Ex. No : 2a	Implementation of the Stop-and-Wait Protocol
Date :	

Aim :-

To implement and execute stop and wait protocol using java.

Algorithm :-

Sender :

Step 1 : Start the program.

Step 2 : Set Sequence = 0

Step 3 : Accept new packet and assign a sequence to it.

Step 4 : Send packet sequence with sequence number.

Step 5 : Set timer for recently sent packets.

Step 6 : If error free acknowledgment from receiver and Next Frame Expected -> sequence then sequence -> Next Frame Expected.

Step 7 : If time out then go to step 3. Step 8 : Stop the program.

Receiver :

Step 1 : Start the program.

Step 2 : Next frame expected = 0, repeat step 3 forever.

Step 3 : If error free frame received and sequence -> Next Frame Expected, the pass packet to higher layer and Next Frame Expected -> Next Frame Expected +1 (modulo 2).

Step 4 : Stop the program.

Program :-

Sender.java import

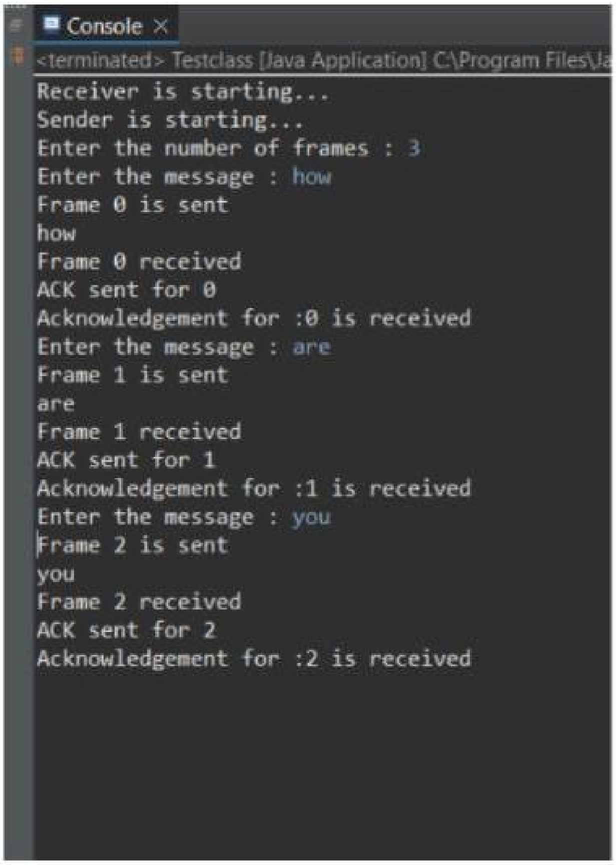
```
java.io.*; import
java.net.*;
public class Sender {
public static void main(String[] args)
{ String serverAddress = "localhost";
int serverPort = 9876;
int timeout = 5000; // Increased timeout
try (DatagramSocket socket = new DatagramSocket()) {
InetAddress serverInetAddress = InetAddress.getByName(serverAddress);
String[] messages = {"Message 1", "Message 2", "Message 3"};
for (String message : messages) { byte[]
sendData = message.getBytes();
```

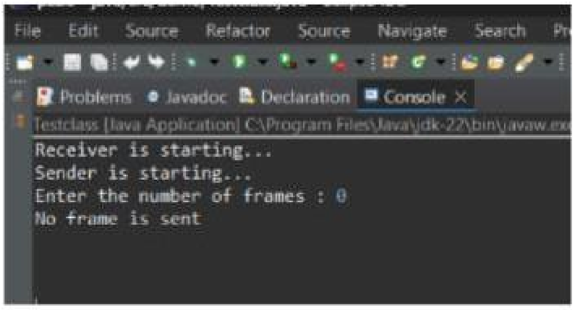
```
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
serverInetAddress, serverPort);
socket.send(sendPacket);
System.out.println("Sent: " + message);
socket.setSoTimeout(timeout);
boolean ackReceived = false;
long startTime = System.currentTimeMillis();
while (System.currentTimeMillis() - startTime < timeout) { try
{
byte[] receiveData = new byte[1024];
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
socket.receive(receivePacket);
String ackMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
if (ackMessage.equals("ACK"))
{ ackReceived = true;
System.out.println("Received ACK for: " + message);
break;
}
} catch (SocketTimeoutException e) { System.out.println("Timeout
expired, resending: " + message); socket.send(sendPacket); // Resend
if no ACK is received
}
}
if (!ackReceived) {
System.out.println("No ACK received after timeout, retrying...");
socket.send(sendPacket);
}
}
} catch (Exception e)
{ e.printStackTrace();
}
}
}
```

Receiver.java import

```
java.io.*; import
java.net.*;
public class Receiver {
public static void main(String[] args) { int
serverPort = 9876;
try (DatagramSocket socket = new DatagramSocket(serverPort))
{ byte[] receiveData = new byte[1024];
while (true) {
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
socket.receive(receivePacket);
String receivedMessage = new String(receivePacket.getData(), 0,
receivePacket.getLength());
System.out.println("Received: " + receivedMessage);
Thread.sleep(500);
InetAddress senderAddress = receivePacket.getAddress(); int
senderPort = receivePacket.getPort();
String ackMessage = "ACK";
byte[] sendData = ackMessage.getBytes();
DatagramPacket sendPacket = new DatagramPacket(sendData, sendData.length,
senderAddress, senderPort);
socket.send(sendPacket);
System.out.println("Sent ACK for: " + receivedMessage);
}
} catch (Exception e)
{ e.printStackTrace();
}
}
}
```

Output :-





Result :-

Thus, the program for stop and wait protocol was implemented and executed successfully.

Ex. No : 2b	Implementation of the Sliding Window Protocol
Date :	

Aim :-

To implement and execute sliding window protocol using java.

Algorithm :-

Sender :

Step 1 : Start the program.

Step 2 : Implement .net and other necessary packages.

Step 3 : Declare objects for input/output and socket to receive the frame and send acknowledgment.

Step 4 : Wait for the client to establish connection.

Step 5 : Receive the frame one by one from the socket and return the frame number as acknowledgment within the thread sleep time.

Step 6 : Send acknowledgment for each receiving frame and continue the process until acknowledgment for all frames are sent.

Step 7 : Receive the acknowledgment of last frame. Step

8 : Stop the program.

Receiver :

Step 1 : Start the program.

Step 2 : Import .net and other necessary packages.

Step 3 : Create objects for server socket, socket to send the frames and display the server address when the server is connected.

Step 4 : Get the number of frames to be sent, from the user.

Step 5 : Send the first frame to server using the socket.

Step 6 : When one frame is sent, wait for the acknowledgment from the receiver for the previous frame.

Programs :-

Sender.java

```
import java.util.LinkedList;
import java.util.Queue; public
class Sender { private int
windowSize;
private Queue<Integer> window; private
int totalPackets;
private int nextSeqNum;
```

```
public Sender(int windowSize, int totalPackets)
{ this.windowSize = windowSize;
this.totalPackets = totalPackets;
this.window = new LinkedList<>();
this.nextSeqNum = 0;
}
public void startTransmission(Receiver receiver) {
while (nextSeqNum < totalPackets || !window.isEmpty()) {
while (window.size() < windowSize && nextSeqNum < totalPackets)
{ System.out.println("Sender: Sending packet with sequence number " + nextSeqNum);
window.add(nextSeqNum);
receiver.receivePacket(nextSeqNum);
nextSeqNum++;
}
receiver.acknowledge(window);
window.clear();
}
}
}
```

Receiver1.java

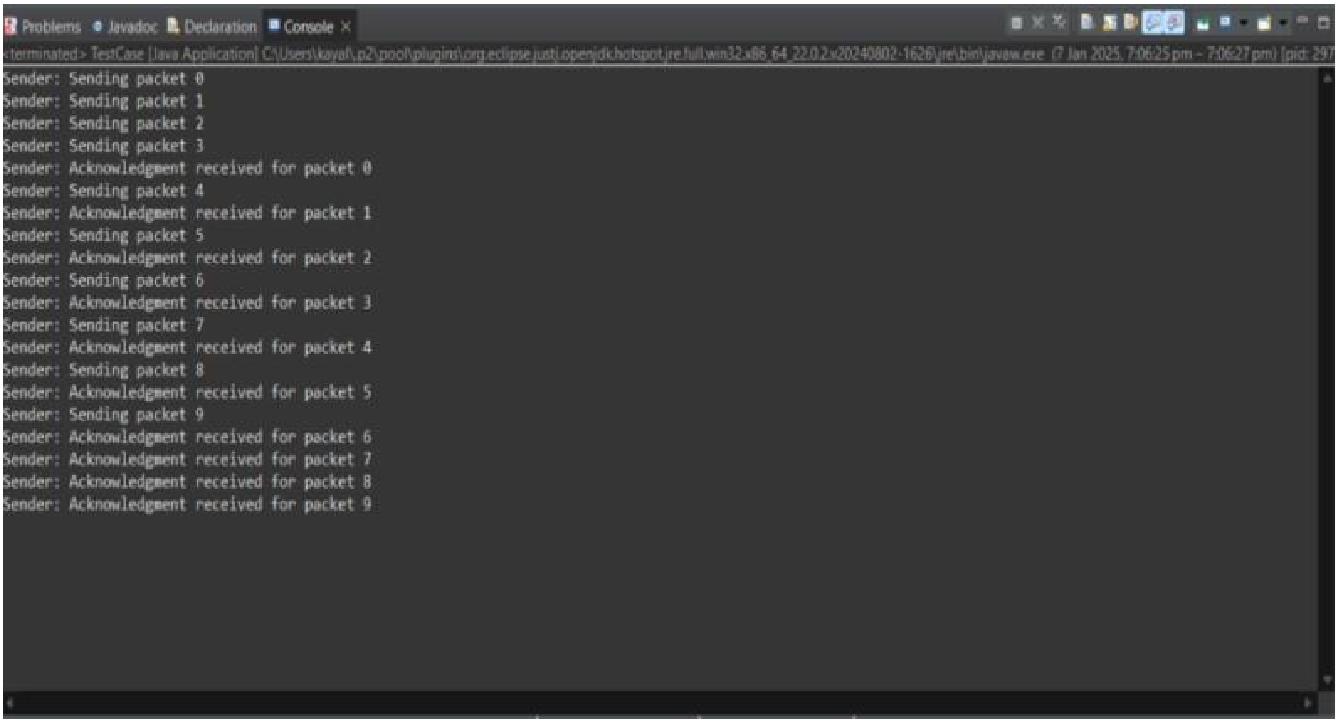
```
import java.util.Queue;
public class Receiver {
public void receivePacket(int packet) {
System.out.println("Receiver: Received packet with sequence number " + packet);
}
public void acknowledge(Queue<Integer> window) { for
(Integer seqNum : window) {
System.out.println("Receiver: Acknowledging packet with sequence number " + seqNum);
}
}
}
```

Main.java

```
import java.util.Scanner;
public class Main {
public static void main(String[] args)
{ Scanner scanner = new Scanner(System.in);
System.out.print("Enter the window size: ");
```

```
int windowSize = scanner.nextInt();
 System.out.print("Enter the total number of packets: ");
 int totalPackets = scanner.nextInt();
Sender sender = new Sender1(windowSize, totalPackets);
Receiver receiver = new Receiver1();
sender.startTransmission(receiver);
scanner.close();
}
}
```

Output :



Result :-

Thus, the program for sliding window protocol was implemented and executed successfully.

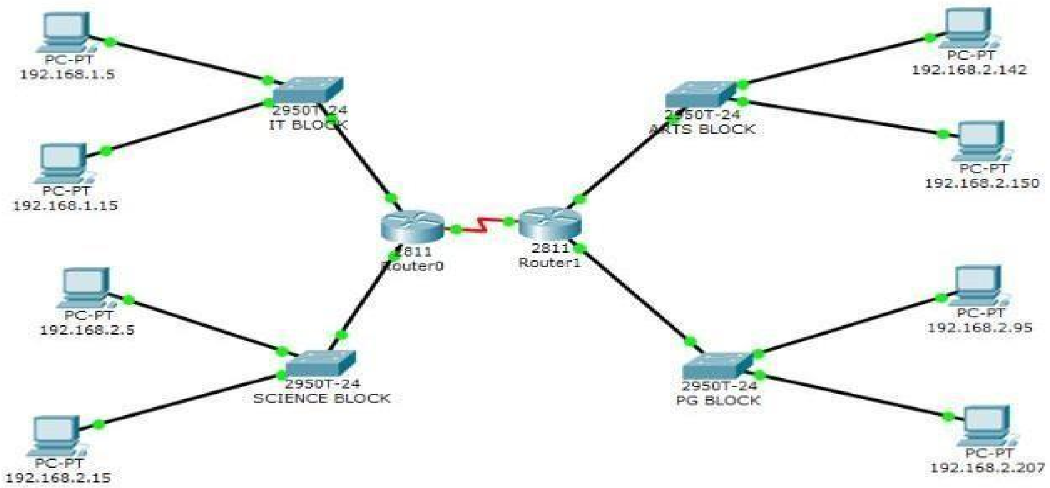
Ex. No : 3	IMPLEMENT SUBNETTING TECHNIQUES TO OPTIMIZE NETWORK ADDRESSING USING CISCO PACKET TRACER
Date :	

Aim :-

To implement subnetting and find the subnet masks using Cisco Packet Tracer.

Implementing an IP Addressing and Subnetting Topology

Diagram :-



OBJECTIVES :-

- Subnet an address space based on the host requirements.
- Assign host addresses to devices.
- Configure devices with IP addressing.
- Verify the addressing configuration.

Background / Preparation :-

In this activity, you will subnet the private address space 192.168.1.0/24 to provide enough host addresses for the two LANs attached to the router. You will then assign valid host addresses to the appropriate devices and interfaces. Finally, you will test connectivity to verify your IP address implementation.

Step 1: Subnet an address space based on the host requirements.

You are given the private address space 192.168.1.0/24. Subnet this address space based on the following requirements

IT BLOCK – 136 host

Science Block – 95 host Arts Block:

51 host

PG Block: 24 host

Step 2: Assign host addresses to devices.

What is the subnet address for subnet 0? 255.255.255.0 What is the subnet address of it ? **55.255.255.126** .

What is the subnet address for subnet 2? **255.255.255.192**

What is the subnet address for subnet 2? **255.255.255.224**

Assign subnet 0 to IT Block, assign subnet 1 to Science Block assign subnet 2 to arts Block and assign Subnet 3 to PG block.

This address is assigned the FastEthernet0/0 interface on Customer Router. What is the first address in subnet1? **192.168.2.1**

This address is assigned the FastEthernet0/0 interface on Customer Router. What is the first address in subnet2? **192.168.2.129**

This address is assigned the FastEthernet0/1 interface on Customer What is the first address in subnet3? **192.168.2.193**

What is the first address in subnet0? **192.168.1.1**

This address is assigned the FastEthernet0/2 interface on Customer

Assign address for serial connection for Router 0 to Router 1

Serial 0/0 for Router 0? **192.168.2.225**

Serial 0/0 for Router 1? **192.168.2.226**

Step 3: Configure devices with IP addressing.

Configure the LAN interfaces on Customer Router with IP addresses and a subnet mask.

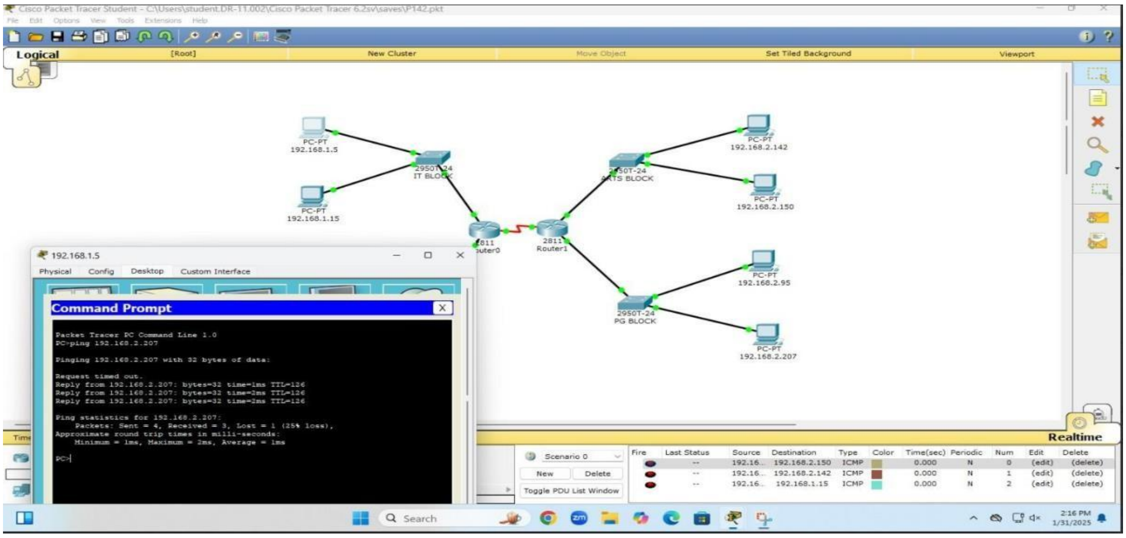
For a better view of the commands, you can increase the size of the window. To resize the window, place your mouse pointer over the bottom border of the window. When the cursor turns into a double headed arrow, click and drag.

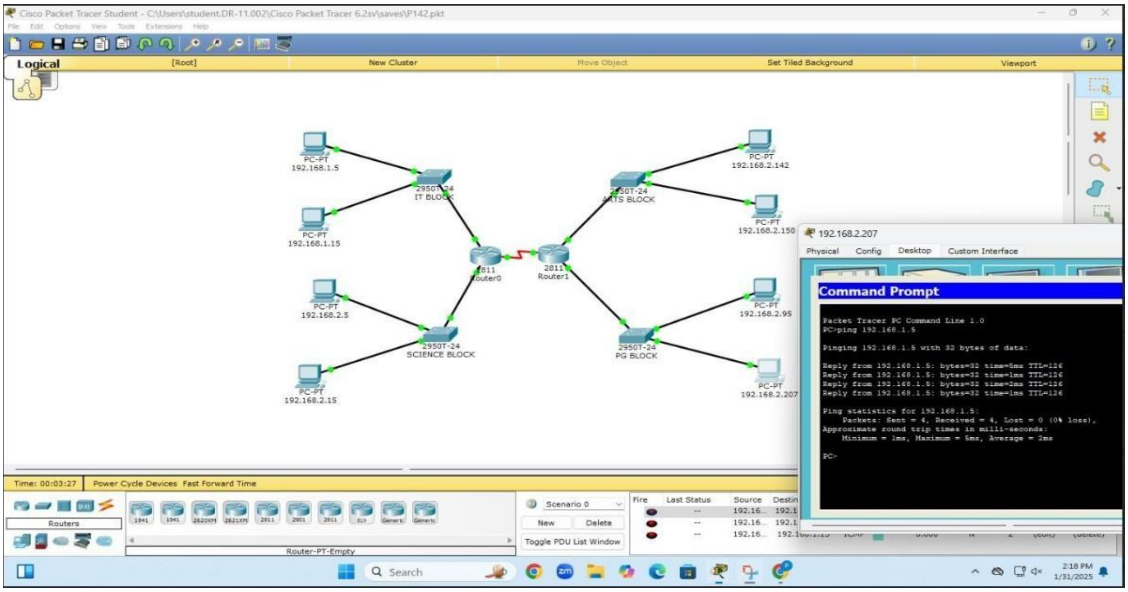
Step 4: Verify the addressing configuration.

a. TestconnectivitybetweenHost1, Host3, ISP Workstation, and ISP Server. You can usetheAddSimplePDUtooltocreatepingsbetweenthedevices.Youcanalsoclick Host 1 or Host 3, then the Desktop tab, and then Command Prompt. Use the ping command to test connectivity to other devices. To obtain the IP address of another device, place your mouse pointer over the device.

b. Check results. On the Connectivity Tests tab, the status of each test should be successful.

OUTPUT :-





Result :-

Thus the concept of implementing of subnetting has been executed successfully and its output verified.

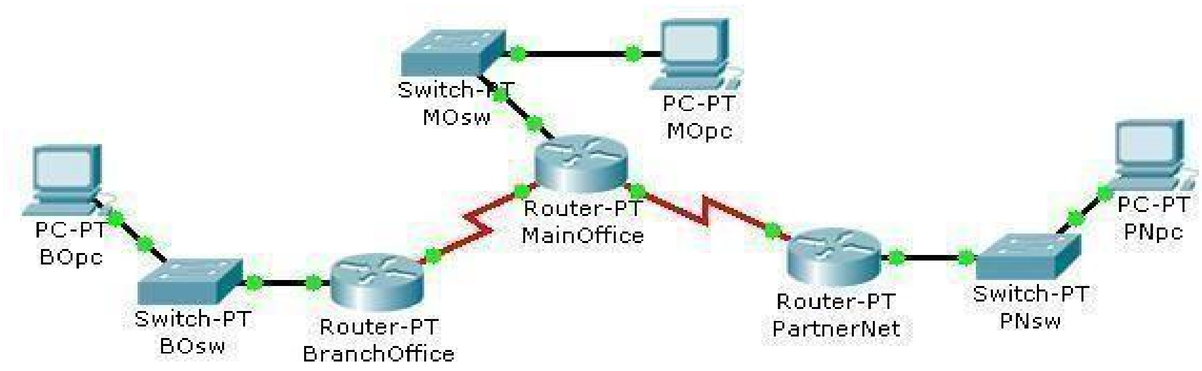
Ex. No: 4A	DESIGN AND IMPLEMENT ROUTING ALGORITHMS FOR EFFICIENT DATA TRANSMISSION
Date :	

AIM:

To write a program for Implementing distance vector routing algorithm.

Configuring Static and Default Routes

Topology Diagram



Objectives

- Configure static routes on each router to allow communication between all clients.
- Test connectivity to ensure that each device can fully communicate with all other devices.

Background / Preparation

- This topology represents a small WAN. Each device in this network has been configured with IP addresses;
- however, no routing has been configured. The company management wants to use static routes to connect the
- multiple networks.

Step 1: Test connectivity between the PCs and the default gateway.

To determine if there is connectivity from each PC to its configured gateway, first use a simple ping test.

c. Click BOpc and go to **Desktop > Command Prompt**.

d. From the command prompt, type the **ipconfig** command. Note the IP address for BOpc and the default gateway address. The default gateway address is the IP address for the Fast Ethernet interface on Branch Office.

Ping 192.168.1.1, the default gateway address for the Branch Office LAN, from the command prompt on BO pc. This ping should be successful.Click PNpc and go to **Desktop > Command Prompt**.

e. From the command prompt, type the **ipconfig** command. Note the IP address for PNpc and the default gateway address. The default gateway address is the IP address for the

Fast Ethernet interface on PartnerNet.

f. Ping 192.168.3.1, the default gateway address for the PartnerNet LAN, from the command prompt on PNpc. This ping should be successful.

g. Repeat steps a, b, and c for MOpc and its respective default gateway, the Fast Ethernet interface on MainOffice. Each of these ping tests should be successful.

Step 2: Ping between routers to test connectivity.
Use a console cable and terminal emulation software on BOpc to connect to Branch Office

a. Test connectivity with MainOffice by pinging 10.10.10.1, the IP address of the directly connected serial 3/0 interface. This ping should succeed.
b. Test connectivity with MainOffice by pinging 10.10.10.5, the IP address of the serial 2/0 interface. This ping should fail.
c. Issue the **show ip route** command from the terminal window of BOpc. Note that only directly connected routes are shown in the BranchOffice routing table. The ping to 10.10.10.5 failed because the BranchOffice router has no routing table entry for 10.10.10.5.
d. Repeat steps a through d on the other two PCs. The pings to directly connected networks will succeed. However, pings to remote networks will fail.
e. What steps must be taken to reach all the networks from any PC in the activity?

Step 3: Viewing the routing tables.
You can view routing tables in Packet Tracer using the Inspect tool. The Inspect tool is in the Common Tools bar to the right of the topology. The Inspect tool is the icon that appears as a magnifying glass.
Step 4: Configure default routes on the BranchOffice and PartnerNet routers.
To configure static routes for each router, first determine which routes need to be added for each device. For the BranchOffice and the PartnerNet routers, a single default route allows these devices to route traffic for all networks not directly connected. To configure a default route, you must identify the IP address of the next hop router, which in this case is the MainOffice router.

a. From the **Common** toolbar, click the **Select** tool.

b. Move the cursor over the red serial link between the BranchOffice router and the MainOffice router. Notice that the interface of the next hop is S3/0.

c. Move the cursor over the MainOffice router and note that the IP address for Serial 3/0 is 10.10.10.1.

d. Move the cursor over the red serial link between the PartnerNet router and the MainOffice router. Notice that the interface of the next hop is S2/0.

e. Move the cursor over the MainOffice router and note that the IP address for Serial 2/0 is 10.10.10.5.

f. Configure the static routes on both the BranchOffice and PartnerNet routers using the CLI. Click the BranchOffice router, and click the **CLI** tab.

g. At the **BranchOffice>** prompt, type **enable** to enter privileged EXEC mode.

h. At the **BranchOffice#** prompt, type **configure terminal**.

i. The syntax for a default route is **ip route 0.0.0.0 0.0.0.0** ***next_hop_ip_address***.
Type **ip route 0.0.0.0 0.0.0.0 10.10.10.1**.

j. Type **end** to get back to the **BranchOffice#** prompt.

k. Type **copy run start** to save the configuration change.

l. Repeat steps f through k on the PartnerNet router, using 10.10.10.5 as the next hop IP address.

Step 5: Configure static routes at Main Office.

The configuration of static routes at the Main Office is a bit more complex because the Main Office router is responsible for routing traffic to and from the Branch Office and Partner Net LAN segments.

The MainOffice router knows only about routes to the 10.10.10.0/30, 10.10.10.4/30, and 192.168.2.0/24 networks because they are directly connected. Static routes to the 192.168.1.0/24 and 192.168.3.0/24 networks need to be added so that the MainOffice router can route traffic between the networks behind the BranchOffice and PartnerNet routers.

a. Click the MainOffice router, and then click the **CLI** tab.

b. At the MainOffice> prompt, type **enable** to enter privileged EXEC mode.

c. At the MainOffice# prompt, type **configure terminal**.

d. The syntax for a static route is **ip route** ***network subnet_mask next_hop_ip_address***: ip route 192.168.1.0 255.255.255.0 10.10.10.2
ip route 192.168.3.0 255.255.255.0 10.10.10.6

e. Type **end** to return to the MainOffice# prompt.

Type **copy run start** to save the configuration change.

Repeat steps a through e from Step 3. View the routing tables and notice the difference in the routing tables. The routing table for each router should have an “S” for each static route.

Step 6: Test connectivity.

Now that each router in the topology has static routes configured, all hosts should have connectivity to all other hosts. Use ping to verify connectivity.

a. Click **BOpc** and click the **Desktop** tab. Choose the **Command prompt** option.
b. Type **ping 192.168.3.2.** The ping should be successful, verifying that the static routes are configured properly.

c. Type **ping 192.168.2.2.** Notice that the result is successful even though you did not specifically add the 192.168.2.0 network as a static route into any of the routers. Because a default route was used on the Branch Office and Partner Net routers, a route for the 192.168.2.0 network was not needed. The default route sends all traffic destined off network to the MainOffice router. The 192.168.2.0 network is directly connected to the Main Office router; therefore, no additional routes needed to be added to the routing table.

d. Click the **Check Results** button at the bottom of this instruction window to check your work.

OUTPUT:

```
Packet Tracer PC Command Line 1.0
C:\>ping 192.168.30.1

Pinging 192.168.30.1 with 32 bytes of data:

Reply from 192.168.30.1: bytes=32 time=8ms TTL=255
Reply from 192.168.30.1: bytes=32 time=4ms TTL=255
Reply from 192.168.30.1: bytes=32 time=4ms TTL=255
Reply from 192.168.30.1: bytes=32 time=4ms TTL=255

Ping statistics for 192.168.30.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 4ms, Maximum = 8ms, Average = 5ms

C:\>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.20.2: bytes=32 time=10ms TTL=126
Reply from 192.168.20.2: bytes=32 time=10ms TTL=126
Reply from 192.168.20.2: bytes=32 time=10ms TTL=126

Ping statistics for 192.168.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 10ms, Maximum = 10ms, Average = 10ms
```

RESULT:

Thus the network of implementation of router (Static Routing) has been executed successfully and its connectivity is verified.

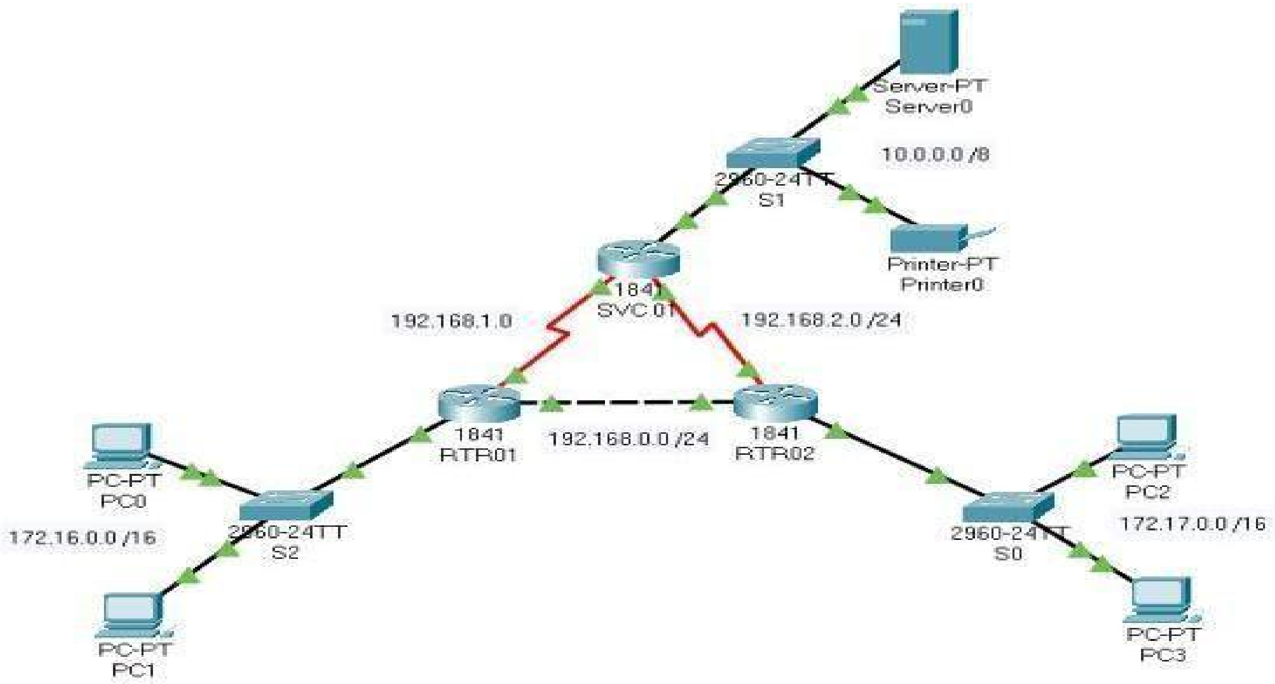
Ex. No: 4B	DESIGN AND IMPLEMENT ROUTING ALGORITHMS FOR EFFICIENT DATA TRANSMISSION (DYNAMIC ROUTING)
Date :	

AIM:

To implement the shortest path routing (DYNAMIC ROUTING) to find the shortest path between the nodes.

Dynamic Routing Configuration

Topology Diagram



Objectives

- Configure routers using basic interface configuration commands.
- Enable RIP.
- Verify the RIP configuration.

Background / Preparation

A simple routed network has been set up to assist in reviewing RIP routing behavior. In this activity, you will configure RIP across the network and set up end devices to communicate on the network.

Step 1: Configure the SVC01 router and enable RIP.

a. From the CLI, configure interface Fast Ethernet 0/0 using the IP address 10.0.0.254 /8.

b. Configure interface serial 0/0/0 using the first usable IP address in network 192.168.1.0 /24 to connect to the RTR01 router. Set the clock rate at 64000.

c. Configure interface serial 0/0/1 using the first usable IP address in network 192.168.2.0 /24 with a clock rate of 64000.

d. Using the **no shutdown** command, enable the configured interfaces.

e. Configure RIP to advertise the networks for the configured interfaces.

f. Configure the end devices.

i. Server0 uses the first usable IP address in network 10.0.0.0 /8. Specify the appropriate default gateway and subnet mask.

J. Printer0 uses the second usable IP address

Step 2: Configure the RTR01 router and enable RIP.

a. Configure interface Fast Ethernet 0/0 using the first usable IP address in network 192.168.0.0 /24 to connect to the RTR02 router.

b. Configure interface serial 0/0/0 using the second usable IP address in network 192.168.1.0 /24 to connect to the SVC01 router.

c. Configure interface Fast Ethernet 0/1 using the IP address 172.16.254.254/16.
d. Using the **no shutdown** command, enable the configured interfaces.
d. Configure RIP to advertise the networks for the configured interfaces. f. Configure the end devices.

i. PC0 uses the first usable IP addresses in network 172.16.0.0 /16.
ii. PC1 uses the second usable IP address in network 172.16.0.0 /16.
iii. Specify the appropriate default gateway and subnet mask on each PC.

Step 3: Configure the RTR02 router and enable RIP.

a. Configure interface Fast Ethernet 0/0 using the second usable IP address in network 192.168.0.0 /24 to connect to the RTR01 router.

b. Configure interface serial 0/0/0 using the second usable IP address in network 192.168.2.0 /24 to connect to the SVC01 router.

c. Configure interface Fast Ethernet 0/1 using the IP address 172.17.254.254 /16. d. Using the **no shutdown** command, enable the configured interfaces.
e. Configure RIP to advertise the networks for the configured interfaces. f. Configure the end devices.

i. PC2 uses the first usable IP addresses in network 172.17.0.0 /16.

ii. PC3 uses the second usable IP address in network 172.17.0.0 /16.

iii. Specify the appropriate default gateway and subnet mask on each PC.

Step 4: Verify the RIP configuration on each router.

a. At the command prompt for each router, issue the commands **show ip protocols** and **show ip route** to verify RIP routing is fully converged.
The **show ip protocols** command displays the networks the router is advertising and the addresses of other RIP routing neighbors.
The **show ip route** command output displays all routes know to the local router including the RIP routes which are indicated by an “R”.

b. Every device should now be able to successfully ping any other device in this activity.

c. Click the **Check Results** button at the bottom of this instruction window to check your work.

OUTPUT:

```
Packet Tracer PC Command Line 1.0
C:\>ping 172.17.0.0

Pinging 172.17.0.0 with 32 bytes of data:

Reply from 192.168.0.2: bytes=32 time<1ms TTL=254
Reply from 192.168.0.2: bytes=32 time<1ms TTL=254
Reply from 192.168.0.2: bytes=32 time=1ms TTL=254
Reply from 192.168.0.2: bytes=32 time<1ms TTL=254

Ping statistics for 172.17.0.0:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 172.17.1.2

Pinging 172.17.1.2 with 32 bytes of data:

Request timed out.
Reply from 172.17.1.2: bytes=32 time=1ms TTL=126
Reply from 172.17.1.2: bytes=32 time<1ms TTL=126
Reply from 172.17.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 172.17.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>
```

RESULT:

Thus the network of implementation of router (Dynamic Routing) has been executed successfully and its connectivity is verified.