ChangeLog:

1. Added parameter contentName to inform\_And\_Update
2. Added return String content to hash function

Pseudocode for P2P Server

---------------------------------------

Required variables:

1) recordList – arraylist of local (content, IPaddr) records

2) requestedRecord – record requested by client

3) DHTserverList – arraylist[4] of DHTserver records (DHTserverNum, IPaddr, status)

4) sharedDir – path directory of shared files

5) jobNum – index of uploaded jobs.

6) transferStatus – (pending, seeding, done) indicates status of upload/download.

7) servSock – server socket for listening to P2P clients’ requests.

main(){

init\_Vars()

init()

cat\_Content()

Try{

Create TCP serversocket(servSock)

Bind TCP socket servSock

While true

If numClients < LIMIT

Create Client\_Thread(client socket: run())

numClients++

If signal QUIT (^C) received

exit()

cat\_Content()

end while

End try

Catch signal (^C)

Exit()

Free allocated memory

Close welcome socket

End catch

}

init\_Vars(){

Create/instantiate new object DHTserverList[4].

Create/instantiate new object recordList.

Create/instantiate variable requestedRecord.

Set numClients to 0

}

init(){

Store the 1st DHT server (DHTserverNum, IPaddr) record.

Create UDP socket

Do

Request (UDP:GETDHT) DHTserverList[0] for IP addresses of remaining servers.

While return message code is not equal to SENDDHT

Retrieve DHT servers 1-3 info from message

Close socket.

}

cat\_Content(){

Read sharedDir and obtain list of files.

For each file in sharedDir

Inform\_And\_Update(FileName)

}

Inform\_And\_Update(contentName){

DHTserverNum = hash(contentName).

If DHTserverNum status is online,

Create UDP socket to DHTserverNum

Send (UDP:INFORM) (record (contentName, IPaddr) to DHTserverNum.

Store the local record (contentName, DHTserverNum, IPaddr) to recordList.

}

hash(content){

sum decimal values of ASCII characters of content name

set x to sum of decimal values

set y to x mod 4

set DHT ID to Y + 1 (actually, just Y for our implementation)

return ID

}

/\* Services P2P client requests \*/

Class Client\_Thread(){

Run(){

Create input buffer

While true

Determine command from client packet

If request message corrupted

Send 400

Else if HTTP version is not 1.1

Send 505

break

Else if command = GET

Determine content file

If content file not found

Send 404

break

Else

Create output buffer

Send content file

Send 200

break

End while

Close socket

}

}

Exit(){

For DHT servers i from 1 to 3

Send QUIT to DHTserverList[i]

}

**DHT\_Server Status Codes**

(Used to communicate between DHT Servers and (P2P Servers or Clients)

1) GETDHT – received from P2P client to query for IPs of other DHT servers.

2) SENDDHT – sent by DHT server to P2P client, contains IPs of other DHT servers.

3) REQINFO – sent by P2P client to DHT server to request records of all P2P servers that have content name.

4) SHUTDOWN – sent from DHT server to DHT server disconnects/closes all sockets and connections with P2P servers and clients

5) INFORM – received from P2P client, contains record (content name, client IP).

6) INFORM – sent by DHT server to P2P client, containing records (content, IPaddr). This is in response to REQINFO from P2P client.

7) QUIT – send by P2P Server to each DHT Server. Each DHT server will delete P2P Server’s content records.

8) CODE404 – sent from DHT server if ip address of client not found.

**P2P\_Client and P2P\_Server Status Codes**

(Used to communicate between P2P Servers and P2P Clients

1) GET – sent by P2P client to request a content file.

2) 200 – sent by P2P server if request succeeded (i.e. client IP found).

3) 400 – sent by P2P server if request message corrupted.

4) 404 – sent by P2P server if content file not found.

5) 505 – sent by P2P server if HTTP version not supported.