

**BMS INSTITUTE OF TECHNOLOGY AND MANAGEMENT
YELAHANKA, BENGALURU - 560064**

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Report on UDP Sockets Programming

Name	Raghavendra K M
USN	1BY18IS093
Semester/Section	5B
Course Code	18CSL57
Course Name	Computer Network Laboratory
Faculty	Prof. Gireesh babu C N
Title	UDP Sockets Programming
Date	28-09-2020

Signature of a Student

Signature of a Faculty

Aim of the experiment:

Write a program on datagram socket for client/server to display the messages on client side, typed at the server side.

Socket programming

Socket programming is a way of connecting two nodes on a network to communicate with each other. One socket(node) listens on a particular port at an IP, while other socket reaches out to the other to form a connection. Server forms the listener socket while client reaches out to the server.

Datagram Socket

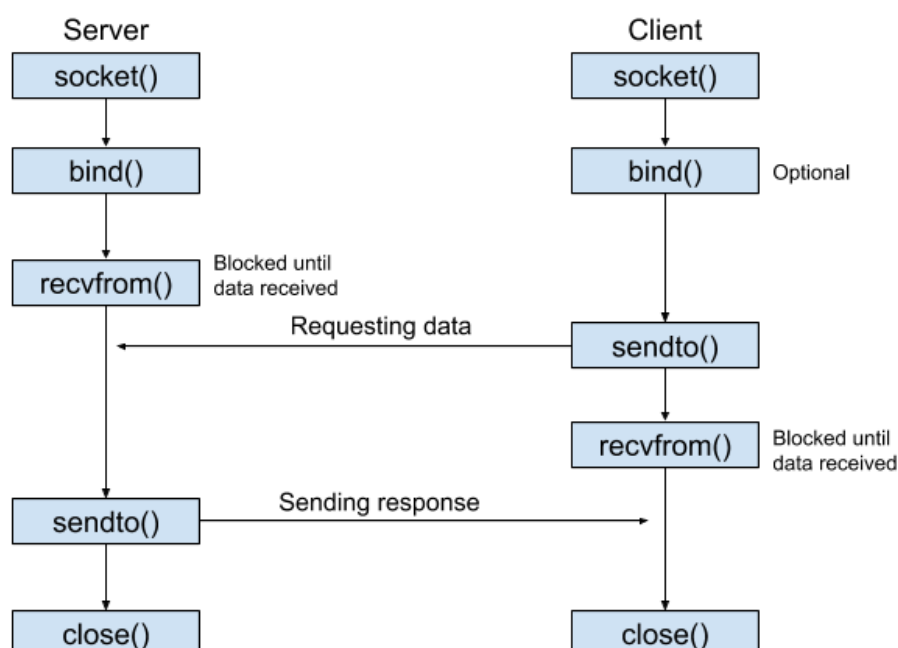
In computer operating systems, a datagram socket is a type of interprocess communication sockets or network sockets which provides a connectionless point for sending or receiving data packets.

Each packet sent or received on a datagram socket is individually addressed and routed. Order and reliability are not guaranteed with datagram sockets, so multiple packets sent from one machine or process to another may arrive in any order or might not arrive at all.

The sending of UDP (User DataGram Protocol) broadcasts on a network are always enabled on a datagram socket.

Working

The UDP socket communication between a server and a client consists of several phases as follows.



- i. **socket()** - Firstly a socket is defined in both server and client. This need not happen at the same time. For the explanation to be comprehensive, I will be discussing the actions of both server and client in each stage.
- ii. **bind()** - Defined socket is assigned an id and a port in the running machine. This is optional for the client socket. This is because even if the client socket is not bound, it will automatically happen whenever the client initiates connecting to the server.
- iii. **recvfrom()** - After binding to a port in the machine, the server socket waits for a connection from a client socket. In the meantime, further execution of the current thread is halt (blocked) until the server socket receives a connection. This is the same for the client socket when waiting for a server response.
- iv. **sendto()** - After connecting with a client, the server socket sends data to the client. This same method is used by the client socket to make a connection request to the server.
- v. **close()** - After successful data exchange, both sockets are closed i.e. system resources allocated for the sockets are released.

Source code:

UDPClient.java

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;

public class UDPClient {
    public static void main(String[] args) throws Exception {
        BufferedReader inFromUser = new BufferedReader
            (new InputStreamReader(System.in));

        DatagramSocket clientSocket = new DatagramSocket();
        InetAddress IPAddress = InetAddress.getByName("localhost");
        byte[] sendData = new byte[1024];
        byte[] receiveData = new byte[1024];
        System.out.println("Enter 'START' to connect to Server");
        String sentence = inFromUser.readLine();
        sendData = sentence.getBytes();
        DatagramPacket sendPacket = new DatagramPacket
            (sendData, sendData.length, IPAddress, 9876);
        clientSocket.send(sendPacket);
        DatagramPacket receivePacket = new DatagramPacket
            (receiveData, receiveData.length);
        clientSocket.receive(receivePacket);
        String modifiedSentence = new String(receivePacket.getData());
        System.out.println("Message received from Server: " + modifiedSentence);
        clientSocket.close();
    }
}
```

UDPServer.java

```
import java.net.*;
import java.util.Scanner;

public class UDPServer {
    public static void main(String[] args) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        System.out.println("Server Started on Port 9876");
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while (true) {
            DatagramPacket receivePacket = new DatagramPacket
                (receiveData, receiveData.length);
            serverSocket.receive(receivePacket);
            receivePacket.getData();
            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();
            System.out.println("Client Connected");
            Scanner input = new Scanner(System.in);
            System.out.println("Enter the message to be sent: ");
            String message = input.nextLine();
            sendData = message.getBytes();
            DatagramPacket sendPacket = new DatagramPacket
                (sendData, sendData.length, IPAddress, port);
            serverSocket.send(sendPacket);
            System.exit(0);
        }
    }
}
```

Outputs:

UDP Server

```
C:\Windows\System32\cmd.exe
(c) 2019 Microsoft Corporation. All rights reserved.

I:\Documents (C)\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\Assignment\UDP Socket\Server>javac UDPServer.java

I:\Documents (C)\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\Assignment\UDP Socket\Server>java UDPServer
Server Started on Port 9876
Client Connected
Enter the message to be sent:
Welcome to UDP SOCKET PROGRAM- Udp

I:\Documents (C)\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\Assignment\UDP Socket\Server>_
```

UDP Client

```
C:\Windows\System32\cmd.exe
(c) 2019 Microsoft Corporation. All rights reserved.

I:\Documents (C)\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\Assignment\UDP Socket\Client>javac UDPCClient.java

I:\Documents (C)\1by18is093\ISE V SEM\Subjects\18CSL57 - Computer Network Laboratory\Assignment\UDP Socket\Client>java UDPCClient
Enter 'START' to connect to Server
START
Message received from Server: Welcome to UDP SOCKET PROGRAM- Udp
```