**Practical 5**

**Aim:**

(Optional) Write a java program using java.net library. Client will send a time request to the server and server will reply with its current timestamp. Client will update its own time using timestamp of a server. (Time-Server using UDP)

**Description:**

****

Fig. 1. Concurrent Client Server scenario

**Program code:**

**EchoClientUDP.java**

//UDP Client

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

public class EchoClientUDP **{**

public static void main**(**String args**[])** **{**

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**();**

String requestTime **=** "Requesting time."**;**

byte**[]** b **=** requestTime**.**getBytes**();**

InetAddress host **=** InetAddress**.**getByName**(**"localhost"**);**

System**.**out**.**println **(**"Attemping to connect to host " **+**

host**.**getAddress**()** **+** " on port 10007."**);**

DatagramPacket dataPacket **=** **new** DatagramPacket**(**b**,** b**.**length**,** host**,** 10007**);**

datagramSocket**.**send**(**dataPacket**);**

System**.**out**.**println**(**"Client: "**+**requestTime**);**

byte**[]** buffer **=** **new** byte**[**100**];**

DatagramPacket reply **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**reply**);**

String word **=** **new** String**(**reply**.**getData**()).**trim**();**

System**.**out**.**println**(**"Server: "**+**word**);**

**}** **catch** **(**UnknownHostException e**)** **{**

System**.**err**.**println**(**"Don't know about host: localhost"**);**

System**.**exit**(**1**);**

**}** **catch** **(**IOException e**)** **{**

System**.**err**.**println**(**"Couldn't get I/O for "

**+** "the connection to: localhost"**);**

e**.**printStackTrace**();**

System**.**exit**(**1**);**

**}**

datagramSocket**.**close**();**

**}**

**}**

**EchoServerUDP.java**

//UDP Server

**import** java**.**net**.\*;**

**import** java**.**io**.\*;**

public class EchoServerUDP **{**

public static void main**(**String args**[])** **{**

DatagramSocket datagramSocket **=** **null;**

**try** **{**

datagramSocket **=** **new** DatagramSocket**(**10007**);**

System**.**out**.**println **(**"Waiting for connection....."**);**

**while(true)** **{**

byte **[]** buffer **=** **new** byte**[**100**];**

DatagramPacket request **=** **new** DatagramPacket**(**buffer**,** buffer**.**length**);**

datagramSocket**.**receive**(**request**);**

System**.**out**.**println**(**"Enroll: 130050130167"**);**

System**.**out**.**println **(**"Connection successful"**);**

System**.**out**.**println **(**"Waiting for input....."**);**

String arrayMsg **=** **new** String**(**request**.**getData**()).**trim**();**

System**.**out**.**println**(**"From: " **+** request**.**getAddress**()** **+** " Port: " **+** request**.**getPort**());**

System**.**out**.**println**(**"Client: "**+**arrayMsg**);**

String currentTimeStamp **=** **new** GetTimeStamp**().**get**();**

byte**[]** sendMsg **=** currentTimeStamp**.**getBytes**();**

DatagramPacket reply **=** **new** DatagramPacket**(**sendMsg**,** sendMsg**.**length**,** request**.**getAddress**(),** request**.**getPort**());**

datagramSocket**.**send**(**reply**);**

String word **=** **new** String**(**reply**.**getData**());**

System**.**out**.**println**(**"Server: "**+**word**);**

System**.**out**.**println **(**"Waiting for connection....."**);**

**}**

**}catch(**Exception ex**)** **{**

**}**

datagramSocket**.**close**();**

**}**

**}**

**Input Output:**

