Change Log:

Added type Content\_Record to required variables

Added type DHT\_Record to required variables

Pseudocode for DHT Server

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

Required variables:

1) recordList – arraylist of local (content, IPaddr) records of type Content\_Record

Content\_Record – data class structure with instance variables (String content, String IPAddr)

2) requestedRecord – record requested by client

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

DHT\_Record – data class structure with instance variables (int serverNum, String IPAddr, String status).

Status should be enum with only 2 values {"online", "offline"}.

4) sharedDir – path directory of shared files

5) jobNum – index of uploaded jobs.

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

Main(){

init\_Vars()

create/fork process – connect\_DHT()

Try {

create UDP socket

While true

If message = GETDHT (NOTE: only for DHT Server[0])

set message code to SENDDHT

send list of other 3 DHT servers IPs to P2P client

send DHT servers information to P2P client (in one UDP packet)

Else if message = INFORM

Determine content of P2P server IP

add\_record(P2P Server IP, content)

Else if message = QUIT

Determine content of P2P server IP

remove\_records()

Else if message = REQINFO

Determine requested content of P2P client IP message

If content not found

set message code to CODE404

send to P2P client

else

set message code to INFORM

send record (content, IPaddr) to P2P client

end while

end Try

catch (interruptedexception ie)

if ie = (^C signal)

shutdown()

end catch

}

Init\_Env\_Vars(){

create recordList – stores records(content, IP address), which will then be hashed to its ID

create DHTServerList – stores records(IP address, status) of other 3 DHT serves

set status of DHT servers 1-3 to “offline”

}

\*\*\* DHT to P2P methods \*\*\*

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)

}

add\_records(P2P Server IP, content){

If record not already in recordList

Add content name, client IP addr to recordList

}

remove\_records(){

remove all records matching client IP addr

}

send\_record(){

retrieve record matching content

}

\*\*\* DHT to DHT methods \*\*\*

connect\_DHT(){

try

if DHT server ID = 0

for DHT servers i from 1 to 3

create TCP ServerSocket SSocket i

create DHT thread get\_DHT(DHT Server i)

Else

create TCP Client Socket CSocket

connect to DHTServer[0] Socket

While true

listen to TCP ping message “Hello” and respond with “Hi”

end while

end try

catch signal ^C

close TCP sockets to all DHT servers

end catch

}

/\* Only used by DHT Server[0] \*/

get\_DHT(DHT Server I, socket){

attach BufferedReader and BufferedWriter to socket

While true

If Ping\_DHT(i) returns “server i online”

set DHTServerList[i].status to “online”

else

set DHTServerList[i].status to “offline”

Wait 1 second

end while

}

Ping\_DHT(Server i){

create TCP socket for server i

send TCP ping message “Hello” to DHT server

if no reply in 1s

return “server i offline”

else

return “server i online”

}

Shutdown(){

send SHUTDOWN signal to other 3 DHT servers

close(disconnect) all UDP sockets to P2P clients and P2P servers

disconnect from other DHT servers

exit application

}

**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.