CS 2400 : Assignment 1

Nikitha CS14B009
Kavya Mrudula CS14B046

May 3, 2016

Introduction:

This assignment involves the following:

1. Finding the list of unique symbols in a text file, computing the probability of occurrence of each symbol and finally the entropy of the text file.
2. Generating a codebook for the text file using Huffman Coding and then sending the data through packets of fixed size through Channel Encoding
3. Generating bit-errors with a probability P_e and sending the message through this channel
4. Decoding the sent message and checking for the error in transmission

Entropy of the Text File:

Defn: Entropy is the average information contained in the message. Here the message is the text file sent across the server.

Entropy is denoted as $H(m)$ and is calculated as follows:

$$H(m) = \sum_{k=1}^{N} P_k \cdot \log \frac{1}{P_k} \qquad P_k = \frac{No.of Occurrences of k^{th} symbol}{Total No.of symbols}$$

File Size	Entropy
64.7 kb	4.68247
65 kb	4.54759
63.8 kb	4.61152
480.4kb	4.61799

Observation: Average Information is observed to be in the range $4.5 - 4.7$

Source Coding:

Huffman Coding: Huffman Coding is an example of optimal prefix coding that is commonly is used for lossless data compression.Being a prefix code, Huffman Coding is uniquely decodable.

ALGORITHM:

1. Form a priority queue with the frequency of each symbol being the key for comparison.
2. Dequeue the two least occurring symbols from the queue and add another special internal node with frequency equal to sum of the frequency of both the leaves.
3. Add this internal node to the queue and repeat step 1
4. The final node left in the queue will be the root to Huffman Tree.
5. To get the prefix code, assign a 0-weight to left and 1-weight to right nodes and find the path to each of the leaves in the tree. The path will give the prefix code for each of the symbol.

Channel Coding:

Cyclic Redundancy Checks: CRC code is generated in the following way:

1. Take the message to be sent and divide it with the CRC polynomial. Append the remainder to message and send it across the channel.
2. Here, we took a 33 bit CRC polynomial and fixed the size of packet as 1kb.
3. To avoid starting zeros, we have flipped the first 33 bits with exor.
4. To decode, the client side must follow the same procedure for error-free communication.

Steps to retrieve the Original Message and Error-Correction in Channel Coding:

1. First, we need to flip the first 33 bits of sent message. The fist 1024 bytes of this message will be our original message and next 32 bits will be the CRC code sent along with the message for error-protection.
2. There are several ways to do the error-checking. We took the first 1024 bytes and divided it with the CRC polynomial. A remainder is obtained.
3. Check if this remainder and the next 32 bits of the sent message are one and the same. If yes, then the transmission was successful.
4. If no, then the transmission failed and request for a re-transmission.

Thus, Channel Coding helps in detecting the error and retrieving the Original Message.

Error Generation and Transmission:

Error Generation: Generation of Error is needed to know how the transmission works within certain channel. We have generated errors with probability $P_e = 0.0043726$. These errors were found while transmitting across the channel. Hence, a re-transmission was asked by the client which was later performed.

Transmission through a Channel: Transmission through the Channel was made using the programs provided. We used UDP protocol to transmit the data from server to the client. The errors were found when $P_e \neq 0$ and were taken care of by transmitting data with no errors upon the request of re-transmission.

Decoding And Error Checking:

Decoding: Code Book generated through Source Coding was used to decode the encoded message and since Huffman code is a prefix code , there is only a unique decoded message for a given encoded message.
Steps involved are:

1. Start with reading message byte by byte which was retrieved through the decode steps mentioned in Channel Coding. Reading byte by byte, we stop when the string we read so far is code word for a certain symbol in the codebook.Since this is prefix code, there would not be any other symbol with this code as a prefix and hence replace this string read so far with the symbol in the code book.
2. After you replace the string with the symbol, discard the string and start a fresh string and continue with where you last read the string.

Detection: You can compare the final decoded file with the original message to get the exact error in the transmission, but most of the cases, the Client does not has the original message. In that case, we use CRC bits as mentioned earlier.