

CIS 4930 Internet Networking Technologies, Spring 2023

Programming Assignment #2

Due: Please submit your source codes and report (**in PDF format**) via CANVAS by 10:40am (**firm deadline**) on Monday March 27th. We highly recommend you to finish the submission 30 minutes before the firm deadline.

Introduction:

In this programming assignment, you will measure the timing performance and analysis of a simple client/server implementation for socket communication. It is intended for the students to measure socket performance with different factors. The implementation will have a server program and a client program running on two machines. They communicate with each other using socket communication on the TCP and UDP protocols.

Requirements:

Step 1: Using your PA#1 (TCP socket implementation) as the foundation, convert your source codes into a UDP-socket version. Make sure the UDP works as well as TCP version.

Step 2: Collect (or CREATE) 10 joke memes from Internet and save them on server machine, and revise your source codes to download these 10 joke memes on the client machine in both TCP and UDP versions.

Step 3: Implement a random-number generator for values between 1 and 10 to retrieve these 10 jokes memes in random ordering. This step is critical to ensure server's file system cache does perform in average, thus the average access time at server can be measured.

Step 4: Place the measurement probes (e.g., "gettimeofday" command or similar ones) in proper places to report the following time-components: (1) The total round-trip time (in msec) to retrieve each joke meme remotely in the client process; (2) The TCP setup time including IP address resolution in the client process; and (3) The joke meme access time locally in the server process.

Step 5: Once the measurement is stable, AUTOMATE the measurement 10 times randomly in one run (because user input slows down the execution). After 10 measurements, both server process and client process shall calculate the statistics as (*min, mean, max, stddev*), i.e., similar to PING statistics. You must visually confirm all 10 memes are viewable on the client machines. If there are errors on any meme, take a note.

Experiments

Summarize your statistics/error findings in your report on all three experiments as the following:

Experiment#1 (Baseline Performance): The statistics shall be measured via the “localhost” on the same machine on both TCP and UDP versions. Note there is no connection setup for UDP, the second time-component is thus replaced with DNS-lookup-only for UDP.

Experiment#2 (LAN Performance): Extend the TCP and UDP measurements for actual communication via 100-Mbps Ethernet LAN between two CISE machines in Lab 114. Compare and analyze the statistics between this experiment and Experiment#1.

Experiment#3 (MAN Performance): Extend the TCP and UDP measurements by placing the server process at a CISE machine and the client process at your apartment. Compare and analyze the statistics between this experiment and Experiment#2.

Some useful Linux/Unix commands:

To check your running processes: `ps -u <your-username>`

To kill a process: `kill -9 pid`

To kill all Java processes: `killall java`

To check processes on remote hosts: `ssh <host-name> ps -u <your-username>`

To clean Java: `ssh <host-name> killall java`

Port number

You have to be careful when using a port number, i.e., some port numbers are already reserved for special purposes, e.g., 20:FTP, 23:Telnet, 80:HTTP, etc. Our suggestion is to use the last 4-digit of your UFID as the port number to avoid potential conflicts with other students. Keep in mind that the port number must be less than 2^{16} (=65,536).

Getting the most points

1. The source codes need to be tested on CISE Linux/Unix machines, because we are going to grade your source codes on Linux systems. Please state your testing machines and results in your report.
2. Please try to complete the source codes step by step. After finishing Step 5, your programs will be able to report the statistics accordingly.
3. Try to finish the simplest operations first (e.g. Experiment#1).
4. We will use the “eog” image viewer on Linux systems to view/confirm your joke memes.

5. Have backups whenever your program can do more. You can avoid the situation where your program used to work but not anymore (e.g. after trying to implement exception handling).

Reminder

1. For your programming conveniences, your server program **does not** need to be implemented with thread functionalities (multi-threading) in this assignment. In other words, your server program doesn't need to handle concurrent requests by several clients, i.e., we will test your server program by one client ONLY.

2. All assignments must be done individually. Using blocks of code from other people or any other resources is strictly prohibited and is regarded as a violating of UF Honor Code! Please refer to the UF honor code if you are unfamiliar with it (<http://itl.chem.ufl.edu/honor.html>).

Report

Submitted report file should be named “**report.pdf**” and include the following:

- Your personal information: Full name, UF ID, and Email
- How to compile and run your code under Linux environment.
- Description of your code structure.
- Show some execution results.
- Statistics/Error report on **ALL THREE** experiments
- Lesson learned
- Any additional comments.

Submission Guidelines

1. The name of a source codes for the TCP server program should be named “tcp_server.java” and the TCP client program “tcp_client.java”. UDP version shall be “udp_server.java” and “udp_client.java”. Measurement probes/codes must be clearly marked in source codes.
2. Only submit your Java source code files and report, do not include .obj/.class or executable files in your submission.
3. Include 10 joke memes.
4. Include the report (**with results and discussions**) in PDF format.
5. Zip all your files into a packet: Firstname_Lastname_ID.zip

Grading Criteria (Note the weight of report is increased to 25%):

Correct Implementation / Outputs	50%
Readability / Comments / Code structure	15%
Graceful Termination / Exception handling	10%
Report	25%
Total:	100%