

Programming Project for TCP Socket Programming

with Python Language

CYBR 2159

Objectives: Understand how socket works. Explore socket programming. Have a deep understanding about TCP and UDP protocol in terms of transferring file precisely and efficiently. Learn what a stream is and the differences between input and output stream. Understand how to attach a stream to a device/socket. Exercise how to design and implement a protocol which controls the communication between server and client side.

Description: In this project, you are required to make a program to transfer the same file from client side to server side for one hundred times with TCP/UDP protocol. Every time as long as the file is transferred, compare with the stored standard file to see they are the same. If the received file is the same as the stored file, it means the file is transferred correctly, otherwise incorrectly. Count the transferring error rate (number of times incorrect file received at server side out of one hundred times) for TCP/UDP to see its (UDP or TCP) reliability. At the meantime, count the time in mili-second cost to transfer the same file for hundred times with TCP/UDP.

Detailed Requirements:

1. You need to make two programs including one client side (C), one server side (S).
2. You can finish it by making TCP code.
3. Use non-persistent TCP connection.
4. Please use Python language. Using any other languages will result in zero of your project.
5. Your client and server side program may run in the same hosts, but must use the Internet IP address, other than 127.0.0.0, or “local host”.
6. You client side program sends the same file (see the sample text format file) to the server side for one hundred times.
7. You server side program receives the file. Every time as long as the file is completely received, please store in at the server side in different name and compare it with the standard file (stored at server side in advance) to see if they are the same. If the received file is different from the standard file, count one time incorrect transfer.

8. Use TCP socket to finish file transferring for one hundred times, count the number of time that incurs incorrect transfer, and count the average time used for one hundred times transferring at server side. The time used to deliver the file includes connection setup time, data transferring time, and connection close up time.
9. When running your server side code, it must display at least the following information:
 - a. I am ready for any client side request
 - b. I am starting receiving file “xyz” for the nth time
 - c. I am finishing receiving file “xyz” for the nth time
 - d. The time used in millisecond to receive “xyz” for nth time is:
 - e. The average time to receive file “xyz” in millisecond is:
 - f. I am done
10. When running your client side code, it must display at least the following information:
 - a. I am connecting to server side: server IP address
 - b. I am sending file “xyz” for the nth time
 - c. I am finishing sending file “xyz” for the nth time
 - d. The time used in millisecond to send “xyz” for nth time is:
 - e. The average time to send file “xyz” in millisecond is:
 - f. I am done

Submission through D2L Dropbox (zip the following into one file for submission):

1. Two screenshots to show the running results with one for client side, and another one for server side.
2. Client side Python source code.
3. Server side Python source code.