21BDS0340

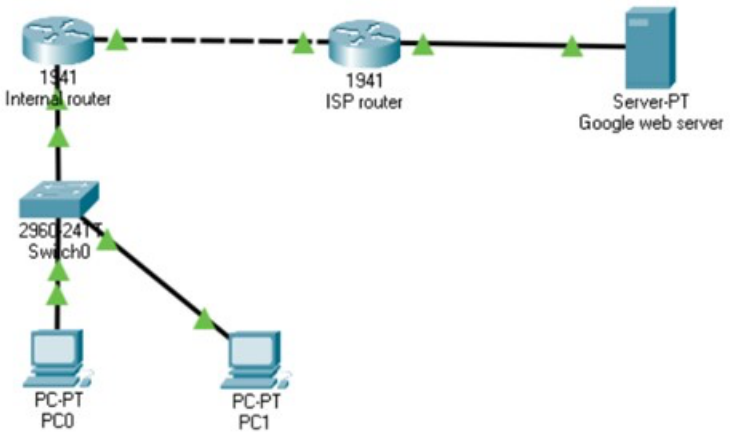
Abhinav Dinesh Srivatsa

Computer Networks Lab

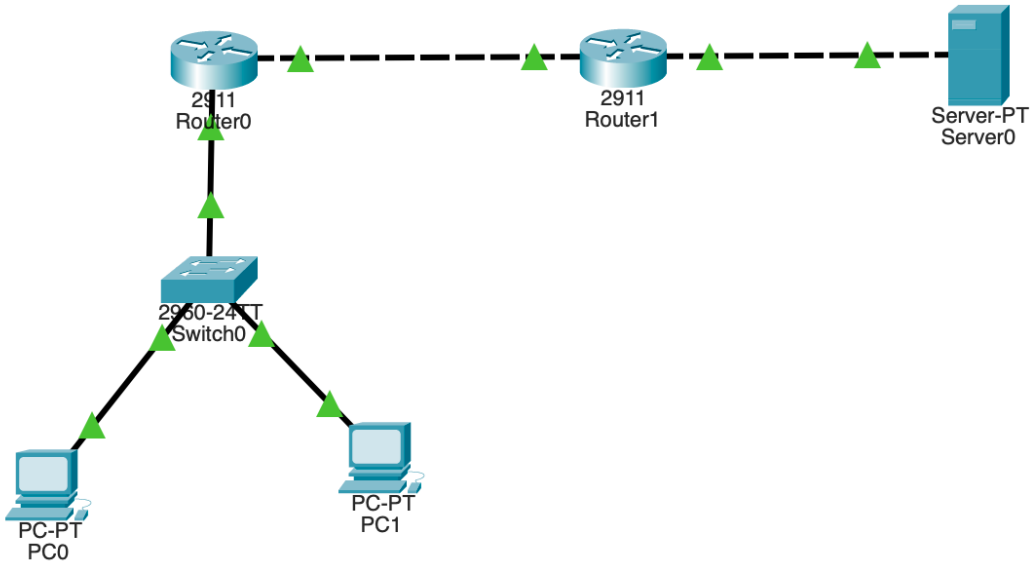
Lab FAT

Question 1

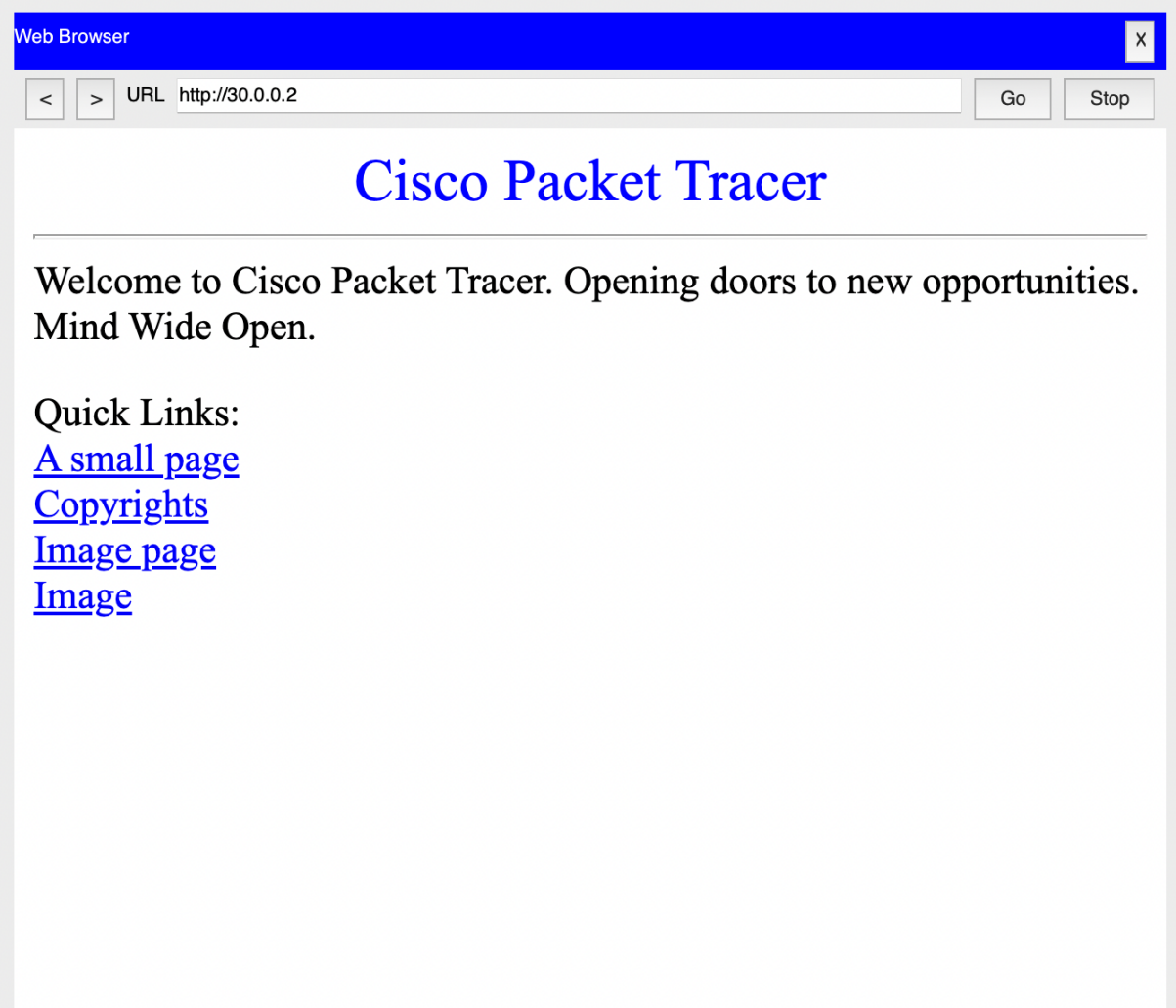
Aim:
Create a webserver similar to given topology. End host machines must be able to access a test HTML page on server using HTTP. Check connectivity between all links and record the output.



My Topology:



Testing HTML Webpage Access:



Result:

The simulation is working as expected, the webpage is available to access. The PCs are at 10.0.0.2 and 10.0.0.3 and the server is at 30.0.0.2. Putting the address 30.0.0.2 in the PCs, show that the connection is working as expected.

Question 2

Aim:
Develop a program for the sender and receiver windows in a network system using Sliding window protocol, where a 3-frame field is used and given the following events. Print the output sequence of frames in transmission:
(a) Frame 0 is sent; Frame 0 is acknowledged.
(b) Frames 1 and 2 are sent; Frames 1 and 2 are acknowledged.
(c) Frames 3, 4, and 5 are sent; Frames 3 and 4 is acknowledged; Timer for Frame 4 expired/ Acknowledgement is lost.
(d) Frames 4, 5, and 6 are sent; Frames 4 through 7 are acknowledged

Output: SenderQ2.java

```
(base) abhi@Abhinavs-MacBook-Pro Lab FAT % java SenderQ2
Waiting for connection...
Received acknowledgement.

Sending packet 0
Sending packet 1
Sending packet 2
Sending packet 3
Sending packet 4 ...packet lost
Sending packet 4
Sending packet 5
Sending packet 6
Sending packet 7

Ending connection.
```

Output: ReceiverQ2.java

```
(base) abhi@Abhinavs-MacBook-Pro Lab FAT % java ReceiverQ2
Initiating connection...
Sending acknowledgement.

Received packet... Message: packet0
Sending acknowledgement.
Received packet... Message: packet1
Sending acknowledgement.
Received packet... Message: packet2
Sending acknowledgement.
Received packet... Message: packet3
Sending acknowledgement.
Received packet... Message: packet4
Sending acknowledgement.
Received packet... Message: packet5
Sending acknowledgement.
Received packet... Message: packet6
Sending acknowledgement.
Received packet... Message: packet7
Sending acknowledgement.
```

Code: SenderQ2.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.util.Arrays;

public class SenderQ2 {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public SenderQ2(int port) throws IOException {
        this.port = port;
        this.socket = new DatagramSocket(port);
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsInt[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsInt[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsInt;
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        System.out.println("Received acknowledgement.");

        this.address = packet.getAddress();
        this.port = packet.getPort();
        return stringToCharArray(Arrays.toString(data));
    }

    private void sendPacket(String dataString) throws IOException {
        byte dataByte[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(dataByte, dataByte.length,
                this.address, this.port);
```

```
        socket.send(packet);
    }

    public static void main(String[] args) throws IOException {
        int port = 5000;
        SenderQ2 sq2 = new SenderQ2(port);
        System.out.println("Waiting for connection... ");
        sq2.receivePacket();
        System.out.println();

        System.out.println("Sending packet 0");
        sq2.sendPacket("0packet0");
        System.out.println("Sending packet 1");
        sq2.sendPacket("1packet1");
        System.out.println("Sending packet 2");
        sq2.sendPacket("2packet2");
        System.out.println("Sending packet 3");
        sq2.sendPacket("3packet3");
        System.out.println("Sending packet 4 ...packet lost");
        System.out.println("Sending packet 4");
        sq2.sendPacket("4packet4");
        System.out.println("Sending packet 5");
        sq2.sendPacket("5packet5");
        System.out.println("Sending packet 6");
        sq2.sendPacket("6packet6");
        System.out.println("Sending packet 7");
        sq2.sendPacket("7packet7");
        System.out.println("\nEnding connection.");
    }
}
```

Code: ReceiverQ2.java

```
import java.io.IOException;
import java.io.UnsupportedEncodingException;
import java.net.DatagramPacket;
import java.net.DatagramSocket;
import java.net.InetAddress;
import java.net.SocketException;
import java.util.Arrays;

public class ReceiverQ2 {
    private static final int PACKET_SIZE = 1024;

    private InetAddress address;
    private DatagramSocket socket;
    private int port;

    public ReceiverQ2(InetAddress address, int port) throws SocketException {
        this.port = port;
        this.address = address;
        socket = new DatagramSocket();
```

```
    }

    private byte[] stringToByteArray(String dataString) throws
UnsupportedEncodingException {
        return dataString.getBytes("UTF-8");
    }

    private char[] stringToCharArray(String dataString) {
        String middle = dataString.substring(1, dataString.length() - 1);
        String elementsString[] = middle.split(", ");
        char elementsChar[] = new char[elementsString.length];

        for (int x = 0; x < elementsString.length; x++)
            elementsChar[x] = (char) Integer.parseInt(elementsString[x]);

        return elementsChar;
    }

    public void sendPacket(String dataString) throws IOException {
        byte data[] = stringToByteArray(dataString);
        DatagramPacket packet = new DatagramPacket(data, data.length, address,
port);

        System.out.println("Sending acknowledgement.");
        socket.send(packet);
    }

    public char[] receivePacket() throws IOException {
        byte data[] = new byte[PACKET_SIZE];
        DatagramPacket packet = new DatagramPacket(data, data.length);
        socket.receive(packet);
        return stringToCharArray(Arrays.toString(data));
    }

    public String[] receiveTransmission() throws IOException {
        String messages[] = new String[10];

        for (int x = 1; x < messages.length; x++) {
            char data[] = receivePacket();
            System.out.print("Received packet... ");

            if (data[0] == '\0')
                break;

            messages[x] = new String(data);
            String index = messages[x].substring(0, 1);
            messages[x] = messages[x].substring(1, messages[x].length() - 1);
            System.out.println("Message: " + messages[x]);

            sendPacket(index);
        }
```

```
        System.out.println("Destroyed connection.\n");

        return messages;
    }

    public static void main(String[] args) throws IOException {
        InetAddress address = InetAddress.getByName("localhost");
        int port = 5000;
        ReceiverQ2 rq2 = new ReceiverQ2(address, port);
        System.out.println("Initiating connection...");
        rq2.sendPacket("initiate");
        System.out.println();

        String messages[] = rq2.receiveTransmission();
        for (int x = 1; x < messages.length; x++)
            if (messages[x] != null)
                System.out.println("Message: " + messages[x]);
    }
}
```

Result:

The sliding window protocol with a window size of 4 has been implemented in code with a sender and receiver. The scenario mentioned in the question has been implemented and the results are matching to the expected output.