COMPENG 4DN4 Lab 3 Report

Aaron Pinto	Raeed Hassan
pintoa9	hassam41
400190637	400188200

March 26, 2023

Server Class

The server class code is run from the terminal and will have a local file sharing directory. It will have two threads, one thread will be responsible for responding to `SERVICE DISCOVERY` broadcasts from clients, and the other thread will handle connections from clients to enable file sharing operations.

The server will have a thread listening for `SERVICE DISCOVERY` broadcasts. When a client requests a `scan` which the server intercepts, the server will respond to the broadcast with the name of the file sharing service.

The server will have a thread that will handle creating IPv4 TCP sockets for client connections. The connection handler will attempt to receive bytes over the TCP connection. If no bytes are received from the connection, then we close the connection. Otherwise, we receive a command input from the client that we can use to perform file sharing operations.

When a TDP connection is established with a client, the server will receive one byte from the client that will contain the command issued by the client. Depending on whether a `list`, `get`, or `put` command is received, the server will handle the operation requested by the client differently.

When a `list` command is received, the server will send eight bytes to the client which encode the response size, and then will send the entire listing response which contains the filenames inside the local file sharing directory.

When a `get` command is received, the server will receive one byte encoding the size of the filename requested, followed by the filename. The server will search for the file, and if found it will send eight bytes to the client which encode the file size, followed by the contents of the file.

When a `put` command is received, the server will receive one byte from the client that encodes the filename size, followed by the filename, followed by eight bytes that encode the file size, followed by the file sent to the server. The file will then be written to the local file sharing directory if the entire file was successfully transferred.

The server will close the connection on the socket when an error occurs, when no response is received from the client, or when the client has closed the connection on their end (`bye` command is issued by the client).

Client Class

The client class code is run from the terminal and will have a local file sharing directory, and be able to scan for and connect to file sharing services to transfer files.

The client will allow the user to scan for `SERVICE DISCOVERY` broadcasts. When the user requests a `scan`, the client will create a UDP socket, and broadcast a service discovery packet. The client will listen for service responses, and will return a list of the available file sharing

services. The client will then be able to `connect` to a file sharing service by establishing a TDP connection.

The client will have file sharing commands available. It will be able to output the contents of the local file sharing directory with the `llist` command. When the client has established a connection with a file sharing server, it will also have access to additional commands, `rlist`, `put`, and `get`. The three file sharing commands will send a packet with the format of the packet determined by the command.

When a `rlist` command is issued, the client will send a single byte integer designated for the `list` command, and receive eight bytes from the server which encode the response size, and then will receive the number of bytes required for the listing response.

When a `get` command is issued, the client will send a single byte integer designated for the `get` command, followed by one bytes that encodes the filename size, followed by the filename. The client will receive eight bytes from the server which encode the file size, and then will receive the number of bytes required for the file. The file will then be written to the local file sharing directory if the entire file was successfully transferred.

When a `put` command is issued, the client will send a single byte integer designated for the `put` command, followed by one byte that encodes the filename size, followed by the filename, followed by eight bytes that encode the file size, followed by the file being sent to the server.

The client will close the connection on the socket when an error occurs, or when no response is received from the server. The client will also close the connection when the `bye` command is issued.

Member Contributions

Aaron contributed to the server class, Raeed contributed to the client class.