Peer-to-Peer File Sharing

CS4089 Project

End Semester Report

B.K.S.Surya Teja (B120791CS)
Bala Pranaya Sujan Ghali (B120756CS)
M.Sasidhar Reddy (B120624CS)
Guided By: Sreenu Naik Bhukya

February 24, 2016

1 Introduction

We are always interested in finding out how data is transferred across the Internet, how a file gets transferred with a just a click on the button, this interest motivated us to pursue our project in file sharing/transfer area of computer networks. By looking at the file sharing systems around us, some drawbacks were identified in our hostel LAN. One of the problems identified was, for finding the necessary file the user has to check every system in the network for its presence. We thought of implementing an architecture where user mentions the required file name in search field of the web page and gets it by just typing the output command in the terminal. This can be done by monitoring the shared directories of all registered users and transferring data to requested user from any user who has the requested data. This project gives us a fair idea about the way in which data is transferred across networks, the way it is implemented using client and server side scripts and simultaneously improve our hostel LAN system.

2 Problem Statement

To build the required architecture for the sharing of any file across the different systems within a network and this is done by running client side and server side scripts on every system which facilitates data transfer between requested user and source(user possessing the data). A web interface is built on top of this for user interaction where the data requested from the user is searched across all the other systems in the network and displays a command which when run on requested user's terminal, will transfer the desired data from source user to requested user. As of now few assumptions are made and they are listed below:-

1. All the users must use linux systems only.
2. All the users should have python compiler installed for compiling client and server scripts.
3. Server script should be run whenever the user restarts his system.
4. Every user should give public accessible permissions to the directory he is willing to share.
5. There should not be sub-directories inside the shared directory.
6. There should not be duplicate file names in the shared directory.

3 Literature Survey

The fundamentals of peer-to-peer architecture is presented by G. Camarillo

[1]. Alan Davoust [2] presented a formal model for P2P file sharing. L. Fjeldsted [3] has described BitTorrent Protocol (BTP) which is a protocol for collaborative file distribution across the Internet. The online course by Charles Severance[4] helped us to understand the basics of python programming. The socket programming video by DrapsTV [5] helped us to understand socket programming using python.We read about port numbers, binding of a socket, file transfer between the sockets.

4 Work Done in the previous semester

Our design involves the detailed description of each element required for implementing this file sharing architecture and the way in which these elements are integrated together. Few fundamental elements that are present in our project are client side and server side scripts, web interface for user interaction that is searching the required files and displaying the results after checking in the database.

1. Users:- Before we further elaborate our design let us familiarize with the types of users involved in our project.

    - Master Node:- Within the network consisting of many systems, this is the node where the web page will be hosted in order to facilitate a search mechanism for the file desired and displaying of results of the users who are possessing the file.
    - Node:- These are users in the network who actually are active in file sharing. They always have to run the server script which listens all the time for checking whether any other node is willing to connect. If this user wants a file, he first gets the command from the master node(web page) containing the details of the node possessing the file. Upon passing this command as input to client script in his system, the user gets the data he intended for.

2. Web Interface:- This is the front end displayed page hosted on the master node which provide fields for registration like username,password and directory path which is open for sharing. Upon registration a web page will be displayed to the user consisting of search field for searching and two python scripts(server and client).

3. Text file:- After registering, downloading and running the server script a text file will be generated containing important information regarding the details of the user. This is the text file which contains information about the node like IP_address, MAC_address, file names in the directory which is open for sharing. This text will be later sent to master node to update values in the database.

4. Database:- The text file generated from server script will be later uploaded by the user on the web page(master node) and the data in this file is stored in the database. The database consists of two tables Table1 consists of username,password,hostname,IP_address, MAC_address(Primary key),shared directory path and Table2 consists of MAC_address, filenames where Table2.MAC_address refers Table1.MAC_address and Primary key for Table2 is a combination of MAC_address + file_names.

5. Server script:- This is the script which runs all the time at the user

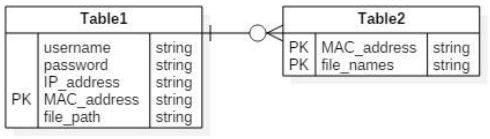

Figure 1: E-R Diagram for the above mentioned tables.

systems which imports socket package and a socket is created. In order to connect with the other users, first the socket binds host name and port number. The host name is obtained using a socket function(socket.gethostname) and the port number which can be anything greater than 1024 and less than 65536 as port numbers less than 1024 are used for other standard protocols, for instance we can take port number as 12345. After connecting,the server script in the source user's system searches for the required file that. On processing this, the required file will be sent to the requested user. Until this process gets completed,no other client script can connect to this server script.

6. Client script:- After searching the required file the user will be provided with the IP address and the destination path of the user containing this file. So, these two variables will be given as input to the client file. The client script import socket package and creates a socket. This socket connects with IP address which was earlier given as input parameter. So,this client script connects to the other user and sends the path of the requested file in its system. Finally, the user system at the other end sends the data and this client script receives it. The client script terminates after the completion of data transfer.

7. Master script:- This is the web code which runs at the master node which acts as a controller between the inputted frontend page values and backend database. Upon request it also pings which the nodes which possess the data and displays the results indicating whether the node possessing the data available for data transfer or not.

These are the elements of our project and are key components in implementation of our project. The step-by-step procedure of the working of our project is stated below:-

1. Every node in the network should register specifying username,password and the directory path open for sharing through the web page hosted in the master node.

2. Once a user is registered he should download the client and server scripts. The user should run the server script indicating that it is ready to accept the connection from other users.

3. The script generates a text file which includes the IP_address of the system, MAC_address and the file names of the shared directory which will be uploaded to the web page(master node) by the user when any changes are made in the directory.

4. Every time a user uploads a text file his corresponding file names are searched in database using his IP_address and MAC_address and replaced with the new file names.

5. Whenever a user requires a file he should login in to the web page hosted and searches for the file required.

6. The master script(web code) takes input from this search field and

checks in the database which are all the nodes possessing this file. A ping message will be sent to the nodes possessing this file to check whether the node is active or not.

7. From the returned value of the ping message, a web page will be displayed listing out the nodes ready which are active and not involved in any other file transfer process.

8. Out of the results displayed,the user has to select one of the active users and after selecting the source user, the requested user will be provided with an IP_address, path of the file in the source user system and the input command for the client script.

9. The requested user runs the displayed command in his linux terminal and the client script runs with given IP_address and file path as the input parameters.

10. A connection will the established between requested user client script and source user server script if the respective port numbers are matched. The file will be transferred from source to requested user and the client script terminates but the server script continues to listens and ready to accept the further connections.

In addition to the above design of our project, the way of establishing a connection between client and server has been implemented, a message has been sent for testing the connection between two systems. The way to get the IP_address, MAC_address and the file names of a directory has been coded.

5 Work Done in the current semester

1. Text File:- A script has been coded which generates a text file containing information about the node like IP_address, MAC_address, file names in the directory which is open for sharing. The script also monitors the changes done in the directory which is open for sharing. his text will be later sent to master node to update values in the database on any changes.

2. Server script:- This is the script which runs all the time at the user systems and has been coded in two versions as follows:-

   - Single Connection:- In this version, only a single connection can be established at an instant.This script firstly imports socket package and a socket will be created.In order to connect with the other users, the socket binds host name and port number. The host name is obtained using a socket function and the port number which can be anything greater than 1024 and less than 65536.The script listens for the other nodes to connect. After connecting,the server script in the source user's system searches for the required file that was requested by the client script of requested user. On processing this, the required file will be sent to the requested user. Until this process gets completed,no other client script can connect to this server script.
   - Concurrent connections:- In this version, more than one connections can be established at an instant.This script firstly imports socket package and a socket will be created.In order to connect with the other users, the socket binds host name and port number. The host name is obtained using a socket function and the port

number which can be anything greater than 1024 and less than 65536.The script listens for the other nodes to connect. Concurrent connections can be facilitated using threads.For this,a new subclass of the Thread class is defined , thread initialisation init method is overridden and the run method to implement what the thread should do when started is overridden. Upon calling start method a new thread is started by calling the run method. After connecting,the server script in the source user's system searches for the required file that was requested by the client script of requested user. On processing this, the required file will be sent to the requested user.

3. Client script: After searching the required file in the web interface, the user will be provided with the IP address and the destination path of the user containing this file. So, these two variables is given as input to this client script. The client script import socket package and creates a socket. This socket connects with IP address which was earlier given as input parameter through the web page along with the destination path of the file. Finally, the user system possessing the file sends the data to this client script on processing the destination path of the file which was sent by client script and writes the received data on the corresponding file. The client script terminates after the completion of data transfer.

4. Web Interface: A part of web interface has been implemented. This is the front end displayed page hosted on the master node and has been coded using HTML,PHP and CSS. The interface provides fields for registration like username, password and directory path which is open for sharing.

6 Future Work and Conclusions

1. Web Interface:- The remaining part of the web interface has to be implemented where upon registration user will be redirected to a web page consisting of search field for searching the required file and two python scripts(server and client) which every user has to download for the transfer of file between the different nodes.

2. Database:- The database has to be created which consists of two tables as earlier mentioned in the E-R diagram where Table1 consists of username, password, hostname, IP_address, MAC_address(Primary key),shared directory path and Table2 consists of MAC_address, filenames where Table2.MAC_address refers Table1.MAC_address and Primary key for Table2 is a combination of MAC_address + file_names.

   This database must be later integrated with the web interface for the retrieval of data from database which is based on the web interface search query.

3. Sub directories:- Extend our project to solve one of our assumption that the case if the shared directory contains sub-directories.

4. Upload of text file:- As per the earlier design, the user has to voluntarily upload the text file upon any changes. So, this extra effort for the user can be minimized by running a script which automatically uploads the text file on any changes occurred for the storing of changed values in the database.

References

[1] G.Camarillo . *P2P file sharing* (2009,November) [Online] Available: https://tools.ietf.org/html/rfc5694.[Accessed: 2015, November 2]

[2] Alan Davoust(M.Eng). *Collaborative Knowledge Construction in a Peer-to-Peer File Sharing Network.* (2009,September) [Online] Available: http://sce.carleton.ca/ adavoust/AlanDavoust-MAScThesis.pdf [Accessed: 2015,November 1]

[3] J. Fonseca,B. Reza and L. Fjeldsted. BitTorrent Protocol – BTP/1.0. (2005,April) [Online] Available: http://jonas.nitro.dk/bittorrent/bittorrent-rfc.html [Accessed: 2015,October 28]

[4] Charles Severance. *Learn to Program and Analyze Data with Python.* [Online] Available: https://www.coursera.org/specializations/python [Accessed: 2015, November 7]

[5] DrapsTV.*Python Advanced Tutorial 6-Networking* (2014,April 29) [Online] Available: https://www.youtube.com/watch?v=XiVVYfgDolU [Accessed: 2015,November 8]