Error Control Codes : Theory and Practice.

Week 1 : Session 1

Error Control codes are integral part of Digital communication system.

OUTLINE

- Briefly discuss Error correcting codes and where do they fit in in the communication system.
- Two types of Error Correcting codes
  - Block codes
  - Convolutional codes

INTRODUCTION

All communication involves three basic steps.

- Encoding a message at its source (Source might be Analog which has to be Digitized)
- Transmitting that message through a Communication medium.
- Decoding the message at its destination

Recall, since the message is sent over Communication channel, the channel introduces Errors (noise in channel / Fading / Interference from other users). So, the receiver has to recover back the signal from the received signal using Error correcting codes to recover original signal from the Noisy received sequence.

INFORMATION THEORY

- The transmission medium in communication is known as channel.
- SHANNON, in his landmark paper in 1948, "A mathematical theory of communication", in Bell System Technical Journal, introduced the concept of Channel Capacity.

  "As long as we transmit at rate less than channel capacity, we can reliably communicate over a communication channel."
- Channel capacity is a measure of the amount of information that can be conveyed between the input X and the output Y of a channel.

Shanon showed that there exists error correcting codes with rate less than capacity.

- SHANNON, in his celebrated "Noisy channel coding theorem" proved the existence of channel coding schemes that can achieve an arbitrarily low error probability as long as the information can be transmitted across the channel at a rate less than the channel capacity, C.

However, Shannon did not tell us how to design such codes.

- Example: If the channel capacity of a particular communication link is 2 Gbps (say), then we can communicate over this channel at any desired rate less than 2 Gbps, and achieve arbitrary low error rates.

CODING THEORY

SHANNON, in his theorem, he did not specify how to construct codes with rate close to Channel capacity. It is the effort of coding theorist to come up with codes with rate close to channel capacity.

Shannon did not specify particular codes that achieve this limit with reasonable implementation complexity, however the goal of coding theory is to reach this limit.

So, coding theorist were involved in designing codes with rate close to channel capacity, which can be used to reliably communicate over a communication channel.

So, how do we design an Error correcting code?

Error correcting code is designed by properly adding redundant bits (parity bits) to the source bits. These redundant bits facilitate the detection and correction of transmission (storage) errors.

Error detection → Try to find out whether the message has error or not.

Error correction → First involves Error detection, figure out where error has occured and correct that error.

Channel coding is used in digital communication systems to control transmission errors caused by channel noise, fading, interference.

Also, in digital storage systems, channel coding is used to control errors caused by storage medium defects, dust particles, radiation.

CHANNEL CODING / ERROR CORRECTING CODES

Example : Repetition Codes

In Repetition codes, the redundant bits are nothing but repeatation of the information bits.

Rate, R is the ratio of the information bit and coded bit.

Rate, $R = \frac{1}{2}$ code means "for 1 information bit, we are sending 2 coded bits".

⇒ For 0 , 00 will be sent
For 1 , 11 will be sent

Similarly, Rate $R = \frac{1}{3}$ code means "for 1 information bit, we are sending 3 coded bits".

⇒ For 0 , 000 will be sent
For 1 ; 111 will be sent

Now, let us see an example to illustrate how we can detect errors, using Repetition codes.

Assume, we are sending 0 1 0 0 sequence.

Now, code the information sequence using Rate, $R = \frac{1}{2}$ repetition code.

0 1 0 0

↓

00 11 00 00

Let's say, the first bit is received in error.

What we received is

10 11 00 00

Since Rate $R = \frac{1}{2}$, decoding is done every 2 bits.

10 11 00 00
↑ Not expected.
can expect 00 (or) 11.

So, clearly there is an error. Therfore, using $R = \frac{1}{2}$ Repetition code, we can detect single bit error.

Can we correct the error?

NO, coz we are not certain that 10 is either 00 (or) 01. So, we can't correct the error in this case.

Now, let us see an example for $R = \frac{1}{3}$ code.

Assume, we are sending 0100 sequence.

Now, code the information sequence using Rate, $R = \frac{1}{3}$ repetition code.

0 1 0 0
↓
000 111 000 000

Let's say, the first bit is received in error.

What we received is

100 111 000 000

While decoding, we need to look at 3 bits at a time.

100 111 000 000
↑ Not expected.
can expect 000 (or) 111.

So, clearly there is an error.

Can we correct the error?

YES, coz it is more likely that 000 got changed as 100, rather than 111 got changed to 100. Hence 100 can be decoded as 000.

Using $R = \frac{1}{3}$ repetition code, we can correct single bit error which was not possible using $R = \frac{1}{2}$ repetition code.

MOTTO OF CHANNEL CODING

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.

— Solomon Golomb.

Now, let us see the DC System and see where does the Error Control coding blocks fit in.

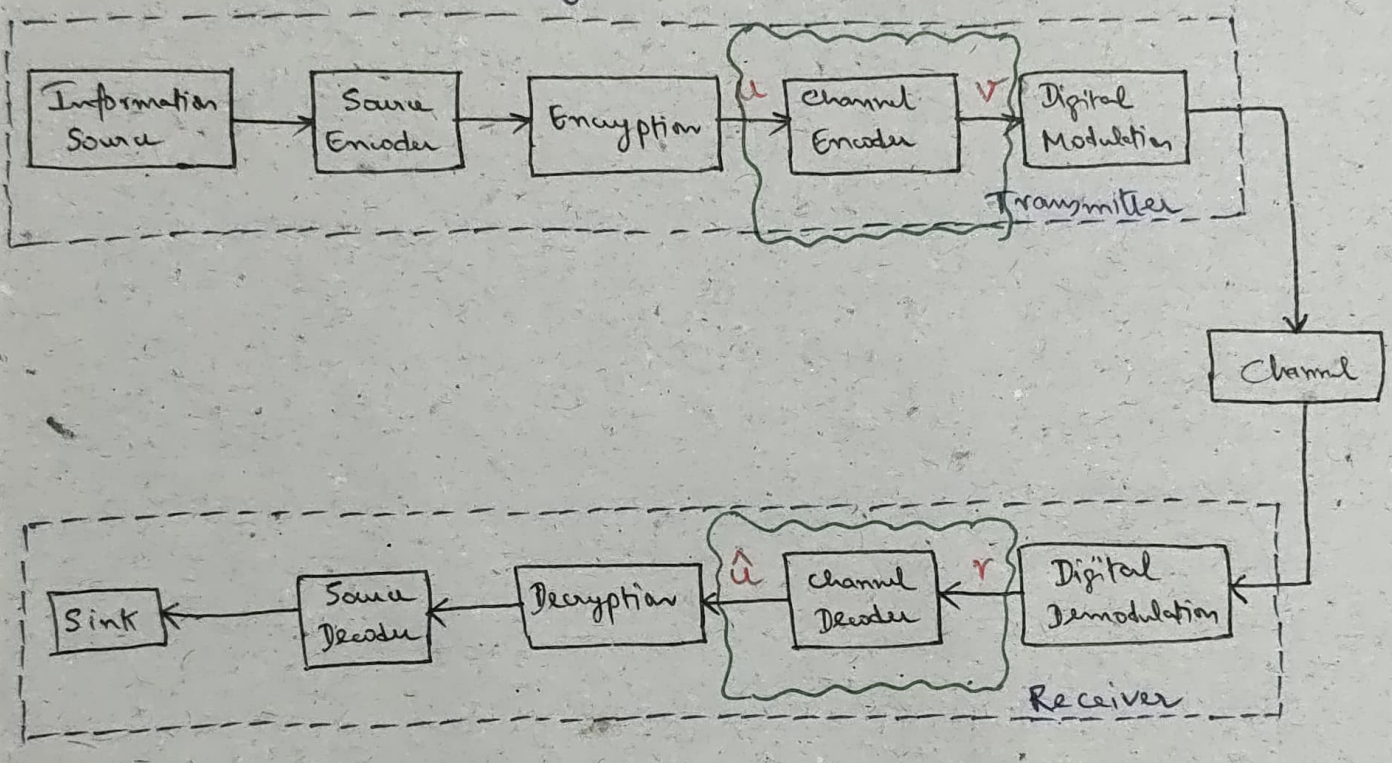


Fig. Block diagram of digital Communication System.

Information Source

The information source may be ANALOG. If so, the analog signal is quantized to make into DIGITAL bit stream.

Source Encoder

The DIGITAL information bit stream is fed into the Source Encoder. Source Encoder is a Data Compression module, which will get rid of redundant information bits.

Encryption

Now, this compressed information sequence is encrypted for security.

Channel Encoder / Error control coding Block:

The Encrypted data (u) is fed into Channel Encoder.

The output of Channel Encoder is the coded sequence (v).

How is 'v' related to 'u' ?

The coded sequence 'v' is a function of the information sequence, as well as we add some redundant bits called Parity bits.

And, the ratio of NO. of Information bits to the NO. of coded bits is known as RATE,

$$\Rightarrow \text{Rate} = \frac{k}{n} = \frac{\text{Information block length}}{\text{coded block length}}$$

Digital modulation

Now, these coded bits are modulated and sent over the communication channel.

This communication channel will introduce noise, errors as a result of passing through this transmission medium.

Digital Demodulation

At the receiver, what we get is the Noisy modulated version of the information sequence.

The first operation at the receiver is the demodulation of the information sequence and get back the Noisy version of the coded sequence.

Channel Decoder

The channel Decoder takes the Noisy coded sequence as input and produces the Estimate of the sequence. This estimate $\hat{u}$ is expected to be same as what we have transmitted without any error. (ie) $\hat{u} = u$.

Decryption

Once we have estimated the information sequence, we decrypt the sequence and we get back compressed receive sequence.

Source Decoder

The objective of the Source Decoder is to decompress and get back the original sequence.

In this course, we are concerned about 2 blocks.

* Channel Encoder
* Channel Decoder

Channel Encoder

How given 'k' information bits are coded into 'n' coded bits. is the objective of this block.

Channel Decoder

Given the noisy received sequence, how to get the good estimate of the information sequence. is the objective of this block.

SOURCE CODING

- Function: To minimize the number of bits per unit time (using data compression) 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.

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

CHANNEL CODING

- **Function**: To correct transmission errors introduced by the channel.
  (we pad some redundant bits to the information bits and we make use of those redundant bits at the receiver to correct errors)
- 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, Reed-Solomon codes (used in CD), CRC codes.
- The encoded sequence, that is the output of the channel encoder is referred to as **Codeword**.

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).

CHANNEL

- Now, these waveforms are sent over the communication medium.
- The physical transmission medium (Channel); it can be Wireline / Wireless
- The channel corrupts the transmitted waveforms due to various effects such as Noise, Interference, Fading and multipath transmission.
- **Examples**: Binary Erasure Channel (BEC), Additive White Gaussian Noise (AWGN).

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.

"From noisy waveform, if we directly get the sequence of bits, it is known as HARD demodulation. On the other hand, if the demodulator outputs are unquantized (or) quantized to many levels, we call it SOFT demodulation".

What is the advantage of SOFT demodulation over HARD demodulation?

In HARD demodulation, since the noisy waveform is directly converted to sequence of bits (0's and 1's), the receiver doesn't know how much the probability of 0's and 1's occurs. For example,

Let's say P(0) in case A is 0.51 and P(1) = 0.49
P(0) in case B is 0.99 and P(1) = 0.01

In case of HARD demodulation, in both the above cases, the receiver will convert the waveform into 0 as $P(0) > 0.5$. In case B, it is very much likely that the transmitted bit is 0, whereas in case A, the transmitted bit could be 1 very well as P(1) = 0.49.

In HARD demodulation, both the waveforms are converted to 0. "And in case B, we are more confident that the transmitted bit is 0". This information is not available in case of HARD demodulation.

In SOFT demodulation, we not only keep the bit sequence, but we also keep the information alive with what probability a particular bit is going to be 0 (or) 1.

This is why, in SOFT demodulation, the performance is much better.

⊙ Soft demodulation has significant improvement in performance than Hard demodulation.

CHANNEL DECODING

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

⊙ If the estimate $\hat{u}$ is not same as the transmitted sequence u, (ie) $\hat{u} \neq u$, then decoding errors have occured.

⊙ 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.

BER → If 'x' number of bits are received in error out of 'N' bits that are transmitted, then the BER $= \frac{x}{N}$.

→ (ie) The Fraction of bits that are received in error.

FER → 'Frame' is a Block of data. When we say Frame is in error, if any bits in that block of data is in error.

- The BER is defined as the expected number of information bit decoding errors per decoded information bit.,
- The coded sequences can be broke 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.

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.

SOURCE DECODING

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

2 TYPES OF ERROR CORRECTING CODES / CHANNEL CODES

① BLOCK CODES

② CONVOLUTIONAL CODES

BLOCK CODES

⊙ As the name suggests, we send data in Blocks.

(ii) The information sequence is partitioned into message blocks of k information bits each, represented as.

$$\text{Message Block}, u = \underbrace{u_0, u_1, \ldots, u_{k-1}}_{k\text{-bits}}$$

⊙ Now, the encoder maps each block of k information bits to an n bit codeword.

$$\text{Codeword}, v = \underbrace{v_0, v_1, \ldots, v_{n-1}}_{n\text{-bits.}}$$

⊙ Therefore, Our Coded Sequence is the block of n-bits.

⊙ The main property of Block code is that the encoder is MEMORYLESS.

How we map this block of k-bits (Message Block) to n-bits (codeword), depends only on those k-bits, and does not depend on how we have encoded the previous blocks of data.

When we say, the encoding process is memoryless, it does'nt depend on past encoding, and it only depends on the current k-bits.

⊙ The ratio of information bits to the coded bits is known as code rate, R.

$$R = \frac{k}{n}.$$

And, $(n-k)$ → No. of redundant bits (also known as parity bits) added to each message to protect against errors.

⊙ If we are considering BINARY code words (i.e. the inputs are in 0's and 1's) and if we are considering our information sequence of length k, then basically there are 2^k possible information sequences.

Now, these 2^k possible information sequences are coded into block of 'n' bit.

Therefore, we say "The set of 2^k code words of length n is called a 'binary (n, k) block code"

where; n → Output codeword length

k → Input information sequence length.

⊙ The codeword sequence, in general, can be non-binary, but we only consider Binary codes since they are the most commonly used in practice.

EXAMPLE OF BLOCK CODE

Let us consider information sequence of length $k = 3$,
Codeword of length $n = 6$

$\therefore$ code rate, $R = \frac{k}{n} = \frac{3}{6} = \frac{1}{2}$

Message			Codewords					
U_1	U_2	U_3	V_1	V_2	V_3	V_4	V_5	V_6
0	0	0	0	0	0	0	0	0
1	0	0	0	1	1	1	0	0
0	1	0	1	0	1	0	1	0
1	1	0	1	1	0	1	1	0
0	0	1	1	1	0	0	0	1
1	0	1	1	0	1	1	0	1
0	1	1	0	1	1	0	1	1
1	1	1	0	0	0	1	1	1

Modulo-2 Addition

$0 + 0 = 0$
$0 + 1 = 1$
$1 + 0 = 1$
$1 + 1 = 0$

Encoding equations (How 3 bits are mapped to 6 bits)

$V_4 = U_1$ | $V_1 = U_2 + U_3$
$V_5 = U_2$ | $V_2 = U_1 + U_3$
$V_6 = U_3$ | $V_3 = U_1 + U_2$

CONVOLUTIONAL CODES

- Unlike Block codes, A convolutional encoder process the information sequence continuously.
- The n-bit encoder output at a particular time depends not only on the k-bit information sequence, but also on m previous input blocks.
  - (ii) a convolutional encoder has a memory of order m.

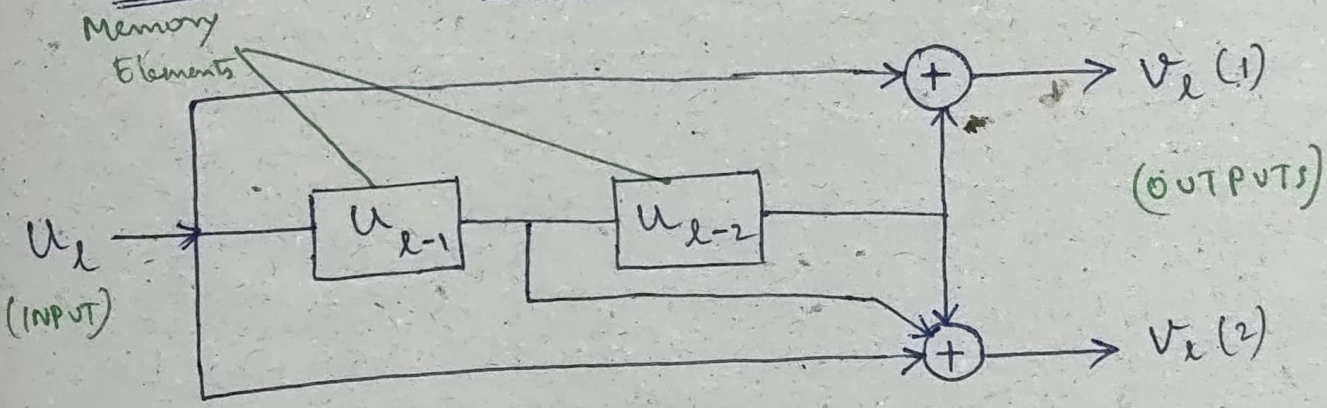
So, to get n-bit output, the convolutional encoder not only depends on the k-bit information sequence, but also depends on m past input blocks.

⊙ The set of sequences produced by a k-input, n-output encoder of memory order m is called an (n, k, m) convolutional code.

So, to describe a convolutional code, we not only need the output codeword length 'n', input information sequence length 'k', but we also need to specify the memory 'm' of the convolutional code.

⊙ Typically, the values of n and k are much smaller for convolutional codes compared to the Block codes.

EXAMPLE OF CONVOLUTIONAL CODE



Here, $k=1$, $n=2$ and $m=2$; Rate $= \frac{1}{2}$ convolutional code

Outputs, $v_l(1) = u_l + u_{l-2}$

$v_l(2) = u_l + u_{l-1} + u_{l-2}$

So, this is (2, 1, 2) convolutional encoder.

Note: Typically, the values of k and n are much smaller for convolutional code than that of the block code.

In convolutional code, the $k = 1/2/3$

$n = 2/3/4$

In Block code, the k and n values are in 100's, 1000's (or) even 10,000's.

Week 1: Session 2

DECODING STRATEGIES

Given that we have received a NOISY information sequence, how do we estimate the information bits from that?

- The decoder produces an estimate $\hat{u}$ of the information sequence u, based on the received noisy sequence r.

If $\hat{u} = u$, there is no error. (or) Rather we have to Minimize $\hat{u} \neq u$. With this objective, we need to design Decoder.

- Equivalently, the decoder can estimate $\hat{v}$ of the code sequence and then use inverse encoder mapping to find the information sequence $\hat{u}$ corresponding to $\hat{v}$.

(a) Estimating $\hat{u}$ is equivalent to estimating the codeword $\hat{v}$, why because there is one-to-one mapping between u and v. (b) k-bit information sequence is mapped to n-bit code sequence. So, if we can estimate the code sequence, we can estimate the information sequence back also.

⊙ Decoding rule

"Assignment of estimate of codeword $\hat{v}$ to each of the received sequence r".

(or)

"Assignment of estimate of information sequence $\hat{u}$ to each of the received sequence r".

Our guiding principle is, we have to minimize Probability of error.

⊙ The average Probability of error is given by

$$P(E) = P(\hat{v} \neq v) \text{ (or) } P(\hat{u} \neq u) = \sum_r P(E|r) \cdot P(r)$$

$$= \sum_r P(\hat{v} \neq v | r) \cdot P(r)$$

Sum over all received sequence.

Probability of Error given the received sequence 'r'.

Probability of received sequence 'r'.

⊙ We have to choose $\hat{v}$ such that $P(\hat{v} \neq v | r)$ is minimized for each r.

⊙ Minimize $P(\hat{v} \neq v | r)$ is equivalent to

Maximize $P(\hat{v} = v | r)$.

⊙ Baye's rule

For each r, Compute $P(v/r) = \frac{P(r/v) \cdot P(v)}{P(r)}$

for every v, and choose v that maximizes $P(v/r)$.

↑ clearly, $P(r)$ does not depend on the selection of v.

⊙ So, we can equivalently say that, maximizing $P(v/r)$ is same as maximizing $P(r/v) \cdot P(v)$, since $P(r)$ does not depend on v.

⊙ Hence, the decoder which maximizes $P(v/r)$ is known as Maximum a-posteriori Probability (MAP) decoder.

A MAP decoder chooses $\hat{v}$ such that $P(v/r)$ is maximized.

⊙ Note that, If all codewords are equally likely, and $P(v)$ = a constant, in this case, maximizing $P(v/r)$ is same as maximizing $P(r/v)$

$P(v)$ is same for all v.

⊙ Hence, the decoder which maximizes $P(r/v)$ is known as Maximum Likelihood (ML) decoder.

A ML decoder chooses $\hat{v}$ such that $P(r/v)$ is maximized.

⊙ For a discrete memoryless channel (DMC) where each received symbol depends only on the corresponding transmitted symbol.

$$P(r/v) = \prod_{i} P(r_i/v_i).$$

(Product of...)

⊙ Since, $\log x$ is a monotone increasing function of x,

maximizing $P(r/v)$ is equivalent to

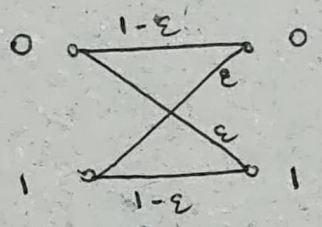
maximizing $\log P(r/v)$.

EXAMPLE : ML rule for BSC.

What is BSC?

In a BSC, we have 2 inputs and 2 outputs, (0, 1) (0, 1)

with probability $1-\varepsilon$, the bits are received correctly, And with cross-over probability ε, the bits are received in error.

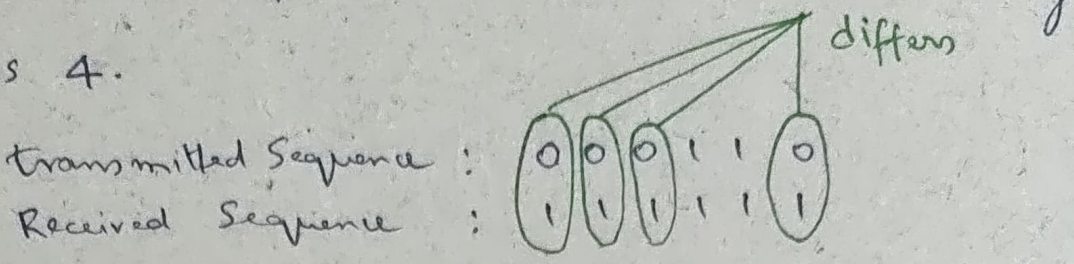


⊙ For a codeword of length n transmitted on a BSC channel with cross over probability p, what should be the ML decoding rule?

⊙ Hamming distance $d(r, v)$ between the received sequence r and the transmitted codeword v is defined as "the number of positions for which the received sequence differs from the transmitted sequence".

(i) $r_i \neq v_i$

Say for example, the transmitted sequence is 000110, and the received sequence is 111111. Then the Hamming distance is 4.

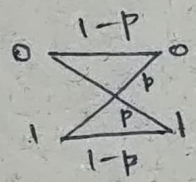


There are 4 bit locations where the received sequence differs from the transmitted sequence.

Therefore, $\boxed{d(r,v) = 4}$

Now, as we know in a BSC, we have 2 inputs & 2 outputs (0,1) (0,1)

with probability $1-p$, the bits are received correctly, And with Cross over probability of p, the bits are received incorrectly.



If we have a block of n information bits, let's say out of those n information bits, $d(r,v)$ number of bits are received incorrectly.

The Probability of receiving incorrect bits $\Big\} = p^d$

Remaining $n-d$ bits are received correctly.

The Probability of receiving correct bits $\Big\} = (1-p)^{n-d}$

We get, $A \cdot p^d \cdot (1-p)^{n-d}$

constant

By taking log, we get

$$d \cdot \log p + (n-d) \log (1-p)$$

$$\Rightarrow d\left(\log p - \log (1-p)\right) + n \log (1-p)$$

$$\Rightarrow d \log \left(\frac{p}{1-p}\right) + n \log (1-p)$$

$$\Rightarrow \boxed{\log P(r/v) = d(r,v) \log\left(\frac{p}{1-p}\right) + n \log (1-p)}$$

Now, ML decoder will choose $\hat{v}$, which will maximize $P(r/v)$.

And the second term "$n \log (1-p)$" depends on the codeword length n and cross over probability p. It is independent of selection of v, whereas the first term depends on v.

Typically, the cross-over probability $p < 0.5$.

Then $\left(\frac{p}{1-p}\right)$ will be a fraction ranges between 0 and 1.

Therefore, $\log \left(\frac{p}{1-p}\right)$ would be a Negative quantity.

So, in order to maximize $\log P(r/v)$, we have to maximize $-d(r,v)$ (Negative of Hamming distance)

Maximizing Negative of Hamming distance $\Leftrightarrow$ Minimizing Hamming distance between the Received sequence 'r' and the transmitted codeword 'v' $\Leftrightarrow$ ML Rule for a BSC.

⊙ For each r, choose $\hat{v}$ as the codeword v which minimizes the Hamming distance $d(r, v)$.

DIFFERENT WAYS IN WHICH WE CAN APPLY ERROR CODING SCHEME

① FORWARD ERROR CORRECTION (FEC)

⊙ In one-way system, transmission takes place only in one direction. (ie) from transmitter to receiver

⊙ The error correcting codes used in such a system are referred as Forward Error Correction (FEC) codes.

"FEC is used in transmission scheme where there is one-way communication. I am trying to communicate to you, but there is no communication path from you to me. In those situation, we use FEC codes".

When I am trying to encode a message sequence, I should encode in such a way that most of the errors are corrected when the packet is sent, coz there is no communication from the receiver to the transmitter. If the packet is not correctly received, the transmitter will have no knowledge about it.

② AUTOMATIC REPEAT REQUEST (ARQ)

⊙ In two-way system, there exists a feedback path from the receiver to the transmitter.

⊙ Error correction can be achieved for two-way system using error detection and retransmission, also known as ARQ.

"Unlike FEC, in ARQ, the packet is sent over the communication link, and there is a feedback link to the transmitter from the receiver".

The receiver will send ACK (if the packet is correctly received) or NACK (if the packet is not received correctly). Upon receiving the NACK, the transmitter will retransmit the package.

③ HYBRID AUTOMATIC REPEAT REQUEST (HARQ)

- For channels with feedback, one can use ARQ protocols in combination with FEC to improve system performance. These types of schemes are known as hybrid-ARQ (HARQ) schemes.
- In HARQ protocols, the transmitted data is encoded for both error correction and error detection. If the receiver detects an error after decoding, it sends a NACK to the transmitter, which then retransmits the data.

" The technique of combining FEC and ARQ is known as HARQ. This is also used in two-way channel, where FEC coded packet is sent to the receiver, which will try to detect error; if detected will try to correct it. If even after correction, if it fails to correct all the errors, it will send NACK, and the transmitter will send the packets again"