PROXY
Assignment 2
Summer Research Intern NUS

Dipen Kumar
Third-year undergraduate
Computer Science & Engineering
Indian Institute of Technology Delhi

How to run the program –

1. "client.py" will take the path along with the name of the input file as command line argument.
2. Input file for "client.py" should have the same format as specified in question. That is –
   <file name along with path to download>
   <new_line> <proxy1-ip-address/hostname> <space> <port>
   <new_line> <proxy2-ip-address/hostname> <space> <port>
   <new_line> <proxy3-ip-address/hostname> <space> <port>
   <new_line> <server-ip-address/hostname> <space> <port>
3. "server.py" will take port number as a command line argument
4. First run "server.py" in every proxy and the server by using the following command
   ```
   $ python server.py <port>
   ```
5. Run client.py in the client by using the following command –
   ```
   $ python client.py input.txt
   ```
   Where "input.txt" has the exactly same format as mentioned in point-2
6. Python version used was "Python 3.7.6"

NOTE –

My implementation for TCP connection is not resilient to disconnections. Thread will terminate when time limit exceeds and is been received on the connection. TCP connection will close after some time and hence if any data is then sent or received, it will raise an exception in that particular thread running that particular TCP connection while keeping unaffected the remaining TCP connections and their thread which were concurrently running. Hence if a client request a file to download and if the network connectivity is poor then TCP connection will end before entire file was downloaded and raises an exception. This will result in no download or partial/incomplete download of the targeted file and the client once again need to send download request. This was generally observed for large file with size greater than 5MB.

Detailed implementation of client.py and server.py is explained in their respective files using comments.