

Assignment -3

(100 points)

Due on October 31, 2020 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, installations, test scripts, source code etc. that you do in this assignment. The subsequent set of assignments will build on the work you do in this assignment. You need to turn in the following:

1. Observation Report (PDF or Document detailing your program output and analysis as necessary. Include any instructions or tools that you used to conduct the experiments. Cite all the referenced resources appropriately.
2. Zip folder containing the source code, support scripts that you have implemented and also the instructions to run the program.

You are free to choose the programming language of your choice. I would recommend 'C' for rich programming experience and understanding the low-level system APIs, but even languages like Python, Java, or Go would be fine too.

Prerequisite: Tool setup (20 points)

1. Install Mininet on your computer/laptop/MacBook (host node): Follow the Link < <http://mininet.org/download/>>. Prefer Option1 mode to setup Mininet. Download Mininet VM and create new VM in VirtualBox say "Mininet-VM1" in VirtualBox. Add the NAT Network Interface. Boot up the VM and check the ifconfig and connectivity to outside network either by trying to ping to your local host IP address or trying to ping some website.
 - a. Capture the output of 'ifconfig' command and try to identify the Interface address, IP Address, Subnet Address details. What specific observations can you make here?
 - b. Run Wireshark or TCPdump tool on your local host node (Windows/MAC/Linux desktop or laptop that you are using) and capture the traffic exchanged with the Ubuntu VM. What kind of packets do you see?
 - c. Install the Networking tools "iperf3" – very simple, lightweight, cross-platform network performance benchmark tool on your host node.

Note: If you are running Mininet on native Linux machine, then you can skip reporting (a) and (b).
2. Choose around 5 of your favorite novels (text format) from Project Gutenberg. Prefer large files >>1MB.
 - 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.

Client-Server Programming (TCP and UDP Sockets) (80 points)

3. Implement as simple Server (Mini-Library) and Client (Reader) Programs in the language of your choice. **(20 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 requested 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.
- (b) UDP Server: Also, implement the same server with UDP protocol that listens on same port 12345, say udp_server.c
- (c) TCP Client: Client program would accept the book name to be downloaded either as argument or an interactive input from the user. 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.
- (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 for different buffer size options (1Byte, 32Bytes, 1KB, 32KB, 64KB respectively) for the largest file. (Record the following defined A, B, and C metrics). **(20 points)**

- A. Time taken by TCP client/server to download the file vs Time taken by UDP Client/server to download the same 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.
- B. What is the throughput for TCP and UDP modes for each of the cases?
- C. Diff/compare the files with the original files on the server. Do you observe any differences?
 - a. Use "wc" tool to find high level differences in terms of line/word and character count
 - b. Use "diff" tool to list the differences if any.

Template Reporting for the following combinations for the largest file and choose the optimal size for Read/write that helps achieve best throughput and report A (time), B (Throughput), and C (difference observed or not).

Read/Write Buffer Size	TCP Results	UDP Results
1 Byte		
32 Bytes		
1 KB		
32 KB		
64KB		

For the subsequent experiments, fix the client/server buffer size to 32 Bytes and

choose any one file (possibly largest file) and evaluate the following: **(20 points)**

- D. Evaluate and Analyze the impact of Enabling and Disabling Nagle's Algorithm i) only on Server ii) Only on Client iii) Both on Client and Server.
 - E. Evaluate and Analyze the impact of Enabling and Disabling Delayed ACK i) only on Server ii) Only on Client iii) Both on Client and Server.
 - F. Study for the combination (Enabling or Disabling Both the options) on i) only on Server ii) Only on Client iii) Both on Client and Server end.
 - G. Now change the client/server buffer size to the one that provided the best throughput earlier and Reiterate D, E, F. In each case, please present the results as either a table or appropriate plot and provide your analysis.
4. Modify the Server program such that every time it transmits data, it sleeps for about 100 micro seconds. **(20 points)**
- a. TCP: Run two copies of TCP clients (that download two different files), but single TCP server. Observe and list the client process TCP ports, server process tcp ports.
 - b. Repeat the same with UDP clients.
- Present your analysis on the port allocation on the client/server side in each case. Overall impact on throughput and download completion time in each case. Describe if you find any interesting observations in this experiment.
- c. Launch only the Client programs without starting the server programs. What do you observe? Any significant differences for TCP and UDP? And What happens when the Server programs are launched later (say after 30 seconds of launching the client program? Illustrate the differences observed if any.
 - d. Further, check if you can deploy both the TCP and UDP servers and run the TCP and UDP clients at the same time. Possible? Not possible? IF yes why? and if not, why not?

In the next iteration, you would need to run the client and server programs with Mininet.