31341

Assignment 2

Title: Hamming Code/CRC

Problem statement: Write a program for error detection and correction for 7/8 bits ASCII codes using Hamming Codes or CRC. Demonstrate the packets captured traces using Wireshark Packet Analyzer Tool for peer to peer mode.

Prerequisites:

1. Object oriented Programming Concepts
2. Wireshark Packet Analyzer Tool

Objectives:

1. To understand how error detection and correction works with Hamming Codes and CRC.
2. To capture packets using Wireshark in peer to peer mode.

Outcome: The student will be able to:

1. Write a program for error detection and correction
2. Capture packets using Wireshark

Theory:

In digital systems, the analog signals will change into digital sequence (in the form of bits). This sequence of bits is called as "Data stream". The change in position of single bit also leads to catastrophic (major) error in data output. Almost in all electronic devices, we find errors and we use error detection and correction techniques to get the exact or approximate output.

· Types of Errors

In a data sequence, if 1 is changed to zero or 0 is changed to 1, it is called "Bit error". There are generally 3 types of errors occur in data transmission from transmitter to receiver. They

31341

are:

1. Single bit errors
2. Multiple bit errors
3. Burst errors

1. Single bit data errors:

The change in one bit in the whole data sequence, is called "Single bit error". Occurrence of single bit error is very rare in serial communication system. This type of error occurs only in parallel communication system, as data is transferred bit wise in single line, there is chance that single line to be noisy.

2. Multiple bit data errors:

If there is change in two or more bits of data sequence of transmitter to receiver, it is called "Multiple bit error". This type of error occurs in both serial type and parallel type data communication networks.

3. Burst errors:

The change of set of bits in data sequence is called "Burst error". The burst error is calculated in from the first bit change to last bit change. Here we identify the error form fourth bit to 6th bit. The numbers between 4th and 6th bits are also considered as error. These set of bits are called "Burst error". These burst bits changes from transmitter to receiver, which may cause a major error in data sequence. This type of error occurs in serial communication and they are difficult to solve.

· Error detecting codes

In digital communication system errors are transferred from one communication system to another, along with the data. If these errors are not detected and corrected, data will be lost .For effective communication, data should be transferred with high accuracy. This can be achieved by first detecting the errors and then correcting them. Error detection is the process of detecting the errors that are present in the data transmitted from transmitter to receiver, in a communication system. We use some redundancy codes to

31341

detect these errors, by adding to the data while it is transmitted from source (transmitter). These codes are called "Error detecting codes".

- Types of Error detection
  1. Parity Checking
  2. Cyclic Redundancy Check (CRC)
  3. Longitudinal Redundancy Check (LRC)
  4. Check Sum

1. Parity Checking

Parity bit means nothing but an additional bit added to the data at the transmitter before transmitting the data. Before adding the parity bit, number of 1's or zeros is calculated in the data. Based on this calculation of data an extra bit is added to the actual information /data. The addition of parity bit to the data will result in the change of data string size. This means if we have an 8 bit data, then after adding a parity bit to the data binary string it will become a 9 bit binary data string. Parity check is also called as "Vertical Redundancy Check (VRC)". There is two types of parity bits in error detection, they are

  1. Even parity:

  If the data has even number of 1's, the parity bit is 0. Ex: data is 10000001 -> parity bit 0 Odd number of 1's, the parity bit is 1. Ex: data is 10010001 -> parity bit 1.

  2. Odd parity:

  If the data has odd number of 1's, the parity bit is 0. Ex: data is 10011101 -> parity bit 0 Even number of 1's, the parity bit is 1. Ex: data is 10010101 -> parity bit 1.

2. Cyclic Redundancy Check

A cyclic code is a linear (n, k) block code with the property that every cyclic shift of a codeword results in another code word. Here k indicates the length of the message at transmitter (the number of information bits). n is the total length of the message after

adding check bits. (actual data and the check bits). n, k is the number of check bits.The codes used for cyclic redundancy check there by error detection are known as CRC codes(Cyclic redundancy check codes). Cyclic redundancy-check codes are shortened cyclic codes. These types of codes are used for error detection and encoding. They are easily implemented using shift-registers with feedback connections. That is why they are widely used for error detection on digital communication. CRC codes will provide effective and high level of protection.

· Error correcting codes

The codes which are used for both error detecting and error correction are called as "Error Correction Codes". The error correction techniques are of two types. They are,

- Single bit error correction
- Burst error correction

The process or method of correcting single bit errors is called "single bit error correction". The method of detecting and correcting burst errors in the data sequence is called "Burst error correction".Hamming code or Hamming Distance Code is the best error correcting code we use in most of the communication network and digital systems.

· Hamming Code

This error detecting and correcting code technique is developed by R.W.Hamming. This code not only identifies the error bit, in the whole data sequence and it also corrects it. This code uses a number of parity bits located at certain positions in the codeword. The number of parity bits depends upon the number of information bits. The hamming code uses the relation between redundancy bits and the data bits and this code can be applied to any number of data bits.

· How the Hamming code actually corrects the errors?

In Hamming code, the redundancy bits are placed at certain calculated positions in order to eliminate errors. The distance between the two redundancy bits is called "Hamming distance". To understand the working and the data error correction and detection mechanism of the hamming code, let's see to the following stages. Number of parity bit. As

31341

we learned earlier, the number of parity bits to be added to a data string depends upon the number of information bits of the data string which is to be transmitted. Number of parity bits will be calculated by using the data bits. This relation is given below.

$2P >= n + P + 1$

Here, n represents the number of bits in the data string.

P represents number of parity bits.

For example, if we have 4 bit data string, i.e. $n = 4$, then the number of parity bits to be added can be found by using trial and error method. Let's take $P = 2$, then

$2P = 22 = 4$ and $n + P + 1 = 4 + 2 + 1 = 7$

This violates the actual expression.

So let's try $P = 3$, then

$2P = 23 = 8$ and $n + P + 1 = 4 + 3 + 1 = 8$

So we can say that 3 parity bits are required to transfer the 4 bit data with single bit error correction.

· Where to place these parity bits?

After calculating the number of parity bits required, we should know the appropriate positions to place them in the information string, to provide single bit error correction. In the above considered example, we have 4 data bits and 3 parity bits. So the total codeword to be transmitted is of 7 bits (4 + 3). We generally represent the data sequence from right to left, as shown below.

bit 7, bit 6, bit 5, bit 4, bit 3, bit 2, bit 1, bit 0

The parity bits have to be located at the positions of powers of 2. I.e. at 1, 2, 4, 8 and 16 etc. Therefore the codeword after including the parity bits will be like this

D7, D6, D5, P4, D3, P2, P1

Here P1, P2 and P3 are parity bits. D1 –– D7 are data bits.

· Constructing a Bit Location Table

In Hamming code, each parity bit checks and helps in finding the errors in the whole code word. So we must find the value of the parity bits to assign them a bit value.By calculating and inserting the parity bits in to the data bits, we can achieve error

correction through Hamming code.

Algorithm

Sender site:

1.Get message and convert it to binary string

2.Create H matrix

3.Compute redundant Hamming bits

4.Concatenate message and Hamming bits

5.Send frame

Receiver site:

1.Receive frame

2.Divide frame to message and code bits

3.Generate H matrix

4.Compute Hamming bits from message

5.Perform XOR operation between received and computed Hamming bits

6.If XOR result is all zero binary

  Message have no errors

7.Else

  Generate H matrix with identity

  Perform on matrix columns XOR operation with previous XOR result

  If some result is all zero bit

    1.The column number is faulty bit position

    2.Change frame bit on founded position to opposite

    3.Compute Hamming bits for frame message part

    4.If corrected frame Hamming bits equals computed Hamming bits

      Message is corrected

    5.Else There is more then one bit error cannot correct

  Else

    1.There is more then one bit error cannot