

**Project**  
**Foundations of Computer Networks**  
**Srinath Kanna Dhandapani (sd2689)**

**Protocol Messages:-**

After the program for server and proxy is run with its corresponding arguments, client sends the message with the details consisting of the task it would like to perform on the server and also the details about the type of connection it would like to set up with the server. The **message sent from client** looks like:-

Type of request to the server (REQ)	Type of connection to the server (TYP)	Username	Password	Time to set on the server (TIME)
-------------------------------------	--	----------	----------	----------------------------------

REQ- The request to the server can be 0, 1 or 2.

0 - To get the time from server in calendar format (flag is set for not -z or -T in command line)

1 - To get the time in epoch format (flag is set for -z in command line)

2 - To set the time in epoch format (flag is set for -T in command line)

TYP- The type of connection to the server can be 'u' or 't'.

u - For UDP connection to the server or proxy (flag is set if -u or not argument in command line)

t - For TCP connection to the server or proxy (flag is set if -t in command line)

Username- Username to set the time in server (required for REQ-2).

Password- Password to set the time in server (required for REQ-2).

Time – Time in epoch format to set on server (required for REQ-2).

Proxy reads the message from the client and establishes a connection to the server based on the TYP field. If the proxy is run on specific mode either TCP or UDP with -t or -u in command line then it directly connects to server in that mode.

Once the message reaches the server it reads the message from client and creates a return message consisting of the details requested by the client. The **return message** from server looks like this:-

The data requested by the client (Data)	
Proxy n IP Address	RTT for message from the proxy to server
Proxy n-1 IP Address	RTT for message from the proxy to server
Proxy 1 IP Address	RTT for message from the proxy to server

Data – The data sent to client can be

Epoch time (For -z in command line)

UTC time (For no argument in command line)

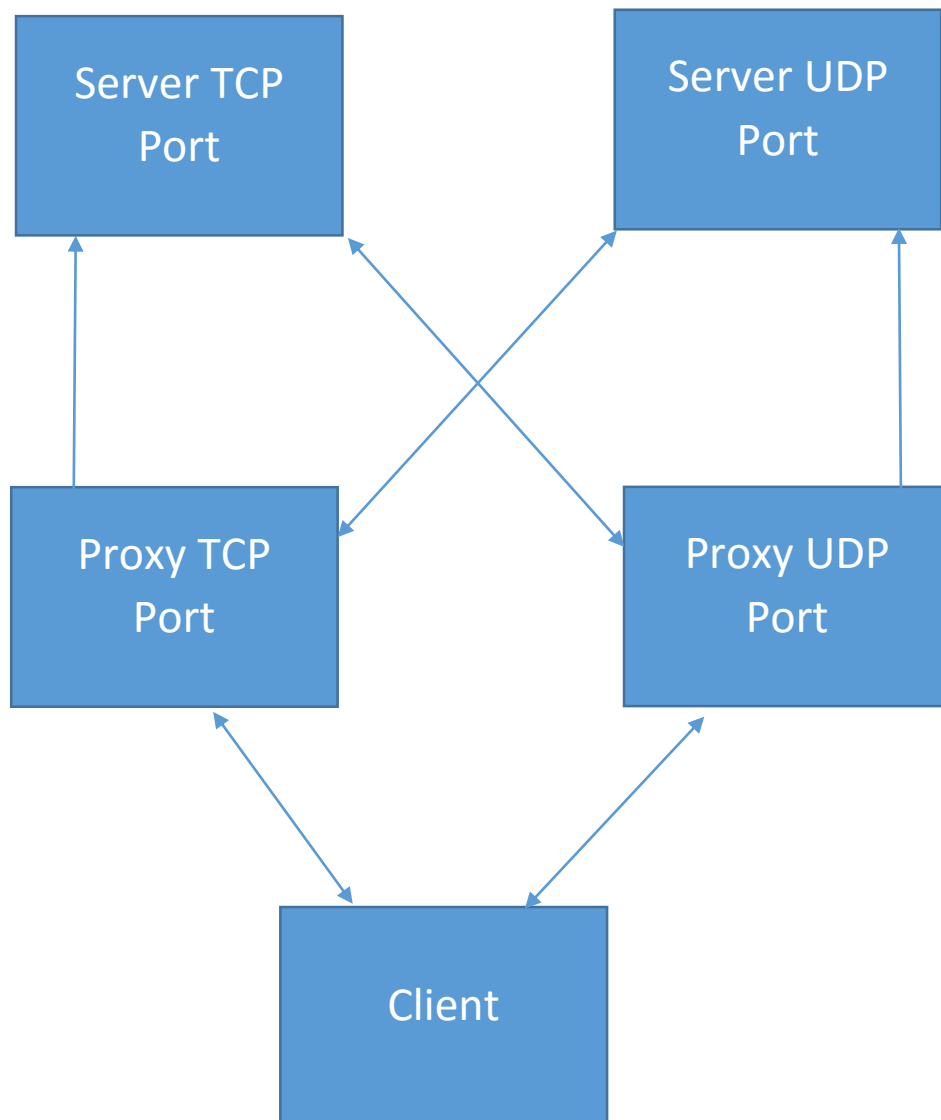
“Valid User: The time is set on server as: xxxx” if the username and password is valid

“Invalid User:” if the username and password is invalid

The details of the proxy IP Address and the RTT time for the message to reach server from that proxy is appended on the packet for all the proxies on the way back till client.

Once the message reaches the client back the message is parsed for the corresponding details and displayed to the client.

**Architecture diagram:-**



### **Error Codes:-**

- If the user tries to perform any other operation with the server then an IO Exception is thrown specifying invalid operation.
- If the user does not have valid authentication to change the time in server then a response from server is sent indicating "Invalid User:"
- If the response takes longer than usual and if the server is busy serving other clients then a request time out exception is thrown back as "request-time out:"

### **Implementation:-**

A single binary is designed to play three different roles client, server and proxy. The first argument from the command line is used to differentiate among the three. When the user runs as a server then two threads are created for two ports one for TCP and the other for UDP, to act as server and handle requests from client and proxy. These threads run continuously serving the incoming requests. They respond back to the proxy or client in the same channel of connection.

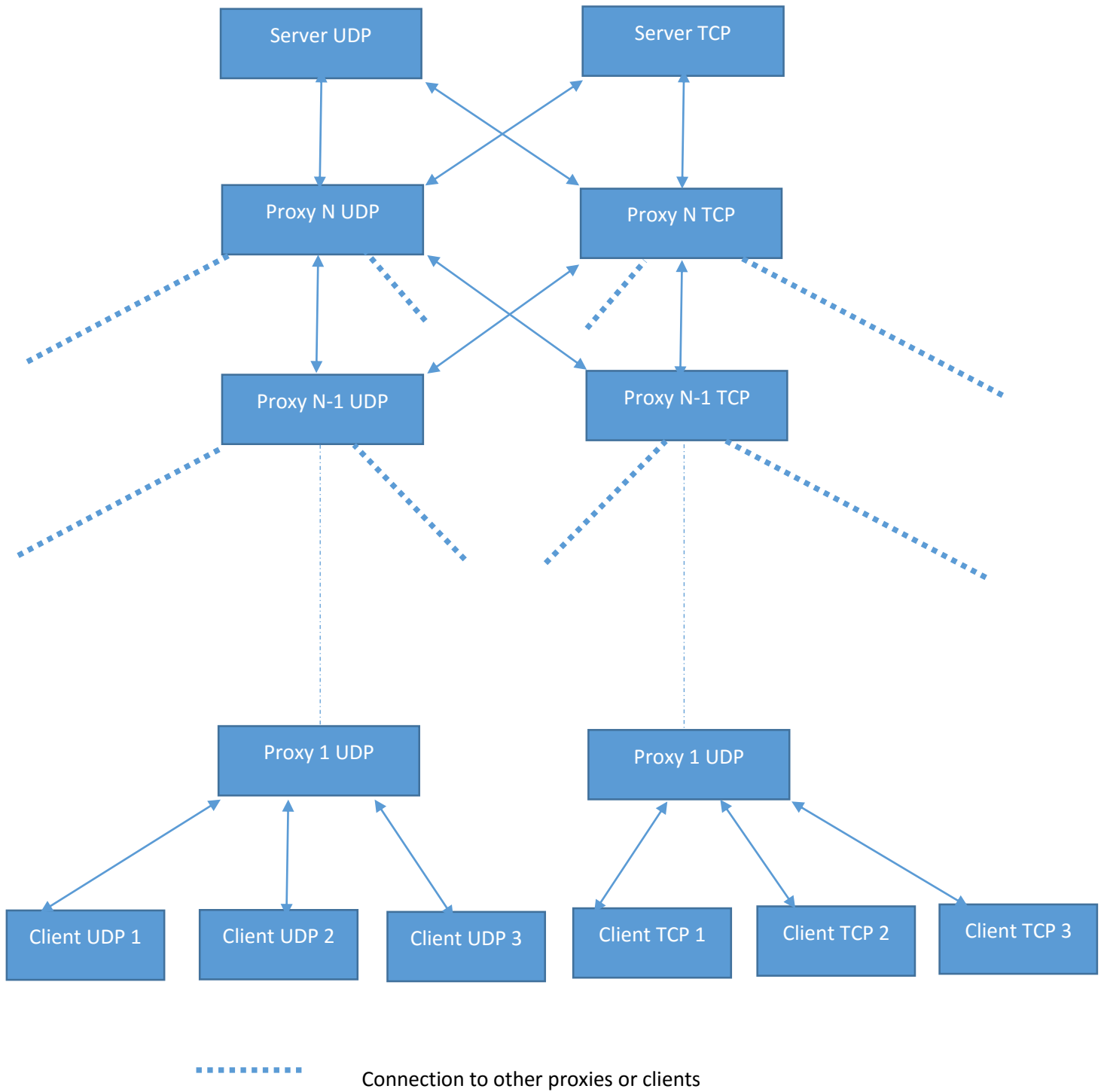
A proxy similar to server when started creates two threads for TCP and UDP and acts as server to the client. After getting the request from the client it passes the request to server in the channel mentioned while starting the program as -t for TCP and -u for UDP. If the channel of communication with the server is not mentioned then the message is sent to server in the channel specified by the client. This detail about the client is taken from the request message in proxy. If neither of them have specified the channel of communication then the message is transferred in UDP channel. Proxies run continuously on their respective ports and support multiple connections to other proxies and clients.

A client creates the message for server depending on its request to the server along with other details like username, password, time it wants to set on the server and the type of communication channel it uses. This message is sent to the server or proxy and waits for the response back. After the response is received the received message is broken down to get actual requested data and the list of IP address in its way along with their hop times. The client can request to query n times using -n followed by the query count. There are three kinds of operations that a client can perform with the server.

- 1) Get the time in epoch format.
- 2) Get the time in calendar format.
- 3) Set the time in epoch format.

Client can send the message to the proxy or server through two kinds of communication channel either by TCP or UDP.

**Flow diagram:**



### Sample use cases:-

1) Input: -s -T 5 5000 5001  
          -c 129.19.21.23 5000

Output:-

Message received from server is 1970-01-01T00:00:00.005Z

S.NO	IP Address	Time Taken
1	129.19.21.23	227

2) Input: -s -T 5 --user usr --pass pw 5000 5001  
          -c 129.19.21.23 -T 99 --user usr --pass pw -t 5001

Output:-

Message received from server is Valid User: Data set on the server is99

S.NO	IP Address	Time Taken
1	129.19.21.23	36

3) Input: -s -T 5 --user usr --pass pw 5000 5001  
          -p 129.19.21.22 --proxy-udp 5000 --proxy-tcp 5001 4000 4001  
          -c 129.19.21.21 -n 2 -u 4000

Output:-

Message received from server is 1970-01-01T00:00:00.005Z

S.NO	IP Address	Time Taken
1	129.19.21.22	134
2	129.19.21.21	145

Message received from server is 1970-01-01T00:00:00.005Z

S.NO	IP Address	Time Taken
1	129.19.21.22	1
2	129.19.21.21	1

### Failure case

4) Input: -s -T 5 5000 5001  
          -c 129.19.21.23 -T 10 --user usr --pass pw -t 5001

Output:-

Message received from server is Invalid User:

S.NO	IP Address	Time Taken
1	129.19.21.23	27