



Objective of this assignment:

- To get you familiar with developing and implementing TCP or UDP sockets.

What you need to do:

1. Implement a simple TCP Client-Server application
2. Implement a simple UDP Client-Server application
3. Collect and analyze round trip time measurements for each client.

Objective:

The objective is to implement a simple client-server application using a safe method: start from a simple **working** code for the client and the server. You must slowly and carefully *bend* (modify) little by little the client and server alternatively until you achieve your ultimate goal. You must bend and expand each piece alternatively like the way a blacksmith forges iron. From time to time save your working client and server such that you can roll-back to the latest working code in case of problems. If you do not use this "baby steps" strategy, you may end up with a ball of wax not working hard to debug.

For this programming assignment, you are advised to start from the simple echo client and server to implement a very simple application.

Part I: TCP "Capitalize" Client-Server

Implement the following Client-Server application that will use two programs: a client program [myFirstTCPClient.java](#) and [myFirstTCPServer.java](#)

a) **Client: myFirstTCPClient.java**

This program must take two arguments: a hostname H and a port number P. The hostname h is a name or a decimal dotted-quad IP address of the server Sv. The port number P is any valid port number where the server Sv is binds to. On Tux machines, a valid UDP or TCP port number is in the range 10010-10200

This client program must:

1) Create a TCP client socket connected with the server Sv running on the machine with hostname (or IP address) h bound to Port number P.

2) Repeatedly perform the following actions:

- i) Prompt the user to enter a sentence S
- ii) Send the sentence S to the server Sv
- iii) Receive the response from the server
- iv) Measure the duration between the time when the sentence S was sent and the time a response was received.
- v) Display the following information: the message received and the time expressed in milliseconds.
- vi) Collect the round trip time.

To implement the client [myFirstTCPClient.java](#), you should consider starting with the program [TCPEchoClient.java](#) (provided on Canvas with this programming assignment). Do not forget to change the name of the class inside the program [TCPEchoClient.java](#) to match your program name.

b) **Server: myFirstTCPServer.java**

This server program must take one argument: a port number P. The port number P is any valid port number.

This server program must:

- 1) Create a TCP server socket
- 2) Wait for a client to connect, receive a message, display it with the IP address and port # of the client, capitalize the message, display the message, and echo back the "**capitalized**" message.

To implement the server [myFirstTCPServer.java](#), you should consider starting with the program [TCPEchoServer.java](#) (provided on Canvas with this programming assignment). Do not forget to change the name of the class inside the program [TCPEchoServer.java](#).

Part II: UDP "Capitalize" Client-Server

Repeat Part I using UDP sockets. Call the client and server programs `myFirstUDPClient.java` and `myFirstUDPServer.java`, respectively.

To implement the server (respectively, client) `myFirstUDPServer.java` (respectively, `myFirstUDPClient.java`), you should consider starting with the program [`UDPEchoServer.java`](#) (respectively, [`UDPEchoClientTimeout.java`](#)) (provided on Canvas with this programming assignment). Do not forget to change the name of the class inside the program.

Data collection and analysis

For **each** client (UDP and TCP), report separately the min, average, and max round trip time. Include screenshots of your TCP and UDP client and server executing on the Tux machines. Screenshots on machines other than the Tux machines will not receive any credit.

Report

- Write a report that will report your results. The report should not exceed 2 pages.
- Your report must contain the following information:
 - whether the programs work or not (this must be just ONE sentence)
 - the directions to compile and execute your program
 - the information this assignment asks you to report (minimum, average, and maximum round trip times)
 - required screenshots of the execution of TCP and UDP clients and servers.

What you need to turn in:

- Electronic copy of your source programs (standalone)
- Electronic copy of the report (including your answers) (standalone). Submit the file as a Microsoft Word or PDF file.

Grading

1) TCP client is worth 20% if it works well: communicates with YOUR server. Furthermore, screenshots of your client and server running on Tux machines must be provided. The absence of screenshots or Screenshots on machines other than the Tux machines will incur a 7.5% penalty.

2) TCP client is worth 5% if it works well with a working server from any of your classmates.

All other server and clients (TCP server, UDP client, and UDP server) will be graded the same as the TCP client (20% + 5%).