Assignment -2 (50 points)

Due on September 18, 2021 23:59

General Instructions:

This is the first of the programming and hands on networking assignment. Make sure to save the work and store the configuration, setup installations, source code that you do in this assignment. The subsequent set of assignments will build on the work you do in this assignment.

You are free to choose the programming language of your choice (C/C++, Python, Java, Go, or any other). I would recommend 'C' for rich programming experience and understanding the low-level system APIs.

Prerequisite: Tools and Data Preparation (0 points)

1. Wireshark/Tcpdump tools are necessary to analyze the traffic.
2. Choose around 5 of your favorite novels (text format) from Project Guttenberg or any reliable source.
    a. Example: War and Peace: https://www.gutenberg.org/files/2600/2600-0.txt. You are going to use these files for building the Socket programs.
    b. Prepare a test file (eg. Test.txt) of size=10KB. Refer "**dd**" or "truncate" command (man dd or man truncate) for more details. Try to populate contents from one of the downloaded text files. < Necessary to verify the content.>

Client-Server Programming (TCP and UDP Sockets) (25 points)

3. Implement as simple Server (Mini Library) and Client (Reader) Programs in the language of your choice. **(10 points)**
   (a) TCP Server: The server program implemented as say <tcp_server.c> would listen on a port **12345**. Upon request from a client, it will try to index to the corresponding file, read it from the system and transmit the contents of requested file to the client. Upon completion of transfer, it would close the connection. For simplicity, you can assume the files are locally accessible within the same directory as that of server program.
   (b) TCP Client: Client program would accept the "book name" from user as input (can be hardcoded in CLI parameter) to be downloaded and request the server to download the book. The client will then save the book on the local filesystem with name "<Name_of_Book>+<Protocol=TCP>+<ClientProcessPID>.txt".

   (c) UDP Server: Also, implement the server with UDP protocol, say udp_server.c for same port 12345.
   (d) UDP Client: Also, implement the same with UDP protocol, say udp_client.c. Here the Protocol would instead be UDP.

Now, evaluate the following aspects: **(15 points)**

1. Compute the time taken by TCP client/server to download the file and the time taken by UDP Client/server to download the same file. List as a table, for: **(5 points)**
   a. The largest of the 5 files you chose.
   b. For 10KB file.

   Note: you need to compute the time on the client side beginning from the first operation (connect in case of TCP) till the end of transfer (close) operations.

2. What is the throughput achieved for TCP and UDP modes? **(5 points)**

3. Diff/compare the files with the original files on the server. Do you observe any differences? **(2.5 points)**

4. Use "wc" tool to find high level differences in terms of line/word and character count and Use "diff" tool to list the differences if any. **(2.5 points)**

Experimentation with socket configurations (25 points)

For the subsequent experiments, fix the server buffer size to 1024 Bytes and choose any one file (10KB) and evaluate the following:

A. Now set the Client buffer size (TCP and UDP) to 2048 bytes. How many packets do you see are exchanged between client and server on Wireshark? Is there any impact on the downloaded file size, diff output for the TPC and UDP clients? In each case, please present the results as a table provide your analysis. **(5 points)**

B. Now set the Client buffer size (TCP and UDP) to 512 bytes. How many packets do you see are exchanged between client and server on Wireshark? Is there any impact on the downloaded file size, diff output for the TPC and UDP clients? In each case, please present the results as a table provide your analysis. **(5 points)**

C. Now set the Server size (TCP and UDP) buffer size to 512 bytes and Client buffer size (TCP and UDP) to 2048 bytes. How many packets do you see are exchanged between client and server on Wireshark? Is there any impact on the downloaded file size, diff output for the TPC and UDP clients? In each case, please present the results as a table provide your analysis. **(5 points)**

D. Reset all the buffer size to 1024 and modify the client programs (TCP and UDP) such that every time client receives data, it sleeps for about 10 milli seconds. Again, analyze the result for same parameters**. (5 points)**

E. Further, check if you can deploy both the TCP and UDP servers and run the TCP and UDP clients at the same time (concurrently). Possible? Not possible? IF yes why? and if not, why not? **(5 points)**

In the next iteration, you would be introduced with Mininet (a network emulator).