

# Report on UDP and TCP Server Implementation and Experiments

## Personal Information

Full Name: Eric deQuevedo

UF ID: 5951-6698

Email: dequevedo.eric@ufl.edu

## Compiling and Running the Code under a Linux Environment

Ensure you have JDK installed. If not, install it using the following command:

```
sudo apt-get install openjdk-11-jdk
```

## Compiling the Code

move into folder and compile the code using the following commands:

- `javac udp_server.java`
- `javac udp_client.java`
- `javac udp_client_test.java`
- `javac tcp_server.java`
- `javac tcp_client.java`
- `javac tcp_client_test.java`

## Running the Code

To run the code, open separate terminal windows for the server and client. outside the file after compiling the code. The server must be run first, followed by the client or test

For the UDP server:

- `java Eric_deQuevedo_59516698.udp_server 128.227.205.239`

For the UDP client:

- `java Eric_deQuevedo_59516698.udp_client 128.227.205.239`

For the UDP client test:

- `java Eric_deQuevedo_59516698.udp_client_test 128.227.205.239`

**For the TCP server:**

- `java Eric_deQuevedo_59516698.tcp_server 128.227.205.239`

**For the TCP client:**

- `java Eric_deQuevedo_59516698.tcp_client 128.227.205.239`

**For the TCP client test:**

- `java Eric_deQuevedo_59516698.tcp_client_test 128.227.205.239`

**Note:** Before running each experiment, make sure to kill any previous processes by running `killall java` in the terminal. This will ensure that the ports are not in use and the experiments can be conducted properly.

## Description of Code Structure

The code is divided into four parts: UDP server, UDP client test, TCP server, and TCP client test. The servers listen for incoming requests and respond with jokes, while the client test code measures the round-trip time for retrieving a joke from the server using either the UDP or TCP protocol.

## Statistics/Error Report on All Three Experiments

### Experiment#1 (Baseline Performance)

The first experiment measures the statistics of the round-trip time for communication between the server and client on the same machine using both UDP and TCP protocols. The results are as follows:

#### UDP:

Local Joke Meme Access Time (in ms):

- Min: 1.0 ms
- Mean: 6.6 ms
- Max: 57.0 ms
- StdDev: 16.800000000000004 ms

#### TCP:

Local Joke Meme Access Time (in ms):

- Min: 1.0 ms
- Mean: 8.5 ms
- Max: 69.0 ms
- StdDev: 20.2 ms

## Experiment#2 (LAN Performance)

The second experiment measures the statistics of the round-trip time for communication between the server and client via 100-Mbps Ethernet LAN between two CISE machines in Lab 114.

### UDP:

Local Joke Meme Access Time (in ms):

- Min: 1.0
- Mean: 5.0
- Max: 31.0
- StdDev: 8.706319543871567

### TCP:

Local Joke Meme Access Time (in ms):

- Min: 2.0 ms
- Mean: 16.4 ms
- Max: 193.0 ms
- StdDev: 41.6 ms

## Experiment#3 (MAN Performance)

The third experiment measures the statistics of the round-trip time for communication between the server process at a CISE machine and the client process at an apartment.

### UDP:

Local Joke Meme Access Time (in ms):

- Min: 2.0
- Mean: 23.4
- Max: 58.0
- StdDev: 16.16910634512619

### TCP:

Local Joke Meme Access Time (in ms):

- Min: 3.0 ms
- Mean: 16.7 ms
- Max: 134.0 ms
- StdDev: 39.10255746111755 ms
- Median: 4.0 ms

## **Lessons Learned**

Implementing both TCP and UDP servers and clients provided valuable insights into the differences between these two communication protocols. The experiments highlighted the trade-offs between reliability and speed in TCP and UDP and how network conditions can impact their performance. Furthermore, working with the CISE Linux/Unix machines and utilizing SSH to access the VMs helped build a greater understanding of the available tools and the underlying architecture of computing networks. This has greatly improved our understanding of how these systems work and how to effectively utilize them. with the technologies of ssh

## **Additional Comments**

The code provided is a basic implementation of UDP and TCP servers and clients. In real-world applications, more complex error handling, security measures, and optimizations may be required. The experiments were conducted on CISE Linux/Unix machines and the results were recorded and analyzed. The hands-on experience with these systems, along with the lessons learned, will be valuable in future projects and endeavors in the field of computing networks. As I greatly learned about the capabilities and limitations of these protocols, I am excited to continue learning more about the field of computing networks and the various applications of these protocols.