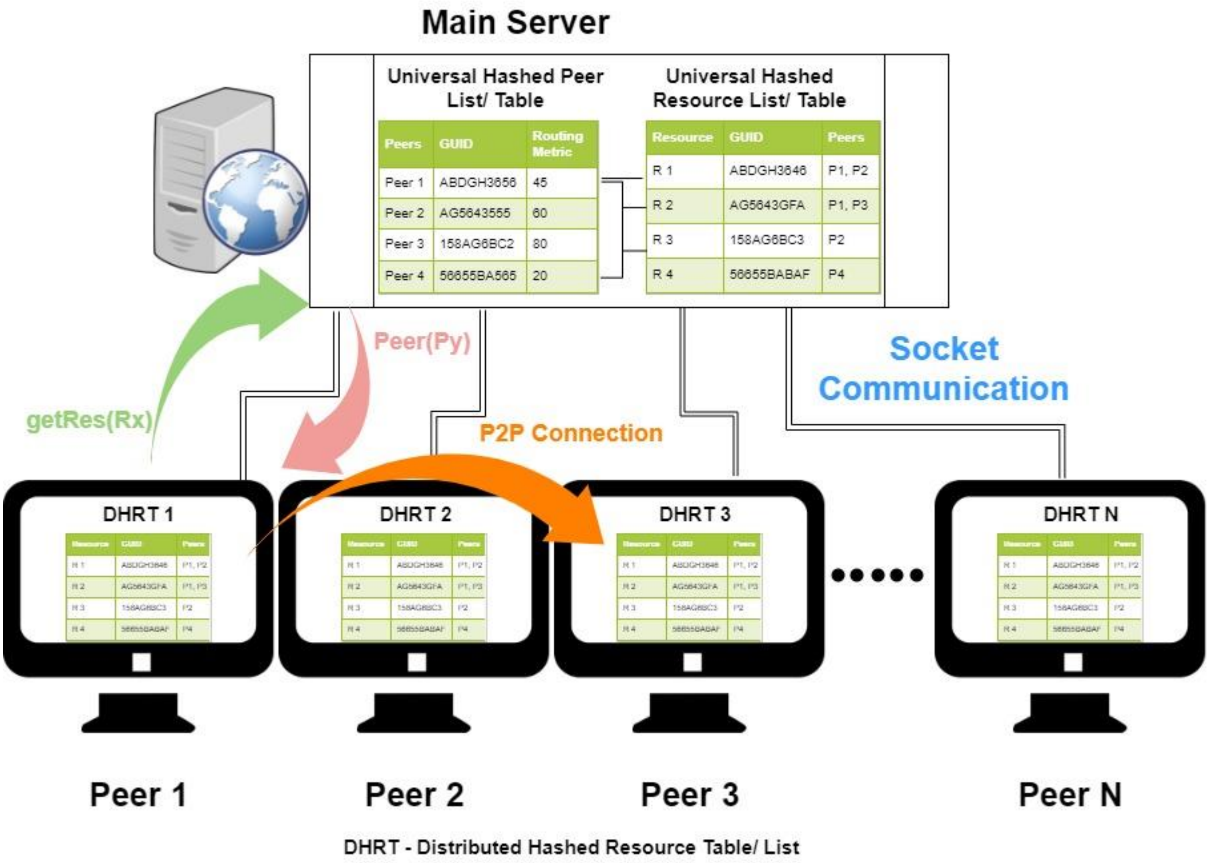
Assignment 3: Practical (40% of final grade)

The task of this assignment is to design and implement a Generation 2 Peer-to-Peer (P2P) system that establishes the file sharing with a Distributed Hash Table (DHT), as explained in the lecture slides. The main intention of this assignment is to replicate the functions of real-world P2P systems, employing simple constructs such as socket programming, hash functions, hash tables/ arrays/ array lists. Fig. 1 illustrates a sample design the students can follow to establish the P2P communication. The system should have a Main server for maintaining GUIDs of Peers and Resources. These could be implemented using arrays, arraylists, or similar constructs (i.e. in Java). The hashes can be generated with SHA-1 algorithm. The clients or peers of the system should each possess a Distributed Hashed Resource Table (DHRT). These DHRTs are only updated upon the requests made to the server regarding resource sharing. These resources can be files of any format that can be stored under a peer directory or even a simple text-based paragraph that carry a considerable weight. However, UHRT is updated all the time; and upon adding new resources, UHRT should be advertised of the changes. When a Peer wishes to request a resource Rx, it should forward the getRes(Rx) request to the server. Then the server will attempt to browse the relevant Rx within its UHRT, and locate the corresponding peers in possession of it. Depending on the Routing Metric (i.e. a value generated based on the distance and bandwidth of each peer connection. For the implementation, these values can be assumed.) a peer to share the resources should be selected. Once the selection is made, both the requested peer and shared peer are notified on how to reach each other by the server. Ultimately, requested peer will establish a P2P connection with the sharing peer, that leads to sharing or sending of the resource to the requested peer. Upon receiving the content, requested peer should save the file or write the content to a directory or a file.


Main Server
Universal Hashed Peer List/ Table
Universal Hashed Resource List/ Table
getRes(Rx)
Peer(Py)
P2P Connection
Socket Communication
DHRT 1
DHRT 2
DHRT 3
DHRT N
Peer 1
Peer 2
Peer 3
Peer N
DHRT - Distributed Hashed Resource Table/ List

Peers	GUID	Routing Metric
Peer 1	ABDGH3656	45
Peer 2	AG5643555	60
Peer 3	158AG6BC2	80
Peer 4	56655BA565	20

Resource	GUID	Peers
R 1	ABDGH3646	P1, P2
R 2	AG5643GFA	P1, P3
R 3	158AG6BC3	P2
R 4	56655BABAF	P4

Figure 1 - Sample Design of the Intended P2P System

The main tasks of this assignment are specified below.

1) The server will be able to support multiple peer connections.

2) The server should generate the GUIDs for both connecting peers and resources. The initial resource possession can be assumed.

3) The server should maintain the UHPT and UHRT as indicated in the Fig. 1. This can be combined if you wish.

4) When a peer connects to the server, its GUID is generated by the server, and the table will be updated. The resources (i.e. their GUIDs) in their possession are advertised to the server.

5) Peers should maintain the DHRTs. It can be assumed that all the DHRTs have the same content as the UHRT in the beginning.

6) When the getRes(Rx) request is sent to the server, server will search its UHRT for the appropriate resource and locate its possessed peers from the UHPT. We can assume that the peer is aware of the resources' GUID. A selection function should be designed and implemented if there are more peers possessing the requested resource. (i.e. Using a routing metric or any other routing overlay method).

7) Send the selected peers' IP Address/ port no. or relevant API to the requested peer by the server.

8) Establish a P2P connection between the peers under discussion. This can be established via exposing a socket/ port of the sharing peer, or any other method that ensure the P2P nature. You can refer:

http://freesourcecode.net/javaprojects/78126/P2p-messenger-in-java#.YW1vNxrMIuU

https://mark0110.github.io/P2P-Networking/

(Sharing the resources through the server connection is acceptable, but certain amount of marks are allocated for ensuring the P2P nature and the creativity of the student)

9) Share the resource or send the content of the file saved in a directory under the sharing peer. These files can be created and saved under each peer at its execution or linked from a pre-existing location. If the student finds its difficult to implement file handling, even a text-based paragraph specified at the imitation is still acceptable. Though, there will be marks allocated for the creativity aspect.

10) Upon receiving the shared file/content, it should be saved or written to a directory under the peer.

It is acceptable for the above functionality to be provided via a command-line interface. Optionally, for extra credit, you may provide a simple Graphical User Interface to perform these tasks. But the majority of the marking rubric is focused on the DHT, Peer discovery, and P2P communication.

Deliverables

a) Design document, describing the design of your system, performance statistics, outlining the pros and cons of your system and giving the rationale for any design decisions you made (Maximum 4 pages);

b) 10 min video on the presentation of the use of your system (i.e. introduction to your system/ implementation of DHTs, getRes(), P2P session, and P2P file sharing succession. specially the completed file sharing part);

c) Code with comments/ Project file.

When submitting, each student should submit a folder with 3 files. Design document (.pdf / .doc / ...etc), recording of presentation (.mp4) and all the program files compressed into a single file (.zip / .rar / .7gz....etc). Please follow these instructions to successfully complete the submission.

Submission deadline: 21st November 2021