**How to verify the program:**

Run server first by the command.

*python rcmdd\_udp.py <port number>*

*(*for UDP)

*python rcmdd\_tcp.py <port number>*

(for TCP)

And then start the client by running,

*python rcmd\_udp.py <IPv6address of server> <PORT NUMBER OF SERVER> <Number of executions> <Delay in seconds> <Unix command>*

(for UDP)

*python rcmd\_tcp.py <IPv6address of server> <PORT NUMBER OF SERVER> <Number of executions> <Delay in seconds> <Unix command>*

(for TCP)

In the server program, IPv6address of server is hardcoded as “fe80::30a4:2fff:fe04:40ce”. So this should be given as *IPv6address of server* in the client command.

The Unix command will be executed by the server *execution\_count*times, with the gap between two executions specified by *time\_delay*and returns the time of execution at server and the result of these commands back to the client, which is displayed by the client.

Both server and client display these as:

Execution time: {}, Result: {}

The client program ends once all the response is received from the server after the server has executed the command *execution\_count*times. The client program ends by displaying “All work has been done”. The server continues serving client requests indefinitely.

For TCP, the server also displays additional message such as “Status” which says either connected or closed depending on whether the server has connected to any client or not.

In TCP implementation, the session can also be terminated by entering *rcend* on the client terminal. The server stops the execution of the Unix command and the connection with the client will be closed by the server. The client program also ends after displaying “Terminating session... “. At the server, it would say status “closed” and display the message “Session terminated by rcend at client terminal”.

I have verified the program with Unix commands such as “date”, “hostname”, “ifconfig, “ls -l”.

**Arsha Aboo**