GV-NAP File Sharing System

CIS457: Project 2 **Fall 2018**
Date Posted: 10/17/2018
Date Assigned: 10/19/2018
Due Date: 11/20/2018 **Point Value**: 100 pts
This is a group work assignment of 3-4 students

Introduction

In the first project you implemented a file transfer protocol that supports multiple client connections. In this project, your file transfer protocol will form the basis of a file sharing system that allows users to access a distributed data storage system based on simple keyword search. The specification of this project relies on the principles of using the centralized directory indexing service to implement a P2P architecture. The system consists of two parts:

- The first part is the host system, which can query the server for files using keywords. The host also has a file transfer client and server. The ftp client allows a user to access files stored at the remote user locations. The ftp server is responsible for providing file transfer services requested by a remote client.
- The second part of the system is the centralized server, which provides a search facility that can be used to perform simple keyword searches. The result of the search is the location of the remote resource.

The project assignment suggests a method of implementation such as described in the following sections. However, you are **free** to implement it in anyway you like provided that you deliver the same required functionality as outlined in the following sections.

Implementation

The project consists of two components such as given in Figure 1.

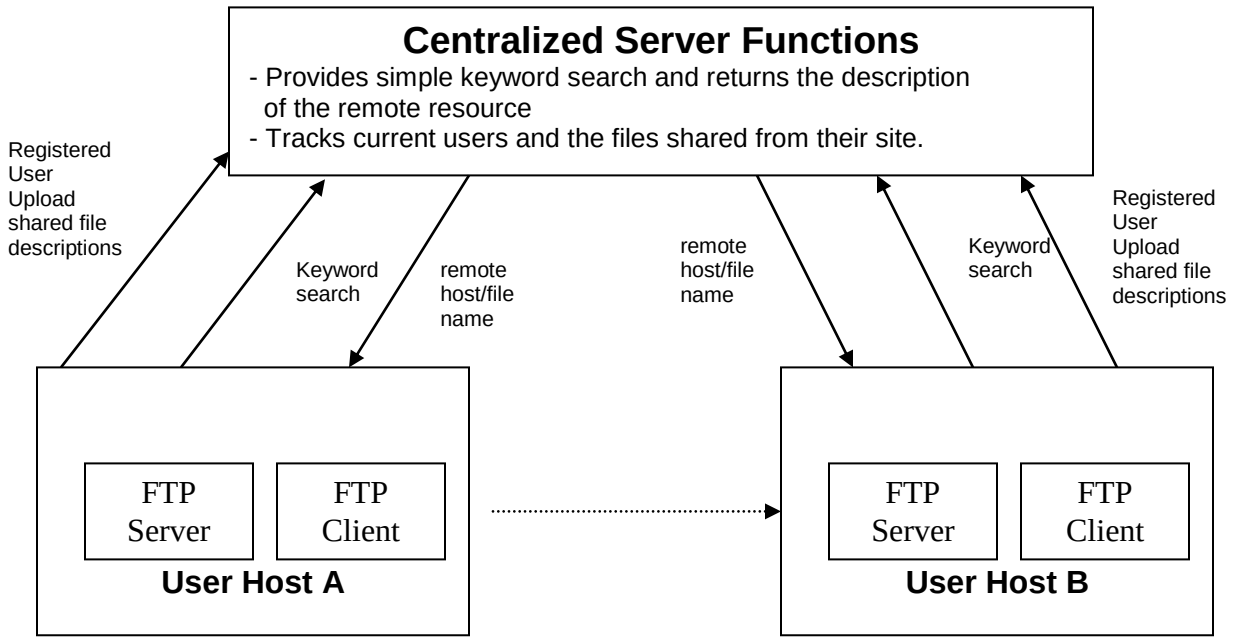

Figure 1: Architecture of the GV-NAP file sharing system.

1. Host Implementation

The host is a GUI application. The user must specify *the hostname and port of the server* in order to connect to it. Once a connection is initiated with the server, the host sends the *three* pieces of information *(username, hostname, and connection speed)*. After that, the host uploads the *file descriptions* (for example a file called *"filelist.xml"* in the current working directory). These descriptions include the file name and the description of each shared file. The shared files are also located in the current working directory. The descriptions for these files are used for the keyword searches. After user and file information are successfully registered, the client is now able to do keyword searches. Once the search is executed, the remote host and file name of all matching files of the keyword are given back by the server.

After getting the results, the client can contact the remote ftp server to retrieve the file. The ftp client can only be used to "get" a file and not "put" a file. When the client wants to disconnect from the NAP server, client can click "Disconnect", which sends out a QUIT command to the server.

2. Centralized Server Implementation

The server tracks current users and the files shared from their site. Once a connection is initiated from a host, it stores *the username, connection speed, and hostname* in a *"users" table*. After this, it acknowledges to the host that it has received this information and then waits for the *shared file descriptions* to be uploaded. Once it receives the *file descriptions*, it'll be stored in a *"files" table*.

It also provides simple keyword search and returns the description of the remote resource. On receipt of a keyword search, the server performs the search in the *"files" table* and returns the *resource location* of the remote files. The resource location includes the *remote hostname, port number, remote file name, and the connection speed* (the host name and connection speed is retrieved from the corresponding entry in the "users" table).

The server allows multiple clients to register and upload their descriptions at the same time. It also allows multiple hosts to query for files at the same time. It tracks the availability of the remote resources too. If a host "unregistered" from the system, the associated file descriptions and user information are deleted from the system. A screen capture of the required server side of the application is given in Figure 2.

The project points' distribution are summarized below:

- 40Pts: user registration
- 30Pts: Keyword search and obtaining a match
- 30Pts: P2P file download
- 10Pts will be deducted if a single thread application is created at the time of the project demo.

40 Points

30 Points

30 Points

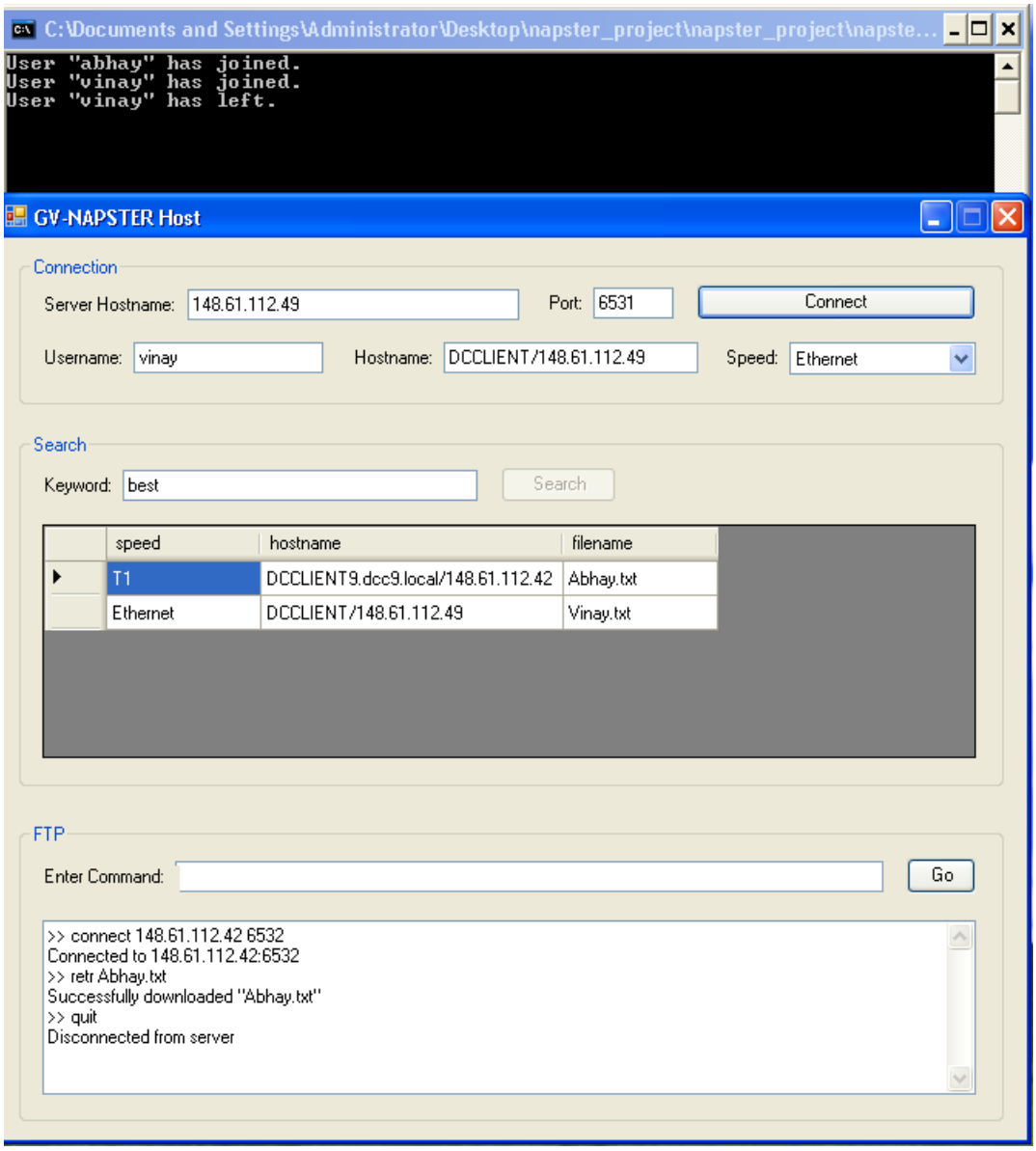

Figure 2: Sample GUI Application

Design Requirements Checklist

1. Your FTP server at the remote sites should support concurrency using multi-threading to allow multiple clients to connect at the same time.
2. Your centralized server should allow multiple users to upload their descriptions at the same time, i.e. you need a concurrent centralized server.
3. When a user registers with the centralized server for the first time, he/she may be prompted for the capacity of the link to the Internet (e.g. modem, Ethernet, T1, T3 etc).
4. The centralized server should be implemented to:
    - Perform keyword searches on the uploaded file descriptions based on a user supplied keyword.
    - Track the availability of remote resources. For instance, if a user "unregistered" from the system, the associated file descriptions should be deleted at the centralized server.
5. Your code can be implemented in any language. You may use any file transfer protocol such as ftp or http if you wish, ..etc.

Submission information

Turn in a copy of all your programs with a written report showing screen capture for a complete session (user registration, user supplying query, and client downloading the requested files) that is provided by the system.

You must indicate the project completion percentage. The **project completion percentage** describes how complete is your project work and meets the requirements.

The report should include the basic logic of the program, how you implemented each program module, what problems you have encountered and how you resolved them.

The programs should be submitted in two formats, a paper version, and a compressed electronic version that contains all of your source codes. Email me the programs as an attachment @ elsaidm@gvsu.edu

Note:

- Your project must be demoed in order for me to grade it. Until this will happen, your project is on a late submission status (even if you emailed your code and report) for up to 7 days (max) with a 3% late submission each day. After the (7) days, the project will receive "NO" credit.