Unit- Six

IMPAIRMENTS, ERROR HANDLING AND COMPRESSION TECHNIQUES

- ❖ Attenuation & Distortion, Delay Distortion, Noise & Types, interference, crosstalk
- ❖ Types of error & its Detection and Correction Methods
- ❖ Data Compression, Lossless Compression - Run Length Coding, Dictionary Coding and
- ❖ Huffman Coding
- ❖ Lossy Compression - Predictive Coding and Transform Coding

Compiled by: Er. Ganesh Datt Joshi, Lecturer at NAST

Impairment

In communication system, the signal that is received may differ from the signal that is transmitted due to various transmission impairments. For analog signals, these impairments can degrade the signal quality & for digital signals bit errors i.e. altering of bits may be introduced. The basic signal impairments are; attenuation and distortion.

Attenuation

Attenuation means a loss of energy. When a signal travels through a medium its strength decreases with distance. Attenuation is generally exponential and thus is typically expressed as a constant number of describes per unit distance. Amplifiers and repeaters are used to increase the signal strength. Beyond a certain distance, the attenuation becomes unacceptably great and repeaters or amplifiers are used to boost the signal at regular intervals.

Distortion

Distortion means that the signal changes its from or shape i.e. it is unwanted signal fluctuating in signal and in-deterministic manner. Each signal component has its own propagation speed through a medium and therefore, its own delay through a medium in arriving at the destination.

Distortion

Types:

i) **Amplitude Distortion:** distortion in amplitude of signal.

ii) **Phase Distortion:** Change in phase signal. Generally, it occurs

Noise

Noise is defined as the unwanted from of energy which tend to interfere with the proper reception and reproduction of transmitted signals.

Types:

i) External Noise

Noise which is generated external to a communication system is known as external noise. For e.g.: Atmospheric noise, industrial noise, galactic noise noise.

Noise

Types:

ii) Internal Noise

Noise which is generated internally or within the communication system is known as internal noise. For e.g.: shot noise, thermal noise etc.

Crosstalk

It is the phenomenon of hearing another conversation. It is due to unwanted coupling between single paths. It can occur by electrical coupling between nearby twisted pair or coaxial cable (rarely) lines that carry multiple signals.

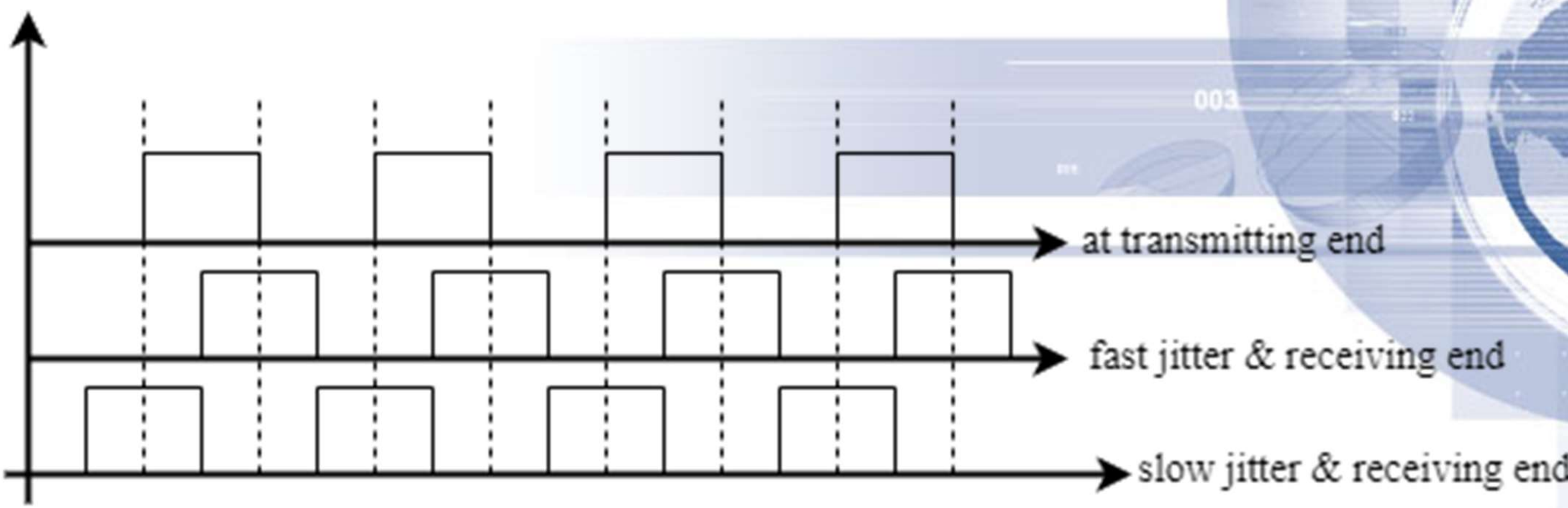
In case of real-life experience of crosstalk, it is while using the telephone, we hear another conversation.

Echo

It is the result of transmitting signals being coupled into return path and feedback to the respective source. It occurs because of coupling mechanism between the two wires and four wires cables. The signal reflected to the speaker and of the circuit is called talker echo.

Jitter

It is the movement of zero crossing of signal from their expected time of occurrence.



Error

In digital transmission system, an error occurs when a bit is altered between transmission and reception i.e. binary '1' is transmitted and a binary '0' is received and vice-versa. There are generally two types of errors they are; single bit error and burst error.

A single bit error is an isolated error condition that alters one bit but doesn't affect nearly bits.

Error

A burst error of length B is a continuous sequence of B bits in which the first and last bits and any number of intermediate bits are received in error. A burst error means that two or more bits in the data unit have changed.

01101001	01100001

Single Bit Error

01101001	00101101

Burst Error

Error Detection Methods

i) Parity Coding

It is the simplest method of coding in which parity bits are added to the message at the time of transmission. At the reception end, receiver checks these parity bit. Errors are detected if the expected pattern of parity bits is not received. Parity may be of two types.

Error Detection Methods

i) Parity Coding

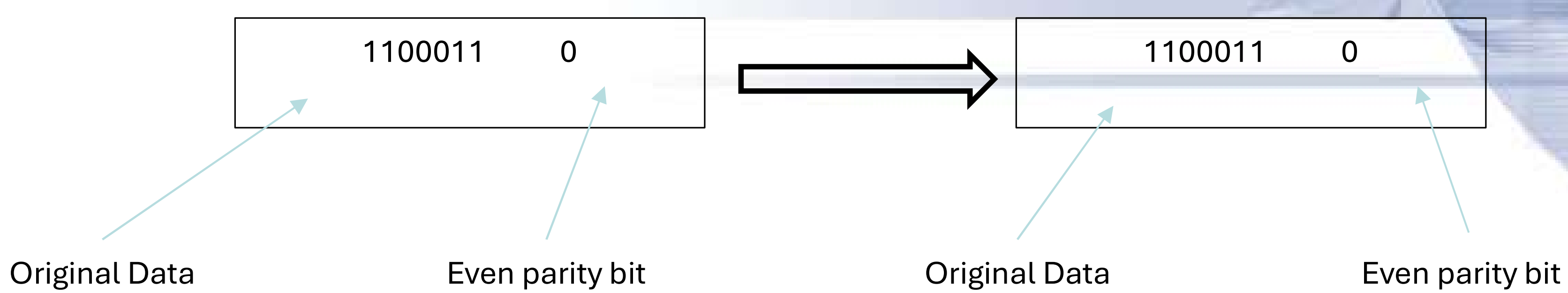
a. Even Parity: The parity of binary word is known as even if it contains even number of 1's. For e.g.: 10010011

b. Odd Parity: The parity of binary word is said to be odd if it has odd number of 1's. For e.g.: 10010010

Error Detection Methods

ii) Vertical Redundancy Check (VRC)

In this method the parity check is done on each character separately. In ASCII code, each character has 7 bits. Also, one redundant digit is added to each character to make the parity odd or even.



ii) Vertical Redundancy Check (VRC)

During transmission of code if error is occurred i.e.

1100011 0 ⇒ 11000111 1

(i.e. odd parity)

If the transmission impairments cause bit flip of '1' to '0' or '0' to '1', the parity check identify it and detect that error has occurred.

Error Detection Methods

ii) Vertical Redundancy Check (VRC)

If two digits are in error then no. of 1's in the received message is even. In this case receiver will detect no error. This is the drawback of VRC. VRC check error only in one direction i.e. either in row or in column.

iii) Longitudinal Redundancy Check (LRC)

In this method, a block of bit is organized in the form of rows. If we want to send 32 bits of data, we arrange them into a list of 4 rows.

Then the parity bit of each column is calculated and a new row of 8 bit is calculated and this becomes the parity bit for LRC as shown in figure. It is complex but can correct double or triple errors.

iii) Longitudinal Redundancy Check (LRC)

11010110	00111101	01101101	11100111

Now,

110101110 → 1 bit parity

110101110 → 1 bit parity

01101101 → 1 bit parity

11100111 → 0 bit parity

iv) Cyclic Redundancy Check (CRC)

It is one of the most common and most powerful error determining technique. In CRC, the user data stream is divided by a predetermined binary number.

The remainder to the number is appeared to the message as a frame check sequence field (FCS). Then the data is transmitted towards the receiving end. At the receiving end another calculation is performed and then result is compared with FCS field.

iv) Cyclic Redundancy Check (CRC)

If the remainders are same then the message is error free and thus accepted by the receiver. For e.g.:

Original message: 1010000

Generator polynomial: $x^3 + 1$

$$\therefore 1.x^3 + 0.x^2 + 0.x^1 + 1.x^0$$

i.e. (1001) which is 4-bit

If CRC generator is of n-bit then append $(n-1)$ zeros in the end of original message.

iv) Cyclic Redundancy Check (CRC)

The data are accepted and found to be correct only on the condition that the remainder of this division is zero.

iv) Cyclic Redundancy Check (CRC)

Sender

```
1001 ) 1010000000
       1001
       ----------
       0011000000
         1001
       ----------
           01010000
            1001
           --------
            0011000
               1001
           --------
               01010
               1001
           --------
               011
```

Message to be transmitted:

1010000000+011=1010000011

Receiver

```
1001 ) 1010000011
       1001
       ----------
       110000011
       1001
       ----------
           01010011
            1001
           --------
            0011011
               1001
           --------
               01001
               1001
           --------
               0000
```

Zero mean data is accepted.

v) Checksum

It is an error detection in which detect the error by dividing data into segments of equal size and then uses 1's complement to find the sum of the segment and then sum is transmitted the data to the receiver and same process 1's done by the receiver at receiver side, all zeros in the sum indicated the correctness of data.

First, data is divided into k's segments in a checksum error detection scheme and each segment has 'm' bits.

v) Checksum

For finding out the sum at the sender's side all segments are added through 1's complement arithmetic. And for determining the checksum we complement the sum.

Along with data segments, the checksum segments are also transferred. All the segments that are received on the receiver side are added through 1's complement arithmetic to determine the sum. Then complement the sum.

v) Checksum

The received data is accepted only on the condition that the result is found to be 0. And if the result is not 0 then it will be discarded

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

v) Checksum

Sender

10011001
11100010
1 01111011
1
01111100
00100100
10100000
10000100
1 00100100
1
Sum: 00100101
Checksum: 11011010

Receiver

10011001
11100010
1 01111011
1
01111100
00100100
10100000
10000100
1 00100100
11011010 (Checksum)
Sum: 11111111
Checksum: 00000000

Conclusion: Data accepted

Error Correction Methods

i) Hamming Codes

A Hamming code is a technique developed y R.W. Hamming for finding out the position of the error bit. It is based on the relationship between the redundant bits and data units and its main advantages is that it can be applied to data units of any length.

i) Hamming Codes

Working:

The position of the data bit and the number of redundant in the original data. The number of redundant bits is deduced from the expression $[2^r > d + r + 1]$

Fill in the parity data bit and redundant bit and find the parity bit value using the expression. $[2^p; where, p = \{0,1,2, \dots \dots \dots n\}]$

Error Correction Methods

Working:

Fill the parity bit obtained in the original data and transmit the data to the receiver side.

Check the receives data using the parity bit and detect any error in the data to and in case damage is present, use the parity bit value to correct the error.

Error Correction Methods

Working:

For e.g.: The data bits to be transmitted is 1011010, to be solved using the Hamming Code Method.

The $data\ bit = 7$

The redundant bit,

$$2^r \geq d + r + 1$$

$$2^4 \geq 7 + 4 + 1$$

$16 \geq 12$, [So that value of $r = 4$].

Working:

Position of the redundant bit, applying the 2^p expression.

$$2^0 = p_1$$

$$2^1 = p_2$$

$$2^2 = p_4$$

$$2^3 = p_8$$

Error Correction Methods

Working:

So, the data with parity

1	0	1	p_8	1	0	1	p_4	0	p_2	p_1

Working:

Finding parity bit

a. p_1 parity bit is deduced by checking all the bit with is in the last significant location.

$$p_1: 1,3,5,6,8,11$$

$$p_1 - p_1: 0,1,1,1,1$$

$p_1 - 0$

b. p_2 parity bit is deduced by checking all the bits with 1's in the second significant location.

$$p_2: 2,3,6,7,10,11$$

$$p_2: p_2, 0,0,01,0,1$$

$$p_2: 0$$

Working:

c. p_4 parity bit is deduced by checking all the bits with 1's in the third significant location.

$$p_4: 4,5,6,7$$

$$p_4: p_4, 1,0,1$$

$$p_4: 0$$

d. p_8, fourth significant location.

$$p_8: 8,9,10,11$$

$$p_8 - p_1, 1,0,1$$

$$p_8 - 0$$

Working:

So, the original data to be transmitted to the receiver side is;

11	10	9	8	7	6	5	4	3	2	1
1	0	1	0	1	0	1	0	0	0	0

Data Compression

It is a digital signal process in which data to be transmitted is compressed to reduce the storage amount in bits. To send audio, video, and other file over the internet we require compression techniques.

❖ Lossless Compression

In this method, compression is done through identifying and eliminating any statistical redundancy. For instance, when we encode a data source before transmitting its size is effectively reduced and data remains intact and unchanged. Some lossless techniques are:

❖ **Lossless Compression**

a. Run Length Coding (RLC)

It is a simple and widely used techniques for lossless data compression. It works by replacing sequence of identical elements with a single element and a count of how many times that element is represented.

a. Run Length Coding (RLC)

It is classified into two types:

i) Encoding (Compression)

- Identify consecutive runs of identical symbols in the data.
- Replace each run with a pair of values; they symbol and the count of consecutive occurrences.

a. Run Length Coding (RLC)

For e.g.: consider a string

A	A	A	A	B	B	B	C	C	D	A	A

The compressed sequence becomes;

A	B	C	D	A
4	3	2	1	2

a. Run Length Coding (RLC)

ii) Decoding (Decompression)

- Reverse the process by expanding each pair back into the original sequence.

  A4 → AAAA

  B3 → BBB

  C2 → CC

  D1 → D

  A2 → AA

The decompressed sequence: AAAABBBCCDAA

a. Run Length Coding (RLC)

ii) Decoding (Decompression)

- RLC is most effective when there is long run of identical symbols in the data. It is commonly used in scenarios where such runs are likely to occur, such as in certain types of images, where large areas of the same color may exist.

❖ Lossless Compression

b. Dictionary Coding

It involves a creating a dictionary of frequently occurring patterns in the input data and the replacing these patterns with shorter codes. The most well-known dictionary coding algorithms are Lempol Zive Weich (LZM) and its variant.

❖ Lossless Compression

c. Huffman Coding

It is widely used techniques developed by David A-Huffman in 1952. It efficiently represents data by assigning variable length code to symbols based on their frequencies in the input. More frequent symbols are assigned shorter codes. While less frequent symbols receive longer codes. This results in a compact representation of the input data.

c. Huffman Coding

For e.g.: Consider the input "ABRACADABRA"

Algorithm:

- Frequency Analysis

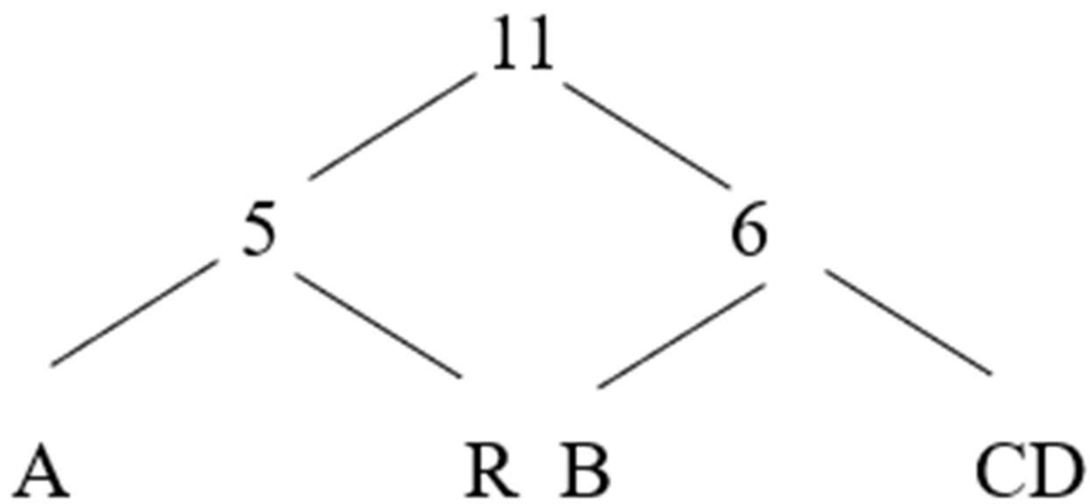
  Frequencies: {'A' : 5, 'B' :2, 'R' : 2, 'C' : 1, 'D' : 1}

  Total: 11

- Tree Construction: Build the Huffman

  Build a binary tree using a greedy algorithm, merging nodes with the lowest frequency until a single.

c. Huffman Coding



Create a priority queue based on symbol frequencies.

c. Huffman Coding

- Code Assignment

  'A' – '0'

  'B' – '10'

  'R' – '11'

  'C' – '100'

  'D' – '100'

- Encoding

  Replace each symbol with its corresponding Huffman code.

- Decoding

Reverse.

c. Huffman Coding

Advantages:

- Optimal compression for data with varying symbol frequencies.
- Used in file compression (ZIP), image compression (JPEG) and data compression.

❖ Lossy Compression

Some part of data is deleted or lost because it identifies and then delete unnecessary information before transmission. Some techniques are described as:

❖ Lossy Compression

a. Predictive Coding

- Based on the idea of predicting the value of data point based on the values of previous data point.
- The prediction error (difference between predicted and actual value) in then encoded and transmitted.

In the decoding process, the predicted values and the encoded errors are used to reconstruct the data.

a. Predictive Coding

Steps:

- Prediction: Predict the values of the current data point based on the values of previous data points.
- Error Coding: Encode the difference between predicted and actual values. The errors are what get compressed and transmitted.
- Decoding: Use the predicted values and the encoded errors to reconstruct the original data.

a. Predictive Coding

For e.g.: Consider a time series of audio samples. The current sample and could be predicted based on the previous sample and prediction error is what is encoded.

❖ Lossy Compression

b. Transform Coding

It involves transforming the data from it original representation to different domain, where its more amenable to compression. It includes the discrete cosine transformation (DCT) and discrete Fourier transformation (DFT).

b. Transform Coding

Steps:

- Transformation: Convert the data from its original representation to a different domain using a mathematical transform.
- Quantization: Reduce the precision of the transformed coefficients, discarding loss significant information.
- Coding: Encode the quantized coefficients using entropy coding techniques.

b. Transform Coding

For e.g.: In image (e.g. JPEG) the DCT is applied to image block. The resulting coefficient are quantized and the quantized coefficient are then encoded.