MITOCW | 1. Overview: information and entropy

GEORGE VERGHESE: The course is called Digital Communication Systems. So I wanted to say a bit about what that means. And the easiest way to do that is to contrast it with analog.

What's analog communication? Well, in analog communication, you're typically focused on communicating some kind of a waveform. So you've got some continuous waveform, typically, an x of t, maybe the voltage picked up at a microphone at the source, and you want to get it across to a receiver.

And it's under this umbrella that you have things like amplitude modulation, frequency modulation, and so on. These are all schemes aimed at transmitting a continuous waveform of this type. So an amplitude modulation, for instance, what you'll do is you'll take a sinusoidal carrier. The carrier carries the information about the analog waveform, and basically, it's a high-frequency sinusoid whose amplitude is varied in proportion to the signal.

I haven't drawn it too well. It's supposed to be constant frequency and just the amplitude varying. So this is something of the type x of t cosine 2 pi fc t, for instance. It's a sinusoid of a fixed carrier frequency with the amplitude varying slowly.

In FM, what you do is you have a fixed amplitude waveform, but you vary the frequency. So what you might do is have high frequency in this part, and then when the signal goes low, the frequency gets lower. And then it gets higher where the signal is high.

So there's a modulation of the frequency, but the amplitude stays fixed. The good thing about this is you can be transmitting at full power all the time, and the information is coded onto frequency, whereas this can tend to be more susceptible to noise. But the focus is on an analog waveform and transmitting that.

Now, in digital communication the focus changes. So in digital, we think in terms of sources with messages. So we have messages. There's a source of some kind that puts out a stream of symbols.

So at Time 1, there's some symbol emitted. At Time 2, there's some other symbol, symbol, symbol. And these are all heading to the receiver.

So already we're thinking of a clocked kind of system. We're thinking of symbols being transmitted at a particular rate. We're thinking of these discrete objects rather than continuous waveforms. And the focus is then on getting a message across as opposed to getting a waveform across with high fidelity.

And that turns out to actually be a big shift in perspective. These symbols, then, will often get coded onto-- for instance, the symbols, if they originally were, let's say an A, B, C, D, for instance-- coding the grades in a class-- you might want to, when you're transmitting them, to adopt those symbols to what your channel is able to take. And maybe your channel is one that's able to distinguish between two states, but maybe not between four states, so you might want to code these onto a channel that-- well, onto strings of 0's and 1's, so that you can impress this on a channel that can respond to just two states.

So you might have a coding step that takes the original symbols, puts out a stream of 0's and 1's. And then you've got the task of decoding the message. The channel might corrupt these streams, and that's another thing that you have to deal with.

So what's made digital explode is the fact that it's really well matched to computation, memory, storage, all the stuff that's advancing rapidly. In the world of analog, you're talking about analog electronics, which is also advancing greatly but doesn't have the same flexibility. Here, you can do all sorts of things with digital and with the computation that's available, and that's growing in power all the time to do more and more fancy things. So digital communication is really most of what you see around you.

So when you talk on the phone, or do computer-to-computer communication, or browse the web, and so on, it's really digital communication that you're talking about-- with one little caveat, I guess. When you get down to the details of how you get a 0 and 1 across on a channel, you're back in the analog physical world. And you tend to be doing things that are much closer to what you worry about on analog channels.

And we'll see that in this course. So for most of what we talk about, we'll be working at the level of the digital abstraction here. But when we come to talking about transmission on a link, and modulation, and demodulation, and the like, we're back in the analog world. And you'll actually get a good feel for some of this in the course of our digital communication.

So to give you a sense of how the course is structured, we'll spend some time, first of all, talking about information-- information in a message and how you measure it, how you code it up. So this is sort of the bits piece of the course. And then we'll talk about how to get these messages across single channels.

So this is a single link source at one end, receiver at the other end. And we'll focus on how you get the data across. And that brings us to the analog world and to the world of signals, so we'll spend time on that. And that's sort of the second third of the course.

And then the last third of the course, which Harry will actually be lecturing, focuses on now when you have interconnected networks. So you've got multiple links, so you might want to communicate from some source here to a receiver that's way across on the network, going through multiple links and multiple nodes. And there are all sorts of issues there.

And there, we're thinking in terms of a packet kind of abstraction. It's packets that we ship around the network with associated logic, and mechanisms for failure detection in the network, and coping with all of that So there are these three sort of chunks to the course, and you'll see that in the notes as well.

So we'll start off with the bits piece. That's more or less Quiz one. Then we'll get to the signals piece. That's more or less Quiz two, and we'll get to the packets piece, and that's more or less Quiz three.

And these will be relatively modular. So you get a chance to make a fresh start on each of them. But you'll find us reaching back to build on ideas developed earlier in the course.

Now, as I think about where digital communication originated, it actually turns out to be largely due to the person who painted this painting. This is called "The Gallery of the Louvre." This is somebody who painted it in around 1830, an American painter.

He was actually born in Charleston, close to here, studied at Yale, made enough of a name for himself that he had commissions to paint portraits and the like. He was actually called to Washington, DC to paint a portrait of the markets, the Lafayette. While he was there, he got a telegram from his father saying that his wife in New Haven was convalescing.

By the time he got to New Haven, which was as soon as he could-- he abandoned the painting he was doing and left for New Haven-- by the time he got there, he found that she had actually died and been buried. And that sort of fortified him for what he decided was his life's work, which was to find better ways to communicate, faster ways to communicate. He didn't want to have to depend on horse riders to carry messages or ships across the ocean.

He's actually painted himself into the middle of that painting. These were some friends that he made in Paris. It's actually the author, James Fenimore Cooper, of *The Last of the Mohicans* fame. He was actually hoping to sell his painting to Fenimore Cooper, but things didn't work out that way.

In any case, this is a huge painting. It's about six feet by nine feet. He wrapped it up to bring it back to the States. It wasn't quite finished.

This was about 1831 or so. And on the boat, he met this person who had a little electromagnet that he was playing with. And they had various discussions, and he got the idea for a telegraph. Anyone with a guess as to the name?

Morse. Now we think of Morse as the Morse code guy, but it turns out that he actually did hugely more than the code. So that's Samuel Morse, looks pretty imposing. He didn't just come up with the code, he actually invented the whole system.

Now, he didn't work in a vacuum. There were people doing related things in different places, but his was the first practical essentially single-wire system. If you look at his patent documents, he's got all the little pieces that it takes to make the system.

A key piece was actually the relay. So he figured out, working with a colleague back in New York, he figured out that with a little battery, you could close an electromagnet or you could power an electromagnet at some distance. But you couldn't have that wire be too long.

So what he arranged was a relay where that electromagnet then pulls another piece of metal, which then closes another switch on a separate circuit, so you can then start to propagate the signal over very large distances. And that was really a key part of his invention. Morse code-- there's actually some discussion as to whether he invented it or it was actually his assistant Vail, but it's called "Morse code" anyway.

The other staggering thing about this story is how soon after the invention-- I mean, his patent was, let's see, 1840, very early in the days of the Patent Office, as you can see from the numbers assigned to the patent. About 15 years later, there were people raising money to lay cable across the Atlantic to carry telegrams. So can you imagine, partly the bravery of these people?

I mean, it's hard enough to think of laying cable across Boston Harbor. And they were prepared to design this cable, load it on a ship, and lay it across the entire Atlantic. They made an attempt in 1857.

It actually turned out to work for about three weeks. That was long enough for Queen Victoria to congratulate President Buchanan, except it took almost all of a day to get the 98 words across from one side to the other. And the reason is when you put a little pulse on one end of a very long cable, it distorts like mad by the time it gets to the other end. So you can barely detect-- if you put a sharp change in one end and you've got a long cable, and if it's a poorly designed cable, it takes a long time to detect the rise at the other end if you detect it at all.

It turns out the person at the American end was the person who would later become Lord Kelvin. He was called plain old Walter Thomson at that point. He had designed a very sensitive way to measure these changes in voltage at the ends of cables.

But the person at the British end was actually a surgeon, a self-taught electrical engineer, who was convinced that the problem was there was not enough voltage on the cable. So he kept cranking up the voltage. When he got to 2,000 volts, the cable failed.

And so there had been celebrations in the street, and there had been fireworks, and all of that. And then people got very angry, and thought this was a scam, and a way to raise money, and all of that. Despite all of the negative press, a year later here was this man again, with enough funding from governments and private sources to make another attempt at the cable.

Anyway, it took a while. It took a good nine years to finally lay a good cable. They'd gone out about 1,200 miles with a cable in 1865 before it broke. They had to start again in 1866.

They managed to lay an entire cable, and then they came back and found the broken end of the 1865 cable, and picked it up, and continued it. So in 1866, they managed to get two cables working. And now it was a lot faster-- eight words a minute.

It was digital communication. It's got all the ingredients of what we see in digital communication today. And then a little while later, there was a transcontinental line, which marked the end of, essentially, the Pony Express trying to carry mail across the continent. Now much more was going to happen on telegraph lines.

There was a transpacific line in 1902, so that meant at that point, you could encircle the globe with telegraph. So it was really a transformative technology. And it was a digital technology because all you were trying to figure out at the other end was whether something was a dot or a dash. It was just basically two states that you were trying to distinguish.

That's his Patent Office documents. Actually, they're interesting to read, but let's just see here-- "Be it known that I, the undersigned, Samuel F. B. Morse, have invented a new and useful machine and system of signs for transmitting intelligence between distant points by the means of a new application and effect of electromagnetism." And then he goes on to describe the equipment and the code itself. This is just a map to show you the kind of distance that they had to lay that first cable over.

Morse code you've all seen. It's gone through some evolution, actually. Actually, Morse originally thought of just a code for numbers, and then he imagined a dictionary at the two ends, and you would just send the number for the word in the dictionary, and someone would look it up at the other end. But then with Vail, they developed this scheme.

You notice the most frequently used letter has the shortest symbol here. It's just a dot. And then if you go to an A, it's a dot, dash, and so on. The T, I think, is a dash. Yeah.

So the choice of symbols sort of matches the expected symbol frequencies in English text. You want the more frequently used letters to have the shorter symbols because there are going to be many of them, and you don't want to be sending along code words with them. But this was Morse code.

Here's another way to represent it. So going to the left is a dot, going to the right is a dash. So a single dot brings you to an E. A dot, dot brings you to an I, dot, dot, dot to an S. Dash dot brings you to an N, and so on. So you can display this code on a graph.

One thing you see right away from this display, and it was clear from the code itself, is you're not going to be able to get away with just two symbols. Because if you're trying to get to an A on this path, you hit an E on the way. And you need something to tell you that you aren't done yet.

So there is a third symbol, and that's the space. So Morse code has dot, a dash, and a space. It's really a three-symbol alphabet, and the space is critical.

If you want to have a code where you can deduce instantly that you've hit the letter that the sender intends, you need all the letters to be at the ends here, at the leaves of the tree. If you have all the code words at the leaves of the tree, right at the ends, then you're not going to encounter any other code words along the way. So you just keep going down the tree, dot to the left, dash to the right, till you hit the code word at the end, and then you're done. But in this kind of arrangement, you need a third symbol, actually, to demarcate the different words.

So this made Morse a very celebrated man. He got patents in various places. He got medals from all over the world, including the Sultan of Turkey.

I believe this one is a Diamond Medal from the Sultan of Turkey. He was celebrated on postage stamps, and all deservedly so. He really made a huge difference to communication, and set digital communication on the current path.

So I've got to skip forward very quickly now to bring me to the part of the story I want to continue with. So we're going to hopscotch over a whole bunch of, again, transformative inventions. There was a telephone in '76, again with Boston connections.

It was really thought of as speech telegraphy at that time, so it wasn't the telephone yet. Bell's patent is titled "Improvement in Telegraphy." There was wireless telegraphy, which was Marconi sending signals from Europe to, actually, Cape Cod, but it was not voice. It was Morse code, basically, dots and dashes.

And then here came analog communication. So this was exactly what I talked about, AM and FM, and then later video images, and so on. So there's a lot of this going on during this period.

A big player in the theory, that company-- this was actually Bell Labs. The Bell Labs is really full of people who made a huge difference to the development of all this. In fact, I had some names listed on a previous slide. Let me see if I-- I passed over them without mention.

But in the development of the telegraph, I've mentioned Lord Kelvin already. He did a lot to model transmission lines and to show how to design better ones. Design of magnets and the invention of the relay-- that was actually Joseph Henry, a Professor at Colombia, after whom the unit of inductance is named. That's the Henry, and various other people. So this technology really was a very fertile kind of ground for people to develop things in.

And if you take other courses in the department, these are names you will encounter all over the place-- Nyquist, Bode, e But the one I want to focus on is Claude Shannon, who's sort of the patron saint of this subject, I would say.

Shannon did his Master's degree here at MIT. It's been called one of the most influential Master's theses ever, because he developed Boolean algebra as a way to design logic circuits. The logic circuits he was talking about were relay circuits of the time, but this was very quickly picked up, and quoted, and applied.

And then he moved on to something else for his PhD. And I don't know the extent to which that's been influential in genetics. But then he joined Bell Labs just about the start of the war years.

A lot of work on cryptography during that time, initially classified but then a declassified version published. During that time, he also had interaction with Alan Turing, who was working on cryptography in England, but had been sent over to Bell Labs to share ideas. And then in 1948, a groundbreaking paper that really is the basis for information theory today.

So it was this that developed a mathematical basis for digital communications. And the impact has been just incredible since then. So that's what we want to talk about a little bit.

Now I have here a checklist that I don't want you to look at now, but the theory that Shannon developed is actually a mathematical theory. It's a probabilistic theory. And if you're going to be doing calculations with probability, you need to know some basics.

What I put down there is a checklist. We don't have a probability prerequisite for this course. We assume you've seen some in high school, some in 601, some elsewhere.

By the way, I should say that all these slides are going to be on the web, or maybe they're on the web already, so you don't have to scramble to copy all this. We'll put the lecture slides up on the web. We may not have them exactly before lecture, because I'm often working right to the bell, but we'll have them after the lecture. So take down whatever notes you want, but you don't have to scramble to get every word here.

By the way, the other thing I should say is that your contract in this course is not with whatever materials on the web or what you find from past terms and so on. It's really with us in lecture and in recitation. So we urge you to come to lecture even though all of this will be posted, because there's other learning that happens, and you have a chance to bring up questions, and hear other people's questions, and so on.

I'm not going to read through that. But I want to have a little picture that you can carry away in your mind for what we think of when we think of a probabilistic model. So we've got a universe of possibilities. We've got outcomes.

These are what are called elementary outcomes. Think of it as rolling a die, for instance, and I get one of six numbers. So each of those is an outcome. So here's the elementary outcome.

I could number them s1 to s n, and they don't have to be finite. It could be an infinite number, a. Continuum we'll see examples of that later.

But if you're thinking of a source emitting symbols to construct a message, then at every instant of time, the source is picking one of these symbols with some probability. So that's the kind of model we're thinking of. So here are the elementary outcomes-- s1, s13, and so on.

You've got events, and events are just collections of outcomes. So events are sets. So this is the event or set a, just a collection of elementary outcomes.

I say that the event has occurred if the outcome of the experiment is one of the dots in here. If the dot is out here, if this is what you got when you ran the experiment, the event didn't occur. So an event is just a set, a subset of these outcomes. We say "the event has occurred" if the outcome that actually occurs is sitting inside, in that set.

And then we can talk about intersections of events. We say that if this event a and this is b, the event a and b corresponds to outcomes that live in both sets. So if I roll a die, and I get a number that is even on a prime, that tells me what that number is.

So if this is the event of getting an even number on rolling a die, and this is the event of getting a prime number on rolling a die, what number do I get when both events have occurred? So I can identify different events, and then I assign probabilities to them. So I can talk about the probability of an event.

And then you can combine probabilities in useful ways. So let's see, there's a lot on here because I wanted, in principle, to fit it all on one slide that you could carry around with you. Probabilities live between 0 and 1.

The probability of the universal set is 1, meaning when you do the experiment, something happens. So it's guaranteed that something happens and therefore, u always happens. So the probability of u is 1.

And then the probability of a or b happening is the probability of a plus the probability of b If a and b have no intersection, if they're mutually exclusive. So we say that two events are mutually exclusive-- actually, let me draw a c over here-- a and c in this picture are mutually exclusive because there's no outcome that's common to the two events. So if one event occurs, you know the other one didn't occur.

And so if I now ask what's the probability of a or c occurring, it's the probability of a plus the probability of c. You'll be doing this all the time in this course. You'll be adding probabilities, but you've got to think-- am I looking at mutually exclusive events? If you've of got mutually exclusive events, then the probability of one or the other happening is the sum of the individual probabilities.

If they're not mutually exclusive, then there's a little correction you have to make. The probability of a or b happening is the probability of a plus the probability of b minus the probability of both happening. All of this is quite intuitive.

Another notion that's important is independence. So we've seen that mutual exclusivity allows you to add probabilities. Independence allows you to multiply probabilities.

So we say that a set of events-- a, b, c, d, e, for instance-- are mutually independent if the probability of a and b and c and d happening is the product of the individual ones. But similarly for any subcollection-- so you're going to call a collection of events independent if the joint probability of their happening in any combination factors into the product of the individual probabilities. And again, this is a computation you'll be doing all the time in different settings, but you've got to think to yourself-- am I applying it to things that are independent? Because if not, then it's not clear you can do this factorization.

We'll come later to talk about conditional probabilities. But the probability of a given that b has occurred-- we can actually sort of see it here-- the probability of a given that b has occurred is this area as a fraction of the whole area there. Sorry, the probability of a given that b has occurred is the probability of a and b over the probability of b.

Given that b has occurred, you know that you're somewhere in here. And what's the probability that a has occurred given that you're somewhere in here? It's the probability associated with that intersection.

One last thing-- expectation. We talk about the expected value of a random variable as being basically the average value it takes over a typical experiment, let's say. And the way you compute that is by computing the average weighted by the associated probabilities. And we'll see an example of that.

I didn't feel right just jumping into Shannon's definition of information without saying a little bit about how you set up a probabilistic model. But with all that said, here's what Shannon had as the core of his story, and building on earlier work by other people.

So if you're thinking of a source that's putting out symbols, the symbols can be s1 up to s n, the information in being told that the symbol s i was emitted is defined as log to the base 2 of 1 over the probability. So what you're trying to come up with is actually a measure of surprise.

Maybe "surprise" is a better word than "information." "Information" is very loaded word. But what you're trying to measure here is how probable is the thing that I am just seeing.

If it's a highly improbable event, I gain a lot of information by being told that it's occurred. If it's a high probability event, I don't get much information by being told that it's occurred. So you want something that's dependent reciprocally on probability. The log is useful because that allows you to have the information given to you by two independent events being the sum of the information in each of them.

And the calculation is just this-- it says that if a and b are independent events, then the information I get on being told that both of them occur is 1 over log p a p b. But that then just becomes the sum of the individual ones. So the advantage of having a log in that definition is that for independent events where the-- I should have actually perhaps written one more here.

Here's the information in being told that both events have occurred. Because they're independent, that joint probability factors into the product of the individual ones, which then factors into the sum of these two logarithms. So here's the information in being told a and b. Here's the information in being told just a occurred, and here's the information in being told that just b occurred. So the log allows things to be additive over independent events.

Now, the base 2 was a matter of choice. Hartley chose base 10, Shannon chose base 2. And he called it the "bit."

So when you measure information according to this formula, with the log taken to the base, 2 you call the resulting number the number of "bits" of information in that revelation. I'm being told that that's the output. Now for this lecture and probably only this lecture, I'm going to try and maintain a distinction between the bit as a unit of information and our everyday use of the word "bit" to mean a binary digit.

It's unfortunate that they both have the same name, because they actually refer to slightly different things. A binary digit is just a 0 or 1 sitting in a register in your electronics, whereas this is a unit of measurement. And the two are not necessarily the same thing. So I'll try and catch myself and say "binary digit" when I mean something that can be a 0 or 1, and "bit" when I'm talking about a measure of information.

But here, for instance, is a case where the two coincide. If I'm tossing a fair coin, so it's a probability 1/2 that it comes up heads, 1/2 that it comes up tails, then log to the base 2 of 1 over 1/2 gives me 1. So there's one bit of information in being told what the outcome is on the toss of a fair coin. And that sort of aligns with our notion of a binary digit as being something that can be either 0 or 1. We don't usually associated probabilities when we use "binary digit," but with "bit," we do.

So Shannon has a measure of information. And there are examples we can talk about there in the notes, so I won't go through them. And I think I've said this already, so I'll pass through that and get to his second important notion, which is the notion of entropy.

The entropy is the expected information from a source. So what we have is expected information from a source or from the output of an experiment, but if you're thinking of a source emitting symbols, this source can emit symbols s1 all the way up to s n, let's say, with probabilities p1 up to p n, let's say. And the sum of those is going to be 1.

If I tell you that s1 was emitted by the source, I've given you an information log 2 1 over p1. If I tell you s1 was emitted, that's the information I've given you.

But if I ask you before you see anything coming out of the source, "What's the expected information, what information do you expect to get when I run the experiment, when I produce a symbol," then you've got to actually average this quantity over all possible symbols that you might get. But it's got to be weighted by the probability with which you're going to see that symbol.

So this is exactly what I had defined earlier as an expected value. So the entropy of the source is the expected information-- or let's say the expected value of information you get when you're told the output of the source. And so if the emission is s1, then the information is this, but that happens with probability p1. If the emission is s2, that carries this information that happens with this probability, and so on.

So this is the entropy. Shannon is borrowing here from ideas developed in thermodynamics and statistical physics. People like Gibbs at Yale in 1900 already had notions of this type. His innovation is in actually applying this to communications, and he has several constructs beyond this.

We'll come to some of them later. But up to this point, he's making a connection with what they do in statistical physics, except they're usually not thinking in terms of information. They're thinking in terms of uncertainty here. And they're not thinking of sources emitting symbols. So this is the entropy.

So for instance, if you've got a case where you have capital N symbols and they're all equally likely, then the probability of any one of them is 1 over N. So what is the entropy? Well, it's going to be summation i equals 1 to N, each probability is 1 over N. I take log 2 1 over 1 over N.

So what does that end up being? That ends up being-- I can take the log 2 N out, and then I've got the summation, 1 over N. And the result is log 2 N. So if I've got equally likely symbols, N of them, then the entropy, the expected information from being revealed what the outcome is, is log 2 N.

It turns out that this is the best possible case in the sense of maximum uncertainty. If you're looking for a source that's maximally uncertain, that's going to surprise you the most when it emits a symbol, it's a source in which all the probabilities of symbols are equal. Symbols are equally likely-- that's when you're going to be surprised the most.

Now you can see this in a particular example here. Let's look at the case of capital N equals 2. So we're just talking about a coin toss. I toss a coin.

I get heads with probability p, some p. I get tails with some probability 1 minus p. Instead of saying "heads" or "tails," I could make it look a little more numerical. I could say C equals 1 for a head, C equals 0 for a tail. That's sort of coding the output of the coin toss.

And now I can evaluate the entropy for any value p you give me. So if you've got a fair coin with p of 0.5, I evaluate the entropy. And I find, indeed, that it's one bit. So the average information conveyed to me by telling me the output of the toss of a fair coin is one bit of information. But if the coin is heavily biased, then the average information or the expected information can be a lot less.

This turns out to be a very tight connection to this idea of coding. So let's actually take an extreme example. I've taken the case now where you've got a terribly biased coin.

It's not p equals 0.5, it's p over 10 to the 24. I picked 10 to the 24 because log to the base 2 of that is easy. So it's a very small probability of getting a head.

In fact, if you were to run 1,024 trials, the law of large numbers, which I haven't put on that one sheet, but you probably believe this-- if I had a coin that had a 1 in 1,024 probability of coming up heads, and I threw a coin 1,024 times, I'm more likely to get a heads once than anything else. And actually in a very long stretch, that's just about exactly the fraction of heads that you get. That's the law of large numbers.

In that case, what is the entropy? So I've got p times log 2 1 over p plus 1 minus p times log 2 1 minus p. I'm just evaluating that parabolic-looking function. It's not quite a parabola.

And I see that I've got just 0.0112 bits of information per trial. So unlike the case of a fair coin-- remember, in the case of a fair coin it equals 0.5, I have an entropy of one bit. That's the average information revealed by a single toss.

Now I'm down to much less. I'm down to 0.0112 bits per trial. And the reason is that this coin is almost certainly going to come up tails, because the probability of heads is so small. So for almost every trial, you'll tell me, "Oh, it came up tails." And there's no surprise in that. There's no information.

There's just the occasional heads in that pile. And when you tell me that came up heads, I'll be surprised. I get a lot of information. But not when I average it out over all experiments. It's actually low average information there.

So if you wanted to tell me the results of a series of coin tosses with this coin, you toss it 1,024 times, and you want to tell me what the result of that set of coin tosses is, it would seem to be very inefficient to give me 1,024 0's and 1's, saying, it was 0, 0, 0, 0, all the way along here. Let me say it this way-- here's one way to code it that would tell me what you got in 1,024 trials.

You could say, well, it was tails, tails, tails, tails, tails, tails, tails, oops, head, tails, tails, tails. So you could give me that 1,024 binary digits with a 1 to tell me exactly where you got the heads. It seems a very inefficient use of binary digits.

A binary digit can actually reveal a bit of information, and here you are using 1,024 binary digits to reveal much less information. In fact, let's see-- 0.012 bits per toss times 1,024 is really all the information there is. And you shouldn't be using 1,024 binary digits to convey that information. If you're sending it over a transmission line, it's a very inefficient use. So can you think of a way to communicate the outcome of this result with something that's much more efficient? Yeah.

AUDIENCE: [INAUDIBLE]

GEORGE VERGHESE: Yeah, just since you're expecting in 1,024 tosses that there'll typically be just a single one, just encode the position where that one occurs. How many binary digits does it take to do that? 10, right?

1,024-- you've got to tell me, is it in position 1, 2, 3, 4? You've just got to be able to count up to 1,024. So if you send me 10 binary digits to tell me where that 1 is, you'll have revealed what the outcome of the sequence of experiments is.

So 10 binary digits over 1,024 trials, so here's the average usage of binary digits-- binary digits per trial if I use your scheme. And that's much more. That's much closer to the actual bits per outcome. And somebody had a question on that?

AUDIENCE: Yeah. Is it actually less than [INAUDIBLE]?

GEORGE VERGHESE: It better not be. And part of it might be that I've rounded this here. Is it a small rounding difference? Did you actually compute something there?

AUDIENCE: 0.0097.

GEORGE VERGHESE: Sorry.

AUDIENCE: 0.0097 [INAUDIBLE].

GEORGE VERGHESE: Oh, is it not 0.0112? OK, good, I'm glad somebody computed that. How did I get that? Sorry.

AUDIENCE: This is also only because a possibility of 1 [INAUDIBLE].

GEORGE VERGHESE: Oh, I see. What you're saying is that this was-- right, you're saying this is 0.99 something. OK, I'm just saying that we're in the ballpark if we try to code just for the single 1. But there will be cases in my experiments where there might be two of these, and then I've got to use a more elaborate coding. I'll use a longer code word. Those are less likely events, so I've got to factor in all those probabilities.

Yeah, good. I'm glad you caught that. I don't want to get too sunk in this because I just want to convey the idea. The idea is that the Shannon entropy actually sets a lower limit to the average length of a code word.

And so when you're trying to do design of codes, you're actually trying to find codes that will get you close to the Shannon entropy limit. So what I want to just briefly mention, and you'll follow up in recitation, is something called Huffman coding, which you might apply to a situation like this. So you're coding, let's say, grades to send to the registrar. A's occur with probability 1/3, B's with 1/2, and so on.

You want a coding whose expected length will come close to the Shannon entropy. So the question is, what's the Shannon entropy? I hope I haven't jumped over too many slides. I have jumped over too many slides. Let's go back and find the Shannon entropy here.

For that particular case, if we compute the entropy, we get 1.626 bits. If you are communicating four possible grades for 1,000 students to the registrar, one way to do it would be to use two binary digits. You can cover all four grades, send 2,000 bits to the registrar.

The entropy says that you've got 1.626 bits per grade on average. So for 1,000 grades, you should be able to get something closer to 1626 bits. So can you communicate a set of 1,000 grades occurring with these probabilities with a code whose expected length is closer to the 1626? That's the task for designing a variable length code.

Now, it turns out that this task was set by Professor Fano, who was a Professor here, retired, but still comes to our weekly lunches, set as a term paper in the course he taught on information theory, actually just three years after Shannon's paper appeared. He posed the problem of designing a variable-length code whose expected length came as close as possible to the Shannon limit.

Huffman struggled with that almost to the end. Fano offered the option of doing a final exam if you didn't have a term paper. He was about to give up on it, and then came up with an idea that turns out to be the optimal variable-length coding scheme for this scenario.

So what he does is, just to very quickly finish with that, he takes the two lowest probability events, groups them together to make a single event that is C or D with probability 1/6. Then in that resulting reduced set, he looks at the two lowest probability events, combines to make them a meta-event with a probability that's the sum of the individual ones, and so on.

So he chases this procedure up-- take the two lowest probability events, combine them into a single one with the probability that's a sum of these individual ones, in the resulting reduced set look for the two lowest probability events, and so on. Build up a tree. The resulting tree then reveals the Huffman code.

The Hoffman code is guaranteed to have an expected length that satisfies this constraint, but actually has an upper bound, too. It's within entropy plus 1 on the upper side. We'll talk next time about how to improve this, but in recitation tomorrow, you'll get practice at a little bit more leisurely pace than I did here with constructing Huffman codes.