Implementation TCP-like protocol within UDP data

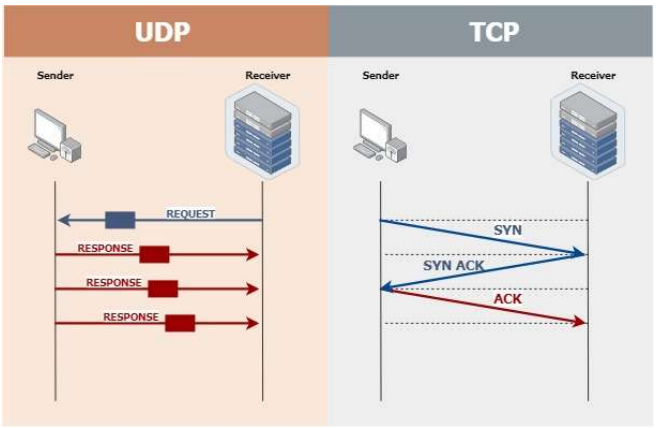
Group Details

Group-5			
S.No	Name	Registration No	Class & Section
1	M. Lokesh	AP18110010281	CSE-E
2	Y. Tarun Sai	AP18110010469	CSE-H
3	B.V.S. Neeraj	AP18110010474	CSE-H
4	M. Sri Krishna Kumar	AP18110010506	CSE-H

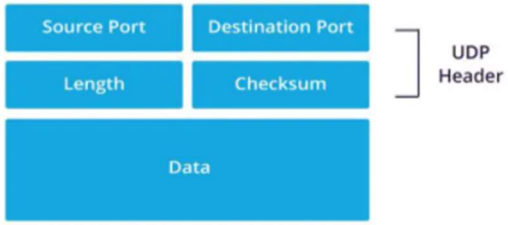
Description:

TCP stands for Transmission Control Protocol and UDP stands for User Datagram Protocol. Both TCP and UDP protocols belong to the Transport layer of the TCP/IP reference model. The key difference between the TCP and UDP is that TCP is a connection – oriented protocol, and UDP is a connectionless protocol.

The communication over TCP happens by implementing a "three – way handshake" between the server/host system and the client/receiver system. The UDP packet starts its communication process by gathering the data in a UDP packet and then adds the corresponding packet's header information. The packet contains all the necessary information like source & destination address, checksum, packet length. Later, the UDP packet is transmitted to its destination.

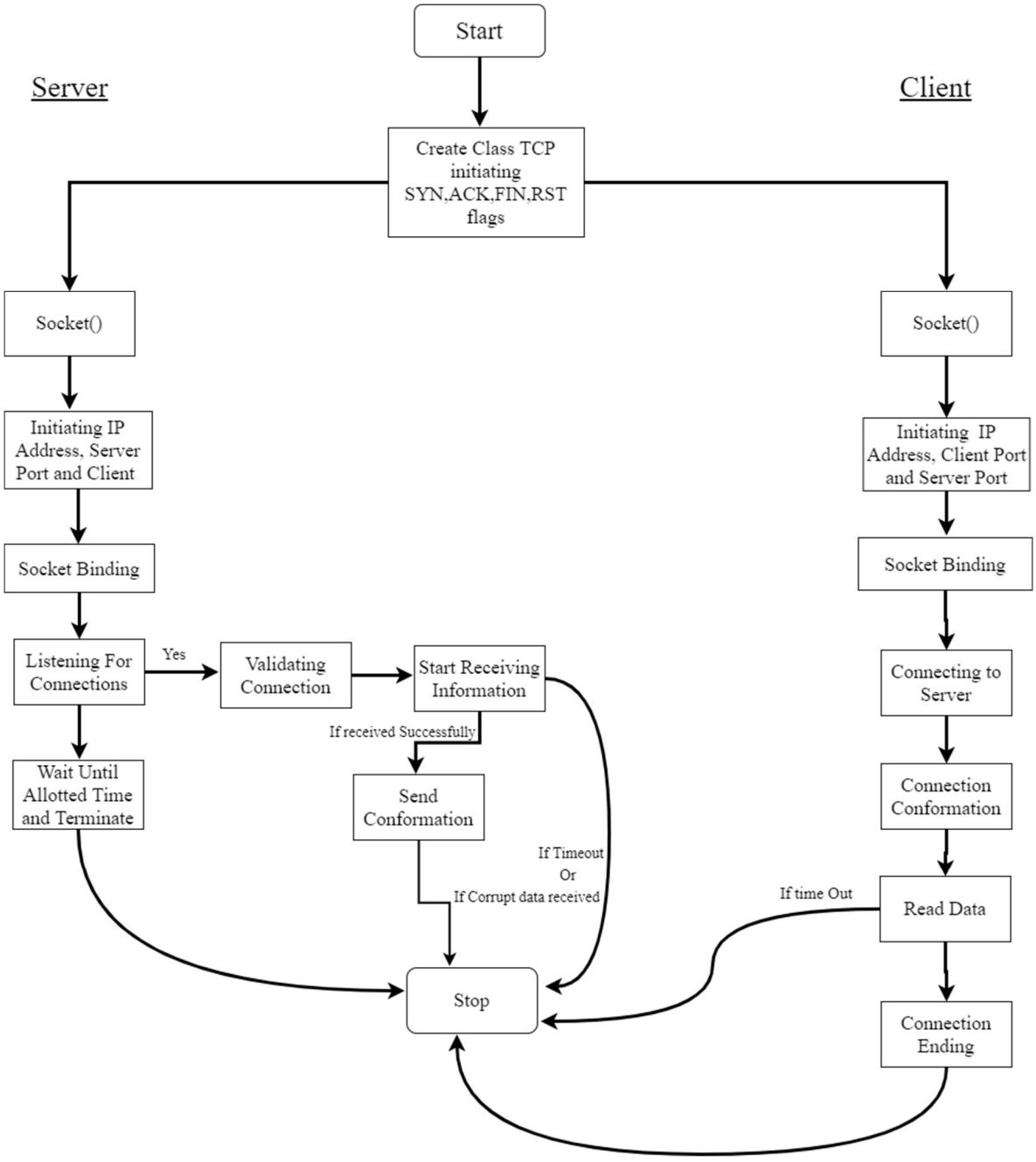


The aim of this project is to develop a model that performs TCP – like protocol within the UDP data.



The approach that has been adopted for developing this project was imitating a TCP behavior over UDP based sockets. During the implementation of the project the cost of overhead is minimal, as a result lightweight packets can be transmitted in quick sessions. The imitation of TCP - like protocol happens in UDP data part of the UDP packet.

Flow Chart:

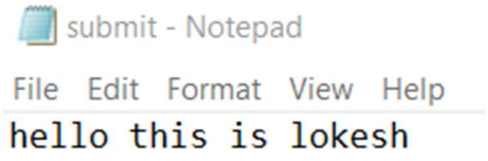


Implementation:

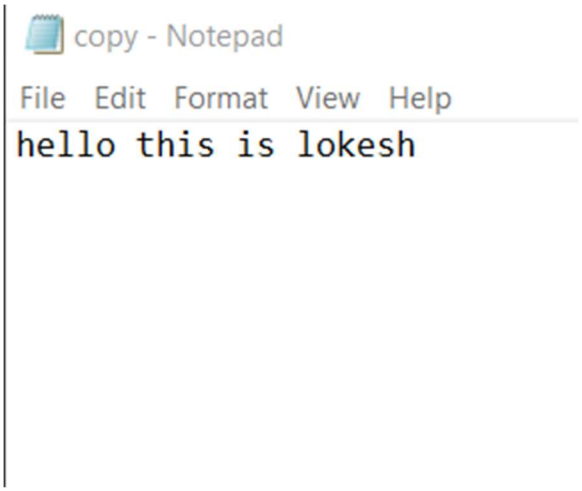
The aim of this project is to develop a TCP – like protocol within the UDP data. While implementing the project we develop a server - client model which basically transmits a UDP Packet. In the UDP Packet the TCP - like protocol is implemented. On the server side, the data/input is entered into a notepad. The data that is entered into the notepad on the server will be successfully transmitted to a separate/new notepad which is defined on the Client side.

Resultant Output:

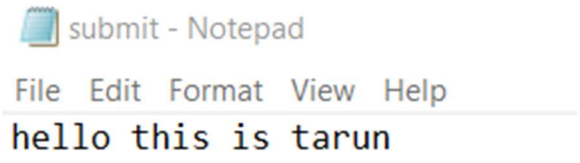
Input-1:



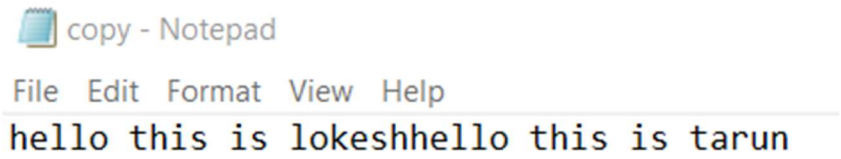
Output-1:



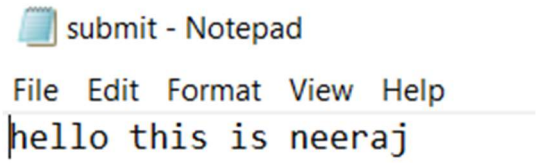
Input-2:



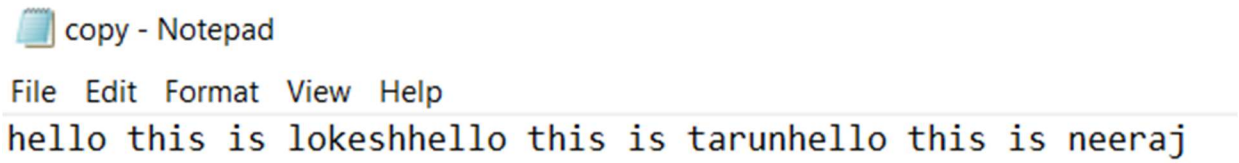
Output-2:



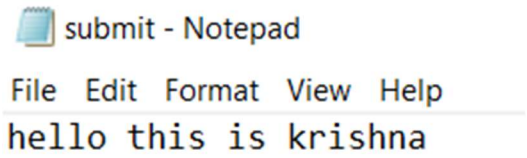
Input-3:



Output-3:



Input-4:



Output-4: