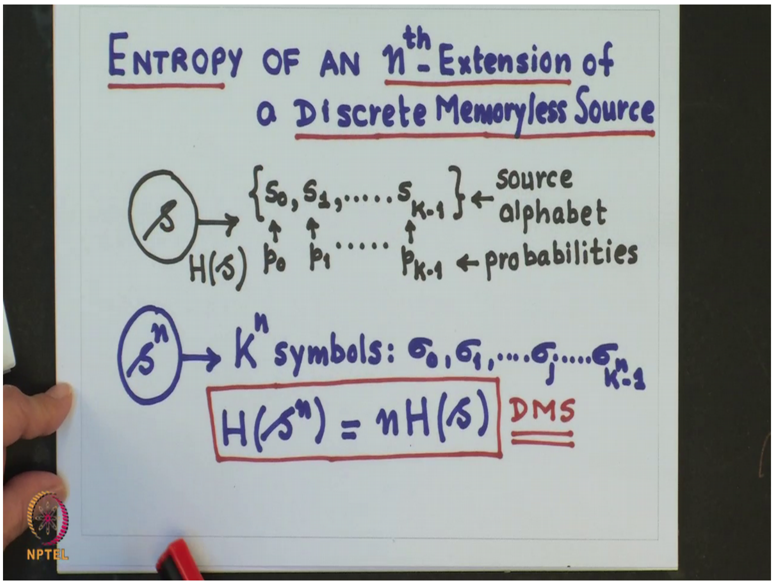
Principles of Digital Communications
Prof. Shabbir N. Merchant
Department of Electrical Engineering
Indian Institute of Technology, Bombay

Lecture – 03
Lossless Sources Coding Theorem

Hello, in the last class we studied entropy of nth extension of a source. What we said was as follows.

(Refer Slide Time: 00:33)

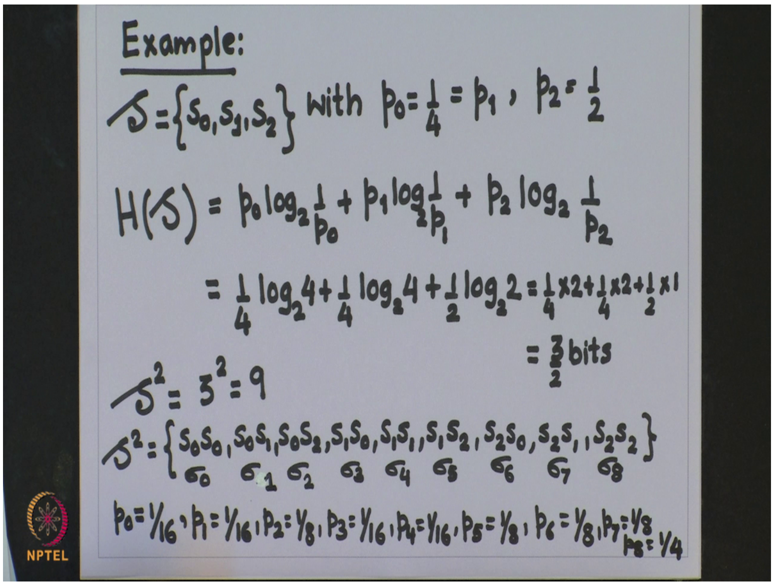


Given a source s with the source alphabet consisting of the symbols and the probabilities assigned to these symbols. What we said was that you can look at the sequence coming out of the source s in terms of group or block of n symbols or subsequence of length n symbols. And then we could consider this block or group of n symbols as new symbols which are the emitted or generated from an another source which we call as nth extension of the original source s.

So, we saw that this source nth extension will have K raised to n symbols and we derived an important relationship that is for a discrete memoryless source, the entropy of nth extension of a discrete memoryless source is equal to n times the entropy of the original source.

Now, let me take an illustrative example to demonstrate this and this will help us in getting a better understanding of this concept.

(Refer Slide Time: 02:00)

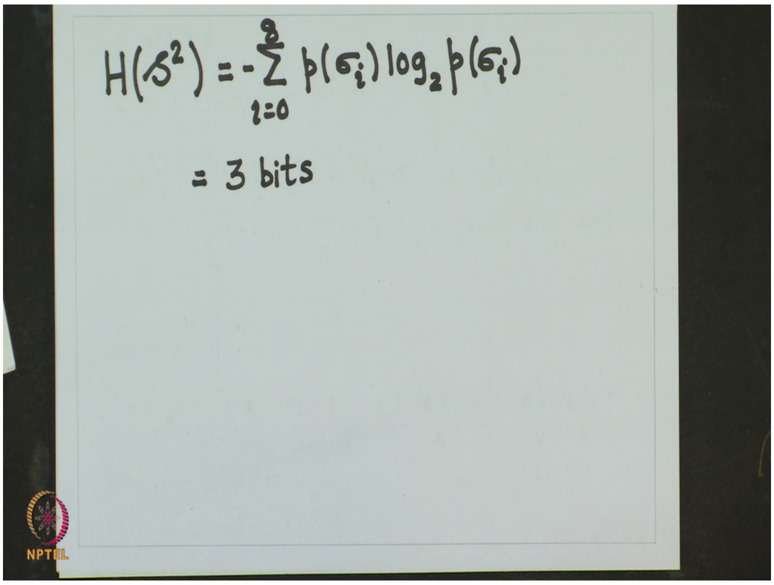


So, let us take an example, let us consider a source with source alphabet consisting of 3 symbols, with the symbol probabilities as follows. Given the symbol probabilities let us evaluate the entropy of the source, this is equal to one-fourth multiply by 2 plus one-fourth multiply by 2 plus half multiplied by 1, this reduces to 3 by 2 bits.

Now, we will evaluate the entropy for the second extension. So, in this case K is equal to 3 and n is equal to 2. So, we will get 9 symbols for the second entropies. The symbols are denoted as follows S 0 S 0, S 0 S 1, S 0 S 2, S 1 S 0, S 1 S 1, S 1 S 2, S 2 S 0, S 2 S 1, S 2 S 2; And the corresponding probabilities for this 9 symbols this sigma 1, sigma 2, sigma 3, sigma 4, sigma 5, sigma 6, sigma 7, sigma 8.

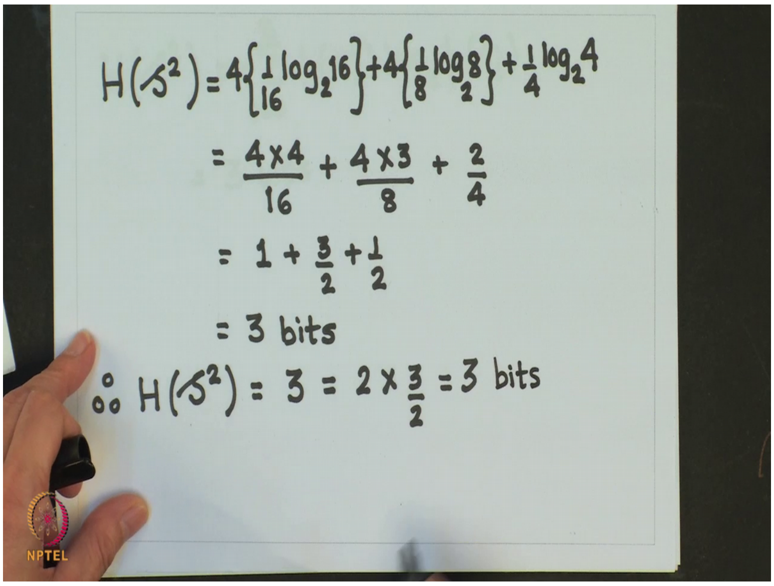
And the probabilities for this symbols would be as follows. We are assuming independent symbols p 8 is 1 by 4. This is sigma 1 and now if we evaluate the entropy of this source we would get it as i is equal to 0 to 8. And then you substitute this values which we have evaluated earlier and if you do that you will get it as 3 bits.

(Refer Slide Time: 06:30)



So, what it shows basically the detailed calculation is just shown on this slide, if you just plug in the values you will get it equal to 3 bits.

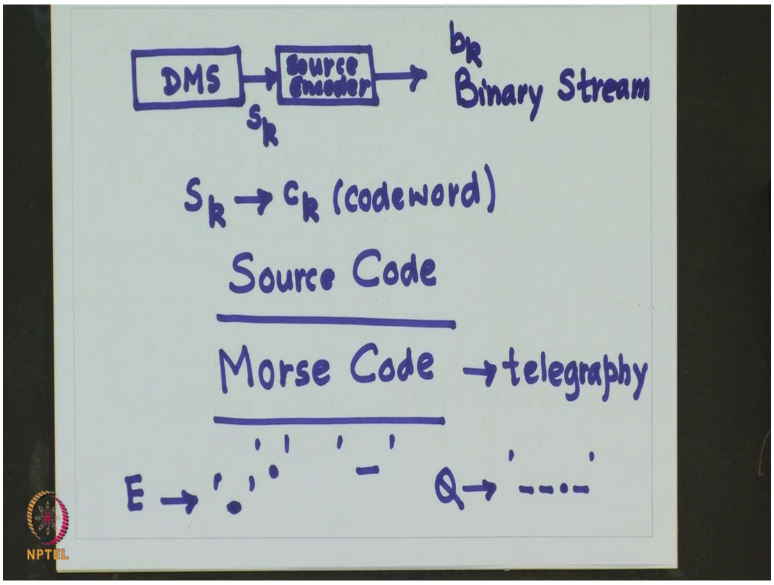
(Refer Slide Time: 07:20)



So, what we have shown that entropy of the second extension of the source turns out to be 3 bits which is equal to n times that is 2 multiplied by the original entropy that is 3 by 2 and is equal to 3 bits, ok, its fine.

We had mentioned earlier that one of the key issues in evaluation of the performance of a digital communication system is the efficiency with which the information from a given source can be represented. Let us address this issue little more in depth. The process of efficient representation of the discrete data which is generated by a discrete information source is known as source encoding. And this task is performed by the device that is known as source encoder.

(Refer Slide Time: 08:43)



So, the input to the source encoder is symbols and these symbols come as an output of a discrete memoryless source and the task of the source encoder is to assign what is known as a code word to each of the source symbols which are input to this source encoder.

So, S K, it gets map to what is known as a code word. This code word is nothing, but a sequence of symbols which are selected from another alphabet called code alphabet, and the symbols for this code alphabet is usually binary that is binary digits 0 and 1. And this set of the code words which we will have for all the source symbols in the source alphabet is known as source code.

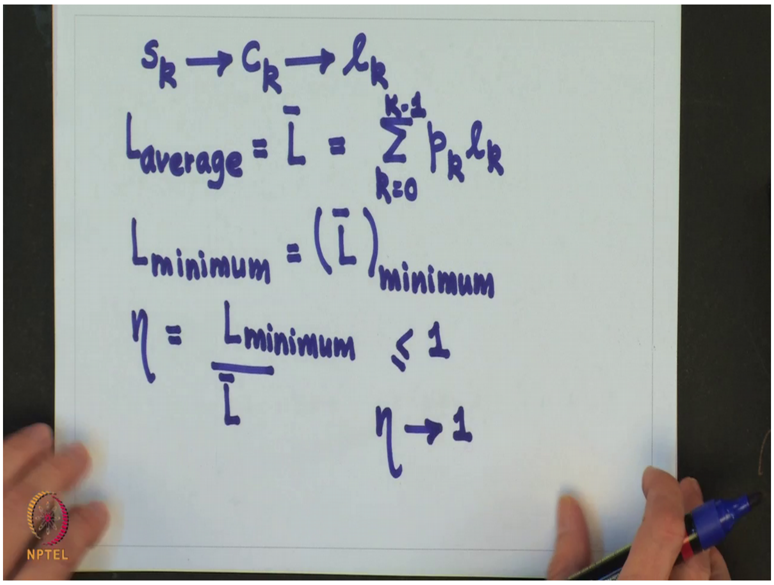
We will see that for an efficient source encoding the source encoder will exploit the source statistics. The source statistics will be exploited as follows. The short code words will be assigned to those source symbols which occur frequently, and long code words would be assigned to those symbols source symbols which occur infrequently. In the process what we will generate is variable length code, what we mean by this is that the code words in this set code source code will not have fixed length the length of each code word could be different, ok.

An example of this variable length code which is also known as VLC is a Morse code. This was used in the past way back in 19 century for telegraphic purposes. Morse code basically uses dot and dashes to encode the letters from the English text. If you take a character or letter e in English texts this occurs very frequently and the Morse code codes it as just one single dot, and if you take another letter say letter Q which does not occur. So, frequently this is coded as two dashes, followed by a dot followed by a dash.

Above primary interest in the formulation of a source encoder would be such that it satisfies the following two requirements. One the code words are in binary form and the second the source code is uniquely decodable. What we mean by this is that we should be able to obtain the source symbols from the output of the encoder.

So, as we said earlier that since the code words are using the binary symbols from the code alphabet the output stream from the source encoder would be a binary stream [music]. So, we had mentioned earlier that we want our source encoding to be efficient. So, the question is now what is an efficient source encoder. Let us address this issue.

(Refer Slide Time: 13:46)



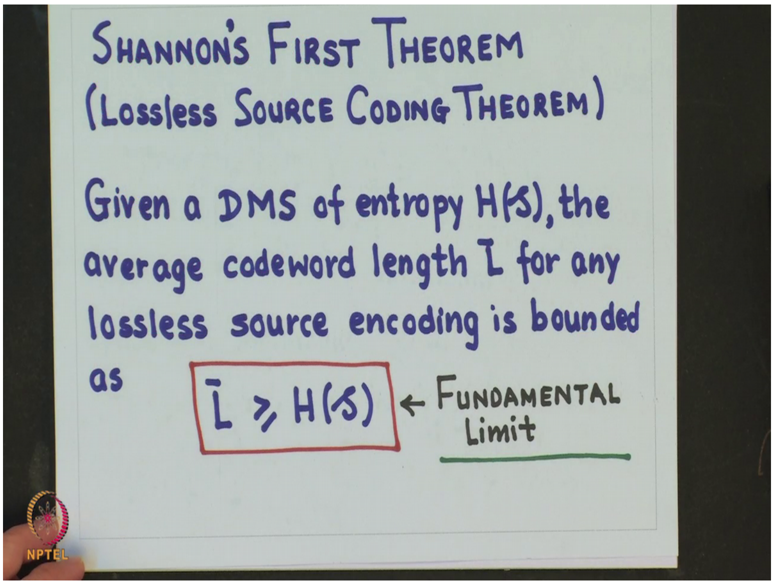
Now, let say that we have a source symbol S k, source encoder assigns the code word to this C k and this is nothing, but sequence of binary digits which we said earlier we will call it as binits. So, this sequence has a length which will indicate l k. So, l k denotes the number of 0s and 1s in the sequence which is assigned to the code word C k, ok.

Now, we can calculate L average, which is the average length of the code word and it is also known as the average length of the code let us indicate by L bar, this will be equal to p k l k k is equal to 0 to capital K minus 1, ok. So, for the given source this is the average length we will get if the source encoder has assigned the lengths to the code words as shown here, ok.

Now, let me assume that L minimum is equal to l bar which is minimum. So, what I am saying is that there exists L average which is minimum. Then the efficiency of a source encoder is defined by eta is equal to L minimum by L average this quantity will be always less than or equal to 1. So, what we want an efficient source encoder to do is that eta should tend towards 1.

So, the next question is how do you find out L minimum for a given source. Can you ensure that that exists L minimum for a given source? And the answer to this basically is provided in the form of Shannon's first theorem which is also known as lossless source coding theorem.

(Refer Slide Time: 16:30)

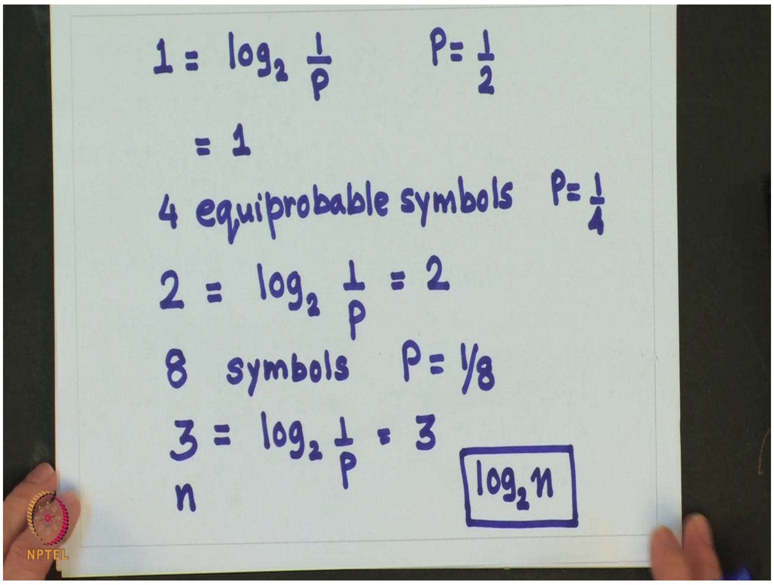


And the theorem says that given a discrete memoryless source of entropy H s then the average code word length for any lossless source encoding is bounded as follows. It is important to note that we are talking about the lossless source encoding and it says that the average length has to be always greater than equal to the entropy of the source and this is the fundamental limit which we will have on the data representation of the information which is being emitted from a given discrete information source.

Now, we will not go into the rigorous proof of this theorem the rigorous proof can be found in one of this books elements of information theory by cover. And Thomas published by Wiley and information theory and reliable communication by Gallagher again published by Wiley. What follows is basically an informal proof which helps us to appreciate this Shannon's first theorem, ok.

So, let me assume that I have a discrete memoryless source and it emits two equiprobable symbols.

(Refer Slide Time: 18:32)



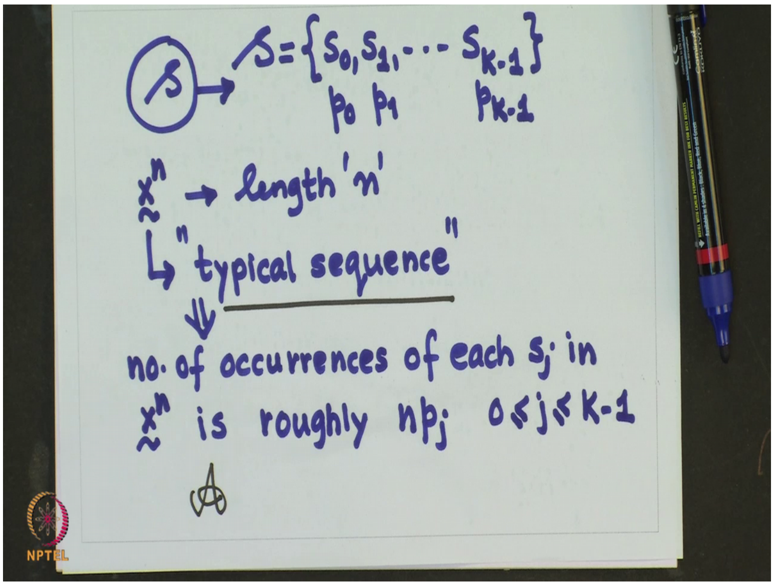
Now, to represent the output of this source we need one binary digit, and that is equal to log of p where p is the probability of occurrence of the symbols from this binary source and we have assumed it is equiprobable probable. So, this is equal to half. So, what I get is equal to 1. So, let us say that I have a source which generates 4 equiprobable symbols.

So, now, p is equal to one-fourth. Now, to represent this 4 equiprobable symbols using binary digits we know that we would require 2 binits and this is again equal to log of is equal to. And if we extend to 8 symbols equiprobable symbols then I will have p is equal to one-eigth and we know that 2 represent 8 symbols I need 3 binits and this is again equal to log of 1 by p and is equal to 3.

So, in general if I have n equiprobable symbols then the number of binits required to represent the outputs from this source would be equal to log to the base 2 of 1 by probability. And since there are n equiprobable symbol the probability is equal to 1 by n, so this will turn out to be log to the base 2 n ok.

So, this we have taken a very restricted case. Now, let me try to extend this to a source a discrete memoryless source again where it generates K symbols and the occurrence of each of these symbols is independent and the probabilities are unequal. So, let us see how do we would carry out the coding for this case ok.

(Refer Slide Time: 21:10)

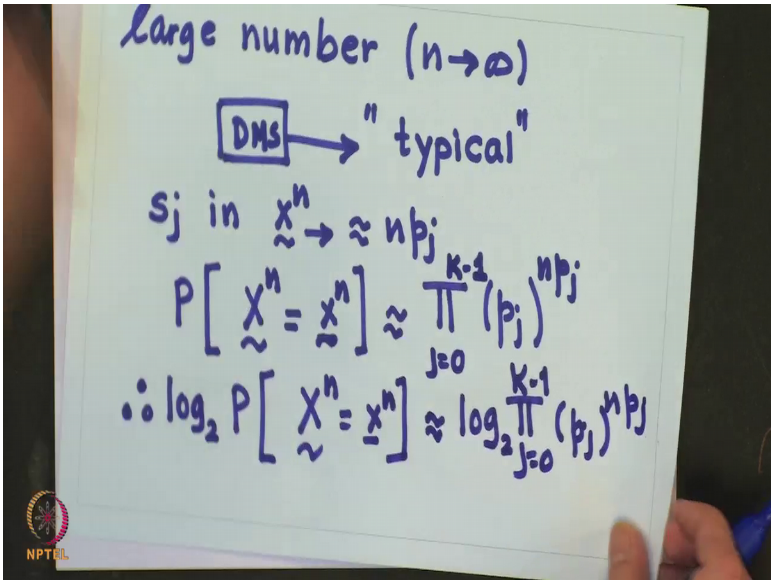


So, I have a source it generates S 0, S 1 repeat. So, I have a source it generates K symbols with the probabilities p 0, p 1, and p K minus 1.

Now, let us consider a sequence of length n and denote this sequence as a typical sequence. What I mean by typical sequence is that number of occurrences of each symbol S j in this sequence of length n is roughly n times p j and this is true for all j greater than equal to 0 and less than equal to K minus 1.

So, what we mean by this typical sequence is that the proportion of K symbols in all the typical sequence will be the same, but in general the order of the symbols in which they will appear in this typical sequence will be different ok. And the set of all typical sequence is denoted by A.

(Refer Slide Time: 23:44)

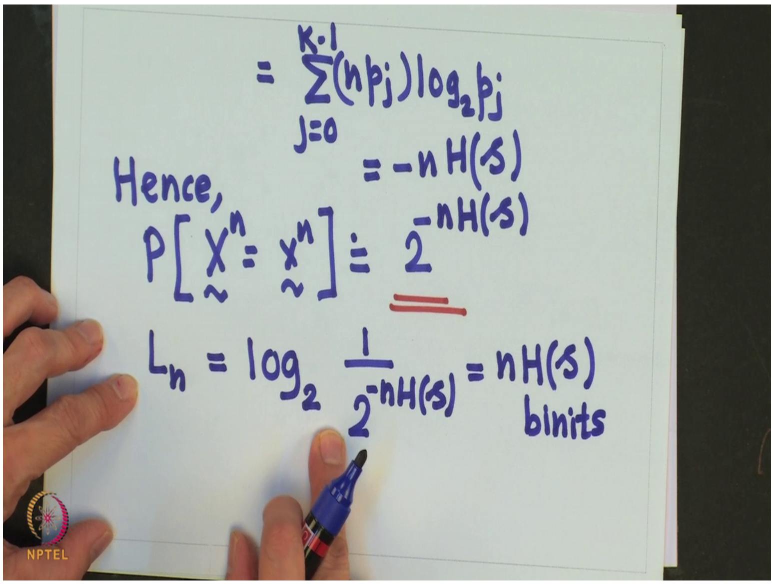


Now, by the law of large numbers which states that with high probability approaching 1 as n tends to infinity the output of the discrete memoryless source which we have talked of will be typical, correct, ok.

Now, since the occurrence of S j in this typical sequence is roughly equal to n p j and since the discrete memoryless shows the probability of observing a particular typical sequence is as follows. This is approximately equal to each symbol occurs with the probability p j there are n symbols.

So, the probability of occurrence of all those n symbols that is S j is equal to this, and since the source is discrete memoryless source this will be equal to the expression given here therefore, I can write this log to the base 2 of.

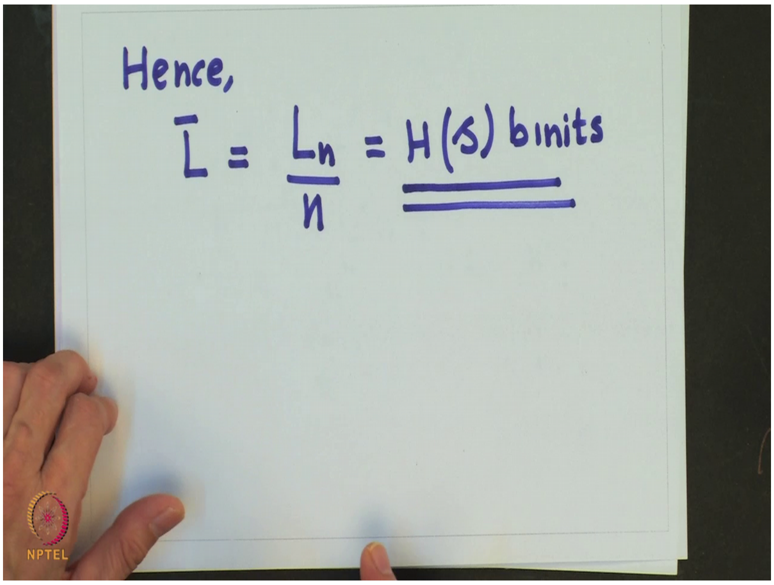
(Refer Slide Time: 26:00)



So, which I can simplify equal to, and hence from the above expression the probability of occurrence of any typical sequence turns out to be equal to if you simplify this is nothing, but n times H s this turns out to be 2 raised to approximately 2 raised to minus n H s.

So, what it states that all typical sequences have roughly the same probability and this common probability is given by this, correct. So, if we consider this long typical sequences as new symbols which are now equiprobable then to encode such typical sequences we need L n binary digits of binits and this will be equal to log to the base 2 of 1 by 2 raise to minus n H s which is equal to n H s binits. Note that this L n is the length that is number binits of the code word required to encode n symbols in sequence.

(Refer Slide Time: 27:50)



Hence L average number of binits required per symbol is equal to L n by n and this will be equal to H s binits. So, what we have shown that by encoding and successful symbols it is possible to encode a sequence of source symbols using on the average H s binary digits per symbol where H s is the entropy of the discrete memoryless source in bits.

So, we mentioned earlier that one of the requirement of a source encoder is to provide uniquely decodable source code. We will examine this in detail in the next class.