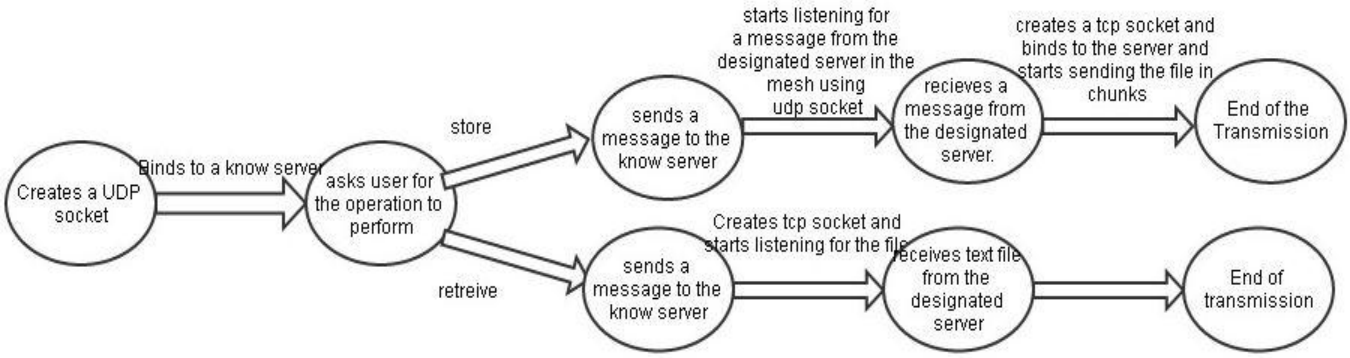
Socket Programming Lab: Distributed File System

Team members:

1. Rajeev Kumar M 110050052
2. Nithin Kumar 110050053
3. Prithviraj M Billa 110050065

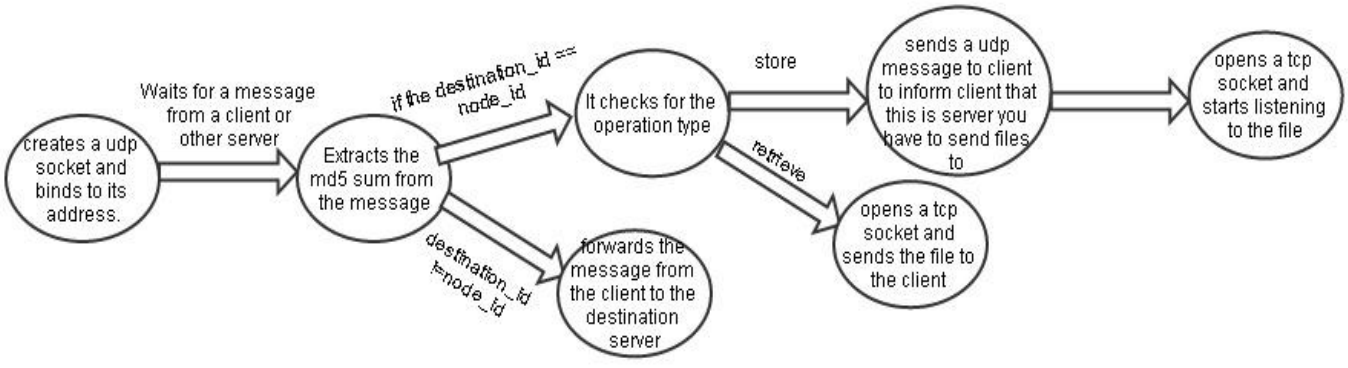
Client:

State diagram of Client:



Server:

State diagram of Server:



Readme:

1. Client creates a UDP socket and binds to a known ip address and port and sends a message to the server with the md5 sum of the file which is to be stored or retrieved. Message type: “ip_addreess_of_the_client:port:md5:store|retrieve”.
2. Server creates a UDP socket and binds to its ip address and waits for a message from the server or other client. Server receives a message from the client and calculates modulo(md5sum,number of servers in the network). If the destination node is not as same as the node id, it forwards the message from the client to the correct server.
3. If the operation is store, server sends a message to the client to inform the client that it has to send the files to this server. Then server opens a tcp connection and starts listening for the file. The client sends the file to the server.
4. If the operation is retrieve, server opens a tcp connection and and starts sending the file to the client. Client opens a tcp socket and starts listening to the requests.

Files:

./lab9

FileMesh.cfg	configuration files for servers
Client.cpp	program to run on client side
Node.cpp	program to run on server side
Input.txt	sample input file for client side

Compiling Instructions:

```
For Client:

$ g++ Client.cpp –o Client

$ ./Client < Input.txt

For Server:

$ g++ Node.cpp –o Node

$ ./Node
```

Input for the Client (in Input.txt)

--Ip_addr of the client

-- Port of the Client

-- filename | md5sum of the file

-- Operation (store | retrieve)

Input for the Node

-- Node ID (starts from 0 to linenumber-1 from the FileMesh.cfg)

-o-