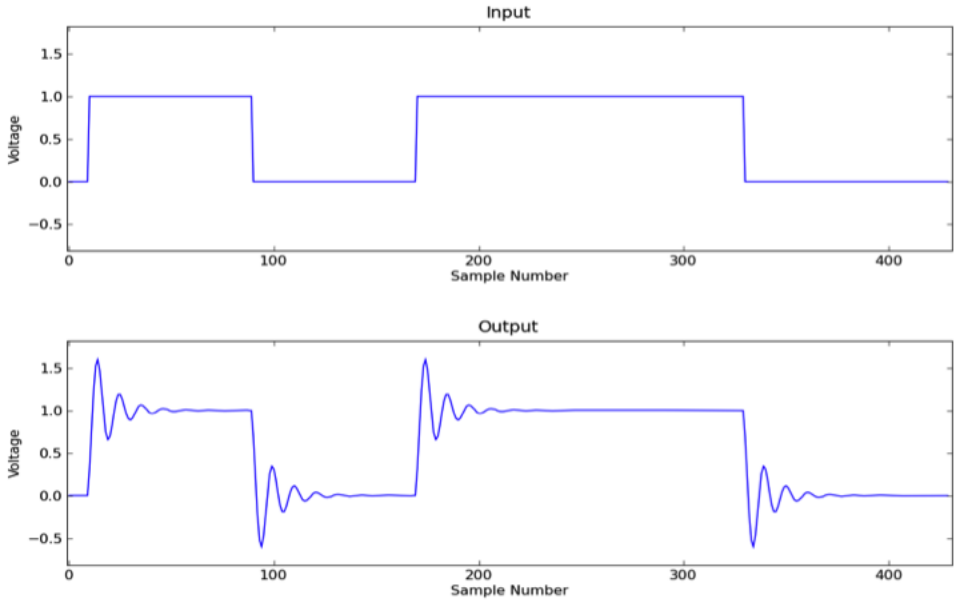

Figure 2-4: A channel showing "ringing".

signal reflections.[2] In 6.02, we will not try to determine the physical source of ringing on a wire, but will instead observe that it happens and deal with it. Figure 2-4 shows an example of typical ringing.

4. **Noise.** In addition to the above effects, all communication channels have noise. In this lecture, we won't spend too much time on understanding and coping with noise, but will save this important topic for future lectures.

Figure 2-5 shows an example of non-ideal wire effects. In the example, the transmitter converted the bit sequence 0101110 in to voltage samples using ten 1 volt samples to represent a '1' and ten 0 volt samples to represent a '0'. If the transmitter and receiver sample rates are the same as used in our IR hardware (four million samples per second or one sample every 0.25 microseconds), then ten samples per bit would correspond to a bit period of 2.5 microseconds. In the example, the settling time at the receiver end of the wire is longer than 2.5 microseconds, and therefore bit sequences with frequent transitions, like 010, may not be received correctly. As can be seen in Figure 2-5, at sample number 21, the wire output voltage is still ringing in response to the rising wire input transition at sample number 10, and is also responding to the wire input falling transition at sample number 20. The result is that the receiver may misidentify the value of the second or third transmitted bit. Note also that the the receiver will certainly correctly determine that the the fifth and sixth bits have the value '1', as there is no transition between the fourth and fifth, or fifth and sixth, bit. As this example demonstrates, the slow settling of the wire output implies

[2]Think of throwing a hard rubber ball against one of two parallel walls. The ball will bounce back and forth from one wall to the other, eventually settling down.

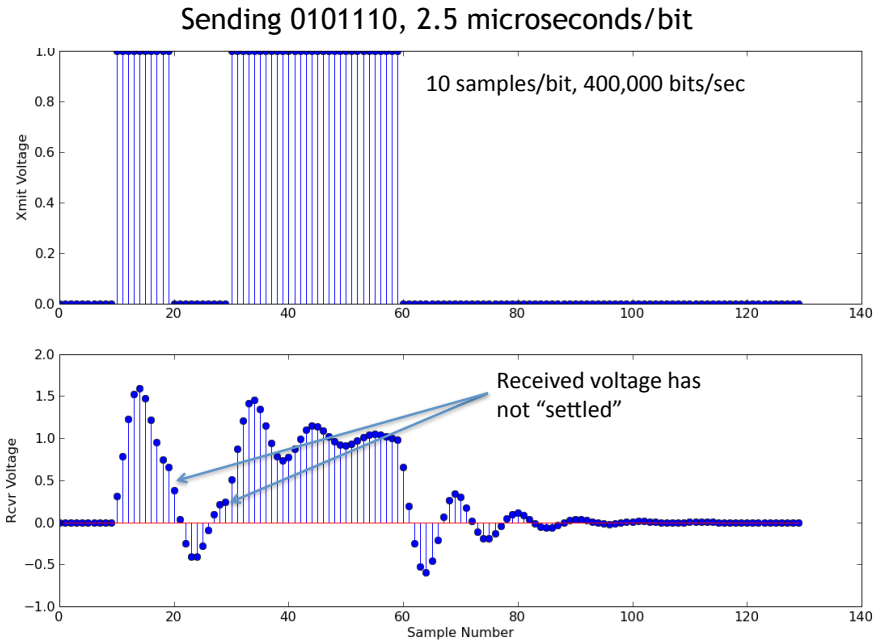

Figure 2-5: The effects of rise/fall time and ringing on the received signals.

that the receiver is more likely to misidentify a bit that differs in value from its immediate predecessors.

If we increase the number of samples per bit, and therefore the bit period, it is clear that there will be more time for the voltage at the receiver end of the wire to settle, reducing the likelyhood that the receiver will misidentify the bit values. In fact, one could increase the number of samples per bit to the point where there is almost no chance of receiver error, but that implies that bits are being transmitted at a very slow rate. In effect, we are trading away speed to gain reliability. Selecting the value of samples per bit is an example of an *efficiency trade-off* that communications systems engineers worry about—a too-low value yeilds a high bit rate but also a potentially high error rate, and a too-high value results in very few errors, but a low bit rate as well. Like Goldilocks, we want something "just right"—few enough samples per bit that we have a high bit rate, but enough samples per bit that errors are infrequent. These infrequent errors will have to be dealt with, and issue will be address in a number of ways in later sections.

■ 2.1.3 Inter-Symbol Interference

There is a formal name given to the impact of long rise/fall times and long settling times, (both cases were shown in the examples above) **inter-symbol interference**, or ISI. ISI is a fancy way of saying that "*the received samples corresponding to the current bit depend on the values of samples corresponding to preceeding bits.*" Put another way, the samples don't just behave independently over a communication channel, but affect each other; and therefore bits, or symbols, **interfere** with one another. Figure 2-6 shows four examples: two for

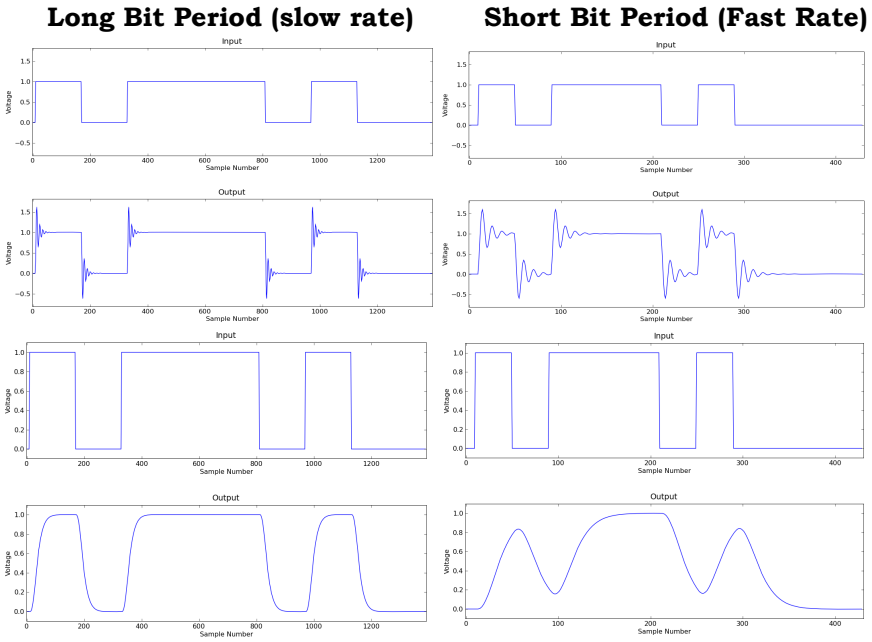

Figure 2-6: Examples of ISI.

channels with a fast rise/fall and two for channels with a slower rise/fall.

We can now state the problem we want to solve: *Given the sequence of voltage samples at the receiver, undo the effects of the channel and accurately determine the sequence samples at the transmitter*. We will develop a set of tools and techniques to solve this problem, which lies at the core of all communication systems. In this lecture and the next, we'll focus on understanding ISI and undoing its effect; we'll study and understand how to deal with noise in subsequent lectures.

■ 2.2 Undoing ISI

Our goal is to develop methods to determine the sequence of transmitted bits using only the voltage samples available at the receiver. As mentioned above, the simplest approach for accomplishing this goal would be for the receiver to pick a candidate voltage sample from each contiguous set of samples-per-bit samples, and then compare these **bit detection samples** to a threshold voltage to determine the transmitted bit. In the next lecture, we will examine a two-step strategy that first generates a, presumably improved, sequence of voltage samples from the received sequence of voltage samples, and then select the bit detection samples from this new sequence.

In order to determine what approach to use to convert received voltage samples to received bits, we need a systematic way to understand the effects of ISI. For example, we would like to know whether a particular choice samples per bit is large enough that bits can be determined reliably by just extracting bit detection samples from the received

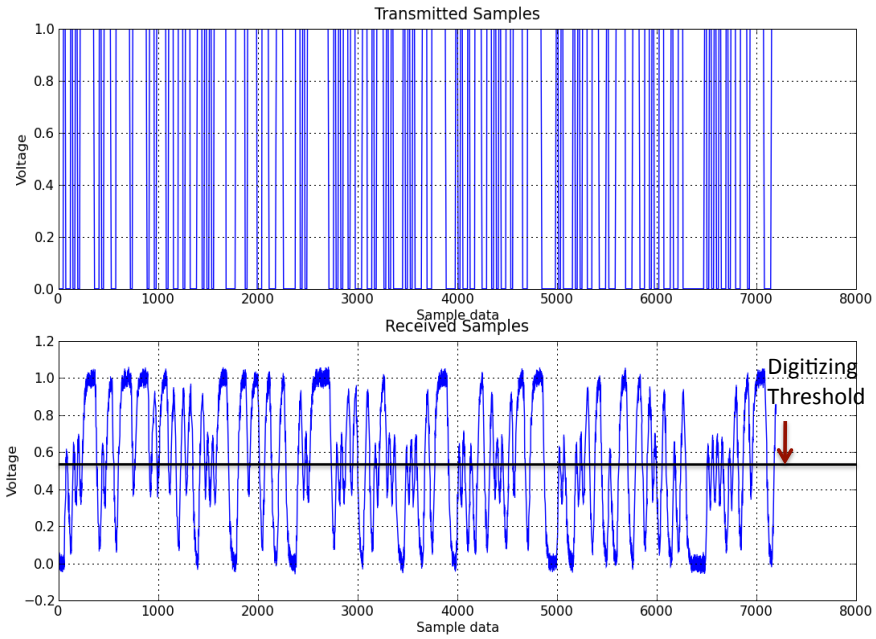

Figure 2-7: Received signals in the presence of ISI. Is the number of samples per bit "just right"? And what threshold should be used to determine the transmitted bit? It's hard to answer these question from this picture. An eye diagram sheds better light.

samples. Or, if the two step approach is used, we would like to know if ISI effects are reduced in the new sequence of voltage samples. A useful tool for visualizing ISI is the **eye diagram**, sometimes also known as an "eye pattern". Eye diagrams are used by anyone who sets out to design or improve the performance of a communication channel.

■ 2.2.1 Eye Diagrams

On the face of it, ISI is a complicated effect because the magnitude of bit interference and the number of interfering bits depend both on the channel properties and on how bits are represented on the channel. The eye diagram is a useful graphical tool to understand how ISI manifests itself and how to make sense of it.

Figure 2-7 shows what the receiver sees (and what the transmitter sent). We have two problems: first, are there too few (or too many) samples per bit? Second, what threshold should the receiver use to infer the bit being sent? The eye diagram is a useful tool to use to solve both problems, particularly the first one.

To produce an eye diagram, take all the received samples and put them in an array of *lists*, where the number of lists in the array is equal to the number of samples in k bit periods. (In practice, we want k to be at least 3, and small; we'll assume $k = 3$ here.) If there are s samples per bit, the array is of size $k \cdot s$.

Each element of this array is a *list*, and element i of the array is a list of the received samples $y[i], y[i + ks], y[i + 2ks], \ldots$. Now suppose there were no ISI at all (and no noise).

Then all the samples in the i^{th} list corresponding to a transmitted '0' bit would have the same voltage value (zero volts is a typical value used to represent a '0' bit), and all the samples in the i^{th} list corresponding to a transmitted '1' would have the same value (one volt is a typical value used to represent a '1' bit). Consider the simple case of just a little ISI, where the previous bit interferes with the current bit (and there's no further impact from the past). Then the samples in the i^{th} list corresponding to a transmitted '0' bit would have two distinct possible values, one value associated with the transmission of a '10' bit sequence, and one value associated with a '00' bit sequence. A similar story applies to the samples in the i^{th} list corresponding to a transmitted '1' bit, for a total of four distinct values for the samples in the i^{th} list. If there is more ISI, there will be more distinct values in the i^{th} list of samples. For example, if two previous bits interfere, then there will be eight distinct values for the samples in the i^{th} list. If three bits interfere, then the i^{th} list will have 16 distinct values, and so on.

Formally, without knowing now many bits interfere, one must produce the above array of lists for every possible combination of bit sequences that can ever be observed. If we were to plot such an array on a graph, we'll see a picture like the one shown in Figure 2-8. In practice, we can't produce every possible combination of bits, but what we can do is use a long random sequence of bits. We can take the random bit sequence, convert it in to a long sequence of voltage samples, transmit the samples through the channel, collect the received samples, pack the received samples in to the array of lists described above, and then plot the result. If the sequence is long enough, and the number of intefering bits is small, we should get an accurate approximation of the eye diagram.

Figure 2-8 shows the *width of the eye*, the place where the diagram has the largest distinction between voltage samples associated with the transmission of a '0' bit and those associated with the transmission of a '1' bit. Another point to note about the diagrams is the "zero crossing", the place where the upward rising and downward falling curves cross. Typically, as the degree of ISI increases (i.e., the number of samples per bit is reduced), there is a greater degree of "fuzziness" and ambiguity about the location of this zero crossing.

The eye diagram is an important tool because it can be used to verify two key design decisions:

1. The number of samples per bit is large enough. If samples per bit is large enough, then at the center of the eye, the voltage samples associated with transmission of a '1' bit are clearly above the digitization threshold and the voltage samples associated with the transmission of a '0' bit are clearly below. *In addition*, the eye must be "open" enough that small amounts of noise will not lead to errors in converting bit detection samples to bits. As will become clear later, it is impossible to guarantee that noise will never cause errors, but we can reduce the likelyhood of error.

2. The value of the digitization threshold has been set correctly. The digitization threshold should be set to the voltage value that evenly divides the upper and lower halves of the eye.

The eye diagram is a great tool for visualizing ISI, and we can use it to determine a suitable number of samples per bit to use when sending data on our channel. But to truly undo the effects of ISI, we need additional tools. The challenge is that there are many

kinds of wires, and ISI could involve arbitrary amounts of history. Fortunately, most wires have three important properties (at least approximately) that we will exploit: **causality**, **linearity**, and **time-invariance**. The next section briefly describes these terms.

■ 2.3 Causality, Linearity, and Time-Invariance

Wires are **causal**: the output changes only after the input changes. There's nothing particularly surprising about this observation—if causality didn't hold over a communication channel, we would have a channel (wire) that could predict the future, and we ought to be working on a way to use that kind of wire to make investment decisions!

Wires are, to a first approximation, **linear**: the output is a linear function of the input. A channel (system) is linear if the following holds. Suppose the output (i.e., the sequence of samples the receiver sees) is Y_1 when the input is X_1 and the output is Y_2 when the input is X_2. Then, if the system is presented with an input $AX_1 + BX_2$, where A and B are scalar constants, the system's output will be $AY_1 + BY_2$. Note that a special case occurs when we multiply an input X (whose output is Y) by a scalar number a; the output would be scaled the same way, producing aY.

By definition, **superposition** can be used to analyze a linear system, and superposition is an amazingly versatile technique. It allows us to construct the output by breaking an input into a sum of simple constituent parts, finding the output of each part, and then adding those outputs together!

Wires are **time-invariant**, which means that if we time-shift the input by k samples, the output also shifts by k samples, but remains unchanged otherwise. Formally, time-invariance means that when presented with an input sequence X, if the system generates an output Y, then when the input to the system is $X[n-k]$, the output will be $Y[n-k]$.

A system (channel) that is linear and time-invariant is also called an **LTI system (channel)**.

■ 2.3.1 Using Superposition

The key insight is that if we have an LTI channel, the response (output) of the channel to any input can be completely characterized by its response to a canonical input, such as the **unit step**, $u[n]$, or the **unit sample**, $\delta[n]$. The reason is that *any* function can be written as a sum of positively and negatively weighted and shifted unit steps, or a sum of positively and negatively weighted and shifted unit samples. Hence, if we know how the channel responds to a unit step or unit sample, we can sum appropriately weighted and shifted unit step or unit sample responses to determine how the channel would respond to any input.

The next lecture will describe this idea in more detail and show how to use this insight to undo the effects of ISI, as long as the amount of extraneous noise is low.

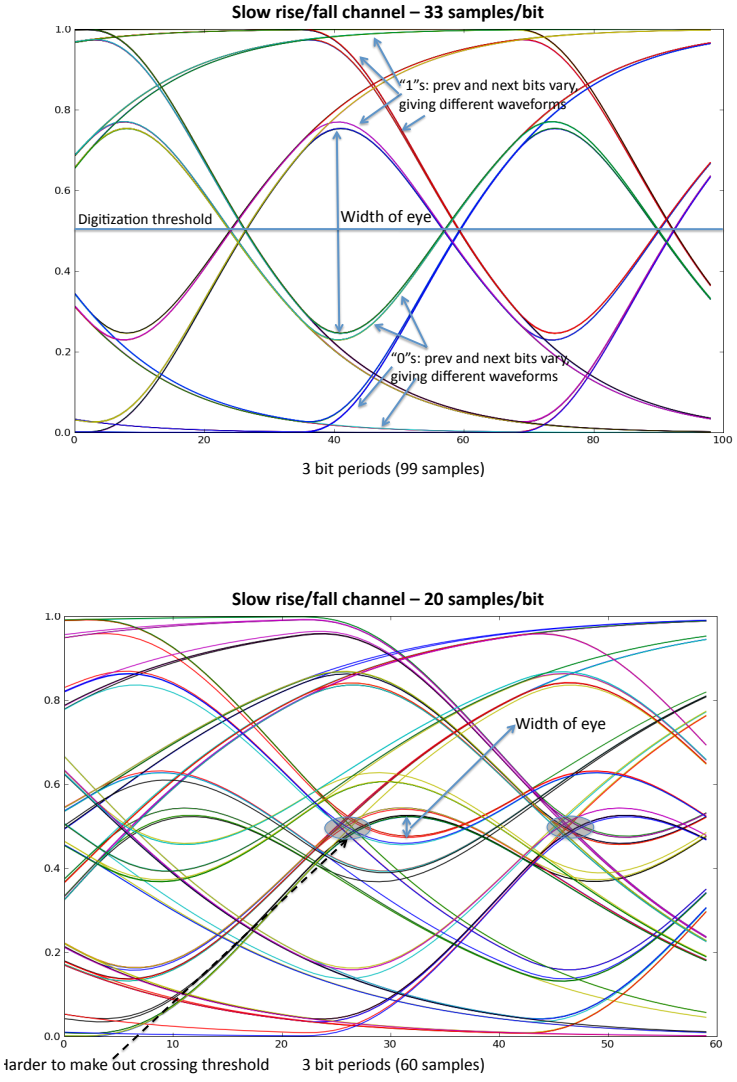

Figure 2-8: Eye diagrams for a channel with a slow rise/fall for 33 (top) and 20 (bottom) samples per bit. Notice how the eye is wider when the number of samples per bit is large.