Project 5: A File Sharing Program

Department of Mathematical Sciences
Computer Science Division
University of Stellenbosch
7600 Stellenbosch

July 2020

1 Introduction

For this project you will be required to implement a P2P file sharing program similar to programs like `DC` and `DC++`. Your program will not have all the functionality that is provided by the above-mentioned programs, but the principles used will give you insights into practical computer networks, and the problems faced when implementing P2P software.

2 Project Specifications

You will need to implement both a client and a server for this project.

2.1 The Server

As with the chat program, the server is a minimal program that orchestrates the actions and interactions of the clients. It does not need a GUI, but system messages such as connection requests, disconnection requests, search request etc. should be displayed. The following aspects must be implemented:

- Clients must remain anonymous and only be known by a nickname.
- Client nicknames must be unique.
- The server does **not** store the list of files shared.
- The server handles all requests from clients.
- The clients only talk to each other after the server has “introduced” them.
- Duplicate nicknames must be handled when clients connect to the server.

2.2 The Client

The client provides an interface to the user. You must implement a minimal GUI or a good text based interface. Your client does not have to implement a chatroom type environment, but you may earn a few extra marks for it. If your client does not implement the chatroom function, you also do not need a list of online users, as all searches must be sent to the server, and the server stores this list. The following must be implemented:

- Multiple clients connect to a single server.
- Clients choose a nickname at connection time.

- Client connections / disconnections must not disrupt the server or other clients.
- Clients must be able to search for files.
- Searches must not only return exact matches, but also close or substring matches.
- Searches may **not** be broadcast. Searches must go via the server.
- A file download / upload progress indicator.
- Pause and resume functionality.

The client does not have to be able to handle concurrent download or upload streams. One download / upload at a time is sufficient. It must however be able to handle one download and one upload at the same time. You can do this by using different ports for download and upload.

2.3 Interaction

When a client requests a file download, the following process should be followed:

1. The client sends a download request to the server with a randomly generated key.
2. The server passes the originating client's address, request and key to the target client.
3. If the target client is accepting download requests, it contacts the originating client with the address and key provided by server.
4. The originating client then accepts and opens the stream if the keys match.

2.4 Security

You may use any form of security, but the amount of marks will be based on the level of security it brings. The most common use would be encrypting the messages, especially the keys, exchanged between the clients and server.

3 Dates

- **Project starts:** 14 October 2020.
- **Project deadline:** 28 October 2020 (13:00).

The report, Makefile and source codes must be submitted as a single tar or zip file whose name has the following form: `GROUP_NUMBER.tar.gz` where `GROUP` is your group name and `NUMBER` is the project number. Your project must be submitted to the GitHub repository `Computer-Science/rw354/2020/Project 5`.

4 Marking Scheme

Client GUI	**4**
Stability	**10**
- *Client*	6
- *Server*	4
Searching	**10**
- *File lists are not stored on server*	2
- *Searches return exact matches*	2
- *Searches return substring / close matches*	3
- *Searches does not disrupt system*	3
File Transfer	**31**
- *Security*	5
- *Single connection stream works*	3
- *Double connection stream works*	5
- *File downloads correctly*	4
- *File uploads correctly*	4
- *Pause / Resume functionality (download)*	5
- *Progress indicator (upload and download)*	5
Housekeeping and style	**5**
Report	**15**
- *Language, spelling and grammar*	3
- *Selection of experiments discussed*	2
- *Description of the experiments*	5
- *Conclusions drawn from the results*	5
TOTAL	**75**

- Stability refers to the general behaviour of your program. It includes things like correct connections / disconnections, clients disconnecting without interrupting the server, searches not interrupting current downloads etc.

- Single connection stream refers to having either an upload or a download stream running. Double connection stream implies that both an upload and download stream is running concurrently.