

REPORT 2

PEER-TO-PEER FILE SHARING SYSTEM

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

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: In order 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.