Socket Programming Project Report

Maria Hajj

December 2020

1 ServerCode.py

In this file is the code for the server that uses TCP to establish a connection between itself and the client. The port chosen for this assignment is 15010 and the buffer size is 1024 and these variables are declared in the beginning of this file.
The code contains a method "handle" which takes as argument the client and it's address (addr) which is defined in all the files as a tuple containing the host IP and the port number(TCP_IP, TCP_PORT). This method prints all the file information like the size then while reading the file cuts the file into chunks then hashes these chunks till the last byte is read and last chunk is formed. The hash will be sent also.
After the method comes the TCP socket set up where the
"(AF_INET, SOCK_STREAM)" indicates that the server creates a TCP socket then associates (bind) the server port number to the socket. Then the server has to "listen" to at least 3 clients (queued connections).
Finally, this code runs the class HeartBeat which is the UDP pinger that checks for request timeouts.

2 ClientCode.py

Same as the ServerCode.py, the code has in the beginning defined variables like the host and the port number then the set up of the TCP connection with the server where the TCP connection has to establish first a connection between the client and the server (3-way handshake) through the "sock.connect(addr)" line.
We have 2 extra variables in this file which are the chunks array to store and rebuild the chunks and the counter called receiver to keep track of the chunks received.
The 3 while loops in this file are to checks if all chunks arrived and that there are no missing chunks while also checking the correctness of the hashing. All of this is done through checking the number of bytes received and the total size which was sent by the server.

At the end of the file, the code checks is the hash is correct.

3 HeartBeat.py

The heartbeat class is a UDP pinger that checks the request if it timed out.
This code was taken from moodle from the solutions provided and made some minor modifications.
The code used to take the host and port number from the command line but now they're known variable. Also added a new boolean variable response that keeps track of the timeout response, if it's true then a timeout occured, else if it's false then there were no timeouts.
At the beginning of the class, the UDP connection is established where the "(AF_INET, SOCK_DGRAM)" indicates that the server creates a UDP socket. It then send a sequence of pings to then calculate the round trip time to know if the message was received on time or if it wasn't received and the request timed out.