In this video, we will consider a classical application of Markov chains, which has to do with the design of a phone system. This is a classical problem, which was posed, analyzed, and solved by a Danish engineer by the name of Erlang. It was more than 100 years ago when phones just started to exist, but the technique remains relevant today to design systems of a similar nature.

As for Erlang, he was trying to figure out how to design the capacity of a phone system. That is, how many lines should we set up for a group of people, say, in a village, to be able to communicate to the outside world? So here is a cartoon of the problem, where these are the phone lines, and we need to decide how many of these lines to set up, let's say, B. How to do that?

Well, we don't want B to be too large, much more than needed, because too many lines would be expensive. On the other hand, we want to have enough lines so that if a reasonable number of people place phone calls during the same period, they will be able to talk and not get busy signals. So if B is 10 and 15 people want to talk at the same time, then 5 would get a busy signal, and that's probably not what you want as an acceptable level of service.

So we would like B to be just large enough so that there is a high probability that no one is going to get a busy signal. So how do we go about modeling a situation like this? Well, we need two pieces of information, one describing how phone calls get initiated, and once a phone call gets started, how long does it take until it ends?

We're going to make some very simple but somewhat plausible assumptions. We will assume that phone calls originate as a Poisson process. We will assume that out of that population, there is no coordination. At completely random times, people pick up their phone independent of each other's. Also, there is nothing special about the various times, and different times are independent. So a Poisson model is a reasonable way of modeling a situation under these assumptions.

We also assume that the rate lambda is known or has been estimated. Now, it might be the case that during the night, the rate would be different than during the day. In that case, you would design the system to meet the largest rate of the two.

For the phone calls themselves, we are going to assume that the duration of a phone call is a random

variable that has an exponential distribution with a certain parameter mu. So 1/mu is the mean duration of a phone call. Duration of phone calls are independent between each other. So here, again, we assume that the parameter mu has been estimated. For example, the mean duration 1/mu could be 3 minutes.

Now, is the exponential assumption a good assumption? So here is the PDF of an exponential random variable with parameter 1 over three. That means that the mean duration is about three minutes here. So if you look at this PDF, it means that most phone calls will be kind of short. There is going to be a fraction of phone calls that are going to be larger, and then a very small fraction of phone calls that are going to be even larger.

So it sounds reasonable. However, it's not exactly realistic in some situations. Typically, phone calls that last a very short time are not that common as opposed to what an exponential distribution would indicate here. So some other distribution might be better, like this one, for example, here, where during a very small period of time the wait corresponding to this very short period of time are kind of small as well.

There are many distributions of this type. I've just provided here one simple example. This one is the Erlang of parameter 2 and 2/3. What it means is that it is the sum of two independent exponential random variables, and each one of them of parameter 2/3. So the mean duration associated with this distribution is also 3 minutes. So this might fit better some practical situation. But here we will keep the simple assumption associated with an exponential distribution.

All right. So let's try now to come up with the models that we can decide how many lines, B, do we want to set up. The Poisson process run in continuous time. And call durations being exponential random variables are also continuous random variables. So it seems that we are in a continuous time universe.

Here is a cartoon of the evolution of the system. So here I have in blue when phone calls get initiated. So this is called 1, a second one, a third, a fourth, and a fifth one. And also, I have represented here the duration of the call. So call 1 lasted that long, call 2 lasted long until here, 3 up to here, 4 here, et cetera.

So when you look at this kind of system in that way, and you run through time in a continuous manner, and here you have 0 line busy. You have 1 line used, 0, 1, then 2 becomes busy, 3, 2, 1, and 0, and so

on and so forth. Also note that if I look at that system at any time t, because of our assumptions of a Poisson process and an exponential duration for phone calls, and a memoryless property associated with these processes, it means that the past really has no information about the future. And so, in some sense, the Markov property is valid.

So it looks like a continuous time Markov process would be needed here. And this is indeed an option, but we have not studied those in this class. So we will discretize time instead and work with a Markov chain. We are discretizing time in the familiar way, the way we did it when we studied the Poisson process. We are going to take the time axis and split it into little discrete time slots, each of duration delta. And delta is supposed to be a very, very small number.

So now under this discretization, by the definition of the Poisson process the probability that we'll see 1 arrival during any time slots of duration delta will be lambda times delta. Also, if at any time, like here we have 1 simple call active, the probability that this call will end during any future time slot of duration delta is mu delta, like here. Indeed, as we have seen in Unit 9, an exponential random variable can be thought of as representing the time until the first arrival of a Poisson process with rate mu.

What if you have i busy calls at the same time? Then the probability of having 1 call ending in a time slot of duration delta will be i mu delta. Like, for example here, this one could correspond to something as 2 mu delta. Indeed, each of the Poisson processes associated with these calls with rate mu can be combined into a merged Poisson process of rate i times mu. And a call completion will correspond to the time until the first arrival of this merged Poisson process.

For example, if I go back here in my situation here at time t, there were still 3 phone calls active. And I represent here the call number 2, call number 3, and call number 4 and their remaining duration. And if you look at these and you combine these 3 associated Poisson processes into 1, you get a merged Poisson process.

And if you look now at the time arrival of the first event, which would correspond to here, it would be an exponential random variable. The duration here would correspond to an exponential random variable of parameter 3 mu. So in that case, if you go back to that representation here, the probability of a departure would be 3 times mu times delta. OK?

So let us continue with our discrete time approximation of our system. Again, we have the village, and

the lines, the B that we would like to decide. We have discretized the time steps. We have made some approximation. And we know that during any of these time slots here, the probability that you would get a new call is about lambda times delta. Lambda is the rate of the Poisson process. And given that you have i calls, the probability that one of these calls ends will be i times mu times delta. OK.

If we want to propose a Markov chain model for this system, we need to specify the states and the transition probabilities. What are the states of the system? If you look at the system at any particular time, the minimum relevant information to collect would be the number of busy lines, something like these 2 lines are busy, or all of them are busy, or none of them are used.

Now, because of our assumptions, again, about the Poisson process arrivals and exponential duration of calls and their memoryless property, that information is enough to fully describe the state of our system in such a way that we get a Markov chain. So the states are numbers from 0 to B. 0 corresponds to a state in which all the phone lines are free. No one is talking. B corresponds to a case where all the phone lines are busy. And then you've got states in between.

Now, let us look at the transition probabilities. Suppose that right now, you are in that state. What can happen next? Well, a new phone call gets placed, in which case the state moves up by 1. Or an existing call terminates, in which case the state goes down by 1. Or none of the two happens, in which case the state stays the same.

Well, it is also possible that a phone call gets terminated, and a new phone call gets placed in the same time period. But when the duration of the time slots are very, very small, the delta here, this event is going to have a negligible probability, order of delta squared. So we ignore it, as we ignore the fact that more than one new call can happen, or more than one call can be terminated during a given slot.

So what is the probability of an upward transition? That's the probability that the Poisson process has an arrival during the slots of duration delta. And as we have seen, this is lambda times delta. So each one of these upward transitions has the same probability of lambda times delta.

How about phone call terminations? If we have i phone calls that are currently active, the probability that one of them terminates becomes i mu delta. So here it would be mu delta, and here B mu delta.

Now, let us analyze this chain. It has the birth and death form that we discussed in the previous lecture. So instead of writing down the balance equation in a general form, we think in terms of frequency of

transitions across some particular cut in this diagram, so for example here.

The frequency with which transition of this kind happen or are observed has to be the same as the frequency of transition of this kind. The frequency of transition of this type will be, if you look at pi i here and pi of i minus 1 here, this transition here will happen with pi i times i mu delta.

And the transition of this type here will be pi i minus 1 times lambda times delta. And the frequency of these transitions have to be the same as the frequency of these transitions, so we have that equals that. And then we can cancel the delta in both, and we are left with this equation here.

So this equation expresses pi of i in terms of pi of i minus 1. So if we knew pi of 0, then we can calculate pi of 1, and then in turn calculate pi of 2, and so on and so forth. And the general formula that comes out of this, after some algebra, is given by this expression, which involves pi of 0.

Now, what is pi 0? Well, we can find it by using the normalization equation, the summation of pi i equals 1. You use this normalization, replace each pi i by their quantities as a function of pi 0, and then we obtain this equation for pi 0. So here, again, we use that normalization. We replaced pi i by their value. We sum to 1, and we obtain pi of 0. And then in turn, from this pi of 0, you can replace the pi of 0 in pi of i, and you obtain a pi of i as a function of B, lambda, and mu.

So if we know B and lambda and mu, we can set up this Markov chain, and we can calculate pi 0, and then pi of i for all i's. We can then answer a question like this. After the chain has run for a long time, how likely is it that at any given random time, you will find the system with i busy lines? Well, it will be pi of i. And also, we can interpret the steady-state probabilities as frequencies. So once I found pi of i, it also tells me what fraction of the time I will have i busy lines. And you can answer that question for every possible i.

Now, we were initially interested in the probability that the entire system is busy at any point in time, in other words, in that state here. So if a new phone call gets placed, it is going to find the system in a random state. That random state is described in steady-state by the probability pi's. And the probability that the entire system is busy is going to be given by pi of B, and this is the probability that we would like to be small in a well-engineered system. So again, given lambda, mu, the design question is to find B so that this probability is small.

Could we figure out a good value for B by doing a back-of-the-envelope calculation? Well, let's suppose

that lambda is 30 calls per minute. And let's assume that mu is 1/3 so that the mean duration is 3 minutes. So on average, a call lasts for 3 minutes, and you get 30 calls on average per minute. Then how many calls would be active on the average?

If a call lasted exactly 1 minute, then at any time you would have 30 calls being active. Now, a call lasts, on the average, for 3 minutes. So by thinking in terms of averages, you would expect that, at any time, there would be about 90 calls that are active, 3 times 30. And if 90 calls are active on the average, you could say, OK, I'm going to set up my B to be 90.

But that's not very good, because if the average number of phone calls that want to happen is, on the average, 90, sometimes you are going to have 85, and sometimes you'll get 95. And to be sure that the phone calls will go through, you probably want to choose your B to be a number a little larger than 90.

How much larger than 90? Well, this is a question that you can answer numerically. By looking at these formulas, if you decide that your acceptable level of service, pi of B, has to be less than 1%, then you will find that the B that you need to design is to be at least 106.

So you actually need some margin to protect against a situation if suddenly, by chance, more people want to talk than on an average day. And if you want to have a good guarantee that an incoming person will have a very small probability of finding a busy system, here 1%, then you will need about 106 phone lines.

So that's the calculation and the argument that Erlang went through a long time ago. It's actually interesting that Erlang did this calculation before Markov chains were invented.