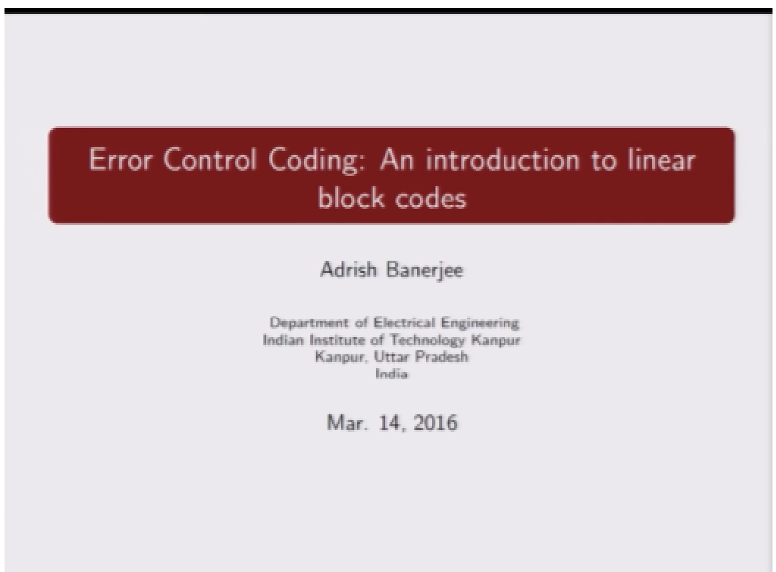
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture – 1A
Introduction to Error Control Coding-I

by
Prof. Adrish Banerjee
Department of Electrical Engineering, IIT Kanpur

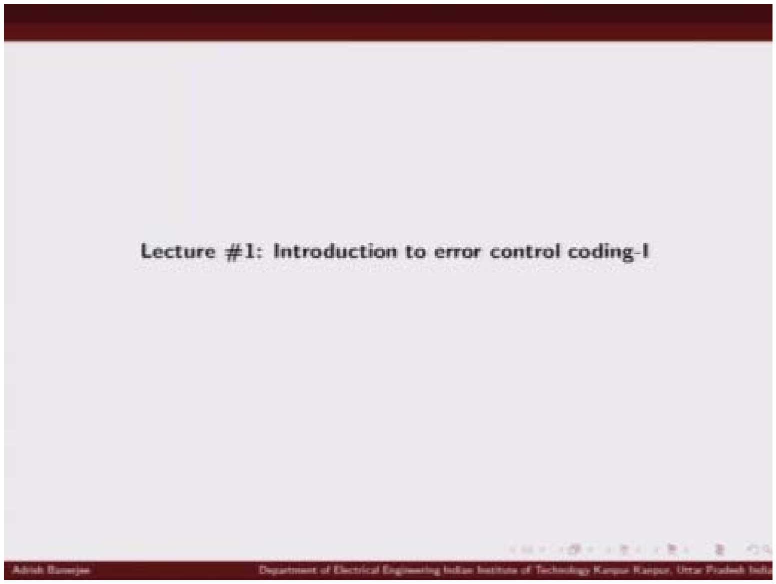
Welcome to the course on error control coding.

(Refer Slide Time: 00:17)



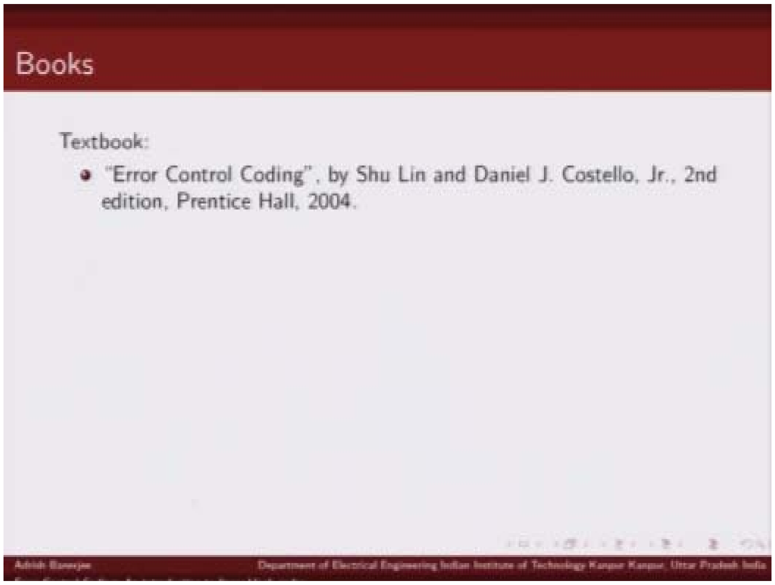
An introduction to linear block codes, I am Adrish Banerjee from IIT, Kanpur. Today we are going to talk about basic introduction to what coding theory is all about. So we will start our lecture.

(Refer Slide Time: 00:31)



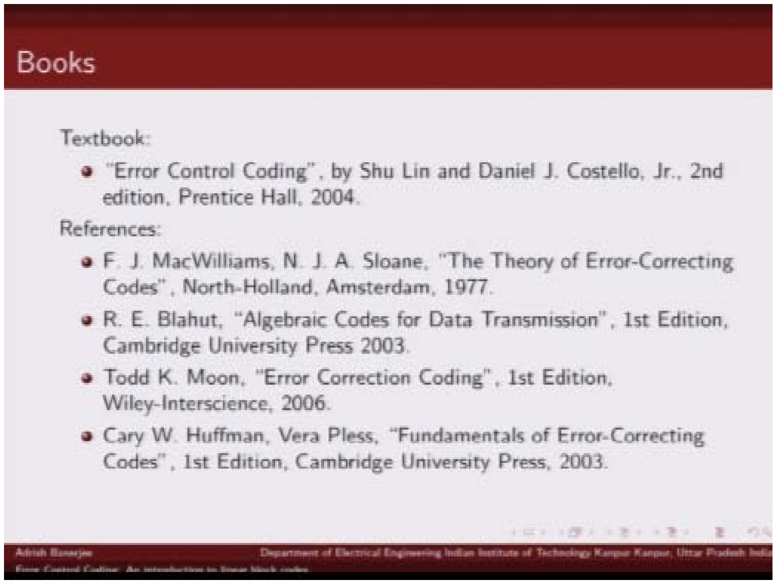
So in the introduction as I said we will talk about what is coding theory, we will illustrate with a very simple example how error correcting codes can be used for error detection and error correction. Before I start my lecture, I would like to talk about the books that we are going to use for this course.

(Refer Slide Time: 00:56)



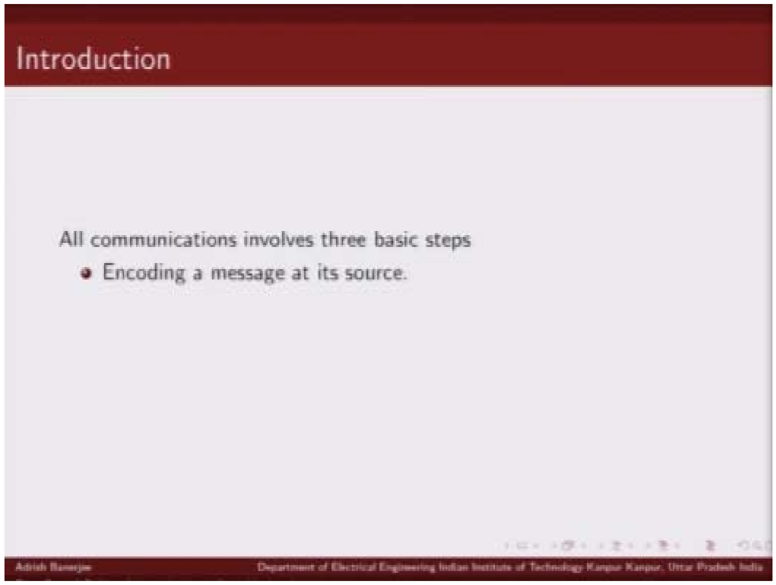
So we are going to follow this book, error control coding by Lin and Costello, the second edition of this book, we are going to follow this book as our textbook.

(Refer Slide Time: 01:08)



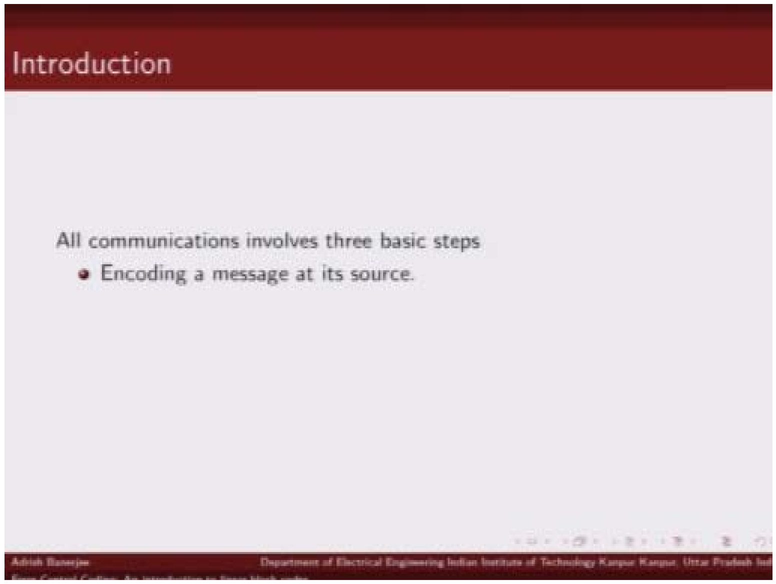
And there are some very nice books which you can use as a reference book, for example this book by Sloane and MacWilliams is a very nice book on block codes. You could also follow this book by Blahut on Algebraic Codes for Data Transmission. This book error control coding by Todd K. Moon, this is also gives a very nice introduction to error correcting codes. Or you could use this book by Huffman and Pless called fundamentals of Error Control Codes.

(Refer Slide Time: 01:42)



So when we talk about communications, communications basically involves three basic steps.

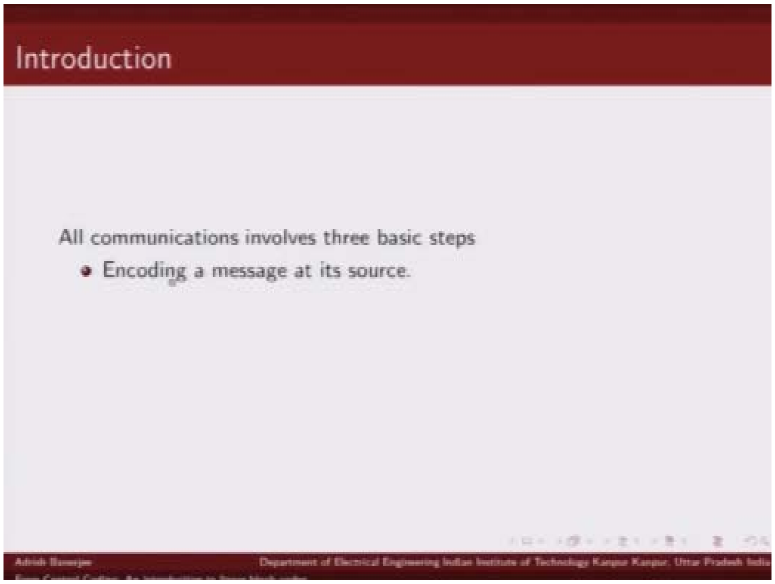
(Refer Slide Time: 01:53)



The first is encoding a message; you have a message source that you want to represent efficiently. Now we can for example consider a speech signal, now if you want to transmit a speech signal, you first have to convert your analog signal to digital signal and then you need to get rid of useless redundancies.

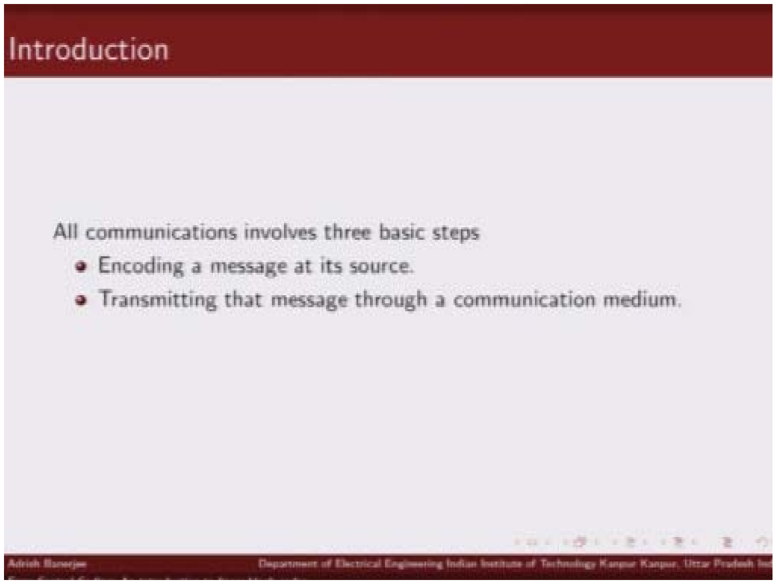
Why, because we want to transmit basically useful information at the – we want – a source inherently has lot of redundancy, and when we try to represent a source we would like to represent a source efficiently in be known number of possible bits.

(Refer Slide Time: 02:34)



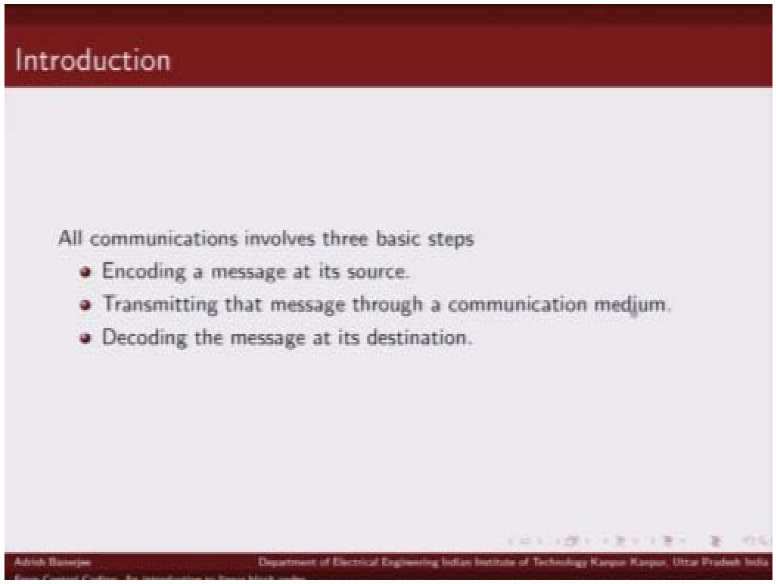
So first step involved in any communication is basically encoding a message.

(Refer Slide Time: 02:39)



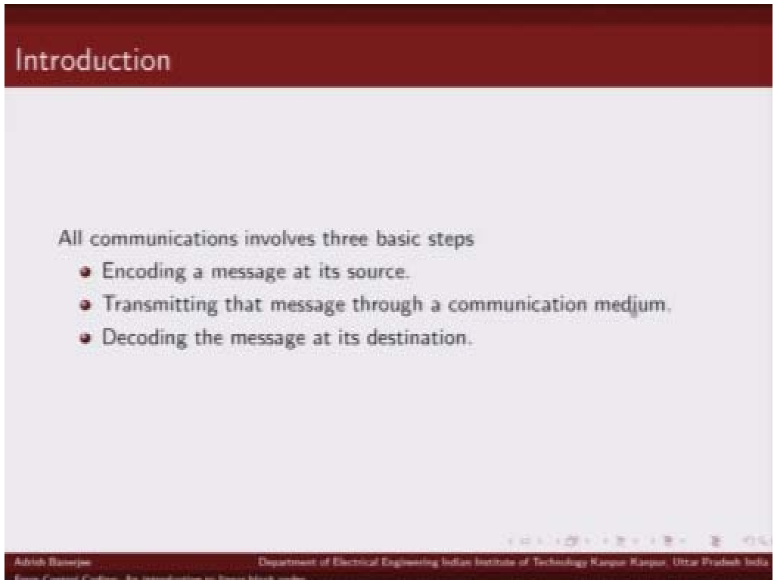
Second is once you have represented your source you want to transmit that source over a communication channel. So the second thing is transmission of the message through a communication channel

(Refer Slide Time: 02:54)



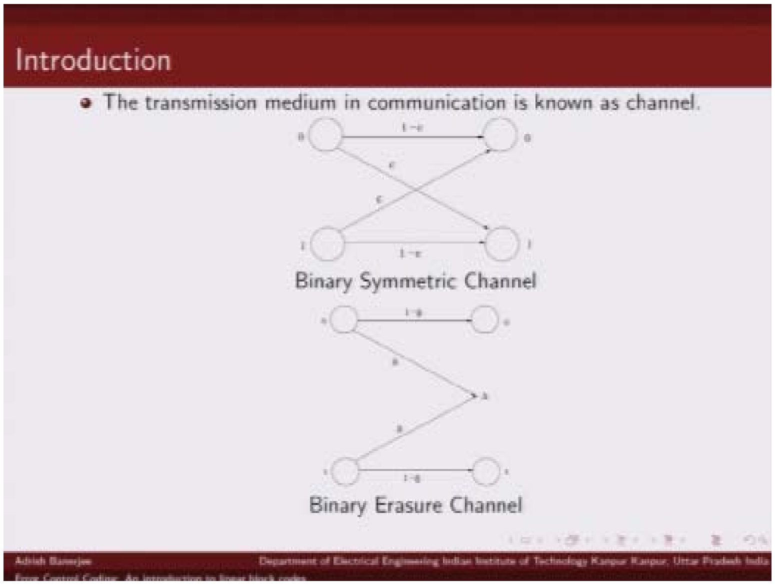
And finally once the receiver received the message it has to decode to find out what was the information that was transmitted. So broadly there are three steps involved in communication encoding, transmission, and then finally decoding.

(Refer Slide Time: 03:14)



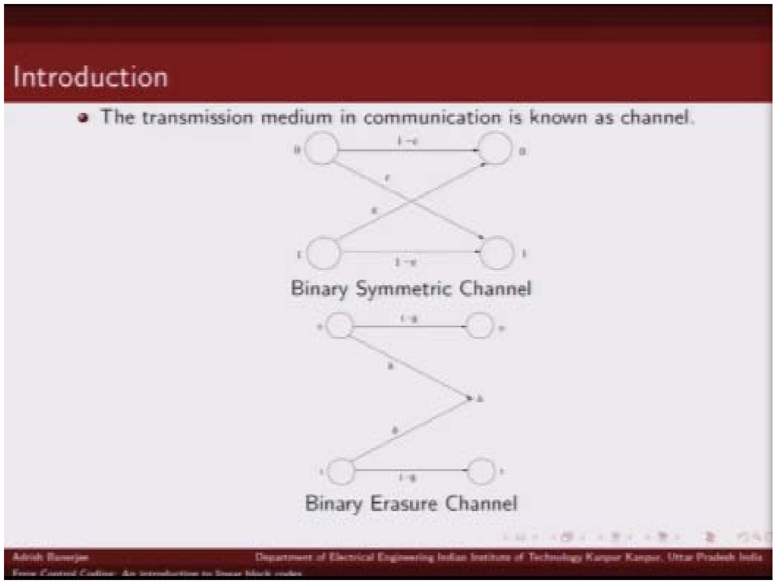
So information theory basically gives us fundamental limits of what is the maximum limit of, like what is the maximum best compression fundamental limit and compression that we can achieve. It also gives us fundamental limits on what is the maximum transmission rate possible over a communication channel.

(Refer Slide Time: 03:37)



So let us spend some time on what is our transmission medium, so the transmission medium over which we want to send a packet that is known as channel and here I have illustrated

(Refer Slide Time: 03:49)

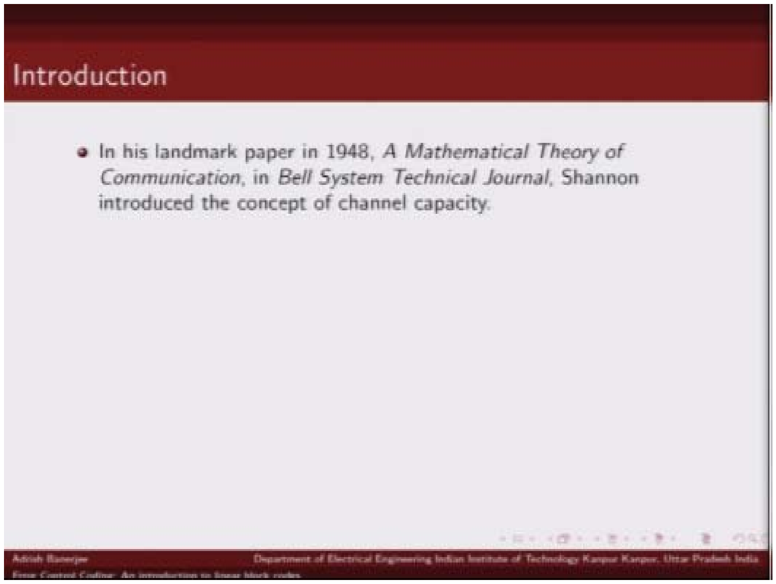


To very simple channel models, the first which is binary symmetric channel, now you can see there are -- it has binary inputs, zeros, and one and similarly it has binary output 0 and 1. Now with probability 1- ε basically whatever you transmitted received correctly at the receiver. So this is a transmitter and this is the receiver side. So if you transmit 0 with probability 1- ε you will receive it correctly, similarly if you transmit 1 with probability 1- ε you will receive it correctly.

And this crossover probability of error is basically given by ε. So this is basically a symmetric channel and this is a binary channel, because the binary input, binary output is known as binary symmetric channel. Another channel which basically is very commonly used to model packet data networks is what is known as binary erasure channel.

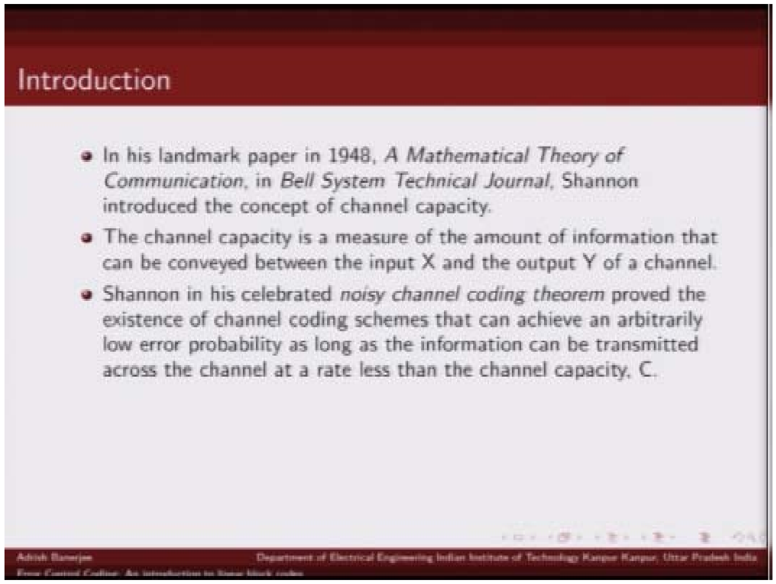
So they are binary inputs 0's and 1's and the outputs are either you receive whatever has been transmitted you receive it correctly or whatever you have transmitted is basically erased. So this Δ that you see basically we are denoting an erased bit using this symbol. So with probability 1-Δ you receive the bit correctly and with probability Δ the bit is erased or lost.

(Refer Slide Time: 05:21)



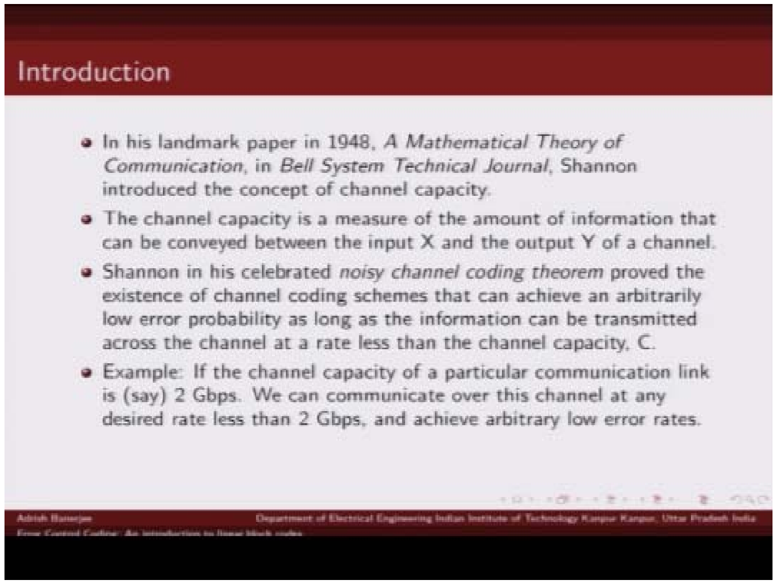
So in his landmark paper in 1948 Shannon introduced this concept of channel capacity, that what is the maximum rate at which we can communicate over a communication link. So a channel capacity is defined as the maximum amount of information that can be conveyed from the input to the output of a channel

(Refer Slide Time: 05:35)



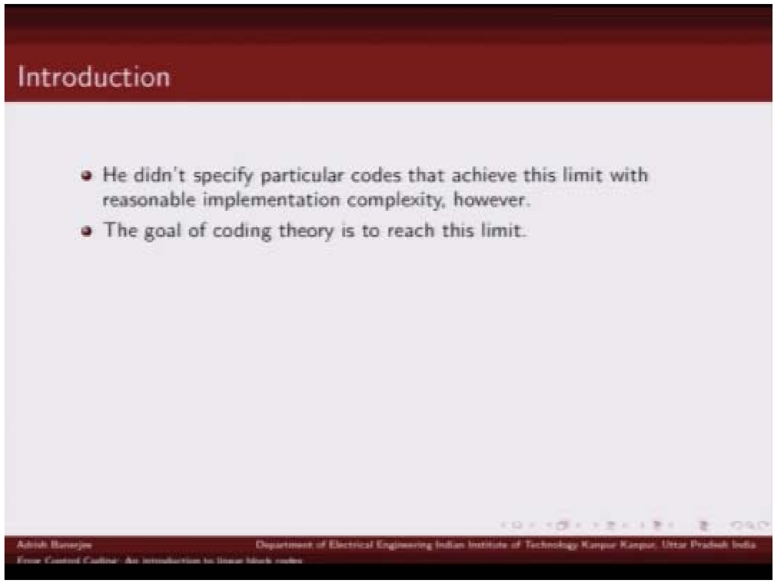
Shannon in his theorem also proved there exists channel coding schemes that can achieve very low arbitrarily very low probability of error as long as the transmission rate is below channel capacity. So Shannon showed that there exists good channel codes as long as the transmission rate is below channel capacity we can achieve arbitrary low probability of error.

(Refer Slide Time: 06: 20)



For example if we talk about channel capacity of a particular link to be 2 Gbps bits per second then basically we should be able communicate at rate any rate up to 2 Gbps or what this communication link without basically and I can achieve very low probability of error at the decoder .

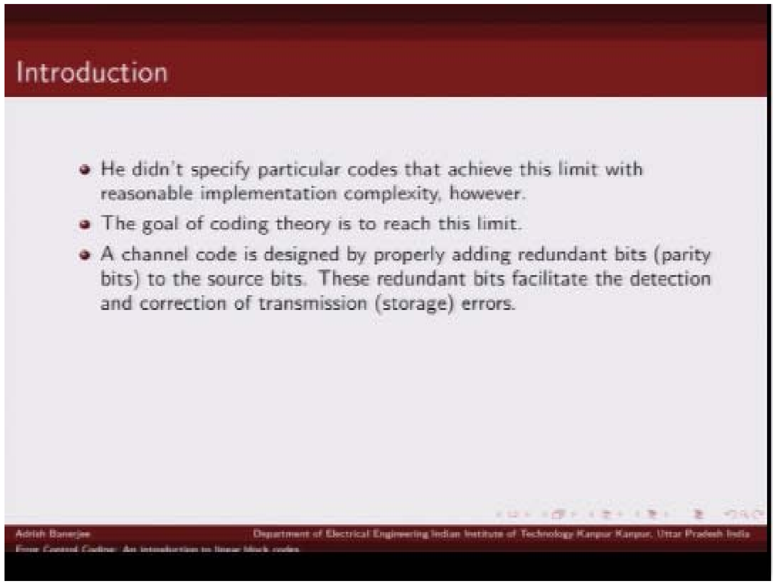
(Refer Slide Time: 06: 39)



Now in his theorem Shannon did not specify how to design such codes which have rate close to capacity and that is where basically error control coding comes into picture, so the goal of error correcting coding theory is to achieve this to design codes which can achieve this limit so basically Shannon has mentioned that we could transmit, we could design as long as we design error correcting codes which have rate less than channel capacity we can at achieve arbitrary low probability of error.

So the goal of the coding theory or error control coding is to design such error correcting codes with rates as close to capacity which can achieve arbitrary low probability of error.

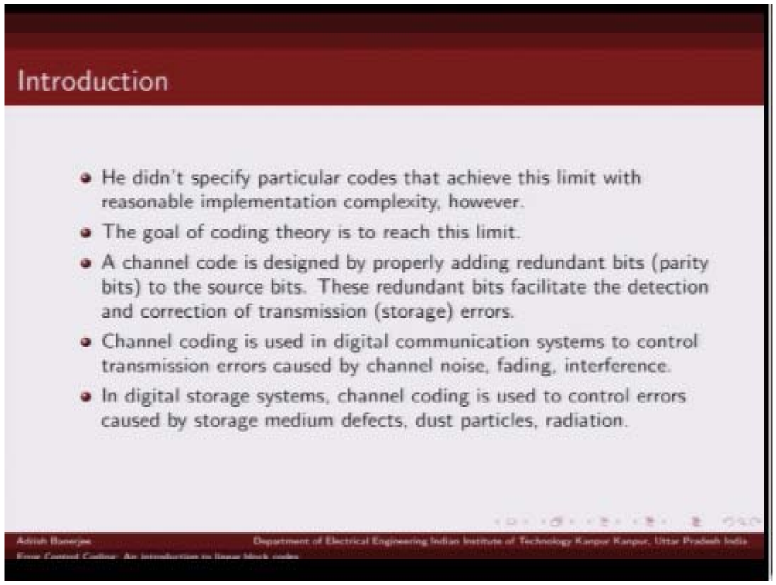
(Refer Slide Time: 07: 36)



And Shannon did not specify how to design such code so basically where coding theory is come into picture, so how do we design an error correcting code? An error correcting code is designed by adding some redundant bits to your message bits, the message bits we call them information bits and those additional redundant bits that we add those are known as parity bits, so error correcting code is designed by properly adding some redundant bits to your message bit and then send this coded massage over a communication link, now we use this.

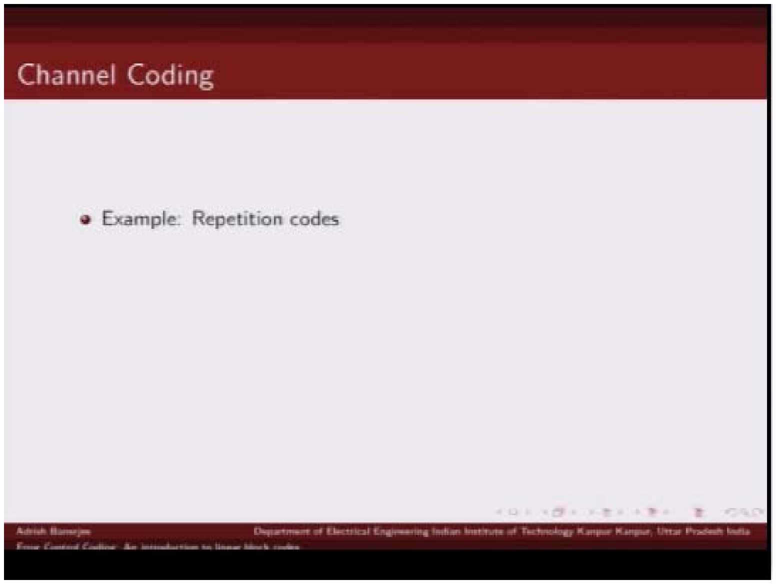
Additional redundant bits to detect error and to correct error.

(Refer Slide Time: 08: 25)



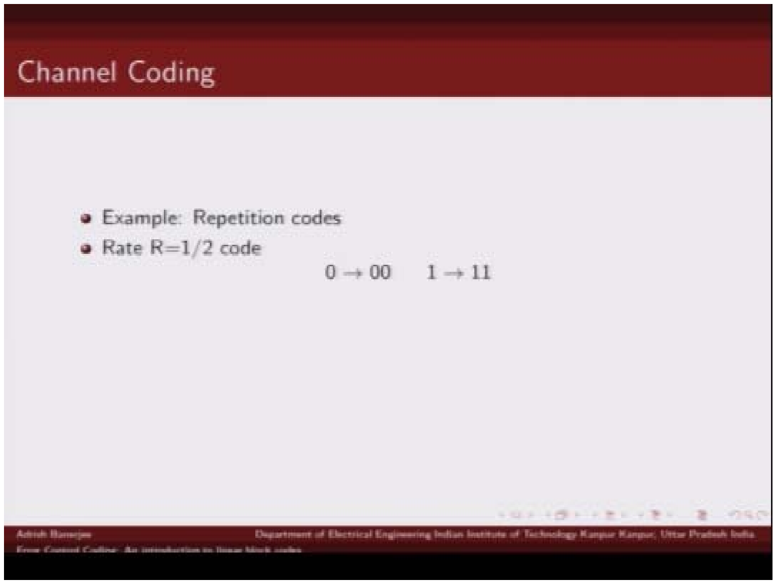
Error correcting codes has wide range of applications in digital communication and storage, I have listed basically a few of the uses for example, when we send a signal over communication link it is get corrupted by noise, fading, interference so to combat the effect of all these basically we use error correcting codes to correct the errors. Similarly in digital storage system you want to correct the error cost due to storage media defects, dust particles, radiations, we use error correcting codes there.

(Refer Slide Time: 09: 04)



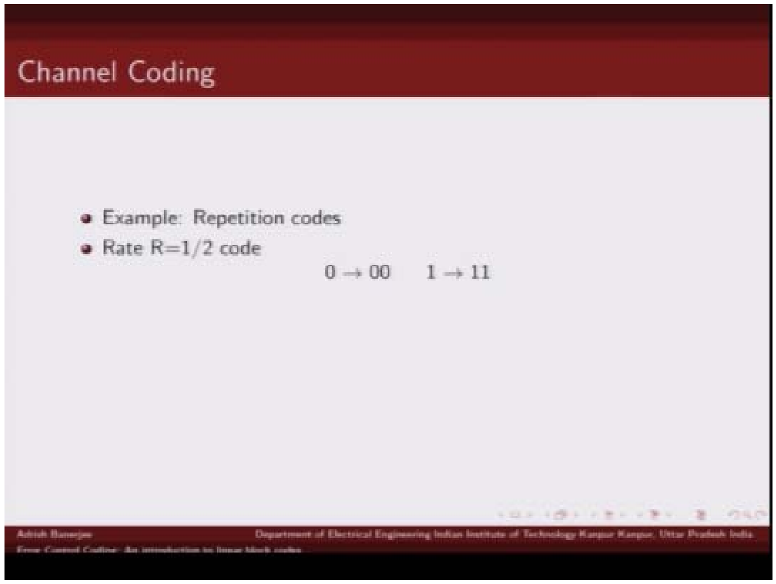
So let us take a very simple example of error correcting codes and illustrate.

(Refer Slide Time: 09: 13)



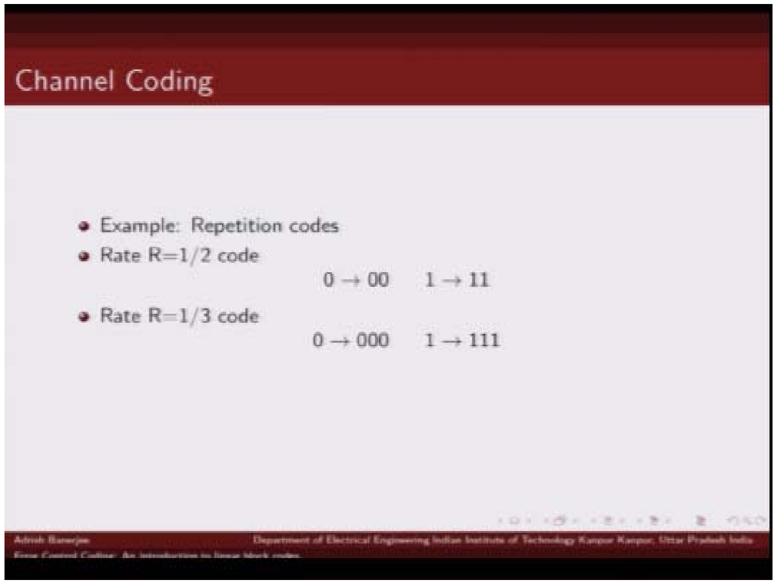
How we can use error correcting codes to detect and correct errors. So example I am going to show you right now is of what is known as repetition code, so the rate is defined as the ratio of number of information bits to number of coded bits, so when I say rate one half code I mean there is one information bit or one message bit and there are 2 coded bits, for example in a repetition code

(Refer Slide Time: 09: 49)



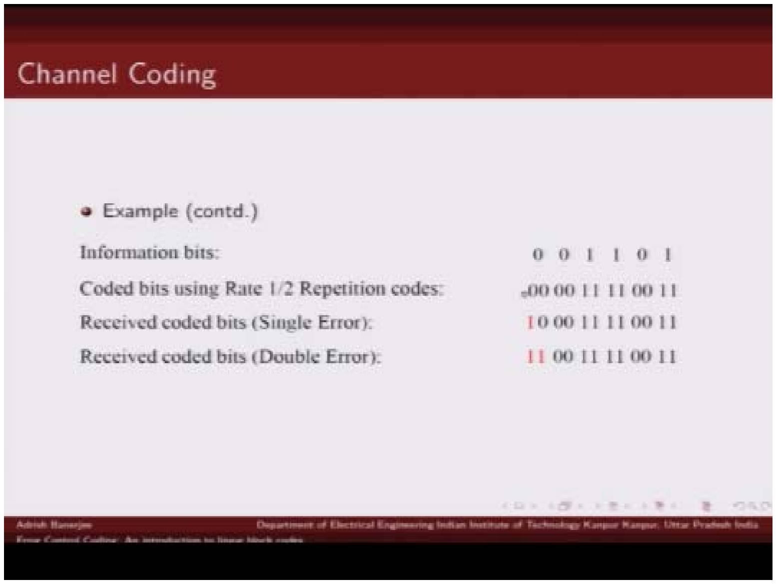
The binary repetition code basically we repeat whatever the information bit is, so a rate half repetition code would be look like this a binary rate half repetition code would look something like this, so for zero we would be transmitting zero, zero and for one we would be transmitting one, one.

(Refer Slide Time: 10: 10)



Similarly for a rate one third repetition code, for zero we will be transmitting zero, zero, zero, and for one we will be transmitting one, one, one. So you can see here in this in rate one half case we are adding one additional redundant bit and in case of rate one third code basically we are adding 2 additional redundant bits. Now how we are going to make use of these redundant bits for error correction error detection, that will be explained in the next line.

(Refer Slide Time: 10: 50)



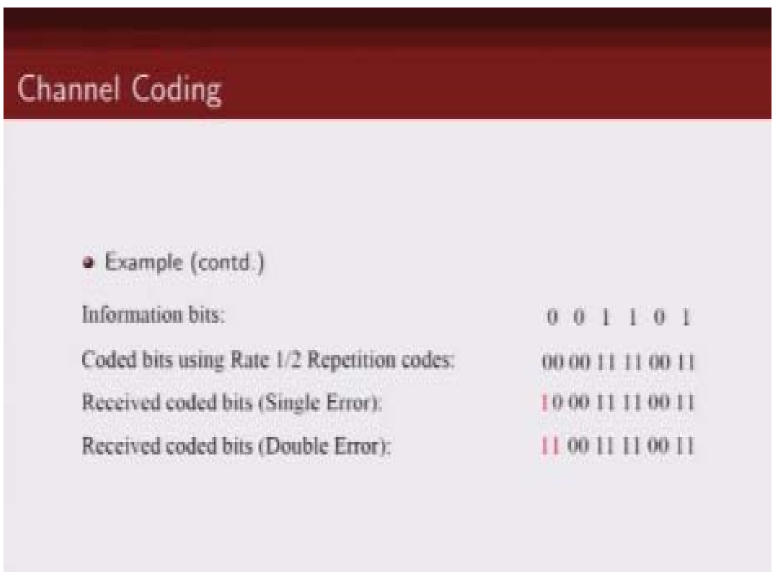
So let us take this example, let us say I want to transmit these set of bits so I want to transmit zero, zero, one, one, zero, one. Now if I use a rate one half repetition code what would be my coded bits? For zero I will be transmitting zero, zero, for zero I will be encoding them as zero, zero, for one I will be encoding them as one, one, and for one I will be sending them as one, one for zero I will be sending as zero, zero and for one I will be sending as one, one .

So this will be my coded sequence okay, now here I have illustrated one case where there is a single error so this was basically the sequence which was transmitted, think of it that this base sequence has been transmitted over a binary symmetric channel and this is what the receive sequence I received. So you can see here this a case of single error, the first bit which was transmitted 0 was received as 1. Now how can I use error correcting codes to detect error? So since it is a rate 1 ½ code for each information bit I am sending two coded bits so at the receiver I will look at two bits at a time, so I will look at first I will look at this 1 0.

Now since this is a repetition code what do you expect? I expect that both the bits should be same right? But here in this case first bit is 1 second bit is 0, which means there is a transmission error so I am able to detect single error, how? Because these bits were encoded using rate half repetition code I expect these two bits to be same, so I know there is an error in the first bit but I do not know whether this is bit 0 or bit 1.

Let us look at other received bits 00 this will be decoded as 0, 11 this will be decoded as 1, 11 this will be decoded as 1 there is no ambiguity, 00 this will be decoded as 0, again there is no ambiguity and 11 this will be decoded as 1, so we can see that using one additional redundant bit we are able to detect single error.

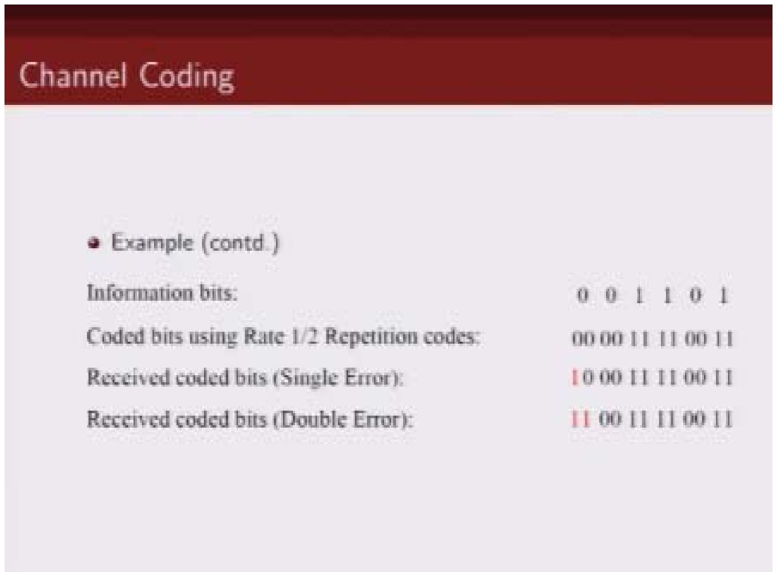
(Refer Slide Time: 13:43)



Now let us look at example for double error, so let us say the first and second bit are received in error so what we have received is basically 11 00 11, 11 00 11, so the first two received bits are in error 11, now let us see whether we can detect error using this rate ½ repetition code, so again we will follow the same logic for decoding, we will look at two bits at a time so first two bits are 11.

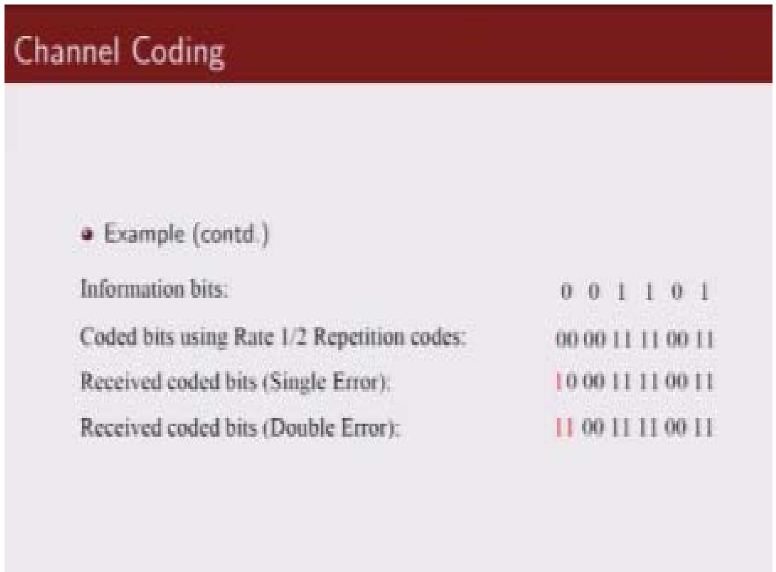
Now since these bits are same we will decode them as 1, but what was transmitted, it was zero. So we can see that this is the case of undetected error, even though these two bits were received in error the decoder is not able to detect this error. So this kind of thing happens when the error pattern is such that it transforms one code word into some other code word.

(Refer Slide Time: 15:00)



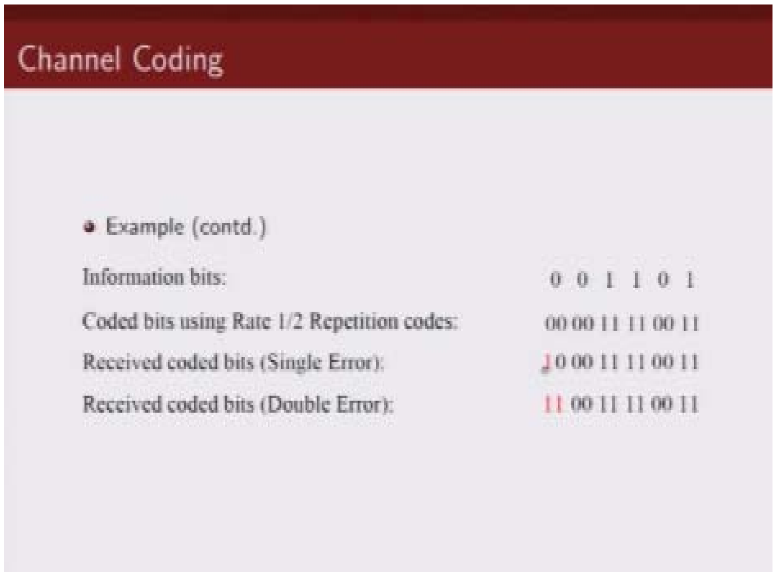
So since 11 is a valid code word for 1 the decoder is not able to detect this error, so this rate 1 ½ repetition code is able to detect single error but it is not able to detect double errors, now let us look at whether it can correct any errors.

(Refer Slide Time: 15:25)



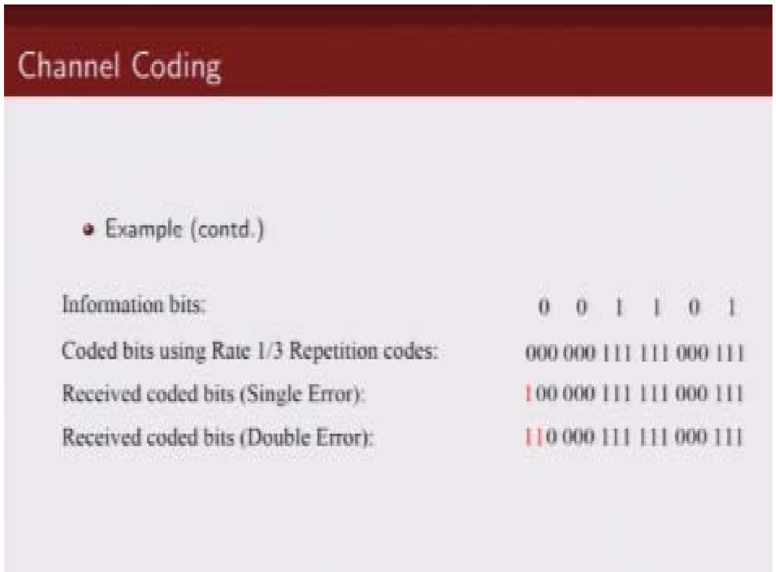
So let us look at this example when we had single error, so note here so what we received was 10, so we were able to detect error that there was an error but can we correct it? No we cannot, why? It is equally likely

(Refer Slide Time: 15:45)



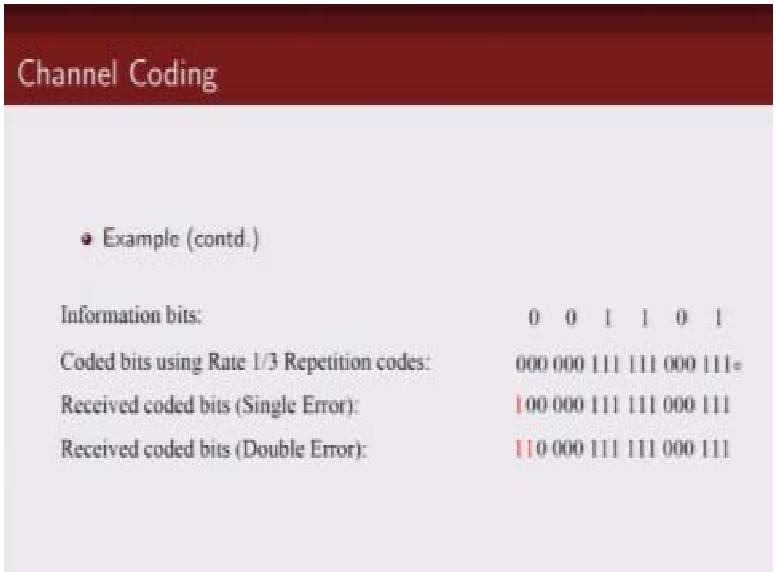
That this 1 that we received was 0 or this 0 that we received was 1, if we are talking about a binary symmetric channel right? So we do not know whether the first bit got flipped to 0, first bit got flipped to 1 instead of 0 or the second bit got flipped to 0 instead of being 1, so this particular rate ½ repetition code cannot correct any errors, it can only detect single errors.

(Refer Slide Time: 16:24)



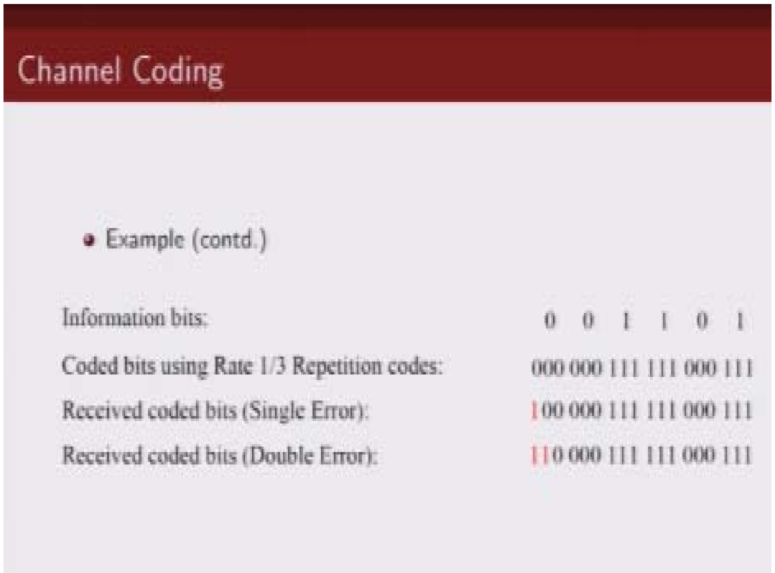
Now let us look at another example, this time we are considering a rate 1/3rd repetition code so what does rate 1/3rd repetition code means? For each and again we are considering binary code so for each bit we are adding so two parity bits and we are repeating the same bit, so for 0 we will be transmitting, we will be coding it as 000, for 1 we will be coding it at 111. So again we consider the same example of transmitting 001101.

(Refer Slide Time: 17:06)



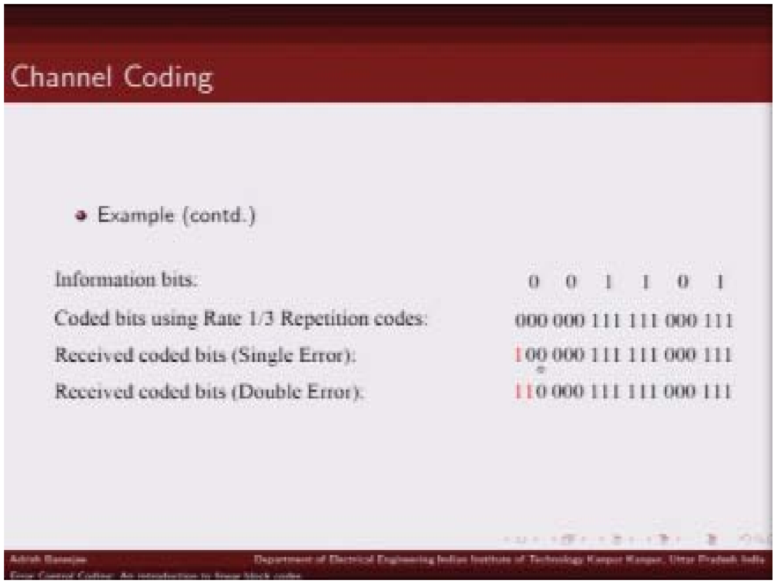
So we are transmitting the same information sequence, this time we are encoding them using rate 1/3rd repetition code so this 0 will be encoded as 000, similarly 1 will be encoded as 111 so we will be transmitting this so this information sequence will be coded in this particular way. Now we will again look at what happens when there are errors at the receive sequence, like we did for rate 1 ½ repetition code.

(Refer Slide Time: 17:43)



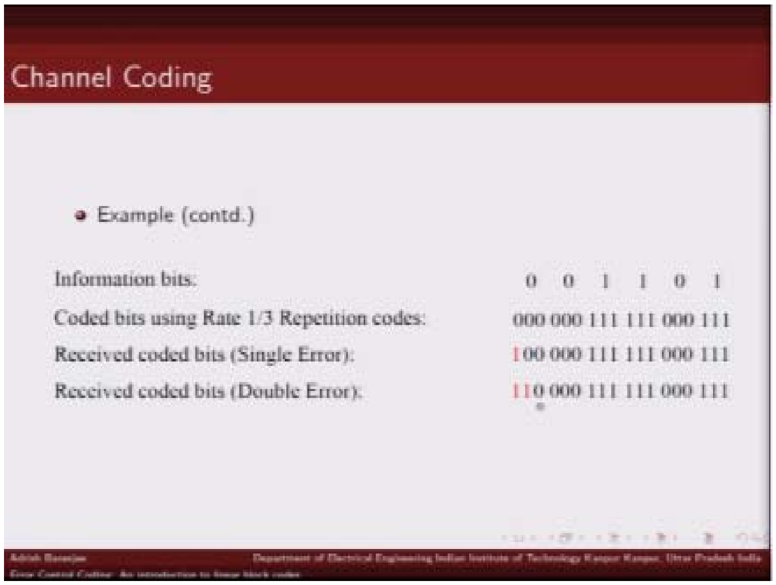
So let us again look at example for single error scenario, so let us say the first bit was received in errors so instead of 0 we received a 1, now let us see whether our rate 1/3rd repetition code can detect single error, so since it is the rate 1/3rd code for each information bit we are sending three coded bits. So we are going at the receiver, we are going to look at three bits at a time. At the decoder we are going to look at three bits at a time, so we will first look at these three bits 100. Now what do we expect, we expect since we are using a repetition code we expect all these three bits to be same.

(Refer Slide Time: 18:35)



But here in this case they are not, because what we are received is 100. Now what is that mean that mean, that means there is a transmission error. So we are able to detect single error using a rate 1/3 repetition code.

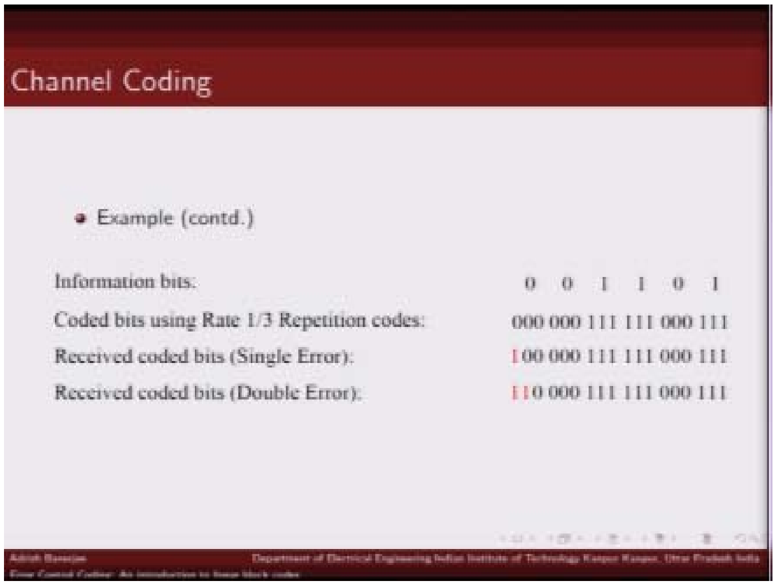
(Refer Slide Time: 18:57)



If we look at other sets of bit 000 there is no error here, 111 no error, again no error here, no error here, no error here. So rate one half repetition code was able to detect single error, even rate 1/3 code is also able to detect single error. Now let us look at double error. So let us consider scenario when the first two bits are received in error so we have 11 and rest of the sequence is this okay. Now can we detect double error?

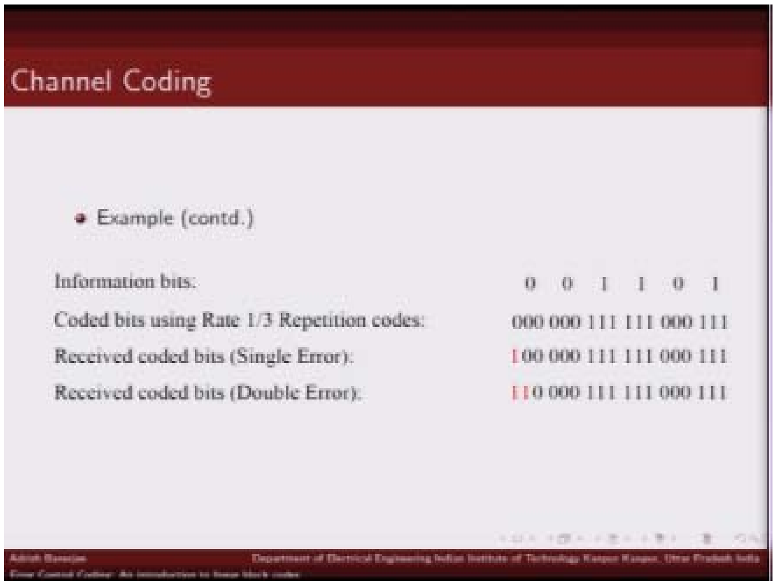
So let us look at, we again look at three bits at a time, so if you look at three bits at a time the first three bits are 110. Now we could see that there is an error why, because either this should have been 000 or 111, but we received 110.

(Refer Slide Time: 20:07)



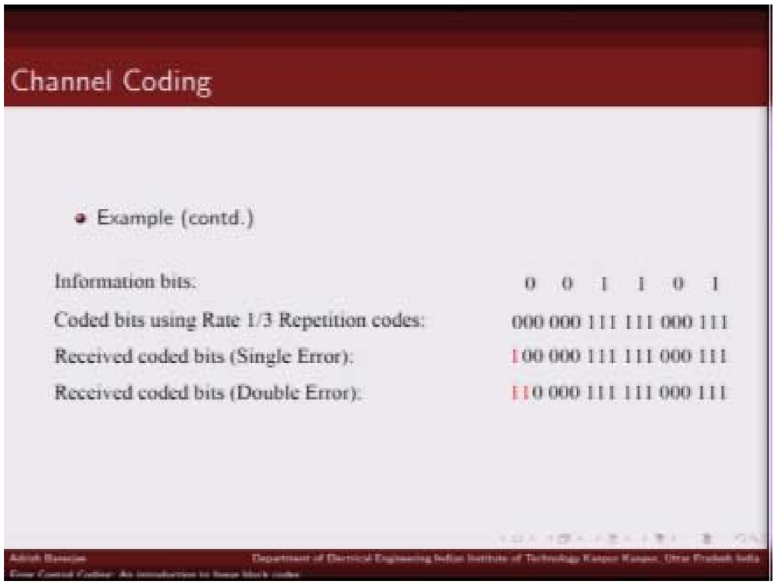
So we are able to detect using rate 1/3 repetition code we are able to detect double errors as well which we were not able to detect using rate one half repetition code. Now let us look at the error correcting capability of this code.

(Refer Slide Time: 20:22)



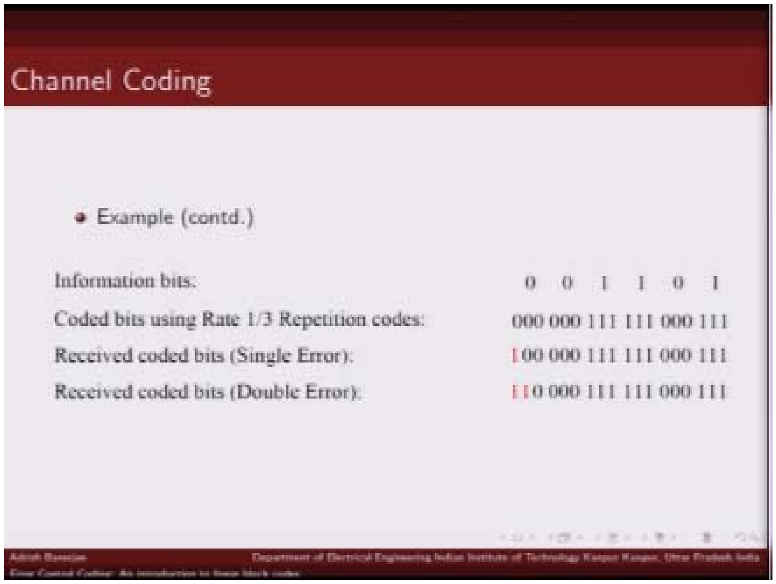
So let us go back again and look at single error situation. Now when the single error happens so something like this let us say one of the bits got flipped, 100. Now can we correct single errors and the answer in this case is yes, why? Because if you look at these three bits two bits are already 0 and one bit is 1. So it is and what are the possible

(Refer Slide Time: 20:52)



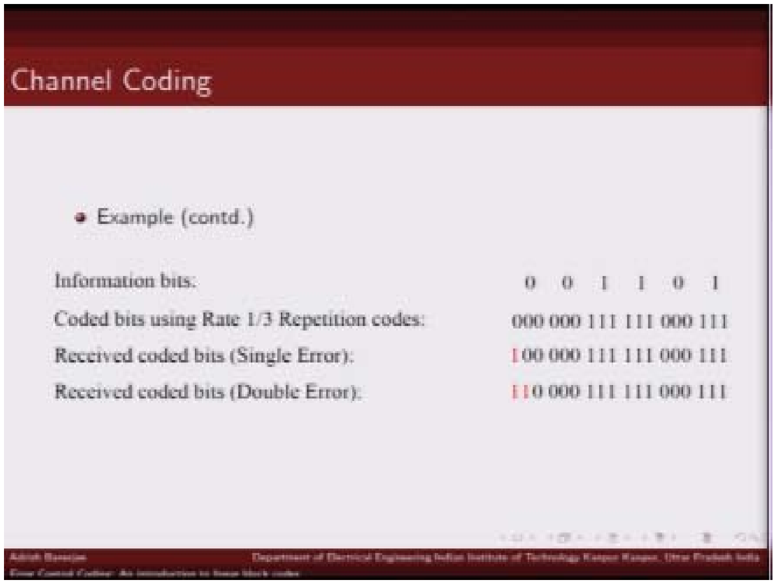
Outcomes this, this could be either 000 or 111. And it is more likely that one bit got flipped, it is more likely that 0 got flipped to 1 rather than two zeros, two ones getting flipped to 0.

(Refer Slide Time: 21:10)



So it is more likely that this bit got flipped from 0 to 1 instead of these two bits getting flipped from 1 to 0. So using majority logic this majority of the bits are 0, we will decode this as 0. So we can see this rate 1/3rd repetition code can correct single error, this was not possible for rate one half repetition code, now can it correct double errors? Now if you look at this 110 it will think that this particular bit got flipped from 1 to 0 so it will decode this as 1. So this cannot correct double errors. So to summarize we saw that rate one half repetition code can detect single error, but cannot correct single error, it cannot detect double errors. Whereas rate 1/3rd repetition code

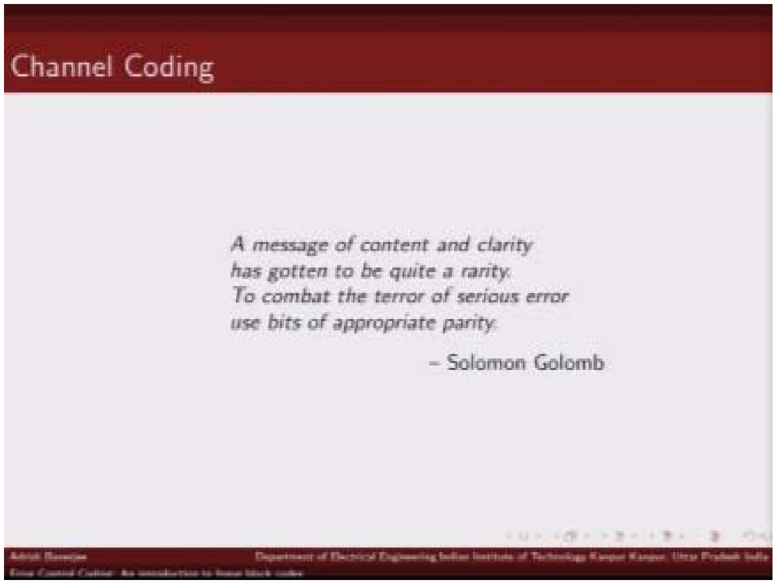
(Refer Slide Time: 22:18)



Can correct single error and can detect single error, it can detect double error but it cannot correct double errors. So why is this code better then the first code, it has certainly has better error detecting capability then the rate one half code? This we will discuss in subsequent lecture, it has to do with separation between the distance separation between the code words and you can see basically in this particular code we are using two redundant bits and in the previous case we were just using 1 redundant case.

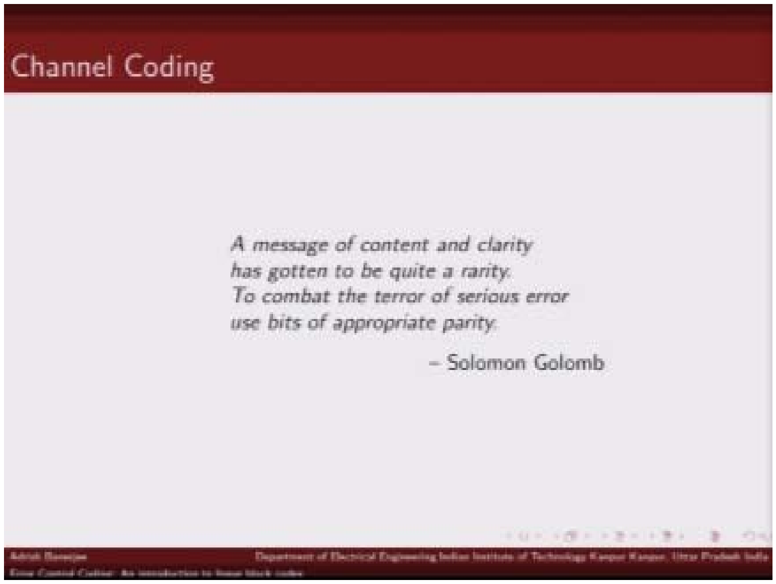
So the error correcting capability and error detecting capability of the code is depending, is dependent on the distance properties of the code and that we will talk about in subsequent lectures. So to summarize it

(Refer Slide Time: 23:08)



I think this quotation by Solomon Golomb rightly captures what error correcting code is all about. So I will read it.

(Refer Slide Time: 23:20)



"A message of content and clarity has gotten to be quite a rarity. To combat the terror of serious error use bits of appropriate parity". Thank you.

<u>Acknowledgement</u>
Ministry of Human Resource & Development

Prof. Satyaki Roy
Co-ordinator, NPTEL IIT Kanpur

NPTEL Team
Sanjay Pal
Ashish Singh
Badal Pradhan
Tapobrata Das
Ram Chandra
Dilip Tripathi
Manoj Shrivastava
Padam Shukla

Sanjay Mishra
Shubham Rawat
Shikha Gupta
K. K. Mishra
Aradhana Singh
Sweta
Ashutosh Gairola
Dilip Katiyar
Sharwan
Hari Ram
Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production