# Assignment - A5

## Problem Statement -

Write a program using UDP sockets to enable file transfers (script, text, audio, video) between two machines. Demonstrate the packets captured traces using wireshark packet analyzer tool using peer to peer method.
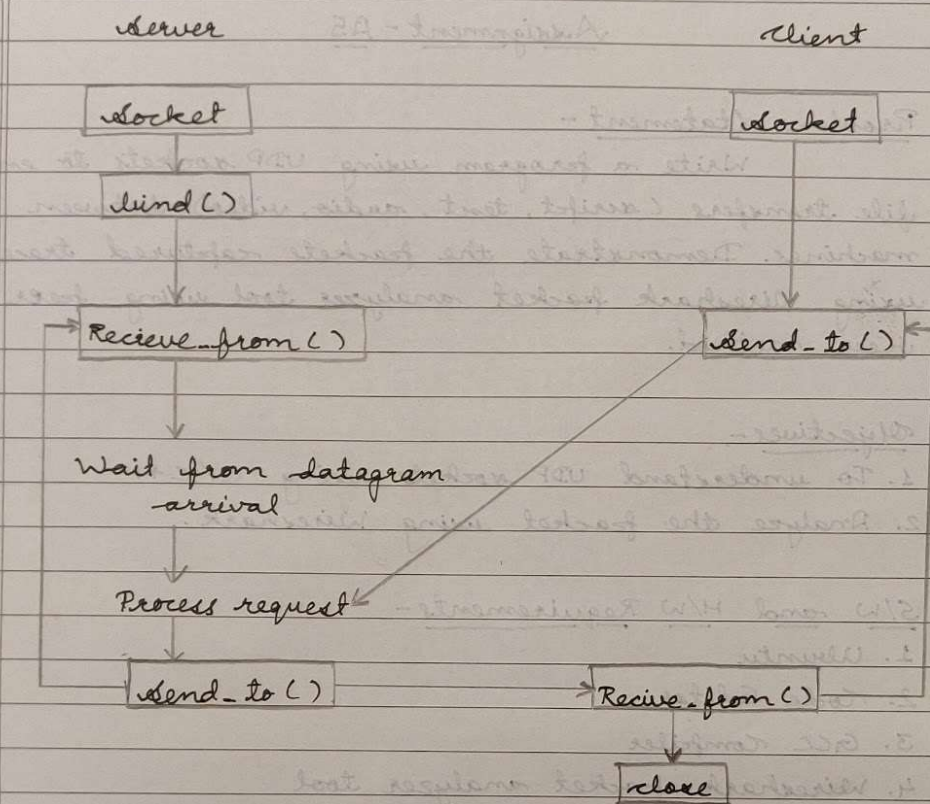
## Objectives -

1. To understand UDP sockets using c/c++
2. Analyze the packet using Wireshark.

## S/W and H/W Requirements -

1. Ubuntu
2. Code Editor
3. GCC Compiler
4. Wireshark packet analyzer tool

## Theory -

Datagram packets are use to implement a connectionless packet delivery service supported by to stream data. Each message is transferred from source machine i.e destination based in information contained within that packet need to have destination address and each packet might be routed differently and might arrive in any order. Packet delivery is not guaranteed.

Server                                          Client

[Socket]                                        [Socket]

[bind ()]

[Recieve-from ()]                    [Send-to ()]

Wait from datagram
        arrival

Process request

[Send-to ()]                [Recive-from ()]

                            [close]

UDP Server -

1. Create UDP Socket
2. Bind the socket to server address
3. Wait until datagram packet arrives from
   client.
4. Process the datagram packet and send a reply
   to client.
5. Go back to step 3.

UDP Client -
1. Create UDP socket
2. Send message to server.
3. Wait until response from server is recieved
4. Process reply and go back To step 2, if necessary
5. Close socket descriptor and exit.

Function :
1. uint socket ( int domain, int type, int protocol)
   DOMAIN - AF_INET or AL_INET6
   Type - SOCK_STREAM or SOCK_DGRAM
   Protocol - 0 means use default protocol

2. int bind ( int socketId, const struct sock-address)
   socketId - File descriptor of socket to be binded.
   sock_address - Structure in which address to bind
                  is specified
                  size of address structure

3. ssize_t sent-to ( int socket , const void *buf , size_t
                     len , int flags, const struct socket-addr,
                     * dest_addr , socketlen_t addrlen)
   len - size of buf application buffer
   flags - Bitwise OR of flags to modify socket
           behaviour.
   dest addr - Structure containing address of
               destination.
   addrlen - size of address structure.

4. ssize_t receive_From (int socketId, void * buf, size_t
                        len, int flags, struct sockaddr,
                        * src_addr, socklen_t * addrlen)
   src_addr - structure containing source address
   addrlen - variable in which address of source
             structure is returned.

Conclusion -
        From this assignment we were able to
implement / understand, the concept of UDP and
hence captures the packet traces using wireshark
Packet Analyzer.