## Lab Assignment – 7

### **Name:** Srujan Patwardhan

### **STD:** TY-D

### **Roll no.:** 02

### **PRN:** 12210847

### **Subject:** Computer Networks

#### Q1. Write a program using UDP Berkeley Sockets for wired/wireless network to enable file transfer (Script, Text, Audio and Video one file each) between two machines. Demonstrate the packets captured traces using Wireshark/Fiddler for traffic analysis tool in peer-to-peer mode.

##### CODE:

##### UDPServer.java:

import *java.io.\**;

import *java.net.\**;

*public* *class* UDPServer {

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

        DatagramSocket socket = null;

        try {

            socket = new DatagramSocket(9876);

            byte[] receiveData = new byte[1024];

            DatagramPacket receivePacket = new DatagramPacket(receiveData, receiveData.length);

            socket.receive(receivePacket);

            String fileName = new String(receivePacket.getData(), 0, receivePacket.getLength());

            System.out.println("Receiving file: " + fileName);

            FileOutputStream fos = new FileOutputStream(fileName);

            BufferedOutputStream bos = new BufferedOutputStream(fos);

            boolean receiving = true;

            while (receiving) {

                receivePacket = new DatagramPacket(receiveData, receiveData.length);

                socket.receive(receivePacket);

                String receivedString = new String(receivePacket.getData(), 0, receivePacket.getLength());

                if (receivedString.equals("END")) {

                    System.out.println("File transfer completed.");

                    receiving = false;

                } else {

                    bos.write(receivePacket.getData(), 0, receivePacket.getLength());

                }

            }

            bos.close();

        } catch (IOException e) {

            e.printStackTrace();

        } finally {

            if (socket != null && !socket.isClosed()) {

                socket.close();

            }

        }

    }

}

##### UDPClient.java:

import *java.io.\**;

import *java.net.\**;

*public* *class* UDPClient {

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

        DatagramSocket socket = null;

        try {

            socket = new DatagramSocket();

            InetAddress IPAddress = InetAddress.getByName("localhost");

            byte[] sendData = new byte[1024];

            String fileName = "E://TY/CN/CNLab/src/main/java/org/example/assignment7/file1.txt";

            System.out.println("Sending file: " + fileName);

            sendData = fileName.getBytes();

            DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

            socket.send(sendPacket);

            FileInputStream fis = new FileInputStream(fileName);

            BufferedInputStream bis = new BufferedInputStream(fis);

            int bytesRead;

            while ((bytesRead = bis.read(sendData)) != -1) {

                sendPacket = new DatagramPacket(sendData, bytesRead, IPAddress, 9876);

                socket.send(sendPacket);

            }

            String endMessage = "END";

            sendData = endMessage.getBytes();

            DatagramPacket endPacket = new DatagramPacket(sendData, sendData.length, IPAddress, 9876);

            socket.send(endPacket);

            System.out.println("File sent successfully.");

            bis.close();

        } catch (IOException e) {

            e.printStackTrace();

        } finally {

            if (socket != null && !socket.isClosed()) {

                socket.close();

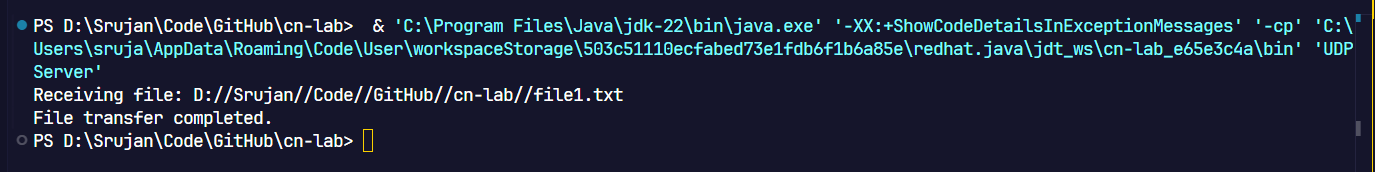
            }

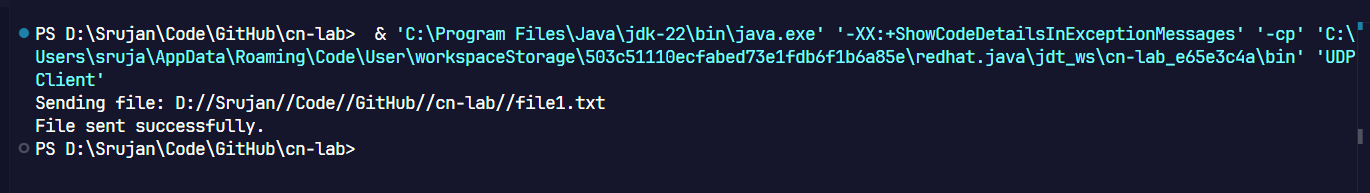
        }

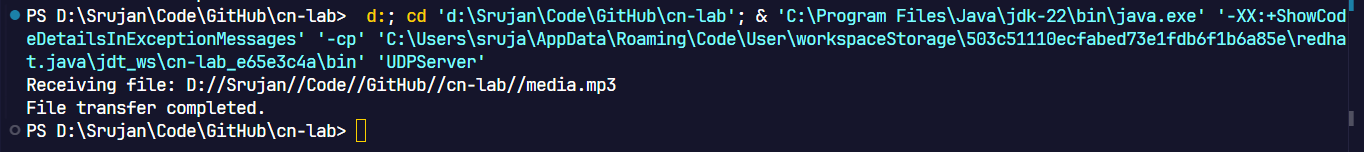
    }

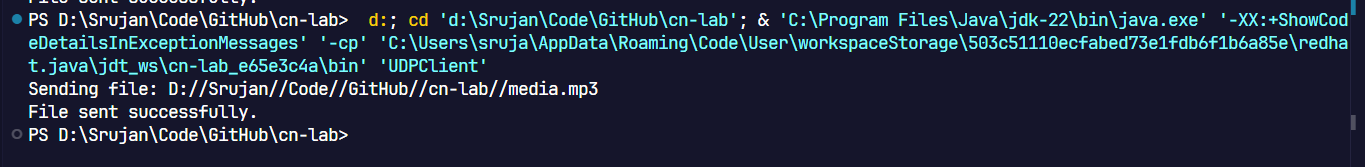
}

##### OUTPUT:

Fig. 1. Text File received by server

Fig. 2. Text File sent by client

Fig. 3. Audio File received by server

Fig. 4. Audio File sent by client

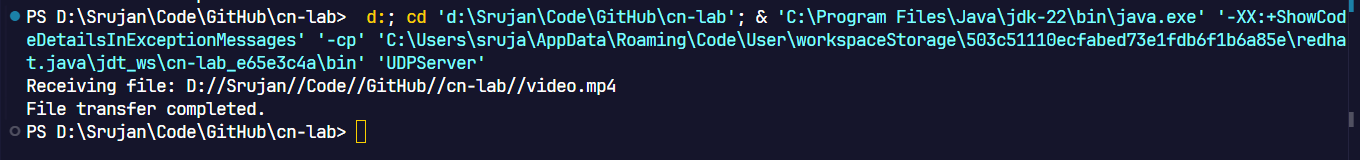
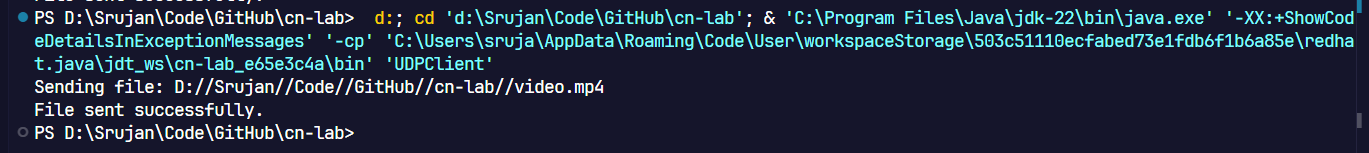
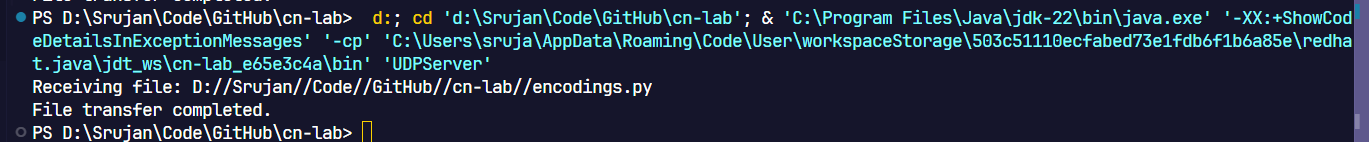
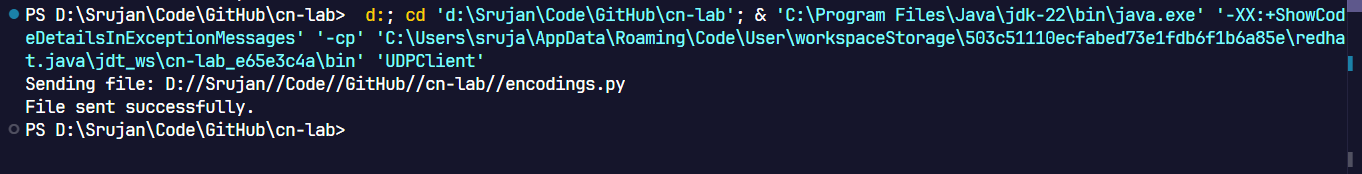


Fig. 5. Video file received by the server

Fig. 6. Video file sent by the client

Fig. 7. Script Received by Server

Fig. 8. Script sent by Client

## Wireshark Capture:

