Report

Comp9331 Assignment
Student name : Yitong Xiao
Student ID : z5151559

1.Program design:

Although I use python3.6 version to simulate DHT, I change the rules for coding in order to fit the environment on CSE, So, my code is also work on python 2.7 or above. And the packet I used is threading, socket and time. In addition, I use different thread to execute different client and server, client and server for TCP and UDP respectively. What's more, In order to describe my thoughts for implement this assignment, I would like to analyze my code by different aim I can realize

(1) For UDP server and client:

In order to implement this action, I need implement udp server and client because of using UDP protocol. So, according to the arguments I input, I know the port and the address of my successors. My client just builds udp socket and send the peer name of itself and the sequence number to the successor, and the interval of sending message is 5 seconds. And then just wait the response which the udp server send. If the client receives the response, which means the successor is alive. The client gets the successor id and the sequence number, and print ''the response has received''. For the sequence number, it is used for the alive detection. If don't receive the message, then the client will lost a sequence number. When it happens five times, then the client will regard the successor as not alive.

When udp server receive the ping message, get the address where the ping message send from, and send it's peer name and the sequence number that it received. And print ''A ping request message was received''

(2) For TCP server and client:

According to the requirements in the spec of the assignment, I use tcp protocol to transfer messages for quit and request function.

First, for request file:

When inputting the request file in the terminal, then it will use tcp client to send message to his first successor, and the message are his peer id and the file id and the peer id who request the file. When tcp server get the message, it will compare the file id, peer id which are in the message and its peer id. Then through the hash function of the file, it will know if the file belongs to itself. If not, it will use tcp client to send the message to his first successor, which are the same with what it got before, just replace his predecessor peer id with its peer id. If the file in its location, then send the message to the peer who request the file, and tell him the file is in its location.

Second, for quit:

When inputting quit in the terminal, it will use tcp client to send messages to his predecessors and successors. Tell the predecessor he will leave the DHT and send his peer Id and his successors to the predecessor. When his first predecessor got the message, just update his second successor. When the second predecessor got the message, update his first successor with his second successor and change his second one with the new peer id. The successor of peer who quit DHT just update its predecessor.

The last requirement for killing a peer:

When udp client one of his successor is not alive using the check sequence number method mentioned above, the use tcp client send message to the alive successor to ask his successors. Actually, there are two different conditions. If the dead peer is the second successor, then update his second successor with the first successor of his first successor, on another condition, then update his first with his second one and update his second one with the second successor of his second successor.

(important note: In my code, the first predecessor always find the dead peer first)

2.Screen demo

Upload my testing demo on Youtube. You can view it with the link below:
Demo link: https://www.youtube.com/watch?v=SEasB-02a1U&t=195s