CSCE 327 Assignment 4 – Ver 1.0.0

Due: April 15, 2019

Objective

- The objective of this assignment is to familiarize yourself with a model different from client server paradigm.

Assignment: Peer-to-Peer File Sharing

In this assignment, you will implement a P2P file sharing application according to the specification given below. The application will consist of two components: (a) a server discovery component and (b) a file transfer component. You can choose any programming language.

Protocol Description

Your P2P protocol will use the centralized index approach. You will implement two programs, a centralized ‘p2pregistery’ and a client/server application ‘p2pservent’. Your protocol will implement the four key operation of P2P applications: join, publish, search, and fetch.

Join:

The registry will listen to a TCP socket at a well-known port. A new servent will contact the registry at this port and indicate its presence. In addition, the registry also listens to a UDP socket and every 60 seconds the servent issues a 'HELLO' message to the UDP port of the registry. If the registry has not received a HELLO message from a servent in 200 seconds, the servent is removed from the list of available servents.

Publish:

After registration, the servent sends a list of files available at the servent to the registry (e.g., all files in a directory specified as command-line parameter).

Search:

A servent looking for a file contacts the registry. The registry looks for the file in its database (matching names) and returns the contact address of the servent if an entry was found (the first matching entry).

Fetch:

The searching servent contacts the servent with the file and requests the transfer of the file across a reliable socket and places the file in the same directory as the shared files.

Implementation Details

You can use code from previous assignments. All communication will be across TCP sockets except the HELLO messages which are UDP datagrams. The registry has to listen to a well-known ports for registration of new servents and for file queries. Each servent also listens for requests for file transfers on a TCP socket.

Error Handling

The following error handling is expected: all return values of the system calls have to be checked and the correct number of command-line parameters for the client (if any). If the registry cannot find a requested file, the client prints a simple error message. Similarly, if a servent does not respond to a file request, the client prints a simple error message.

Evaluation

Start your registry and two servents (on at least two different nodes). Each servent registers at least 3 different files (e.g., simple short text files such as your source files). In your evaluation show the structure or content of your database at the registry after the registration of the two servents.