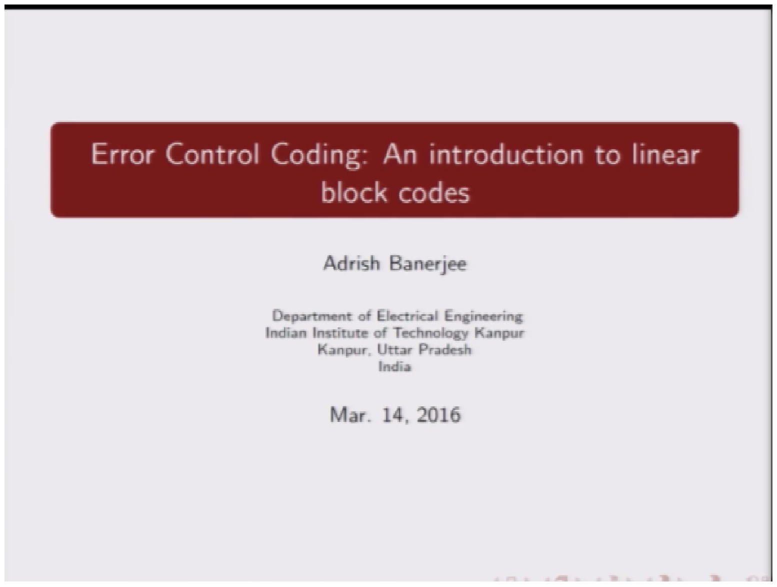
Indian Institute of Technology Kanpur
National Programme on Technology Enhanced Learning (NPTEL)
Course Title
Error Control Coding: An Introduction to Linear Block Codes

Lecture – 1 B
Introduction to Error Control Coding-II

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

Welcome to the course on error control coding, an introduction to linear block codes.

(Refer Slide Time: 00:22)



Error Control Coding: An introduction to linear block codes

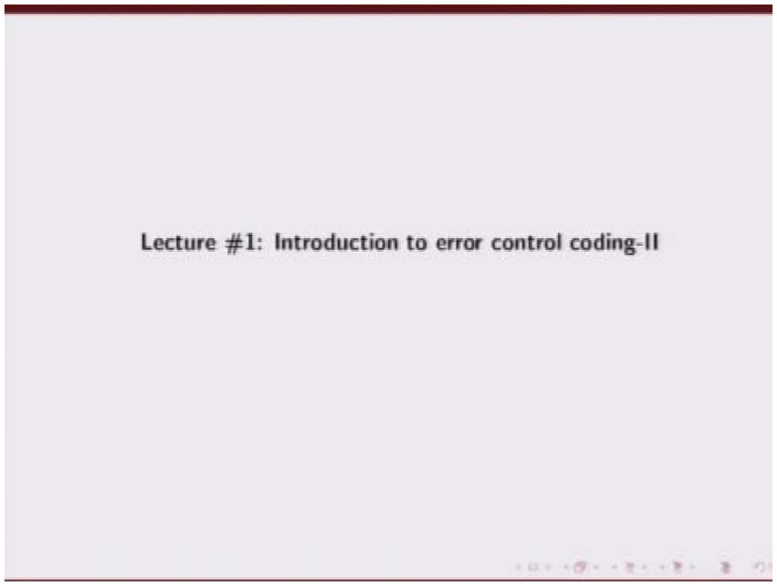
Adrish Banerjee

Department of Electrical Engineering
Indian Institute of Technology Kanpur
Kanpur, Uttar Pradesh
India

Mar. 14, 2016

I am Adrish Banerjee, we are going to continue our introduction to error control coding.

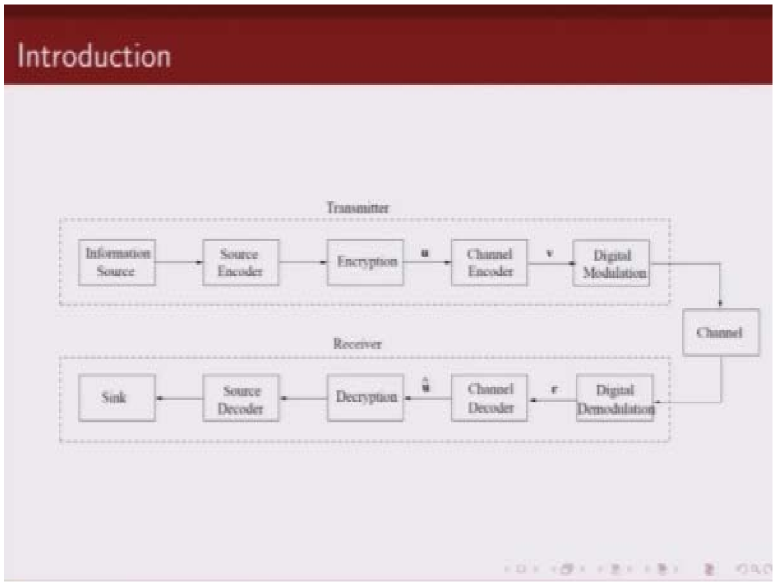
(Refer Slide Time: 00:28)



Lecture #1: Introduction to error control coding-II

This time we are going to show you, where does error control coding fits in, in our digital communication system.

(Refer Slide Time: 00:38)


Introduction

So this is a block diagram of a digital communications system. So we have our information source which is basically the message that we want to transmit. The first block that you see is source encoder, now the source encoder essentially does data compression it tries to represent the source efficiently in terms of minimum number of bits required to represent the source.

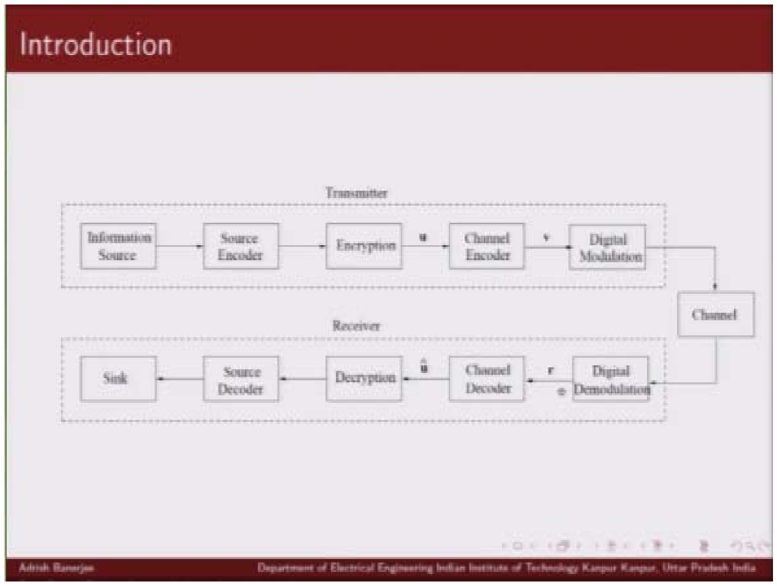
So that is basically done by this block source encoder. Now once you have represented, you have compressed your source, the next block is basically data encryption, this is for secrecy. And then the next block which is our -- of our interest here in this course is what is known as channel encoder.

So it takes into account as input information bits and gives us as output the coded bits. We typically use the symbol U to denote information sequence, and the symbol V to denote the coded sequence. So this course essentially will deal with this block of the digital communication system essentially how to design good error correcting codes and properties of error correcting codes.

Now once we have this encoded sequence you want to send these bits over a communication link so you need to do digital modulation, once you have modulated your signal you want to send it over the transmission medium which is the channel. And then at the receiver you would just have the reverse operation.

So once you get a noisy version of the received sequence you would like to demodulate the signal and the demodulated output is then fed to channel decoder whose job is to estimate information sequence from the received sequence.

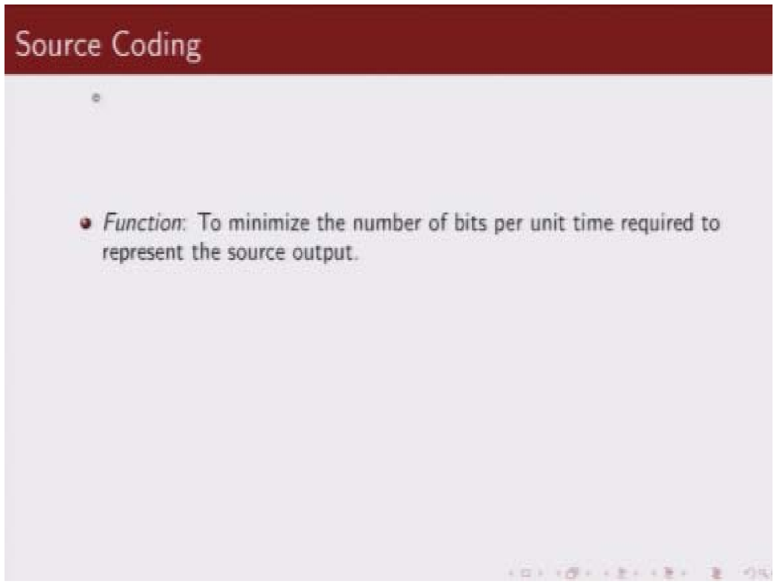
(Refer Slide Time: 03:00)



We use the notation R for received signal a received inputs demodulated signal, and we will use û to denote the estimated information bits. Now if u=û there is no error, however if u is not same as û then there are decoding errors. Now once you decode the information bits then you would like to de-encrypt to get back to your plain text data, so this de-encryption is the opposite action of encryption.

And once you get back your plain text you would like to decompress the signal to get back the original source signal. So that is basically in a nutshell the broad block diagram of a digital communication system. So we can see basically in this course we are interested in this block, we are interested in this block, and of course we are interested in this block as well, because our error control coding strategy depends on what sort of channel we are sending our information bits over okay. So I will now explain in little bit more detail about each of these block diagrams.

(Refer Slide Time: 04:29)

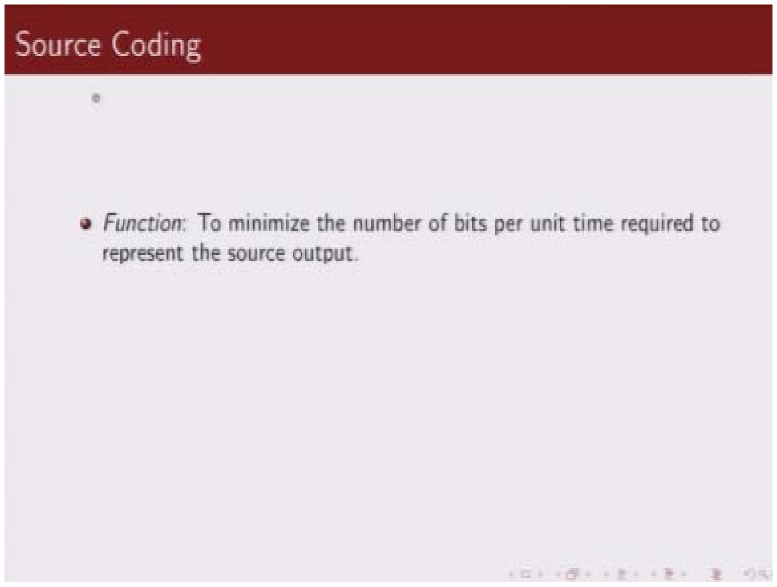


Source Coding

- *Function:* To minimize the number of bits per unit time required to represent the source output.

So the first is basically our sourced encoder and as I said this is essentially does source compression, source coding.

(Refer Slide Time: 04:39)



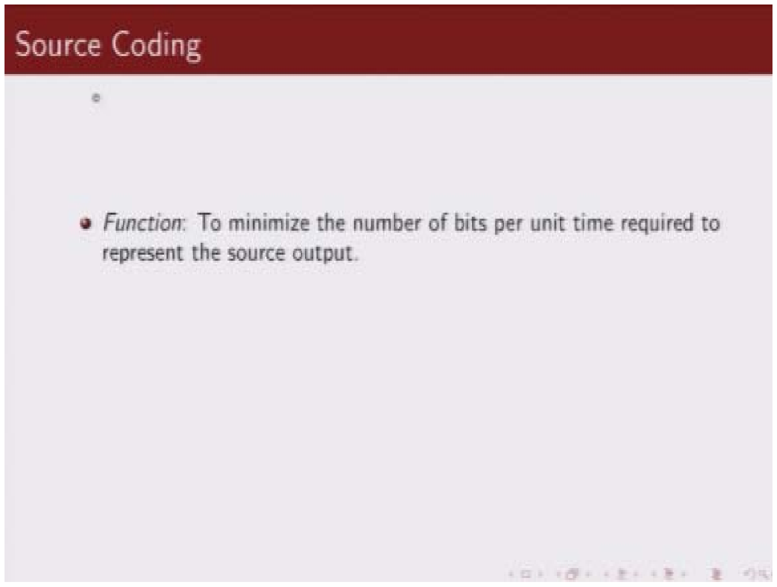
Source Coding

- *Function*: To minimize the number of bits per unit time required to represent the source output.

And its objective is to represent your source in minimum number of bits required. As we know that our source to be call, for example English text, there are a lot of inbuilt redundancies in the text, for example Q, the letter Q is always followed by U. So if we want to transmit let us say some word starting with Q, we do not need to transmit U, because in English text, because Q will be always followed by U.

So like that there are a lot of unnecessary redundancy build in into the source which we would want – which we would like to get rid of, and that is the job of source encoder. So the source encoder will like to represent our source compactly in a minimum number of bits required.

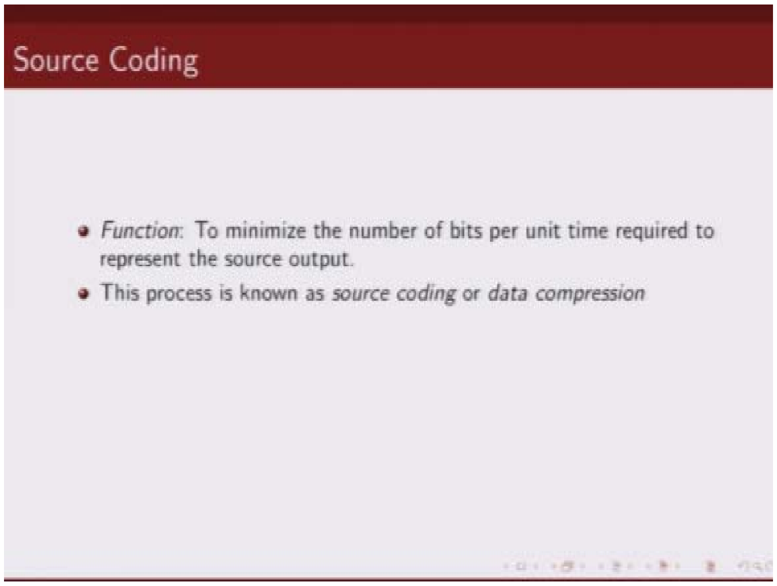
(Refer Slide Time: 05:35)


Source Coding

- *Function*: To minimize the number of bits per unit time required to represent the source output.

That is basically the objective of the source encoder.

(Refer Slide Time: 05:39)

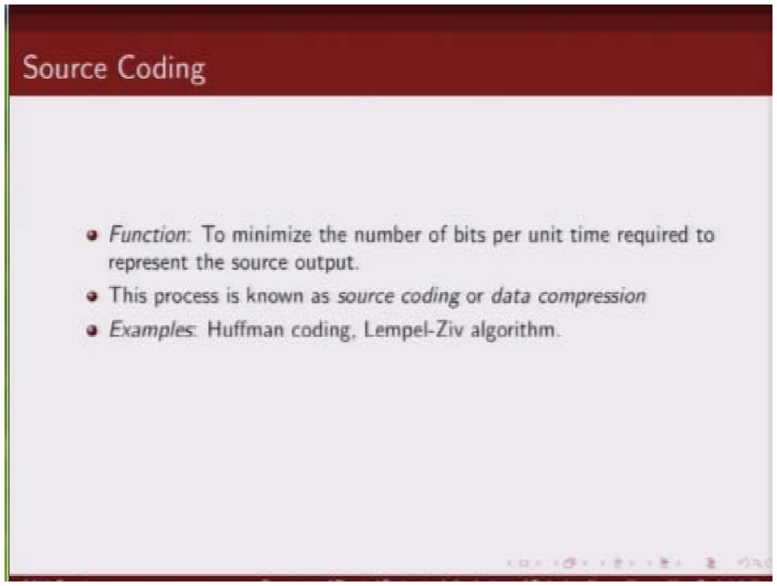


Source Coding

- *Function*: To minimize the number of bits per unit time required to represent the source output.
- This process is known as *source coding* or *data compression*

So it does basically data compression.

(Refer Slide Time: 05:44)

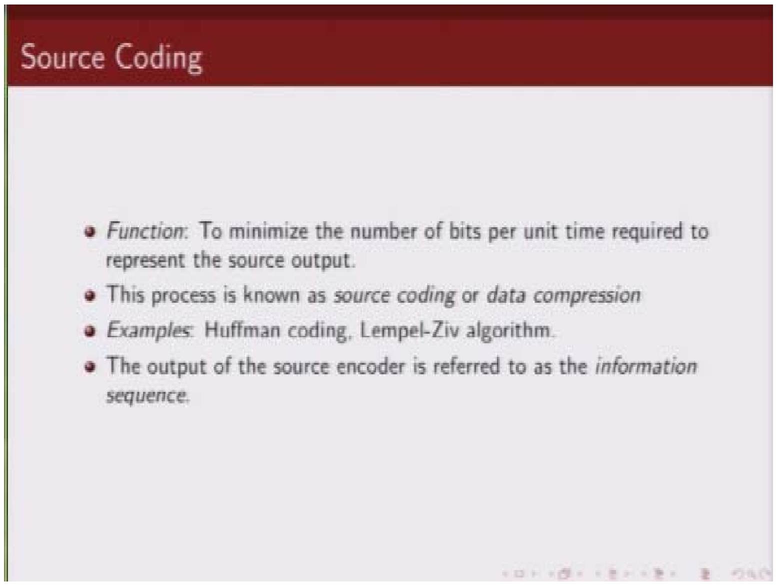


Source Coding

- *Function*: To minimize the number of bits per unit time required to represent the source output.
- This process is known as *source coding* or *data compression*
- *Examples*: Huffman coding, Lempel-Ziv algorithm.

Some examples of data compression techniques is basically Huffman coding, Lempel-Ziv algorithm. We would not be talking about this, because this topics of interest in information theory, we will restrict our discussion to coding theories topics in this course.

(Refer Slide Time: 06:04)

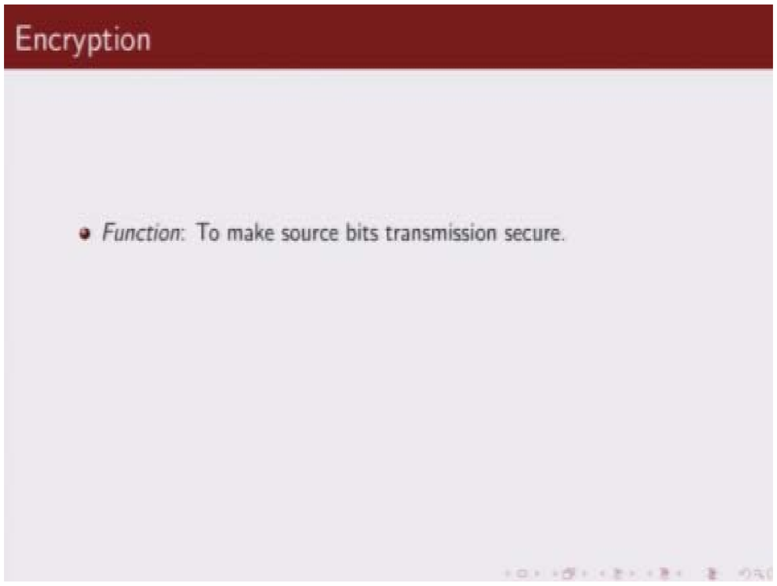


Source Coding

- *Function*: To minimize the number of bits per unit time required to represent the source output.
- This process is known as *source coding* or *data compression*
- *Examples*: Huffman coding, Lempel-Ziv algorithm.
- The output of the source encoder is referred to as the *information sequence*.

And as I said basically the input to our channel encoder, we are referring it as information sequence.

(Refer Slide Time: 06:17)

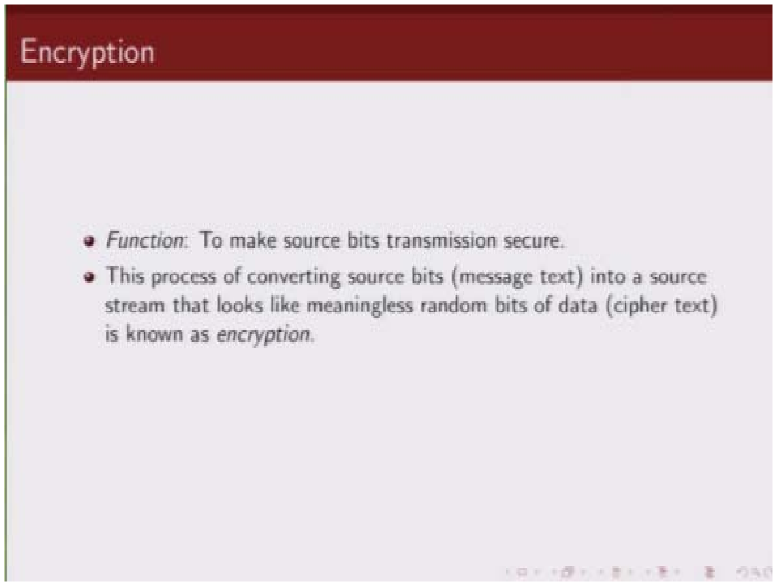


Encryption

- *Function*: To make source bits transmission secure.

Now the next block was of encryption and the objective of the encryption is to make the data's or made the bits secure so that unwanted users should not be able to find out what has been transmitted.

(Refer Slide Time: 06:32)

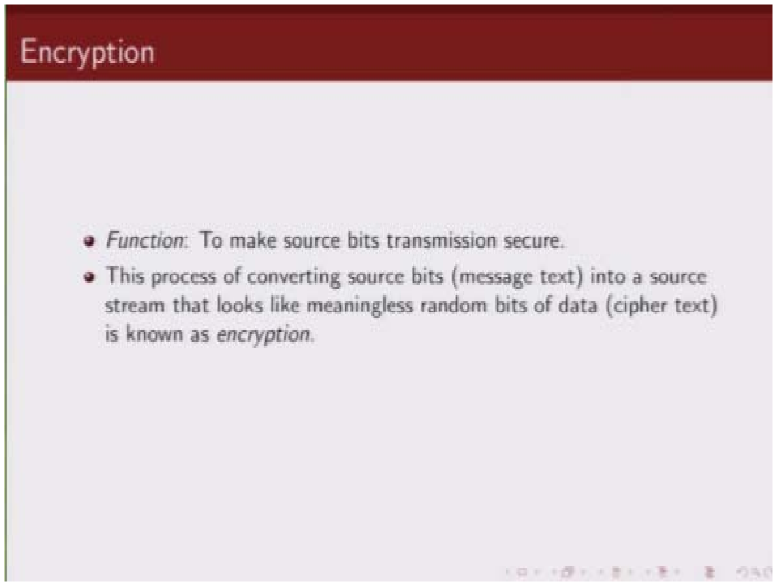


Encryption

- *Function*: To make source bits transmission secure.
- This process of converting source bits (message text) into a source stream that looks like meaningless random bits of data (cipher text) is known as *encryption*.

So in this encryption process what we do we convert our source bit into what we call a cipher text. So we basically -- encryption takes our source bits and it converts it into some random looking bits.

(Refer Slide Time: 06:55)

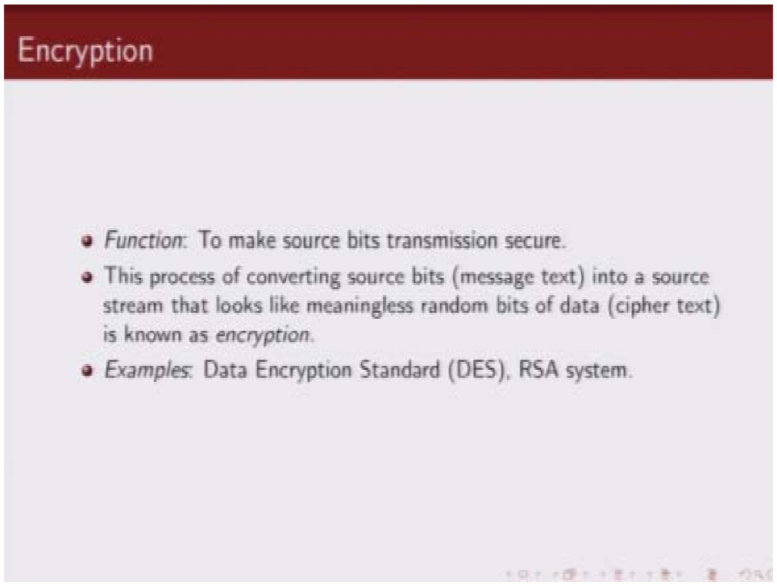


Encryption

- *Function*: To make source bits transmission secure.
- This process of converting source bits (message text) into a source stream that looks like meaningless random bits of data (cipher text) is known as *encryption*.

Which for somebody who does not have a key would appear as a random meaningless data. So it basically converts your message text into what we call cipher text and that is basically encryption.

(Refer Slide Time: 07:10)

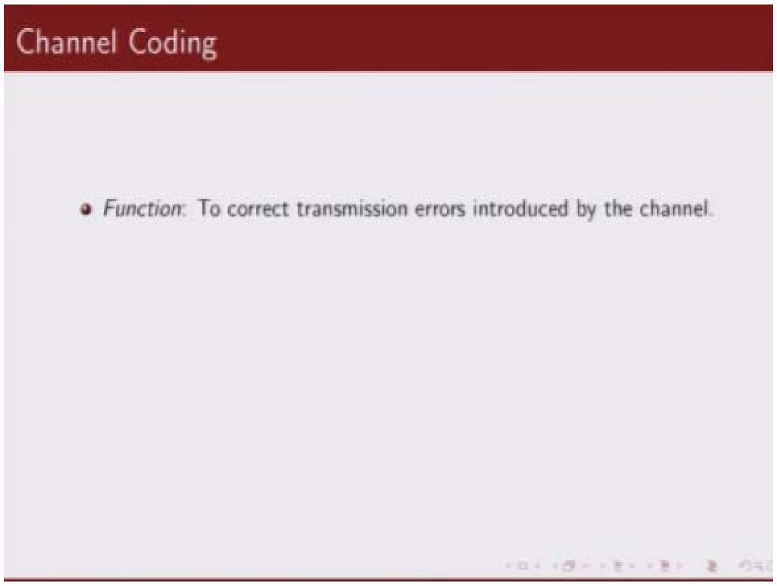


Encryption

- *Function*: To make source bits transmission secure.
- This process of converting source bits (message text) into a source stream that looks like meaningless random bits of data (cipher text) is known as *encryption*.
- *Examples*: Data Encryption Standard (DES), RSA system.

And examples of encryption is data encryption standard, advanced encryption standard, and RSA system. The next block was of channel encoder.

(Refer Slide Time: 07:21)



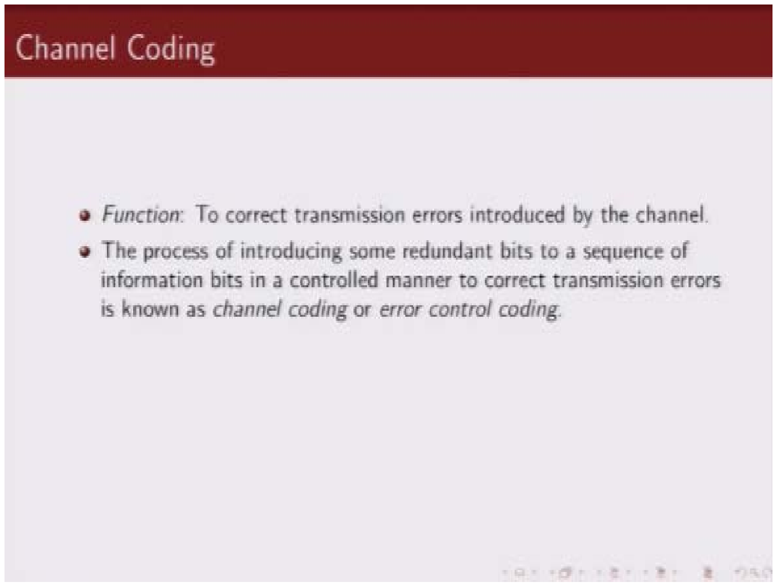
Channel Coding

- *Function*: To correct transmission errors introduced by the channel.

Now this is of interest in this course and what is the objective of this block, we would like to introduce additional redundancies into the source and with the help of these redundant bits we would like to detect and correct error induced by transmission errors or storage areas, basically we would like to correct or detect those errors.

Now if you are wondering why cannot we use the inherent redundant bits which are available in a source and why do not we just get rid of source encoder and channel encoder, just use the plain source which has inherents of redundant bits. The problem with this is we do not have any control over those redundant bits which are inherent in a source whereas in our channel encoder the redundant bits that we are adding have been carefully designed in such a way so as to detect or correct errors.

(Refer Slide Time: 08:22)

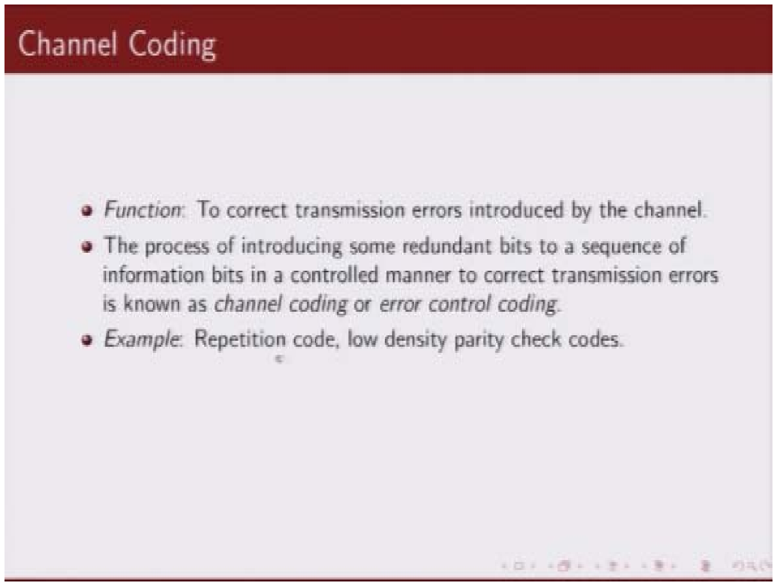


Channel Coding

- *Function:* To correct transmission errors introduced by the channel.
- The process of introducing some redundant bits to a sequence of information bits in a controlled manner to correct transmission errors is known as *channel coding* or *error control coding*.

So this process of basically adding some redundant bits which we call as parity bits to our message bits is basically -- this process is known as channel coding or error control coding, and this is essentially done to detect or correct error which has caused by transmission medium.

(Refer Slide Time: 08:44)

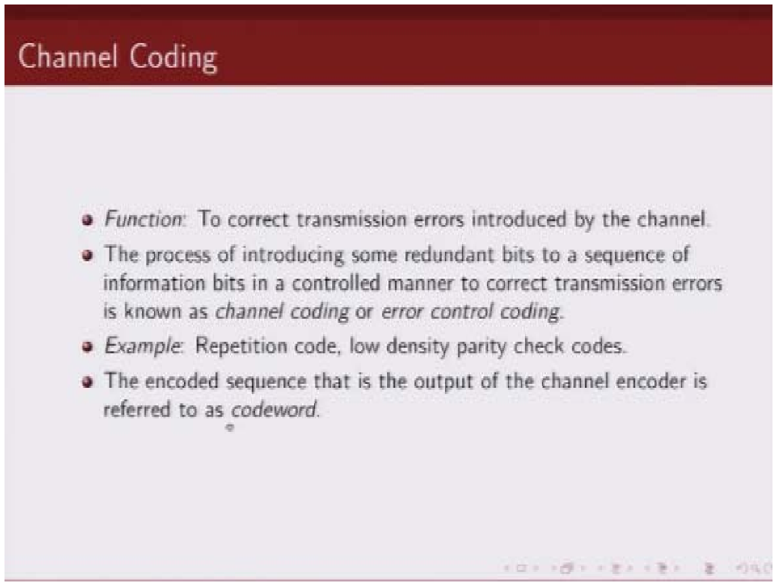


Channel Coding

- *Function*: To correct transmission errors introduced by the channel.
- The process of introducing some redundant bits to a sequence of information bits in a controlled manner to correct transmission errors is known as *channel coding* or *error control coding*.
- *Example*: Repetition code, low density parity check codes.

In the last lecture I already gave an example for simple, very simple code, repetition code, there are throughout this course we are going to talk about some other examples of error correcting codes.

(Refer Slide Time: 08:59)

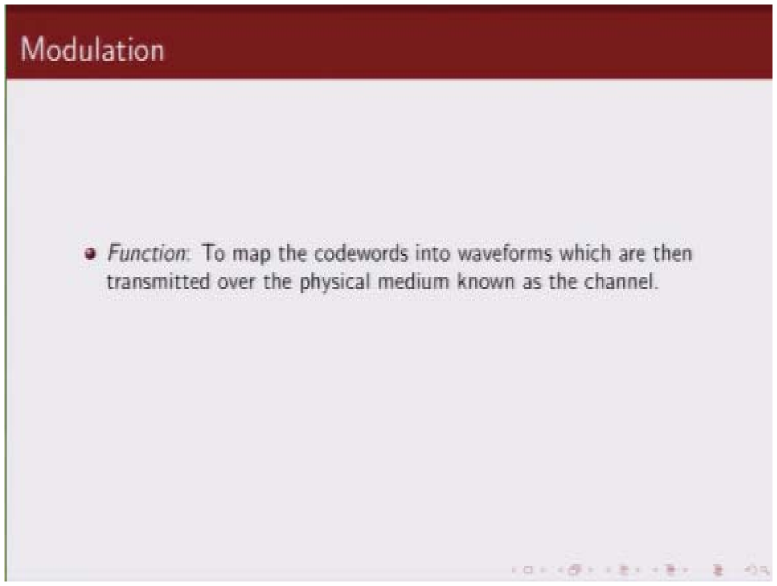


Channel Coding

- *Function*: To correct transmission errors introduced by the channel.
- The process of introducing some redundant bits to a sequence of information bits in a controlled manner to correct transmission errors is known as *channel coding* or *error control coding*.
- *Example*: Repetition code, low density parity check codes.
- The encoded sequence that is the output of the channel encoder is referred to as *codeword*.

We use the term code word to denote the encoded sequence out of the channel encoder.

(Refer Slide Time: 09:09)

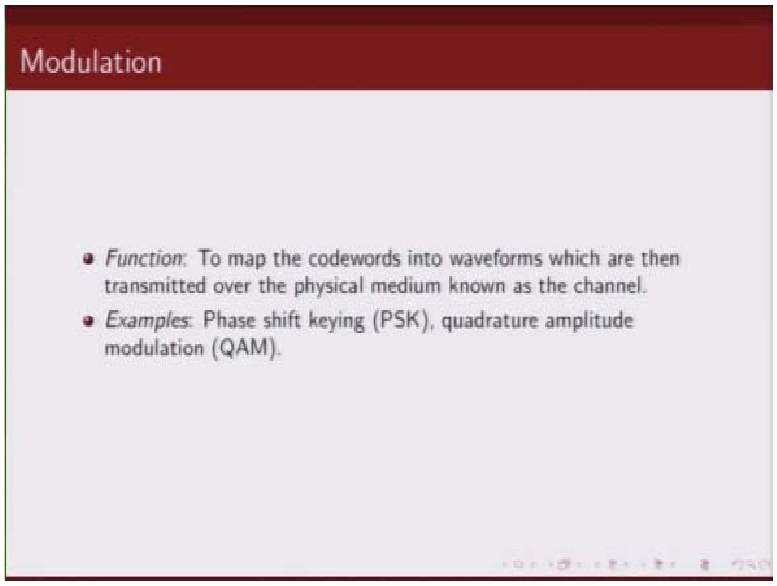


Modulation

- *Function*: To map the codewords into waveforms which are then transmitted over the physical medium known as the channel.

Now once we have the encoded sequence we would like to transmit that sequence over a communication link, so what we need to do is modulation. So modulation maps these code words into actual way fonts which are sent over a communication link and you are familiar with – I am sure you are familiar with digital modulation techniques.

(Refer Slide Time: 09:40)

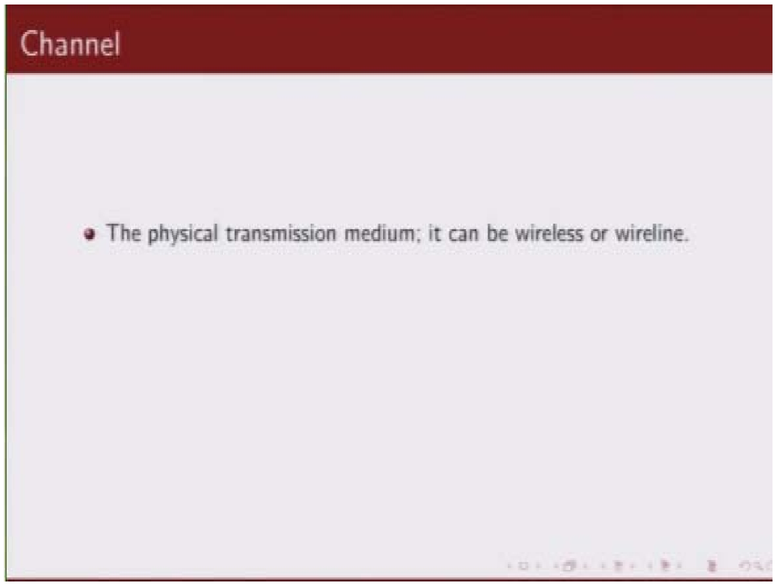


Modulation

- *Function*: To map the codewords into waveforms which are then transmitted over the physical medium known as the channel.
- *Examples*: Phase shift keying (PSK), quadrature amplitude modulation (QAM).

Such as BPSK, binary phase-shift keying, quadrature amplitude modulation or you will also select about analog modulation schemes also.

(Refer Slide Time: 09:43)

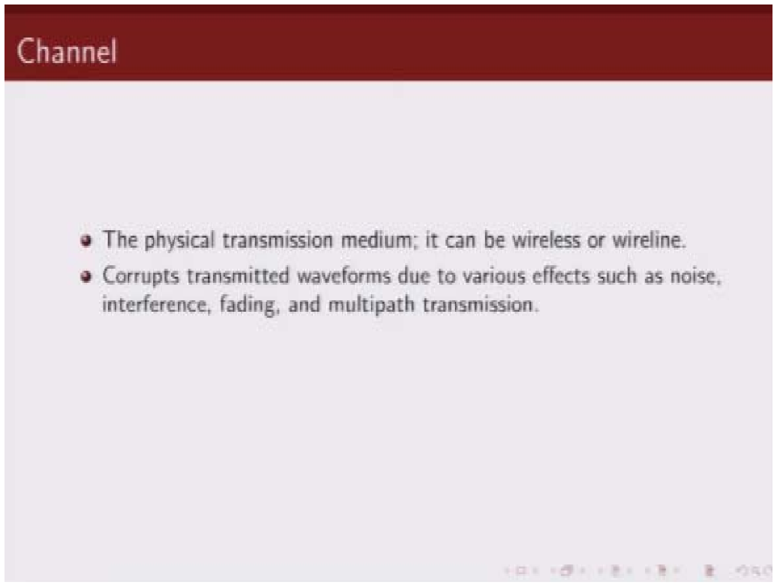


Channel

- The physical transmission medium; it can be wireless or wireline.

The next block in our block diagram was a channel. So what was channel, channel was essentially our transmission medium over which we were sending our coded bits. So the physical transmission medium is known as channel, and this could be a wired medium, for example our old fashion phones or it could be wireless medium.

(Refer Slide Time: 10:13)

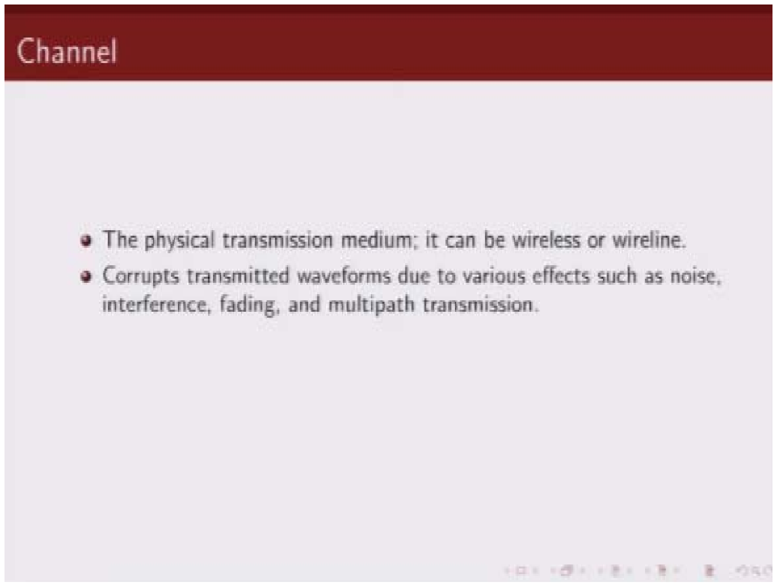


Channel

- The physical transmission medium; it can be wireless or wireline.
- Corrupts transmitted waveforms due to various effects such as noise, interference, fading, and multipath transmission.

Now what effect does channel has on the transmitted sequence.

(Refer Slide Time: 10:21)

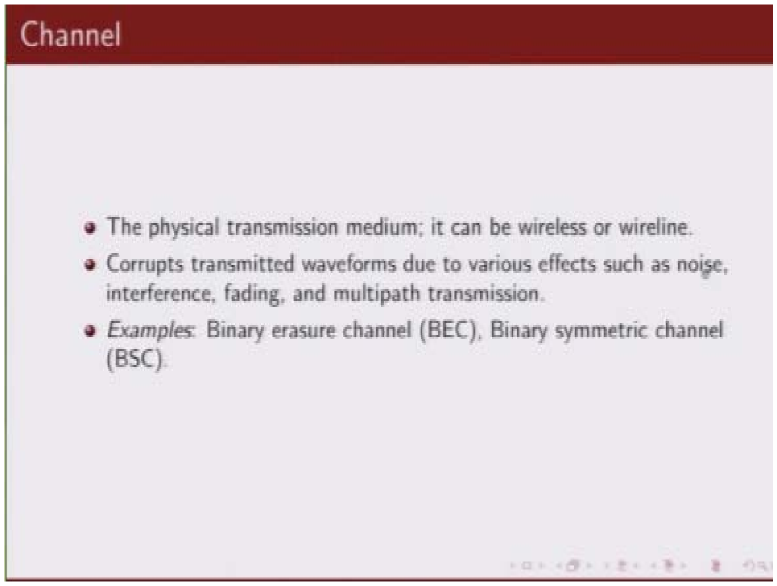


Channel

- The physical transmission medium; it can be wireless or wireline.
- Corrupts transmitted waveforms due to various effects such as noise, interference, fading, and multipath transmission.

As we know basically the channel corrupts the signal in multiple ways, for example noise.

(Refer Slide Time: 10:25)



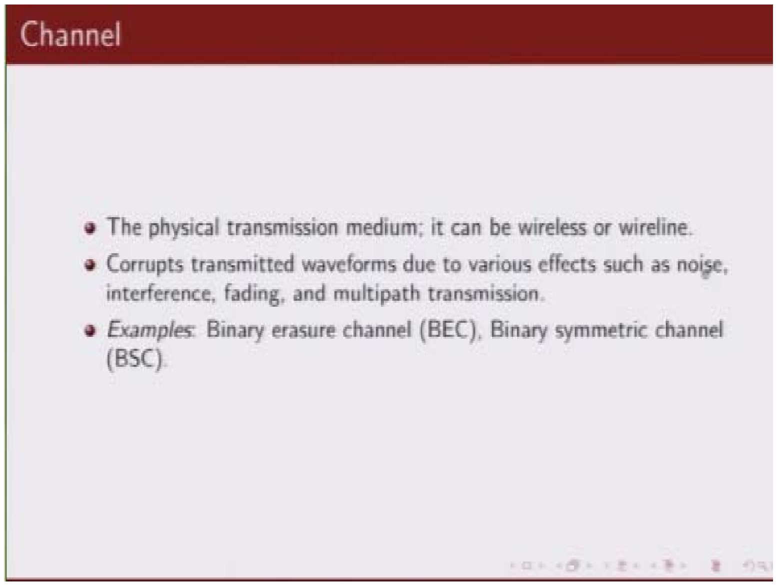
Channel

- The physical transmission medium; it can be wireless or wireline.
- Corrupts transmitted waveforms due to various effects such as noise, interference, fading, and multipath transmission.
- *Examples*: Binary erasure channel (BEC), Binary symmetric channel (BSC).

This basically noise deteriorates the signal interference is basically, because multiple signals clashing with each color fading, basically if you are let us say under some building or something you may not get a very good signal, so that is an example of fading, shadowing, multipath transmission, when you have reflections from multiple buildings, what you receive you send just one signal you get multiple copies with varying attenuation and delay spread.

So essentially channel corrupts our signal that we are transmitting and it results in errors and there are various models of channels which are very popularly used.

(Refer Slide Time: 11:17)

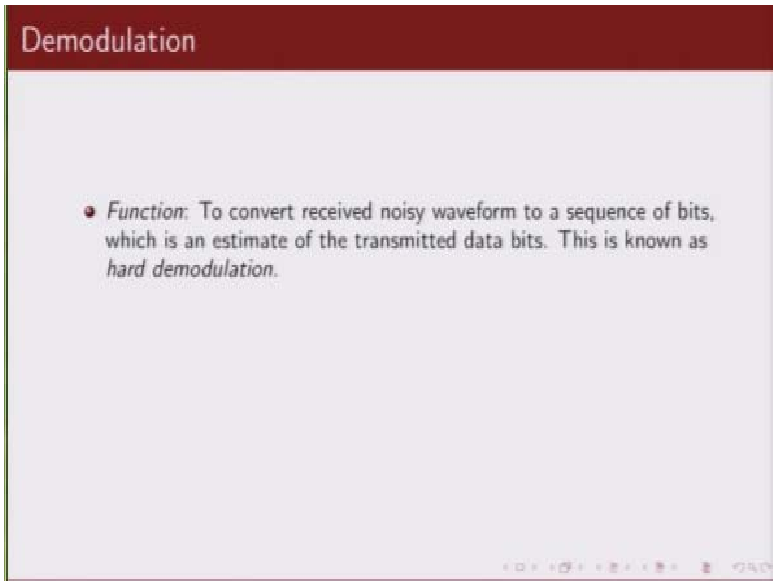


Channel

- The physical transmission medium; it can be wireless or wireline.
- Corrupts transmitted waveforms due to various effects such as noise, interference, fading, and multipath transmission.
- *Examples*: Binary erasure channel (BEC), Binary symmetric channel (BSC).

In the last lecture we did talk about binary erasure channel which is a very good example for two model basically packet data communication system and another channel model what we talked about was binary symmetric challenge, it is a very simple channel model typically used to model additive white Gaussian noise channel with hard demodulation. So these are some examples of channel models.

(Refer Slide Time: 11:46)



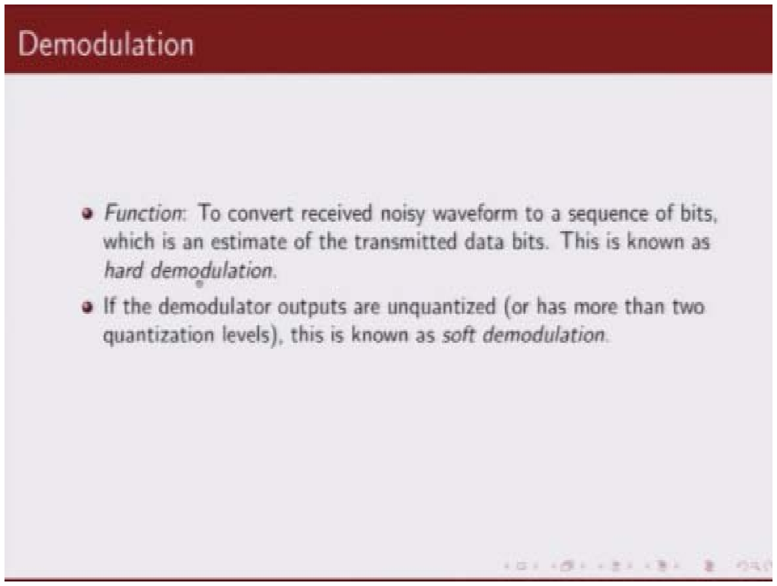
Demodulation

- *Function*: To convert received noisy waveform to a sequence of bits, which is an estimate of the transmitted data bits. This is known as *hard demodulation*.

Now the next block in our additional communication system was demodulation. So what does demodulation does, so as a result of passing through the communication channel what we get is a very noisy way form, and from that very noisy way form essentially you want to get an estimate of what the transmitted bits are.

Now when we try to demodulate or get back original bits zeros and ones that is basically what is known as hard demodulation.

(Refer Slide Time: 12:19)



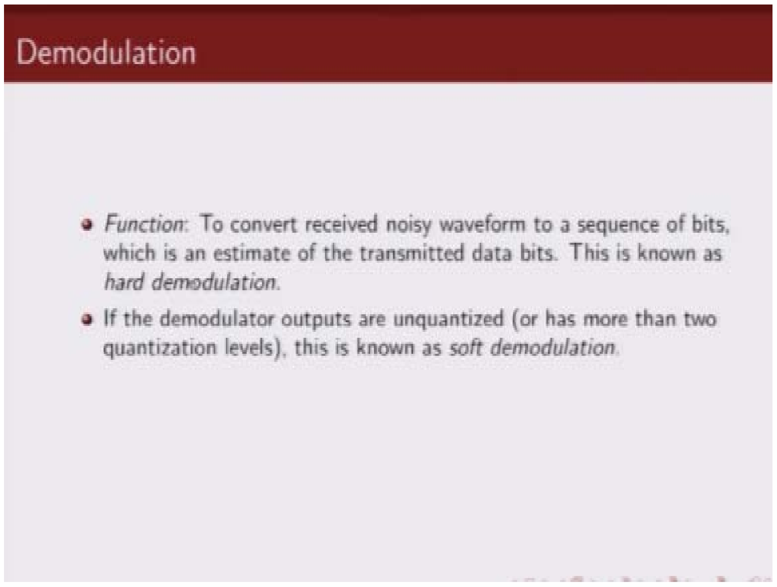
Demodulation

- *Function*: To convert received noisy waveform to a sequence of bits, which is an estimate of the transmitted data bits. This is known as *hard demodulation*.
- If the demodulator outputs are unquantized (or has more than two quantization levels), this is known as *soft demodulation*.

There is also something called soft demodulation, what if we are getting received noisy values of the received sequence instead of quantizing them directly to zeros and ones if we keep it unquantized then this type of demodulation is known as soft demodulation. Now what is the advantage of soft demodulation, in hard demodulation we basically lose lot of information.

For example let us say you consider binary transmission you are sending zeros and ones, and let us say you are using binary phase-shift keying. So let us consider a scenario where zero is mapped 2+1 and one is mapped 2-1. And let us say you received symbol was 0.01.

(Refer Slide Time: 13:18)



Demodulation

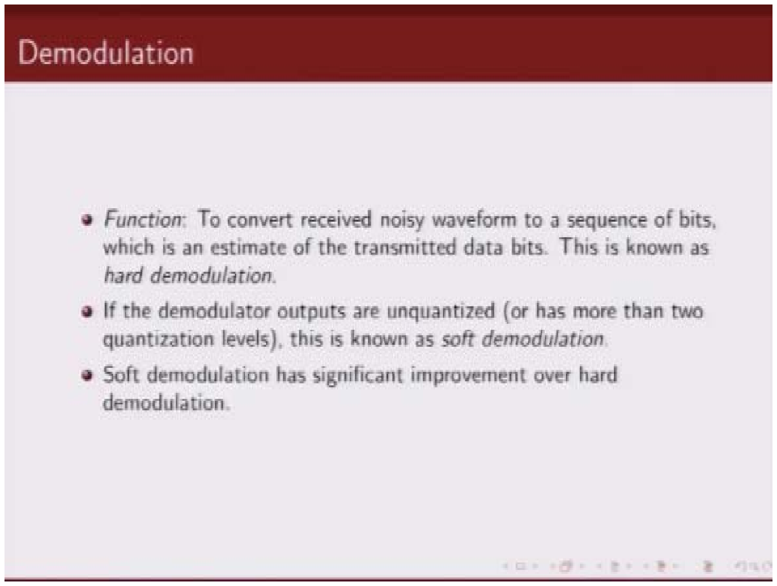
- *Function*: To convert received noisy waveform to a sequence of bits, which is an estimate of the transmitted data bits. This is known as *hard demodulation*.
- If the demodulator outputs are unquantized (or has more than two quantization levels), this is known as *soft demodulation*.

Now if you were doing hard demodulation you would demodulate it to be – because its distance threshold will be zero, you will demodulate it to be zero bit transmitted. But 0.01 could very well be – could have been -0.01 if the noise would have been little bit like this thing, so it is very close to zero.

So in hard demodulation you will lose that information, so if instead of just directly mapping our received sequence it was 0.01 to 0 if you would have just also kept some idea about how sure we are or how close we think it is to zero or one, then that information is useful. And that is captured in soft demodulation. So in soft demodulation we would not just map 0.01 to 0 we will just say it is 0 with probability 0.51 or 1 with probability 5.49, just like I am just giving an example.

So we lose some information when we go for hard demodulation. So typically, basically when you will see basically if we do soft demodulation our performance is roughly 2DB better than if you were just doing a hard demodulation.

(Refer Slide Time: 14:44)

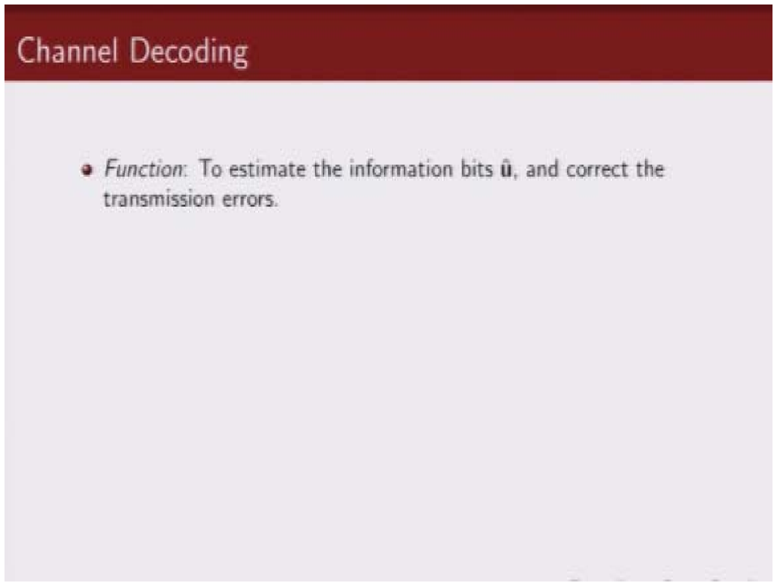


Demodulation

- *Function*: To convert received noisy waveform to a sequence of bits, which is an estimate of the transmitted data bits. This is known as *hard demodulation*.
- If the demodulator outputs are unquantized (or has more than two quantization levels), this is known as *soft demodulation*.
- Soft demodulation has significant improvement over hard demodulation.

So most of the modern coding techniques we will basically, we will be doing soft demodulation and we will pass on these bits soft values to the decoder, channel decoder and we will then ask it to do error correction or error detection.

(Refer Slide Time: 15:02)

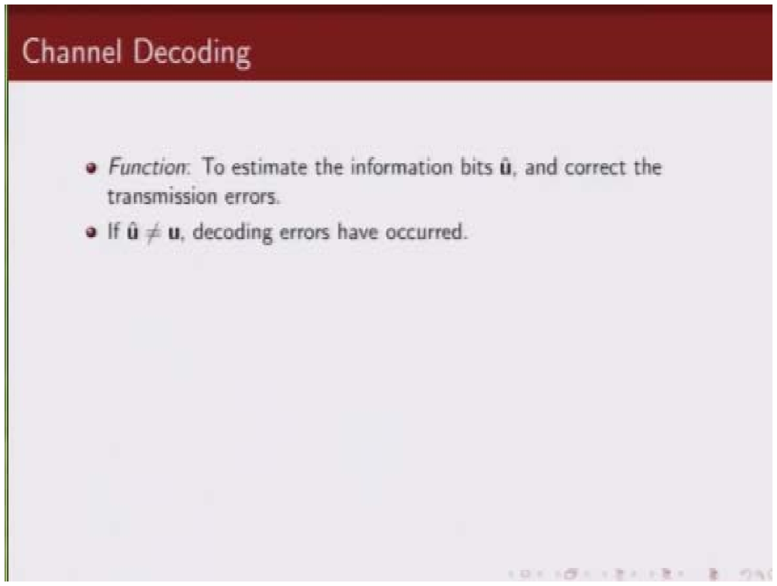


Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.

Software demodulation we have channel decoding, as I said the channel decoder objective is to estimate what was the transmitted information sequence from the received coded sequence.

(Refer Slide Time: 15:21)

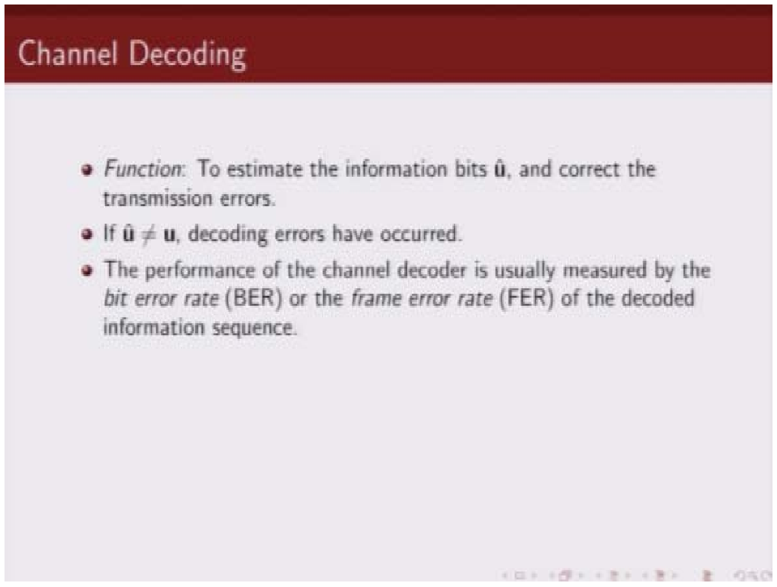


Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.
- If $\hat{\mathbf{u}} \neq \mathbf{u}$, decoding errors have occurred.

And its objective is also to correct any errors which was caused by transmission error. So when do we say an error has occurred, if our decoded bit is not same as transmitted bit, we say that decoding errors have occurred.

(Refer Slide Time: 15:45)

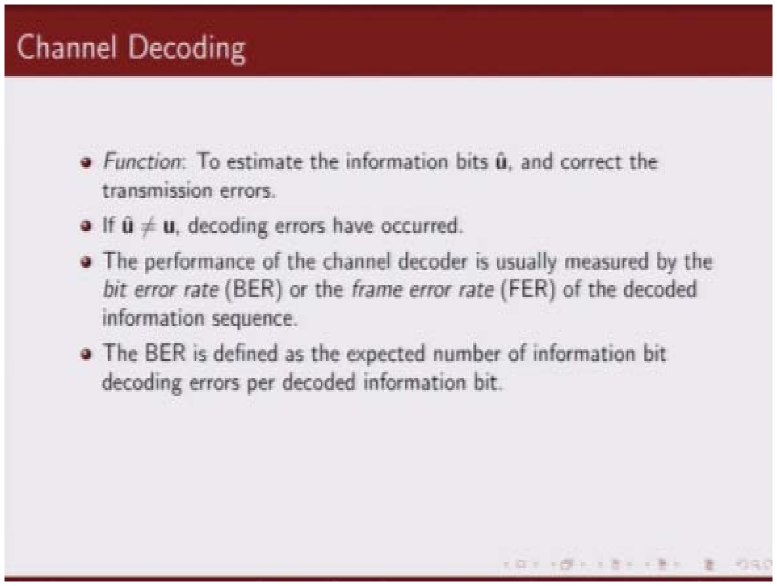


Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.
- If $\hat{\mathbf{u}} \neq \mathbf{u}$, decoding errors have occurred.
- The performance of the channel decoder is usually measured by the *bit error rate* (BER) or the *frame error rate* (FER) of the decoded information sequence.

And how do we quantify these decoding errors, we quantify these decoding errors using what we call bit error rate or frame error rate. So performance of error correcting codes is typically evaluated using these bit error rate and frame error rate. So what is bit error rate and frame error rate?

(Refer Slide Time: 16:11)

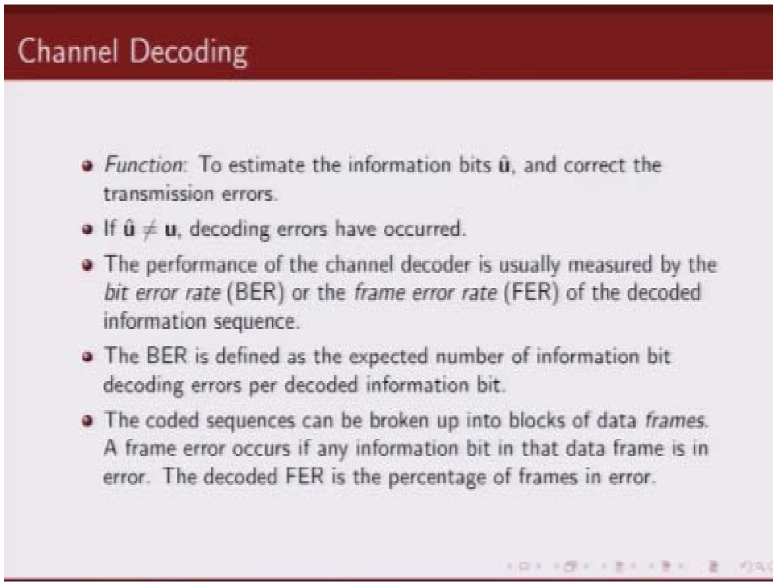


Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.
- If $\hat{\mathbf{u}} \neq \mathbf{u}$, decoding errors have occurred.
- The performance of the channel decoder is usually measured by the *bit error rate* (BER) or the *frame error rate* (FER) of the decoded information sequence.
- The BER is defined as the expected number of information bit decoding errors per decoded information bit.

So we define a bit error rate as the expected number of information bits that are decoded to be in error. So for example if you transmit N information symbols and out of them, X basically our bits are received in error, so the bit error rate would be x/n.

(Refer Slide Time: 16:39)

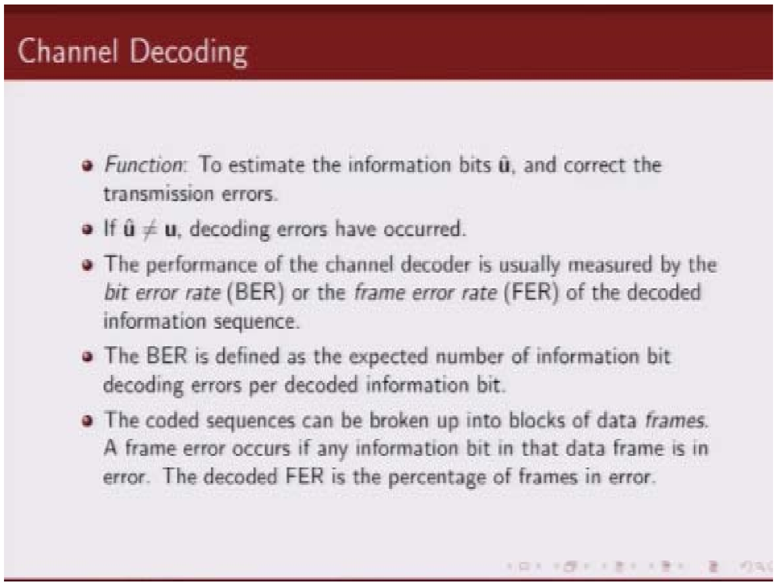


Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.
- If $\hat{\mathbf{u}} \neq \mathbf{u}$, decoding errors have occurred.
- The performance of the channel decoder is usually measured by the *bit error rate* (BER) or the *frame error rate* (FER) of the decoded information sequence.
- The BER is defined as the expected number of information bit decoding errors per decoded information bit.
- The coded sequences can be broken up into blocks of data *frames*. A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.

Similarly when we are talking about packet data communication we are interested in – so for example if any of the bits in the packet, whole packet is an error we say the packet is in error or the frame is in error. So frame error rate is defined as number of frames or number of packets in error divided by total number of packets that we have transmitted.

(Refer Slide Time: 17:06)



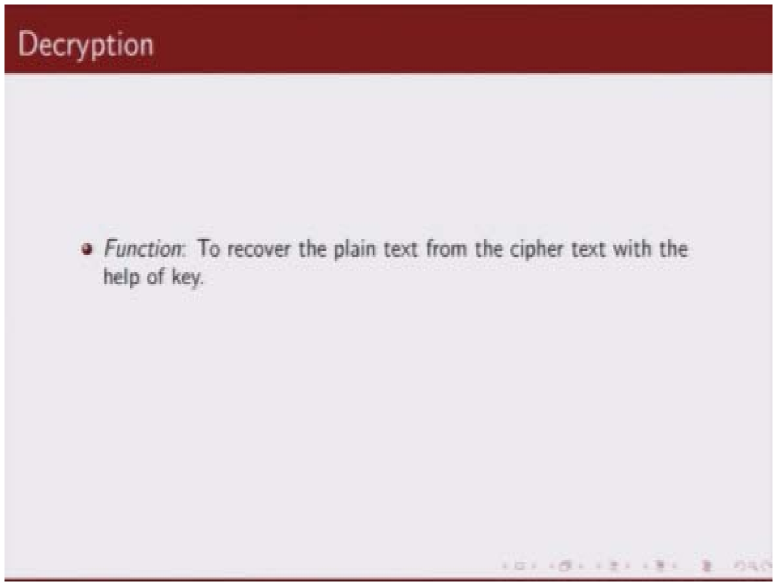
Channel Decoding

- *Function*: To estimate the information bits $\hat{\mathbf{u}}$, and correct the transmission errors.
- If $\hat{\mathbf{u}} \neq \mathbf{u}$, decoding errors have occurred.
- The performance of the channel decoder is usually measured by the *bit error rate* (BER) or the *frame error rate* (FER) of the decoded information sequence.
- The BER is defined as the expected number of information bit decoding errors per decoded information bit.
- The coded sequences can be broken up into blocks of data *frames*. A frame error occurs if any information bit in that data frame is in error. The decoded FER is the percentage of frames in error.

So a frame error rate occurs if any of the bits in your data frame or in your packet is received in error. So it is the percentage of frames that are in error, this is known as frame error rate. So when we compare performance of different error correcting codes what we do is we plot on the Y axis bit error rate or frame error rate versus SNR, signal to noise ratio basically.

So how was the performance of, when we increase signal to noise ratio how does our error correcting codes perform, that is how basically we compare two error correcting, that is one way of comparing two error correcting codes.

(Refer Slide Time: 17:57)

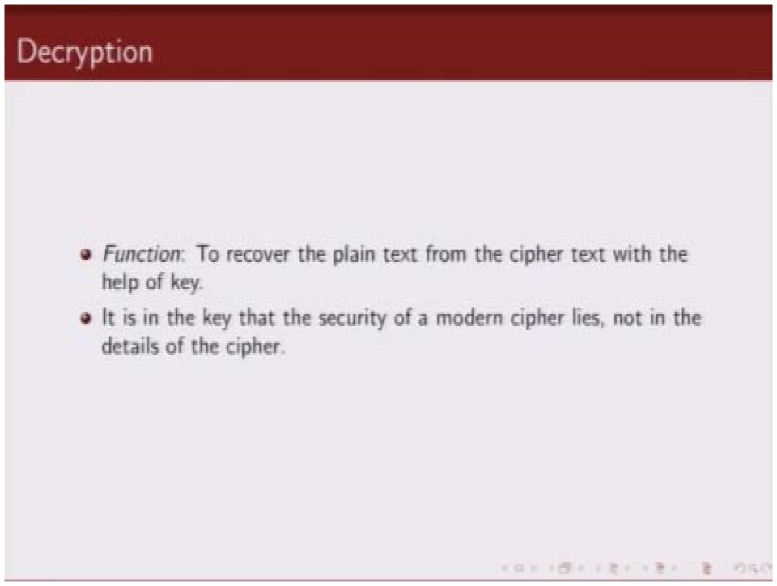


Decryption

- *Function*: To recover the plain text from the cipher text with the help of key.

After channel decoder we have this de-encryption block and as I said its objective is to recover black, back the plain text from the cipher text.

(Refer Slide Time: 18:09)

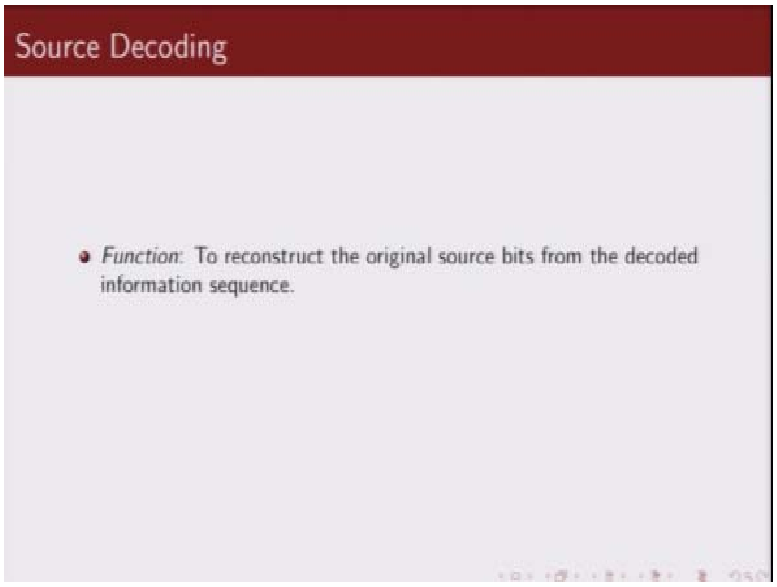


Decryption

- *Function*: To recover the plain text from the cipher text with the help of key.
- It is in the key that the security of a modern cipher lies, not in the details of the cipher.

And essentially the security of the system lies in the key.

(Refer Slide Time: 18:15)

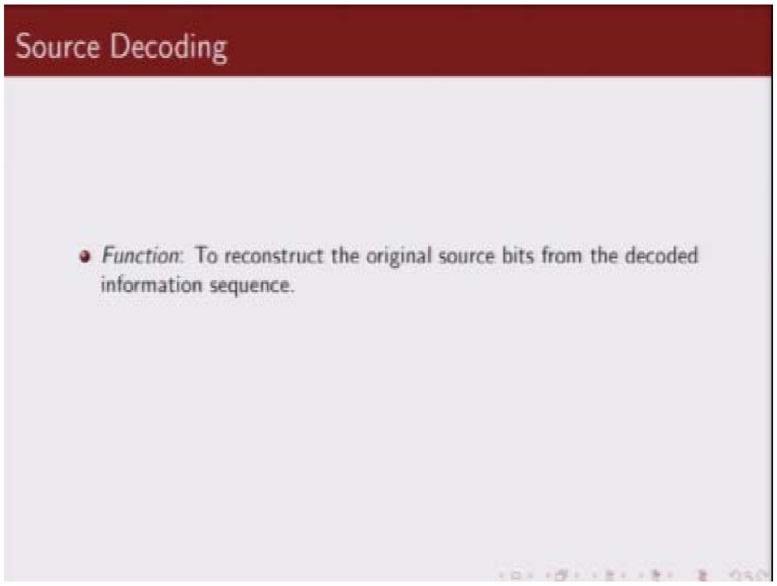


Source Decoding

- *Function*: To reconstruct the original source bits from the decoded information sequence.

And finally we would like to get back our original source, because remember in source encoder what we did was we did data compression. So we have been processing this compressed data, so from the compressed data we would like to get back our original source.

(Refer Slide Time: 18:32)

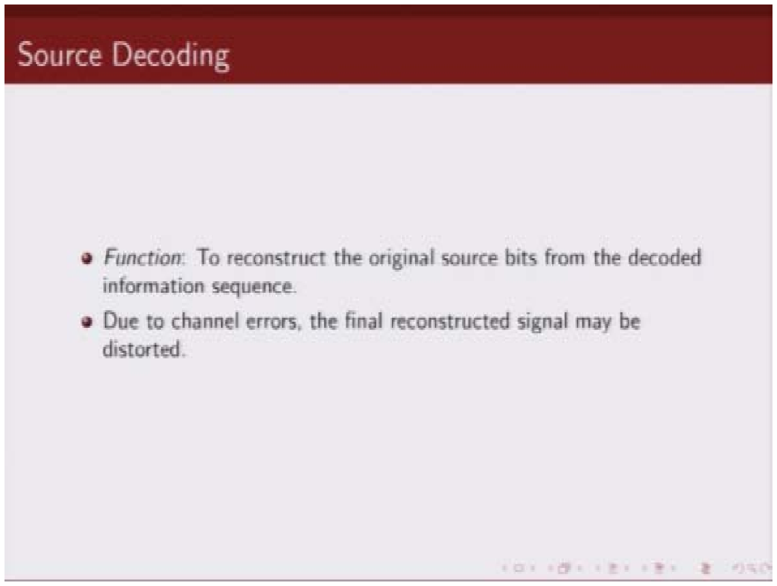


Source Decoding

- *Function*: To reconstruct the original source bits from the decoded information sequence.

And that is basically this source decoder. So it is reconstruct back the original source bits from the decoded information sequence. And of course if there are errors in our final you know decoded sequence would not be the same as what was transmitted.

(Refer Slide Time: 18:51)



Source Decoding

- *Function*: To reconstruct the original source bits from the decoded information sequence.
- Due to channel errors, the final reconstructed signal may be distorted.

So with this I will end this lecture, thank you.

<u>Acknowledgment</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 Gairola
Dilip Katiyar
Sharwan
Hari Ram

Bhadra Rao
Puneet Kumar Bajpai
Lalty Dutta
Ajay Kanaujia
Shivendra Kumar Tiwari

an IIT Kanpur Production