**CN LAB#05**

**Name: Saman Khan**

**ID: 19K-0354**

**Section: H**

1. **Socket Programming:**

**Question#01:**

* **Client Code:**

class client {

public static void main(String args[]) throws IOException {

Socket s = new Socket(InetAddress.getLocalHost(), 8000);

PrintStream dos = new PrintStream(s.getOutputStream);

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));

while (true) {

String str = in.readLine();

System.out.println(str);

if (str.equals("end")) {

s.close(); break;

}

System.out.println("enter message to send: ");

str = stdin.readLine();

if (str.equals("end")) {

s.close();

break;

}

dos.println(str);

}

}

}

* **Server Code:**

class server {

public static void main(String a[]) throws IOException {

ServerSocket ss = new ServerSocket(8000);

Socket s = ss.accept();

BufferedReader in = new BufferedReader(new InputStreamReader(s.getInputStream()));

BufferedReader stdin = new BufferedReader(new InputStreamReader(System.in));

PrintStream dos = new PrintStream(s.getOutputStream());

while (true) {

System.out.println("enter message to send:");

String str = stdin.readLine();

if (str.equals("end")) {

s.close();

break;

}

dos.println(str);

str = in.readLine();

System.out.println(str);

if (str.equals("end")) {

s.close();

break;

}

}

}

}

**Question#02:**

* **Client Code:**

class Client {

DatagramSocket ds, dr;

DatagramPacket dps;

DatagramPacket dpr;

byte buff[] = new byte[1024];

String str, str1, str2;

Boolean i = true;

Sender() throws SocketException {

ds = new DatagramSocket();

dr = new DatagramSocket(8000);

}

public void run() throws IOException {

while (i) {

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

System.out.printf("enter the message: ");

str = in.readLine();

buff = str.getBytes();

dps = new DatagramPacket(buff, buff.length, InetAddress.getLocalHost(), 8001);

ds.send(dps);

dpr = new DatagramPacket(buff, buff.length);

dr.receive(dpr);

str = new String(dpr.getData());

System.out.printf(str);

System.out.printf("Continue? yes/no: ");

str1 = in.readLine();

if (str1.equals("yes")) {

i = true;

}

else {

i = false;

}

}

}

public static void main(String args[]) throws IOException {

Client c = new Client();

c.run();

}

}

* **Server Code:**

class Server {

DatagramSocket ds, dr;

DatagramPacket dps;

DatagramPacket dpr;

byte buff[] = new byte[1024];

String str, str1, str2;

Boolean i = true;

Receiver() throws SocketException {

ds = new DatagramSocket();

dr = new DatagramSocket(8001);

}

public void send() throws IOException {

while (i) {

dpr = new DatagramPacket(buff, buff.length); dr.receive(dpr);

str = new String(dpr.getData());

System.out.printf(str);

BufferedReader in = new BufferedReader(new InputStreamReader(System.in));

System.out.println("enter the message: ");

str = in.readLine();

buff = str.getBytes();

dps = new DatagramPacket(buff, buff.length, InetAddress.getLocalHost(), 8000);

ds.send(dps);

System.out.println("continue? yes/no: ");

str1 = in.readLine();

if (str1.equals("yes")) {

i = true;

}

else {

i = false;

}

}

}

public static void main(String args[]) throws IOException {

Server s = new Server();

s.send();

}

}

1. **TCP:**

**Question#01:**



**IP address:** 172.16.72.241

**TCP port number:** 52607

**Question#02:**

**IP address:** 182.176.156.40

**TCP port number:** 80

**Question#03:**

**IP address:** 172.16.72.241

**TCP port number:** 52607

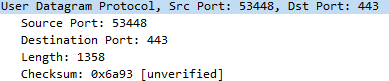
**Question#04:**



Seq = 0 SYN = 1

1. **UDP:**

**Question#01:**

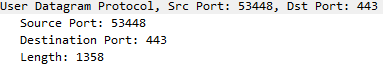


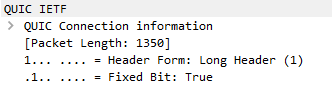
UDP header has the following 4 fields:

* Source Port
* Destination Port
* Length
* Checksum

**Question#02:**







**Question#03:**

The length field specifies the number of bytes in the UDP segment (header plus data). An explicit length value is needed since the size of the data field may differ from one UDP segment to the next. The length of UDP payload for selected packet is 33 bytes i.e (41 bytes - 8 bytes = 33 bytes).



**Question#04:**

The maximum number of bytes that can be included in a UDP payload is (2^16 – 1) bytes plus the header bytes. This gives 65535 bytes – 8 bytes = 65527 bytes.