Project Document
[*FileIt !*]

Computer Network 2
(CS3543)

Project Name	FileIt !	
Date	*2019-04-29*	
Author	*Dhiraj Agarwalla - Es16btech11010* *Siddharth Goel - Es16btech11019*	

Contents

1 SW Development Plan

1.1 Project Overview

1.1.1 Objective and Project Scope

- *In this project we have implemented the core ideas of the **Direct Connect Protocol.***
- *We have implemented from scratch using none of the networking libraries so that we get deep understanding of the networking principles behind the action.*

- *Our project enables users in a network to share files with each other in a blazing fast speed*

Major review items
Central hub (server.py) **It maintains the list of all currently active clients in the network.**
User Program (client.py) **It the main program that the user runs which enables him to share and download**
Fragmentation (upload.py) **It fragments the file into chunks and hashes them. Also creates metafile**
Stitching (stitch.py) **It joins all the downloaded fragments into one intact file using the metafile**

1.2 Assumptions, Dependencies and Constraints

Item	Assumptions, Dependencies and Constraints	Remarks
1.	*User knows the name of the file he/she wants to download*	
2.	*All the clients belong in the same local network*	
3.	*A user does not terminate the client program unless the machine is off.*	

1.3 Roles and Responsibilities

Student Name	Roles and Responsibilities
Siddharth Goel - ES16BTECH11019	**Developer** *Layed out the backbone of the program. Most of the socket programming done. Make server.py and most of client.py*
Dhiraj Agarwalla- ES16BTECH11010	**Developer** *Managed the file side of the program. Made upload.py ,stitch.py, and filled file handling parts of client.py.*

Software Development	**Siddharth Goel** *Server.py , client.py (download, download_query_util, recv_meta,share, share_util, main,download_util)*
	Dhiraj Agarwalla *Stitch.py , upload.py , client.py(recv_meta,share_util, download_util)*

1.4 Development Plan

1.4.1 Development Schedule

Estimated Project Period	25/03/2015 - 30/04/2015
Project Team Size	2
Estimated Man Months	2

Milestone	**1st Review**	**Final Review**
Planned Schedule	2-April-2019	During final exam week.

1.4.2 Development Environment

| Item	Development Environment	Remarks
Program Languages	*Python*	*Used modular design pattern*
Compiler, Build	*Python 3*	*The latest version of python*
Target Kernel	*4.15.0-48-generic LINUX*	*The version is expected to be changed according to new chipset.*
Word Processor for Document Creation	*Google Docs*	
Configuration Management	*Git*	*https://github/dj016/File_it*

2 SW Requirements Specification

2.1 Major Functional Requirements

No	Requirement Id	Function Requirement Name	Description
1	1	server.py	It maintains the list of currently active clients. It sends the list of all currently active users to the incoming clients and also add their ip into the list of active clients. It also updates the list using ping to remove inactive clients.
2	2	client.py	Client implements the sharing and the downloading functionality of the program. For downloading it receives the currently active client list from the server and queries them for the required file name. Then it makes connection with all those users who have the files to download chunks parallely. It also listens for queries by its peers for and services their request by sharing the file chunks.
3	3	upload.py	It divides a file into chunks of max size 1 MB. It renames the chunk by the SHA1 hash of the chunk. It also creates metafile which consist of the list of all the hashes in that order. It automatically manages the directory .
4	4	stitch.py	It is used once the chunks of the file have been downloaded. It uses the metafile to stitch all the chunks together into one intact file. It then deletes all the chunks and also manages directory.

3 SW High & Detailed Level Design

3.1 Overall Architecture
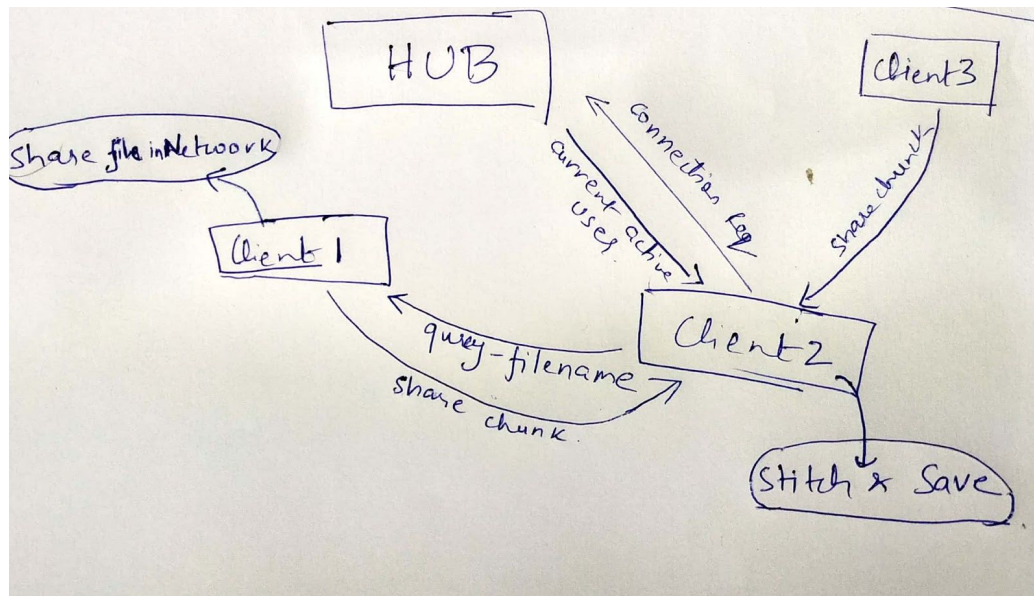
3.2 SW System Operation Design

Server maintains the list of currently active clients. It sends the list of all currently active users to the incoming clients and also add their ip into the list of active clients. It also updates the list using ping to remove inactive clients

Client implements the sharing and the downloading functionality of the program. For downloading it receives the currently active client list from the server and queries them for the required file name. Then it makes connection with all those users who have the files to download chunks parallely.

It also listens for queries by its peers for and services their request by sharing the file chunks.

3.2.1 Module Description

Component	Module	Description
Server	Main	Sends the list of all currently active users to the incoming clients and also add their ip into the list of active clients.
Server	Check_Active	To check if any user has left the network and update the currently active user list
Client	Main	Asks the user which file he wants to share or download to/from the network
Client	Share	Fragments the file by calling myupload function, creates a metafile containing the name of the chunks and store these
Client	Share_util	Answer the query such as (Is present, Send meta file, Send chunk) from the user(who is downloading)
Client	Download	Query to other users to know which of them have the file.and calls download_util function
Client	Recv_meta	Receive meta file from anyone user and store them in a queue.
Client	Download_util	Send request to download the chunks corresponding to hashes stored in the queue
Client	myupload	Divides a file into chunks of max size 1 MB. It renames the chunk by the SHA1 hash of the chunk. It also creates metafile which consist of the list of all the hashes in that order.
Client	Stitch	Used once the chunks of the file have been downloaded. It uses the metafile to stitch all the chunks together into one intact file

3.2.2 Sequence Diagram
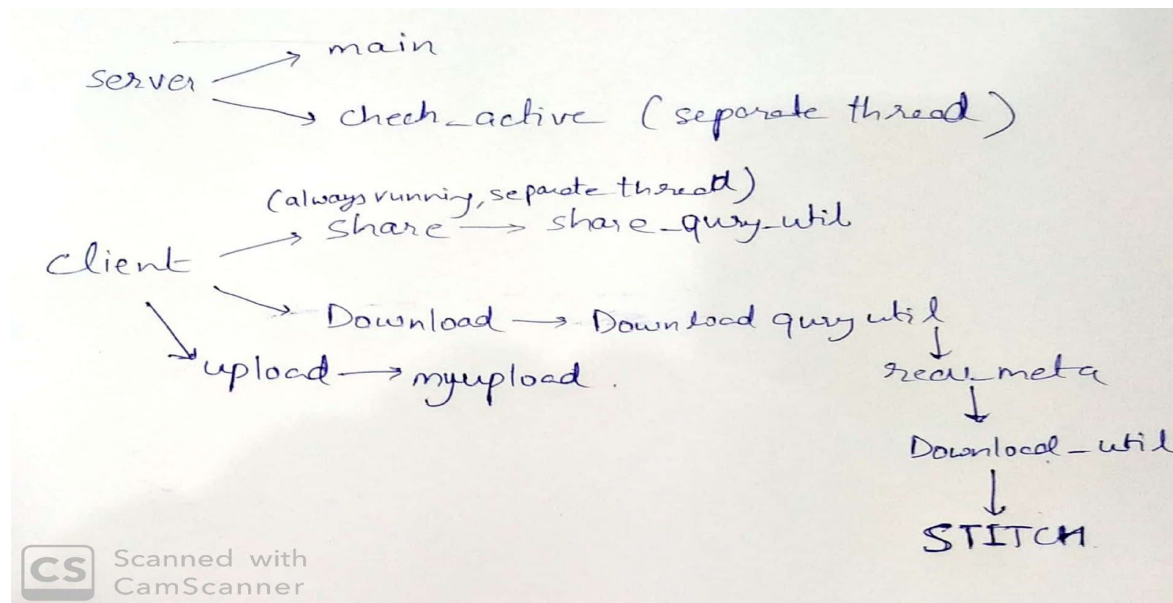
3.3 SW Code Structure

● **Mapping list of modules and files (or folders)**

File name	Module name
Server.py	Server.main()
	Server.check_active()
Client.py	Client.main()
	Client.download()
	Client.download_query_util
	Client.recv_meta()
	Client.download_util()
	Client.share()
	Client.share_util
Upload.py	upload.myUpload()
Stitch.py	stitch.stitck()

4 SW Unit Test Report

4.1 Bugs known at submission date

DATE: *2019.04.29*

S.No	Bug List	Description
1	*Once joined, user remains active until the his system is shutdown/suspended*	*Is included in our assumption*
2	*User and server exit condition*	

5 SW Development Completion Report

5.1 Project Result Analysis

5.1.1 Development Work Promotion Results

Item	Result
Socket Programming- Direct Connect Protocol	*Was able to implement direct connect protocol successfully*
Product Development	*developed a very useful and user friendly file sharing application*
Distributed File Sharing	

5.1.2 Development Results and Utilization

The system can be used in organizations such as colleges and offices where filesharing is very common. It can help a organization to limit its bandwidth usage since all the traffic is local. It also makes downloads very fast and contents easily accessible..

5.1.3 Deliverables List

S.No	Executable Name	Description
1	*: Server.py*	*Run Pyhton3 server.py*
2	*Client.py*	*Run Python3 client.py <server_ip>*
3	*Upload.py*	*Executes inside client.py*
4	*Stitch.py*	*Executed inside client.py*

∎ References

 [1] *https://docs.python.org/2/howto/sockets.html*
[2] *https://en.wikipedia.org/wiki/Direct_Connect_(protocol)*
[3] *www.stackoverflow.com*