Foundation of Computer Networks
Project-1 Report

Amruta Chavan(agc9066)

- **The message Format Exchanged**
  - I have defined a Message class, objects of this class are passed as messages between the client, server and proxy.
  - The Message format is as follows:

Field	Use
`int msgId`	Message Id is used to identify the type of message msgId = 1, then the message is a simple time get request. msgId = 2, then message is a request to set time on server msgId = 3, then message is a response from server msgId =4 , then the message is an error response for the time set request.
String clientIP	This field is used to store the IP address of the client initiating the request
`int clientPort`	This field is used to store the port of the client initiating the request
String msg	The msg string contains the success or failure messages passed from server to client
String userName	The username field will contain the user name that has access to modify the server time. If the request is simple get time request then this field is null
String password	The password field will contain the password to get access to modify the server time. If the request is simple get time request then this field is null
`long time`	If the request is set time request then the time to be set is stored in this value
`int hopCount`	The hop count field is used to keep track of the number of hops from client to server.(one way)
String finalDate	The time obtained from server is in UTC format and it is finally converted to calendar format and stored in this variable

`long startTime`	This variable is used to store the message send time so that the total round trip time can be calculated.

The classes implemented are

1. **TimeServer:** This class is used to start the TCP Server and UDP server at specified ports
   The TCPTimeServer class implements the TCP server
   The UDPTimeServer class implements the UDP server
   The TCP and UDP server are started as two different threads on the specified ports

   The steps followed at the servers are
   a. The server (TCP or UDP) keep listening on its port until it accepts a new connection or in case of UDP server, it listens until a datagram packet is received.
   b. The server then reads the message sent by the client
   c. If the msgId is 1 then server will create a new message with time as the current time set on server, msgId as 3 and message as “03 TIME RETRIEVE SUCCESS”.
   d. If msgId is 2 , then server compares the username and password provided by client with those stored on server. If the username and password are correct then, the time value on server is reset to value in the message and a new message with msgId as 3,message as "03 TIME RESET SUCCESS" and new time is sent to client. If the username and password do not match then msgId is set to 4,message is set is "04 TIME RESET NOT SUCCESS" and this message is sent to client

2. **The TimeProxy class:**
   TCPTimeProxy class is used to start a TCP Proxy
   UDPTimeProxy class is used to start UDP server
   The steps followed at ProxyServer are:
   Like the main server there are two proxy server, TCP and UDP.
   Each server has two sockets open .One socket is used to communicate with client and second is used to communicate with Main Server.
   a. The proxy server will act like a server for the client and it will act like a client for the server.
   b. The hop count is increased by one before sending the message forward.
   c. While receiving the message from server the proxy server doesnot make any changes.
3. **TimeCliet:**

This class has implementation for both TCP and UDP connection.
The message to be sent is formed in this class

The steps followed ny client are:

a. The client creates the message according to requirement
b. The message is sent to server on appropriate port(TCP or UDP);
c. After receiving the final response the client will display the results

- **Example Execution:**
  - **Server Execution::**

```
agc9066@kansas:~/FCN$ javac *.java
agc9066@kansas:~/FCN$ java tsapp -s -T 1000 --user admin --pass paswd 5000 5001
Server started on129.21.37.18
TCP SERVER IS LISTENING
UDP SERVER IS LISTENING
```

  - **Proxy Execution::**

```
agc9066@glados:~/FCN$ java tsapp -p 129.21.37.18 --proxy-udp 5000 --proxy-tcp
5001   4000 4001
PROXY STARTED ON IP::129.21.22.196
TCP PROXY STARTED
UDP PROXYSTRATED
UDP RECEIVED MESSAGE
```

  - **Client Execution::**

```
agc9066@doors:~/FCN$ java tsapp -c 129.21.22.196 -n 1 -t 4001
IN CLIENT TCP REQUEST
MESSAGE SENT IS ::
msgId 1 ClientIP 129.21.37.36msg              Request to get timeclientPort
4000userName     nullpassword    nulltime       0
CLIENT WAITING FOR TCP RESPONSE
```

- **Steps for executing the program:**
  Compile all the files.
  Start the server first using tsapp.
  Start the proxy if any.
  Start the client.

- **Assumptions::**
  The sequence of arguments passed to tsapp will be same as mentioned in project document.