

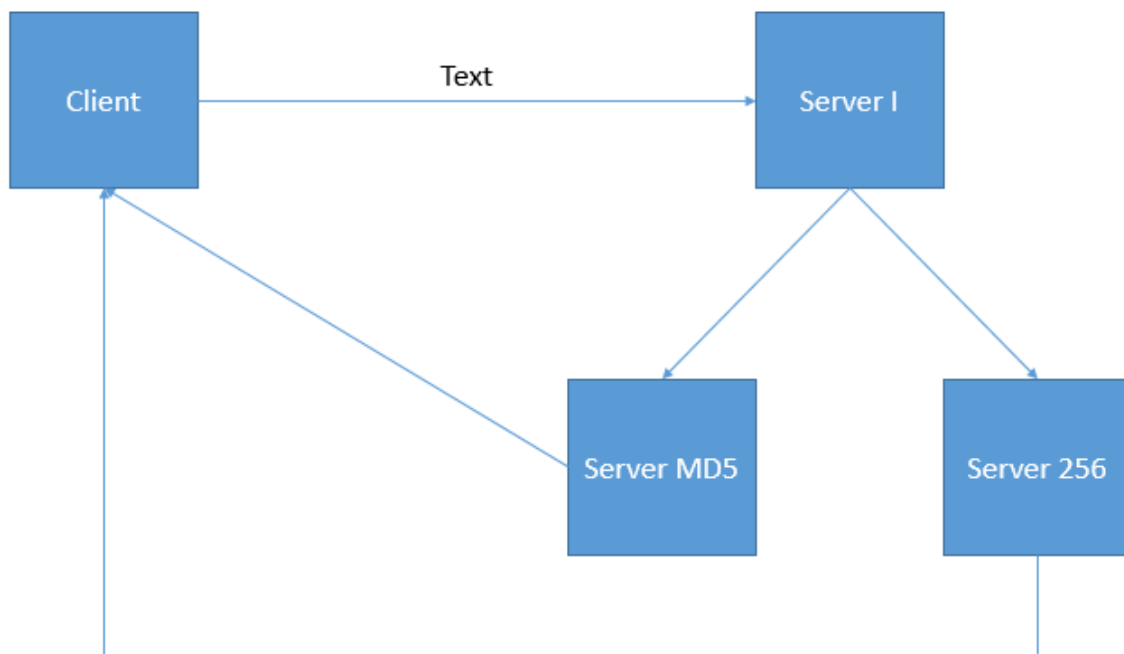
Assignment

The goal of this assignment is to implement a simple client-server application using Java.

The servers will start in passive mode listening for transmissions. The client will then start and contact the server Frontend (on a given IP address and port number). The client will pass the server a string (eg: “network”) up to 80 characters in length.

On receiving an input from a client, the server Frontend should send it concurrently to other servers: server MD5 and server SHA256.

After receiving the input text, server MD5 and SHA256 will return MD5 and SHA256 hash value, respectively, to the client. The client has to wait for both hash values returned before showing them in the terminal.



Protocol

You have to develop a new network protocol. In other words, your task is to tell the client and server how to send and receive a packet. Here are requirements for your protocol:

- Use socket raw (Socket only has IP address, using to transmit IP diagrams).
- You have to attach to the diagrams some information about
 - o Source port: integer
 - o Destination port: integer
 - o Checksum: String
 - o Length: integer
 - o IsACK: boolean
- If the diagram arrives at the destination, the receiver will send back an ACK diagram to the sender to tell it that the diagram has reached the destination.
- If the checksum of the received diagrams doesn't match to the value in field checksum, so the diagram will be ignored, and the receiver doesn't send back ACK.
- After 10 seconds, if the sender doesn't an ACK, it will send the diagram again.

Example

Starting the server:

Assume that you started a server on machine 128.111.49.44, listening to port number 32000. The syntax should look like the following:

```
UET-machine1> java runServer 32000 1600 81 <enter>
```

(32000: port number of server I

1600: port number of server MD5

81: port number of server 256)

Starting the client:

```
UET-machine2> client 128.111.49.44 32000 43593sf435ggkd <enter>
```

(32000: port number of Server I

43593sf435ggkd is the example input message)

Response from server:

Hash value of “43593sf435ggkd”

SHA 256:

fffa835124abbdebc471919b7fc3a041c79f096715359177ffda3fd0e446a788

MD5: 3d36975e129dd93518164803540f39c0

UET-machine2>

At this point, the client should exit.

COMPILING YOUR CODE:

To compile your Java code, use the following commands as an example:

```
javac runServer.java
```

```
javac runClient.java
```

To run your Java code on the machines, use the following commands as an example:

```
java runServer 32000
```

```
java runClient 128.111.49.44 32000 43593sf435ggkd
```

PLEASE NOTE:

- Students should do as 2 men teams. If so, the teams have to submit an extra text file to point out the roles of members.

TURNIN:

You are going to hand in your source files and text file indicating members roles (for teams only). You have to submit before the day you take the final exam. Late submissions will not be accepted.

To hand in your assignment, Compress it as .rar file and send to namnx228@gmail.com. Filename should be your name.

Subject format: Assignment_com_net_x MSV_HoTen

x is the number as your subject format of Wireshark lab submissions mail.

Be careful: Wrong format mails will have no score.

GRADING GUIDELINES

It's okay if you can't do the whole assignment. Therefore, there are some levels of this assignment, and your score will depend on which level you can unlock.

- Level 1: Client sends requests to server Frontend, this server returns the two hash values. No particular protocol required. (4 pts)
- Level 2: Client sends requests to server Frontend, this server then sends the request to server MD5. Server MD5 returns the MD5 hash values to **server Frontend**. Finally, server frontend calculates the SHA256 value of the original input text before returning the result (MD5 and SHA256) to the client. No particular protocol required. (6 pts)
- Level 3: Client sends requests to server Frontend, this server then sends the request concurrently to Server MD5 and SHA 256. The then returns the MD5 and SHA256 hash values to **server Frontend** before returning the result (MD5 and SHA256) to the client. No particular protocol required. (7 pts)
- Level 4: Client sends requests to server Frontend, this server then sends the request concurrently to Server MD5 and SHA 256. The then returns the MD5 and SHA256 hash values to **the client**. No particular protocol required. (8 pts)
- Level 5: Client sends requests to server Frontend, this server then sends the request concurrently to server MD5 and SHA 256. The then returns the MD5 and SHA256 hash values to **the client**. The protocol which comes up above is required. (10 pts)

HINT

- At level 5, the client should have two sockets, one for communicating to server Frontend, and another for listening the reply from server SHA256 and MD5.
- Server SHA256 and MD5 have to know where they should send the result. So, find the way to send client's IP and port which are opened to listen the reply from the servers.

Resources/Help

You may use pieces of code from the Internet to help you do the assignment. You will not be negatively graded for using such code snippets *as long as they are properly cited*. Please email/meet me if you need any help or hints about your assignment. If you need some hints, mail me.

Good luck!