Problem Statement:

Write a program using UDP sockets for wired network to implement

a. Peer to Peer Chat

b. Multiuser Chat

Theory:

- **Introduction:**

  The User Datagram Protocol (UDP) is called a connectionless, unreliable transport protocol. It does not add anything to the services of IP except to provide process-to-process communication instead of host-to-host communication.

- **UDP Features:**

1) Suitable for processes which require simple request response communication with little concern of error and flow control.
2) Suitable for process with internal error and flow control.
3) Suitable transport protocol for multicasting.
4) Used for management processes such as SNMP(Simple Network Management Protocol).
5) Used for some root updating protocol such as routing information protocol (RIP).

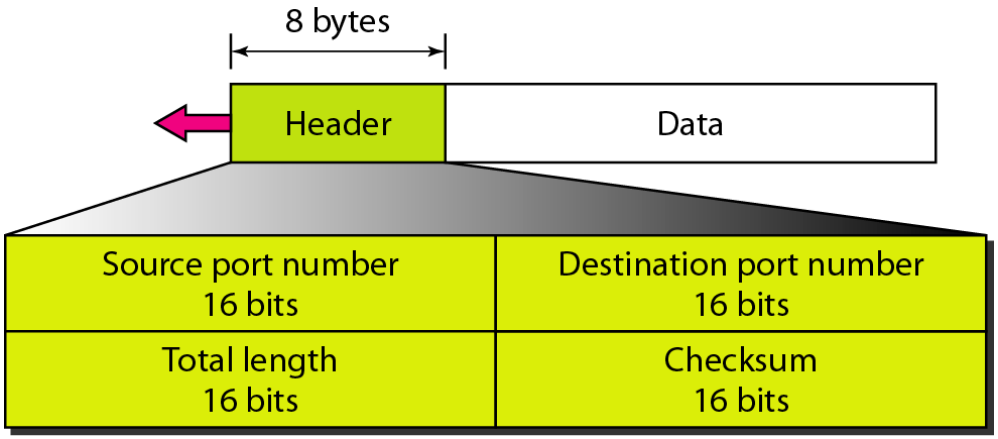

Figure 1 :User datagram format

UDP packets, called user datagrams, have a fixed-size header of 8 bytes. Figure 1shows the format of a user datagram.

The fields are as follows:

1) **Source port number.**

   This is the port number used by the process running on the source host. It is 16 bits long, which means that the port number can range from 0 to 65,535. If the source host is the client (a client sending a request), the port number, in most cases, is an ephemeral port number requested by the process and chosen by the UDP software running on the source host. If the source host is the server (a server sending a response), the port number, in most cases, is a well-known port number.

2) **Destination port number.**

   This is the port number used by the process running on the destination host. It is also 16 bits long. If the destination host is the server (a client sending a request), the port number, in most cases, is a well-known port number. If the destination host is the client (a server sending a response), the port number, in most cases, is an ephemeral port number. In this case, the server copies the ephemeral port number it has received in the request packet.

3) **Length.**

4) This is a 16-bit field that defines the total length of the user datagram, header plus data. The 16 bits can define a total length of 0 to 65,535 bytes. However, the total length needs to be much less because a UDP user datagram is stored in an IP datagram with a total length of 65,535 bytes.

5) **Checksum.**

   This field is used to detect errors over the entire user datagram (header plus data).

- **<u>Socket Programming:</u>**

  Datagram communication:

The datagram communication protocol, known as UDP (user datagram protocol), is a connectionless protocol, meaning that each time you send datagrams, you also need to send the local socket descriptor and the receiving socket's address. As you can tell, additional data must be sent each time a communication is made.

1) public DatagramSocket(int port)
throws SocketException

Constructs a datagram socket and binds it to the specified port on the local host machine. The socket will be bound to the wildcard address, an IP address chosen by the kernel. If there is a security manager, its checkListen method is first called with the port argument as its argument to ensure the operation is allowed. This could result in a SecurityException.

2) DatagramSocket(int port, InetAddress laddr)

Creates a datagram socket, bound to the specified local address.

3) public final class DatagramPacket

This class represents a datagram packet.Datagram packets are used to implement a connectionless packet delivery service. Each message is routed from one machine to another based solely on information contained within that packet. Multiple packets sent from one machine to another might be routed differently, and might arrive in any order. Packet delivery is not guaranteed.

4) DatagramPacket (byte[] buf, int length, InetAddress address, int port)

Constructs a datagram packet for sending packets of length length to the specified port number on the specified host.

5) bind(SocketAddress addr)

Binds this DatagramSocket to a specific address & port.

6) Void close()

Closes this datagram socket.

7) void connect(InetAddress address, int port)

Connects the socket to a remote address for this socket.

8) void connect(SocketAddress addr)

Connects this socket to a remote socket address (IP address + port number).

9) void disconnect()

Disconnects the socket.

10) Commonly used methods of InetAddress class

Method	Description
public static InetAddress getByName(String host) throws UnknownHostException	it returns the instance of InetAddress containing LocalHost IP and name.
public static InetAddress getLocalHost() throws UnknownHostException	it returns the instance of InetAdddress containing local host name and address.
public String getHostName()	it returns the host name of the IP address.
public String getHostAddress()	it returns the IP address in string format.

Conclusion: