COMP9331 Computer Network
Assignment 1 Report

Z5146418 Yuchen Yan

1. Important: This program is written by python 3.6
2. Important: The script code I wrote it by myself, and it is inside the tar file I submited. Put the cdht.py file and script all on Desktop before run them.

1. The steps to implement the program.

(1) Overall structure of the code: there are six threads in my program, which are UDPServerThread, UDPClientThread, UDPresponseSocket, TCPfileServerSocket, TCPfileCLientSocket and also TCPkillSocket. Each thread has different function.

(2) UDPServerThread:

Mean propose of this thread is to receive the ping request message and send the corresponding response message to the corresponding peer. The server socket should bind the server port of this peer. And when receive the request message, print 'A ping request message was received from Peer …'… is the corresponding peer who send the request. And then send the response message to that peer.

(3) UDPClientThread:

This thread just keep sending the ping request message to the first and second successor.

(4) UDPresponseSocket:

This thread is mainly to receive the response messages. And printout the message 'A ping response message was received from Peer …', … is the peer who send the response message.

The last three thread implement the most part of this assignment, including the request file, peer quit and also kill a peer.

(5) TCPfileClisentSocket:

In this thread, when people input quit or request file, the sentence will be accpted by this thread. And when the input is quit the thread will kill all the other threads. And this peer will be quited. When the input is request file and then, if the file is in this thread it will print file filename is here. If the file is not here it will print, 'File request message has been forwarded to my successor.' And send the corresponding message to the first seccussor.

(6) TCPfileServerSocket:

There are several functions which related to the TCPserver socket. Firstly it can receive the file request message from other peers. If it is in this peer printout'file is here, and send the response message to the peer who request the file'

And also if some successor will depart, the peer will also print which peer will leave and its new first successor and second successor.

(7) TCPkillSocket:

In the thread mainly to judge whether its successor has been killed or not and rearrange the first and second successor. When it notice that its successor have been killed then it will print'Peer',str(r) ,'is no longer alive.' And 'My first successor is now peer' 'My second successor is now peer'.

2. The design of my code, and with some explanation of my implementation.

(1) The library are imported:

threading(In order to create different threads.)
time(time.sleep() was used in this program)
socket(the most important one, in order to create the TCP and UDP socket)

(2) The sockets:

Four sockets were created, which are UDPclientSocket, UDPserverSocket, TCPserverSocket, TCPclientSocket. Because the peer to peer structure not only be a client or server, but need the program be both client and server. And the ump and tcp protocol are both used in this assignment so these four was created.

(3) The detail you can see the code in this directory.

(4) the interval between ping message is 10 second. But you can easily change it by change the time.sleep(time). The time inside the sleep method can be changed.

3. The link to my demo.

In this demo Video, I showed the steps one by one to demonstrate the functions of this assignment. Including ping message, request file, quit a peer and also kill a peer.

https://www.youtube.com/watch?v=9NJnjvX8Y1E

4. Clarification:

This assignment was implemented by YUCHEN YAN (z5146418) individually. I didn't outsourcing any other people's code or thoughts.

The is too long to copy the code here. It will use about 3 pages, then the report will be longer than 3 pages. So the code will in another .py file.