CS2400: Assignment 2

Srinidhi Prabhu, CS14B028 Vedant Somani, CS14B053

Objective:
In this experiment, we study the transmission of a text file from a client to a server using User Datagram Protocol(UDP).

Procedure:
1) Given a file, we compute the probabilities of the characters occurring in the file.
2) Using the calculated probabilities, a Huffman code is generated and stored as a code book.
3) Using this code book, a compressed file corresponding to the given file is generated.
4) The compressed file is divided into packets of fixed size(8 bits) and these packets are transmitted from client to server.
5) During transmission of the packets, errors may be introduced in the packets. To identify the errors, the packets are channel encoded using a CRC polynomial.
6) At the server side, we check for errors introduced during transmission(using CRC). If an error is detected, the server requests for re-transmission. If no error is detected, the bits are written to a file.
7) When the transmission by the client is complete, the server decodes the received file using the code book to generate the original text file.

Results:

(a) Compression of files using Huffman code: We compressed large files using a Huffman code generated for the file. We observed the following:

Sl. No.	Actual file size	Compressed file size	Compression rate	Entropy	Average code length
1.	6.9kB	4.9kB	28.9%	4.55	4.93
2.	63.9kB	45.1kB	29.4%	4.69	4.93
3.	165.2kB	113.7kB	31.2%	4.53	4.81
4.	795.8kB	113.7kB	85.7%	0.99	1.00
5.	4.0MB	3.1MB	22.5%	4.94	5.42

The fourth row in the table is the binary file corresponding to the file in the third row. Each 0 and 1 is written as a character, hence the file in the fourth row is much larger than the file in the third row. We observe that it is the redundancy in the file which is important for the compression and not the actual size. Both files represent the same data but in different forms. However, by removing all redundancy, we are able to compress both files to the same size.
The rows in the table other than the fourth row represent general text files. For such files, we observe a compression of nearly 30% using Huffman code.

(b) Channel encoding and receiving at the server:
We divided the compressed file into fixed size packets of seven bits. We channel encoded every packet before transmission using a CRC polynomial. We transmit the channel encoded packets from client to server. Assuming that our channel is noisy, we generated errors in every bit with a probability p.
The server checks for errors in the received packet using a Cyclic Redundancy Check(CRC). If it detects an error, it requests for a retransmission. Otherwise, it computes the original message and writes it to a file. We measure the number of packets transmitted and the number of transmissions for a particular probability. We observe that as the error probability increases, the number of transmissions increases.

The below table gives the results of transmission for a file of size 64kB using CRC polynomial 1011.

Variation of number of transmissions by client w.r.t error probability (number of packets = 113682)

Sl. No.	Error probability	Number of transmissions	Decoded file
1.	0.01	136164	Same as original
2.	0.02	181099	Same as original
3.	0.05	253293	Same as original
4.	0.08	304219	Differs from original
5.	0.1	377612	Differs from original

As we increase the error probability keeping the CRC polynomial fixed, the errors between the decoded file and the actual file increase. This can be explained as follows: As the error probability increases, the errors in a single packet can occur in such a way that they give the same remainder with the CRC polynomial. So, this error would not be detected by the server and we do not receive the correctly encoded packet. In practice, we would like to keep the error probability of a bit flip below 0.05, so that we can decode it correctly at the server.

We also observe that if we channel encode with a CRC polynomial of higher degree, we can successfully decode the file at the receiver even for relatively higher probabilities(compared to CRC polynomial with lower degree). This follows from Shannon's Channel Capacity Theorem.

Success of decoding at the server w.r.t degree of the CRC polynomial

Sl.no	CRC polynomial	Error probability	Decoded file
1.	1011	0.05	Same as original
2.	10110	0.08	Same as original
3.	10110	0.1	Differs from original
4.	101100	0.1	Same as original
5.	101100	0.15	Differs from original