z5097965
Hongyu Chen
COMP9331 assignment report

Designing and Implementing an IoT Data Collection and Sharing Network

Language: Python 3.9

Platform: VLAB

Assignment Specification accomplished:

- Authentication (log-in, log-out, block with 10s)
- Commands: EDG, UED, SCS, DTE, AED, OUT, UVF
- p2p

Application Layer Protocol:

Message format used:

- Header (encoded datatype, eg. ERROR, EDG, UED, COMMAND, etc.)
- Data (encoded content)

During authentication, client will encode the credentials as data and send header+data to the server. Server will check username and password respectively, and give response if any is wrong.

After authentication, the client needs to send command with its argument to server. Client side will check whether command and argument is correct.

Transport Layer Protocol: TCP, UDP

Possible Improvement:

1. Better logic and design
   a. Many if – else statement, makes code too long and hard to debug
   b. Code between the TCP thread and the UDP thread is complex, cause not very familiar with the multithreading method.

How my program works:

1. Run python3 server.py <port> <num trying to reach block> set TCP server
2. Run python3 client.py <IP> <port> <UDPport> in another terminal, set TCP client and UDP server
3. At client terminal:
   i. Ask username and password
   ii. Send username and password to server
   iii. Receive check message from the server:
      a. If username is not in credentials.txt, ask for username and password again, no limit of times
      b. If password is not correct, ask for only password. If give wrong password for <num trying to reach block> times in total. The

username will be block for 10s. Cannot log-in with this username, even in another terminal. close TCP, send UDP port number. Close UDP

c. If already blocked, close TCP, send UDP port number. Close UDP
d. If message is “welcome”, means information is correct: send UDP port number.

At server terminal:

i. Check username and password received, give feedback.
ii. Block if try <num trying to reach block> times, send to client’s UDP server
iii. Give feedback if try to log with blocked username, send to client’s UDP server
iv. Log-in if all correct:
    a. Receive UDP port number
    b. Update edge-device-active file

4. Command between server and client:
    i. Server first checks whether other client is processing. If other in processing, this command needs to wait until the other finishes. Client will check whether the format of command and arguments is right.
    ii. EDG:
        a. Client sends command to server
        b. Server tells client to start generating file
        c. Client tells server done
        d. Server receives done
    iii. UED:
        a. Client sends data in file with correct fileID to server
        b. Server writes received data in a file with correct name
        c. Update upload log file
    iv. SCS:
        a. Client asks for computation operation
        b. Server checks if it can do
        c. Server reads the correct file, and does the operation, and give the answer back
        d. Client displays the answer
    v. DTE:
        a. Client asks to delete uploaded file
        b. Server finds it and delete it in server work directory
        c. Update deletion log file
        d. Server tells client done
    vi. AED:
        a. Client asks for other active edge device
        b. Server look up edge device active file and give information with clients in it except for client asks for this

vii. OUT
   a. Client asks for disconnection
   b. Server send message to client's UDP and ask to disconnect
   c. Client disconnects TCP and UDP

viii. UVF:
   a. Client sends command and argument to server
   b. Server checks whether the other client this client want to send data to is active and give feedback
   c. Client read given mp4 file (maybe other files) and send them via UDP in packets
   d. Destination client receives them via UDP, and write data in correct file name

Possible problem:

1. Split fail, not sure how it will happen. May comes out unexpectedly. During my test with more than 50 times, only appear twice.
2. TCP, UDP receive delay or loss. Because of internet quality or time delay, maybe. Happen very rare times.

Reference:

- https://eecs485staff.github.io/p4-mapreduce/threads-sockets.html
- https://stackoverflow.com/questions/13993514/sending-receiving-file-udp-in-python
- https://blog.csdn.net/ChaoChao66666/article/details/126674607
- https://www.programiz.com/python-programming/datetime/strftime
- https://www.geeksforgeeks.org/socket-programming-multi-threading-python/
- https://stackoverflow.com/questions/4710067/how-to-delete-a-specific-line-in-a-file
- Muti-threaded code provided below assignment specification