COMP3331/9331 Computer Networks and Applications

Implementation of Peer-to-Peer Network using Distributed Hash Table

Assignment Report

Wincent Winarko

Z5240353

Language used: Python 3.7.3

Platform: Linux (CSE Vlab)

How the program works:

1. Initialize the P2P network by initializing at least 4 peer using

   $python3 p2p.py init <peerID> <first successor> <second successor> <ping interval>

Or just run the init_python.sh that initializing 7 peer simultaneously with *ping interval* 15s

   $./init_python.sh

2. Every peer will send ping request to their successors every *ping_interval* and get response if the successors still alive.

3. New peer can join the network by changing the run type “init” to “join” and give known peer instead of the successors

   $python3 p2p.py join <peerID> <known peer> <ping interval>

4. The *known peer* checks whether the new peer is its new successor and if not, it sends join request to it successor until it reach the correct position/peer in DHT. At the correct peer, the peer will contact *peerID* which is the new peer about its successors detail, update *peerID* as new first successor and also tell its predecessors to update the successors with *peerID* as new second successor.

5. There are 2 ways for a peer to exit the network:

   i. Peer Departure (Abrupt) – Press **ctrl+c** in the terminal. After the peer exit abruptly, the predecessor will reach the timeout (Sent 3 ping request and has not receive any ping response).
   ii. Peer Departure (Gracefully) – Typing Quit as an input. The predecessor will be notified by this leave and updates its successors as the first and second successors of leaving peer. Also update the second predecessor of the leaving peer.

6. By giving Store <filename> as input, the peer hashes the filename and finds the correct location to store the file in a list of files.

> **Store <filename>**

7. We can also retrieve a file from DHT by giving filename, same step as storing, the peer hashes the filename and look for the peer that correspond to the key and check whether that peer has the file in its files list. If yes, the peer that has the file will send it to the requested file.

> **Request <filename>**

Possible improvements and extensions:

1. Better logic and cleaner code – has a big function that handle all the incoming tcp request with many if statement.

2. Implement the chord algorithm where each peer have a shortcut to every peer in the network.

3. Better implementation of class for PeerNode class.

4. Implement the sending requested file through tcp connection and make a copy of it.

Problems or Potential Problems:

1. Failed to send and receive the requested file from one peer to another peer. Hence, did not implement them.

- Do not know how to handle the data sent since the function handle the incoming request decode the request immediately to match the substring in if statements.

- Failed to decode the bytes contain data of the requested file. **UnicodeDecodeError: 'utf-8' codec can't decode byte 0x9c in position 72: invalid start byte.**

- Tried the pyPDF2 module to extract the data per page and encode it and send with the filename but the extracted data is blank just an empty string.

2. Printing “Ping requests sent to Peers -1 and -1”at the start of peer join when requesting join after the peers initialized (Peers have not received any ping response yet).

Reference:

- Sample codes at assignment section in webcms