PEER-TO-PEER FILE SHARING SYSTEM

Akash Kumar Singh (16807068)
Yogendra Singh (160829)
Adhip Garg (160042)

GOAL: The project aims to develop an application through which any user running this application on a linux machine can share files with other linux users with the same application installed. Users can get a list of all the files and directories (music files, pictures, etc.) of the connected users that they are willing to share.

IMPLEMENTATION: Due to lack of knowledge about the subject of networking we have taken some help from the internet in order to get an idea on how we can implement it.

For now we have divided the project into coarse phases or milestones that we would like to achieve to have a proper development of the project and to ensure robustness of each component of the system.

1. Discover the other linux systems around.
2. Enable the visibility of the files and directories the user want to share.
3. Get the lists of the files and directories of the other connected users.
4. Able to download files from the other user.
5. Adding extra functionalities and build a GUI around it.
6. Further optimisations.

TIMELINE: For now the timeline is very loose, in a sense that we don’t have any idea of the skill set required to build something like this

1. Week 1: Develop the functionality of discovering the users around.
2. Week 2: Enable the visibility of the files and directories the user want to share.
3. Week 3: Add the functionality to get the lists of the files and directories of the other connected users.
4. Week 4: Enable the feature to actually download the shared files.
5. Week 5: Build GUI around it.
6. Week 6: Further improvement and optimisation.

(2nd Week Report) 24/01/2020

Intro to Socket Programming: This week our aim was to first get some understanding of socket programming and then write the code to achieve this basic functionality. So, we have written a broadcaster which sends a beacon message periodically and a listener which is always listening for this message, in order to connect.

Discover the neighbours: Inorder for the machines to find each other we need to periodically broadcast UDP messages so that anyone in the vicinity can catch those and know our presence. So, for this there should be two programs running on a particular machines at the same time: **Server** and **Client**.

Server: Server must periodically broadcast packets indicating its presence. We use the UDP protocol and include the following information in the packets: Identifier, MAC address, IP address, Port. Using this information the client can then make a TCP connection with the server and start talking with each other.

Client: Passively listen for broadcast beacon messages from nearby nodes. Keep track of which neighbors are currently within range. Note that neighbors will disappear as well as appear. You will want to print out the current list of neighbors so you can see who they are, only printing when there is a change.

We have created a Github repo with the code for broadcast and listener.

Next Step: After we have found a neighbour our aim will be to establish a TCP connection and be able to see the list of shareable files/directories on the connected linux machine.

Response to the comments on report1:

1. For functionality we meant to say that we will be focusing first on using CLI (command line instructions) for testing and using the system but later on we will add a very basic GUI.
2. Second addition to the functionality that we would like to make a distributed file sharing network out of it, like BitTorrent.
3. Optimisation will be done as we go one, on the basis of performance. Upon some searching we found that by flooding and routing using some algorithms can make a performance difference, so we will be looking out for that.