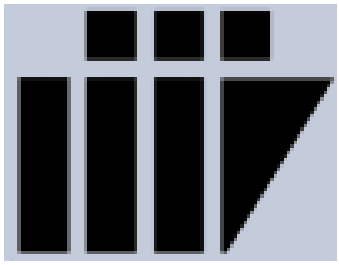
INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, VADODARA



Computer Networks Project

Direct Connect ++

Easy, Quick, Real time

Report By:Gaurav Tolani(201352021)

Pawan Patidar(201352011)

Date:19/11/2015

The Document Contains the motivation behind choosing this project, the procedure and technology we used and learnt while the course of making DC++ , the difficulties we faced and the scopes and future extensions of our project.

CONTENTS

REPORT 1:

Date:23rd August '15

PROBLEM STATEMENT

Communication has been one of the most fascinating area of research for computer scientists in the last couple of decades and a lot of advancement is seen in the same.

We have Planned to make a Client for already build server(apex DC++) that will enable the clients to share files at a lofty rate of about 10-15 mbps. We will also configure the server so that there should be validation through unique ID and Password by each client(IIITV Students).

WHY WE DROPPED THE PROJECT?

Reasons were:

Size of Project-Making a client would not have served the learning purpose of the course because that would have taken not more than a week to be built because server was already made and took just a couple of hours to configure and maintain.

Scope-The primary cause of the this project was to enable the clients share files easily and with confidentiality, but if we take server from a third party the college would not have approved it to use officially.

Licences and Feasibility-For a deemed college like IIIT, Vadodara it is important to be secure and having our own server has many advantages from *operating it to synchronising it with ERP*.

Time Drained:

23rd August '15 - 1st November '15

REPORT 2: Date 1st November '15

PROBLEM STATEMENT

We have planned to create a peer to peer communication software that will be managed by a server and there can be as much as fifty clients in the same network communicate connected to the same server having very high file sharing speed .

Motivation

There is nothing available in college to share and get files from others by wifi and students have to manually go with a storage drive to take data from others, which is time consuming as well as not feasible ever y time.

Also if there is no system like this, everyone downloads the same contents individually which wastes huge data downloaded from college network.

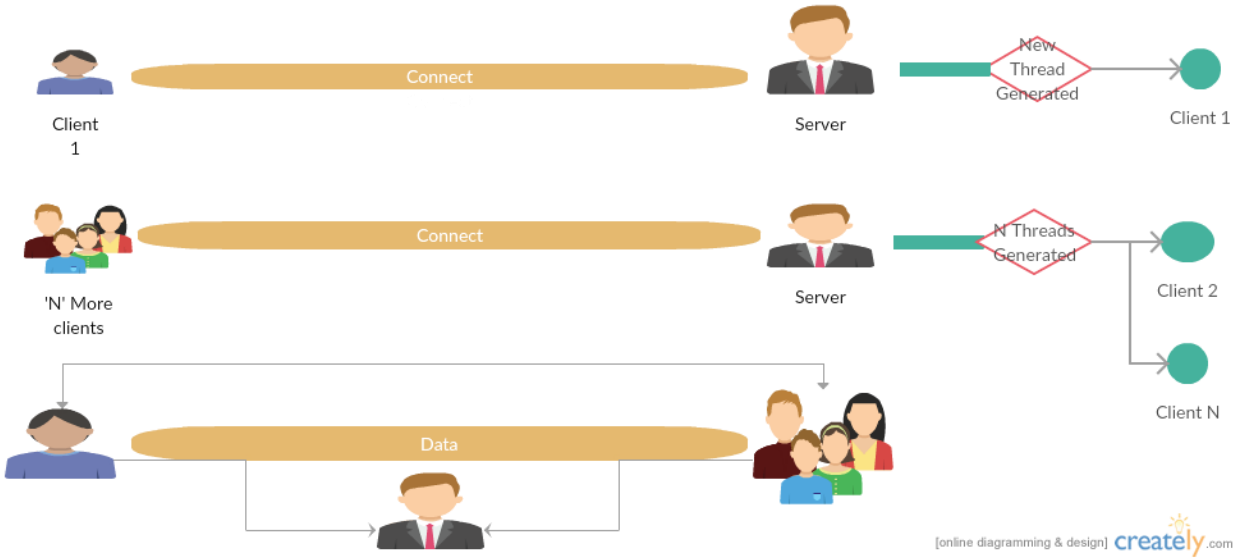
Each time you want something, its not sure that the internet would be working so dc++ can be of great importance at that time.

Solution

We Used Socket Programming in net beans IDE and used multithreading to connect multiple clients to the same server.

P to P: Peer to Peer Connections has the speciality that every client can act as uploader as well as downloader so the role of server is just to establish connection and maintain the connection link between clients.

Activity Diagram:



Database: If we will successfully make a peer to peer channel with no Server storing the database then we won't need any dedicated database for DC++, just a machine in local network to run server and to let the clients connect via it.

But if we store data in server, it will take atleast 1-2 TB of data initially so that transfer of files could be done efficiently(Assuming the admin will delete data after a certain period of time).

Connection: Sockets use IP and Port No to identify the machine to connect, so any client can connect to server having its IP and Port No.

Termination: Whenever the admin wants it can shut down the server and all clients will be disconnected. This gives admin the power to establish as well as terminate the connection on demand.

Tools and Technologies Used

→Socket API

→Java Swing

→ Applet

→Netbeans IDE 8.0.2

→Java JDK 1.8

Difficulties Faced

→Socket Programming was a totally new field for us, so we found it challenging to work in a networks domain as it was totally new for us.

→To store information at a certain directory in each clients machine, it was difficult because we initially we named it as "home/users/intel/document/netbeansprojects" but later we realized that what if the clients doesn't have a folder named netbeans in this specified location, so we implemented the client in such a way that it created a directory if its not there in the clients machine.

→To take the third machine: We were unfortunately only two members in our team instead of three, so every time we wanted to check working of our software in multiple clients, we had to find a machine which was a big problem.

Achievements and Failures:

Achievements:

→We successfully transferred files from one client to others (Broadcast).

→We Achieved about 3 mbps transfer rate of data connected over same network.

→We made the server in such a way that all the transfer of data could be stopped by the server, so if the implementation takes place in college it would be very helpful to restrict certain type of data to transfer.

→We were successful in synchronising our software with GUI made in java swing so that along with the functionality it may look good too.

Failures:

→We were unable to make a system in which no data is stored anywhere, cause in our system data is stored at server but this was just because of the time constraint.

Scope and Future Extensions

The software will serve students as the primary source of file sharing among others and get as much as data download without the use of internet and with very high speed.

Extensions Possible:

-UserId and Password validation Individually for each students so that server could easily track the client.

-Data sharing limit for each UserId so that an individual can share only some specified amount of data in a day of week, that would help in maintaining the data if necessary.

-Functions like restricting the shared file to some users can be a great feature, so that broadcasting could be avoided and private networks can share files among themselves only.

-The whole idea can be implemented on internet in place of intranet but at this time that is not required, intranet would solve the problem if this time is considered.

References

1. W. Richard Stevens, Unix Network Programming, Volume 1: The Sockets Networking API
2. www.Whatis.com
3. You tube
4. Computer Networking: A top down approach, James F. Kurose, Keith W. Ross.
5. www.Creately.com