ASSIGNMENT 2

Team Members:
Madhav Sainanee (2017063)
Jay Rawal (2017240)
Rishabh Sharma (2017087)
Dhruv Kundu (2017146)

Application 7: File Compression

The client in this application sends files to the server which are compressed and sent back to the client using the Lempel-Ziv compression technique.

Protocol at Client End

The client sends the file to be compressed to the server in multiple packets using the connection-oriented TCP protocol.
The client waits for an acknowledgement from the server before sending the next packet.
Initially, we generate a random file and send it to the server in several packets of size 1024 bytes (1 KB). The client disconnects by closing this socket

After the server compresses the file, the client again establishes a connection with the server to receive the compressed file. This time the client receives packets from the server and sends acknowledgements.

Protocol at Sever End

The server receives the file to be compressed from the client in multiple packets using the connection-oriented TCP protocol.
The server sends an acknowledgement every time it receives a new packet.

Then the server proceeds to compress the file.

After compressing, the connection with the client is established again and the file is sent packet by packet.

Observations

Original File Size	Compressed File Size	Sent Packets	Received Packets	Correctly received
92.1 MB	79.5 MB	92100	79500	Yes
42.1 MB	38.3 MB	42100	38300	Yes

Analysis

The established sockets were working seamlessly and we were getting fully lossless data compression. We were getting about 14% compression after performing Lempel-Ziv compression.

This is an iterative server since queries are handled sequentially as opposed to parallel.

References

1. https://www.geeksforgeeks.org/socket-programming-python/

2. https://stackoverflow.com/questions/27241804/sending-a-file-over-tcp-sockets-in-python
3. https://www.tutorialspoint.com/compression-using-the-lzma-algorithm-using-python-lzma
4. http://beej.us/guide/bgnet/html/multi/index.html