



**RAJALAKSHMI**  
**ENGINEERING COLLEGE**  
An AUTONOMOUS Institution  
Affiliated to ANNA UNIVERSITY, Chennai

**DEPARTMENT OF COMPUTER SCIENCE AND DESIGN**

**CS19541 - COMPUTER NETWORKS**

**LABORATORY RECORD**

**Name : Sharvesh S**

**Roll No: 221701052**

**Department : Computer Science and Design**

## EXPERIMENT – 1

**AIM:** - Study of various Network commands used in Linux and Windows

### **BASIC NETWORKING COMMANDS: -**

#### **Windows:**

**arp –a:** ARP is short form of address resolution protocol, It will show the IP address of your computer along with the IP address and MAC address of your router.

---

**hostname:** This is the simplest of all TCP/IP commands. It simply displays the name of your computer.

---

**ipconfig /all:** This command displays detailed configuration information about your TCP/IP connection including Router, Gateway, DNS, DHCP, and type of Ethernet adapter in your system

---

**nbtstat –a:** This command helps solve problems with NetBIOS name resolution. (Nbt stands for NetBIOS over TCP/IP)

---

**netstat:** (network statistics) netstat displays a variety of statistics about a computers active TCP/IP connections. It is a command line tool for monitoring network connections both incoming and outgoing as well as viewing routing tables, interface statistics etc.

e.g.: netstat -r

---

**nslookup:** (name server lookup) is a tool used to perform DNS lookups in Linux. It is used to display DNS details, such as the IP address of a particular computer, the MX records for a domain or the NS servers of a domain. nslookup can operate in two modes: interactive and non-interactive.

e.g.: nslookup www.google.com

---

**pathping:** Pathping is unique to Window's, and is basically a combination of the Ping and Tracert commands. Pathping traces the route to the destination address then launches a 25 second test of each router along the way, gathering statistics on the rate of data loss along eachhop.

---

**ping:** (Packet INternet Groper) command is the best way to test connectivity between two nodes. Ping use ICMP (Internet Control Message Protocol) to communicate to other devices.

1. #ping hostname( ping localhost)

2. #ping ip address (ping 4.2.2.2)

3. #ping fully qualified domain name(ping www.facebook.com)

**Route:** route command is used to show/manipulate the IP routing table. It is primarily used to setup static routes to specific host or networks via an interface.

## OUTPUT: -

```
Microsoft Windows [Version 10.0.22631.4317]
(c) Microsoft Corporation. All rights reserved.

C:\Users\hp>arp -a

Interface: 192.168.0.100 --- 0x5
  Internet Address      Physical Address      Type
  192.168.0.1           b0-95-75-d0-c1-19   dynamic
  192.168.0.255         ff-ff-ff-ff-ff-ff   static
  224.0.0.22            01-00-5e-00-00-16   static
  224.0.0.251           01-00-5e-00-00-fb   static
  224.0.0.252           01-00-5e-00-00-fc   static
  239.255.255.250       01-00-5e-7f-ff-fa   static
  255.255.255.255       ff-ff-ff-ff-ff-ff   static

C:\Users\hp>hostname
DESKTOP-F0DB1M0

C:\Users\hp>ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 1:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . :
```

```
  Connection-specific DNS Suffix . :

Wireless LAN adapter Local Area Connection* 2:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

Wireless LAN adapter Wi-Fi:

  Connection-specific DNS Suffix . :
  IPv6 Address . . . . . : 2406:7400:bb:9a01:68d5:b10:8367:e13e
  Temporary IPv6 Address . . . . . : 2406:7400:bb:9a01:9462:fa5e:eb4:3130
  Link-local IPv6 Address . . . . . : fe80::697a:d049:6a46:ed14%5
  IPv4 Address . . . . . : 192.168.0.100
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : fe80::b295:75ff:fed0:c119%5
                                         192.168.0.1

Ethernet adapter Bluetooth Network Connection:

  Media State . . . . . : Media disconnected
  Connection-specific DNS Suffix . :

C:\Users\hp>pathping

Usage: pathping [-g host-list] [-h maximum_hops] [-i address] [-n]
                [-p period] [-q num_queries] [-w timeout]
                [-4] [-6] target_name

Options:
  -g host-list    Loose source route along host-list.
  -h maximum_hops Maximum number of hops to search for target.
  -i address      Use the specified source address.
  -n              Do not resolve addresses to hostnames.
  -p period       Wait period milliseconds between pings.
  -q num_queries  Number of queries per hop.
  -w timeout      Wait timeout milliseconds for each reply.
  -4              Force using IPv4.
  -6              Force using IPv6.

C:\Users\hp>
```

## **Linux:**

**ip:** The ip command is one of the basic commands every administrator will need in daily work, from setting up new systems and assigning IPs to troubleshooting existing systems. The ip command can show address information, manipulate routing, plus display network various devices, interfaces, and tunnels.

---

**ifconfig:** The ifconfig command was/is a staple in many sysadmin's tool belt for configuring and troubleshooting networks. It has since been replaced by the ip command discussed above.

---

**mtr:** MTR (Matt's traceroute) is a program with a command-line interface that serves as a network diagnostic and troubleshooting tool. This command combines the functionality of the ping and traceroute commands. Just like a traceroute, the mtr command will show the route from a computer to a specified host. mtr provides a lot of statistics about each hop, such as response time and percentage. With the mtr command, you will get more information about the route and be able to see problematic devices along the way. If you see a sudden increase in response time or packet loss, then obviously, there is a bad link somewhere.

The syntax of the command is as follows:

`mtr <options> hostname/IP`

---

**tcpdump:** The tcpdump command is designed for capturing and displaying packets.

---

**ping:** verifies IP-level connectivity to another TCP/IP computer by sending Internet Control Message Protocol (ICMP) Echo Request messages. The receipt of corresponding Echo Reply messages is displayed, along with round-trip times. Ping is the primary TCP/IP command used to troubleshoot connectivity, reachability, and name resolution.

**OUTPUT:** -

**RESULT:** -

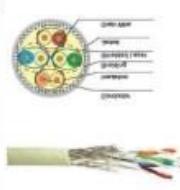
The commands for Linux and Windows has been executed successfully and the output is verified.

## **Experiment No 2**

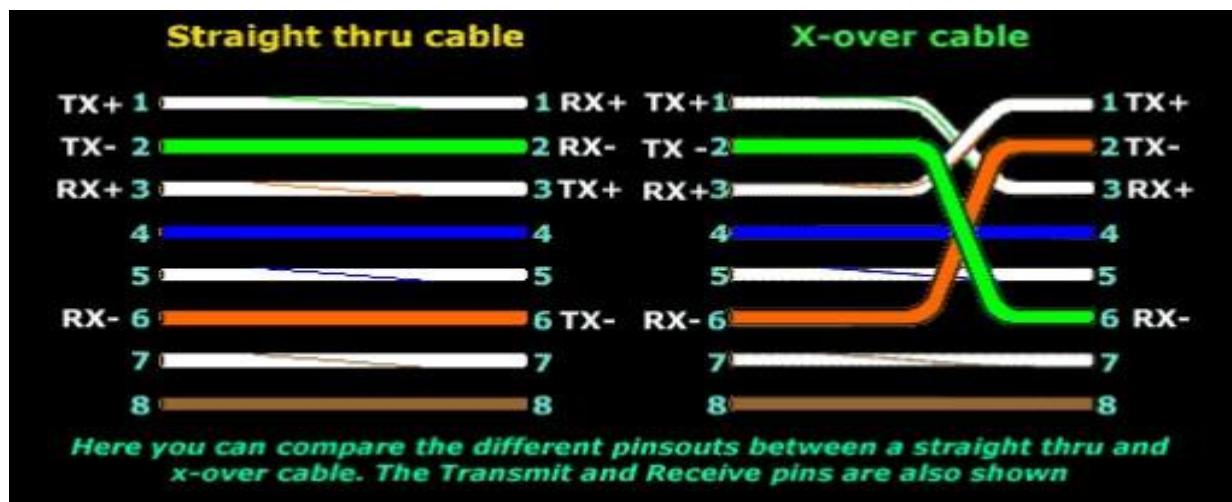
**Aim:** Study of different types of Network cables.

**Different type of cables used in networking are:**

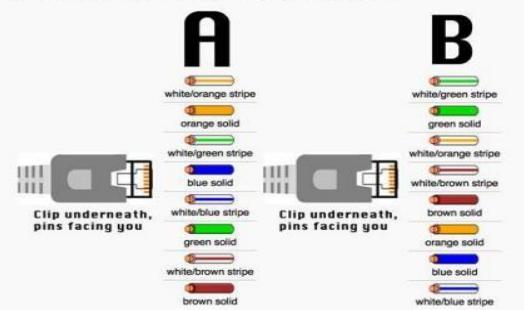
- 1. Unshielded Twisted Pair (UTP) Cable**
- 2. Shielded Twisted Pair (STP) Cable**
- 3. Coaxial Cable**
- 4. Fiber Optic Cable**

Cable type	Category	Maximum Data Transmission	Advantages/Disadvantages	Application/Use	Image
UTP	Category 3	10 bps	<b>Advantages</b> <ul style="list-style-type: none"> <li>• Cheaper in cost</li> <li>• Easy to install as they have a smaller overall diameter.</li> </ul> <b>Disadvantages</b> <ul style="list-style-type: none"> <li>• More prone to (EMI) Electromagnetic interference and noise</li> </ul>	10Base-T Ethernet	
	Category 5	Up to 100 Mbps		Fast Ethernet, Gigabit Ethernet	
	Category 5e	1Gbps		Fast Ethernet, Gigabit Ethernet	
STP	Category 6,6a	10Gbps	<b>Advantages</b> <ul style="list-style-type: none"> <li>• Shielded.</li> <li>• Faster than UTP.</li> <li>• Less susceptible to noise and interference</li> </ul> <b>Disadvantages</b> <ul style="list-style-type: none"> <li>• Expensive</li> <li>• Greater installation effort</li> </ul>	Gigabit Ethernet, 10G Ethernet (55m) Widely used in data centres	
SSTP	Category 7	10Gbps	Gigabit Ethernet, 10G Ethernet (100m)		

Coaxial cable	RG-6 RG-59 RG-11	10-100Mbps	<ul style="list-style-type: none"> <li>High bandwidth</li> <li>Immune to interference</li> <li>Low loss bandwidth</li> <li>Versatile</li> <li><b>Disadvantages</b></li> <li>Limited distance</li> <li>Cost</li> <li>Size is bulky</li> </ul>	Speed of signal is 500m Television network High speed internet connections	
fibre optics cable	Single mode Multi mode	100Gbps	<b>Advantages</b> <ul style="list-style-type: none"> <li>High speed</li> <li>High bandwidth</li> <li>High security</li> <li>Long distance</li> </ul> <b>Disadvantages</b> <ul style="list-style-type: none"> <li>Expensive</li> <li>Requires skilled installers</li> </ul>	Maximum distance of fibre optics cable is around 100meters	



**Straight through network cable:** both sides should be A  
**Crossover cable:** One side A, one side B



**Step 1:** To start construction of the device, begin by threading shields onto the cable.

The crimping tool has a round area to complete this task.

**Step 3:** After, you will need to untangle the wires; there should be four “twisted pairs.”

Referencing back to the sheet, arrange them from top to bottom. One end should be in

arrangement A and the other in B.

**Step 4:** Once the order is correct, bunch them together in a line, and if there are any that

stick out farther than others, snip them back to create an even level. The difficult aspect

is placing these into the RJ45 plug without messing up the order. To do so, hold the plug

with the clip side facing away from you and have the gold pins facing toward you, as

shown.

**Step 5:** Next, push the cable right in. The notch at the end of the plug needs to be just

over the cable shielding, and if it isn't, that means that you stripped off too much shielding. Simply snip the cables back a little more.

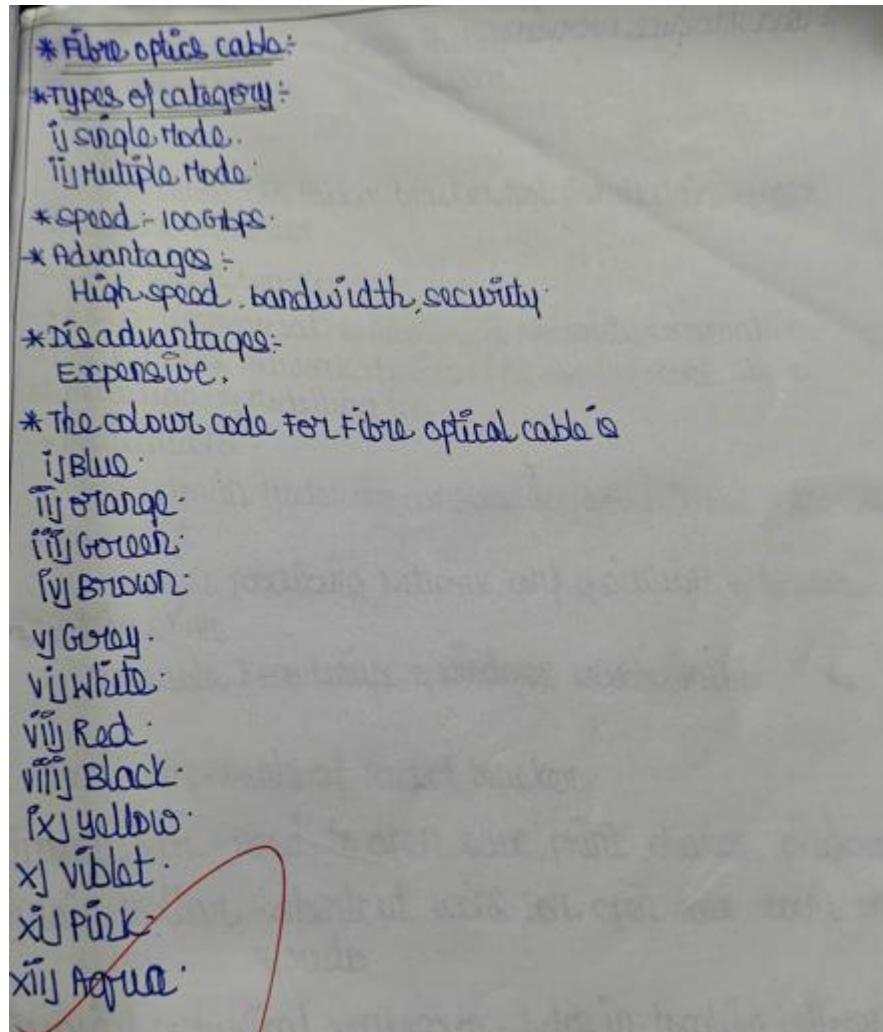
**Step 6:** After the wires are securely sitting inside the plug, insert it into the crimping tool

and push down. It should be shaped correctly, but pushing too hard can crack the fragile plastic plug.

**Step 7:** Lastly, repeat for the other end using diagram B (to make a crossover cables)/

using diagram A (to make a straight through cable)

To test it, plug it in and attempt to connect two devices directly.



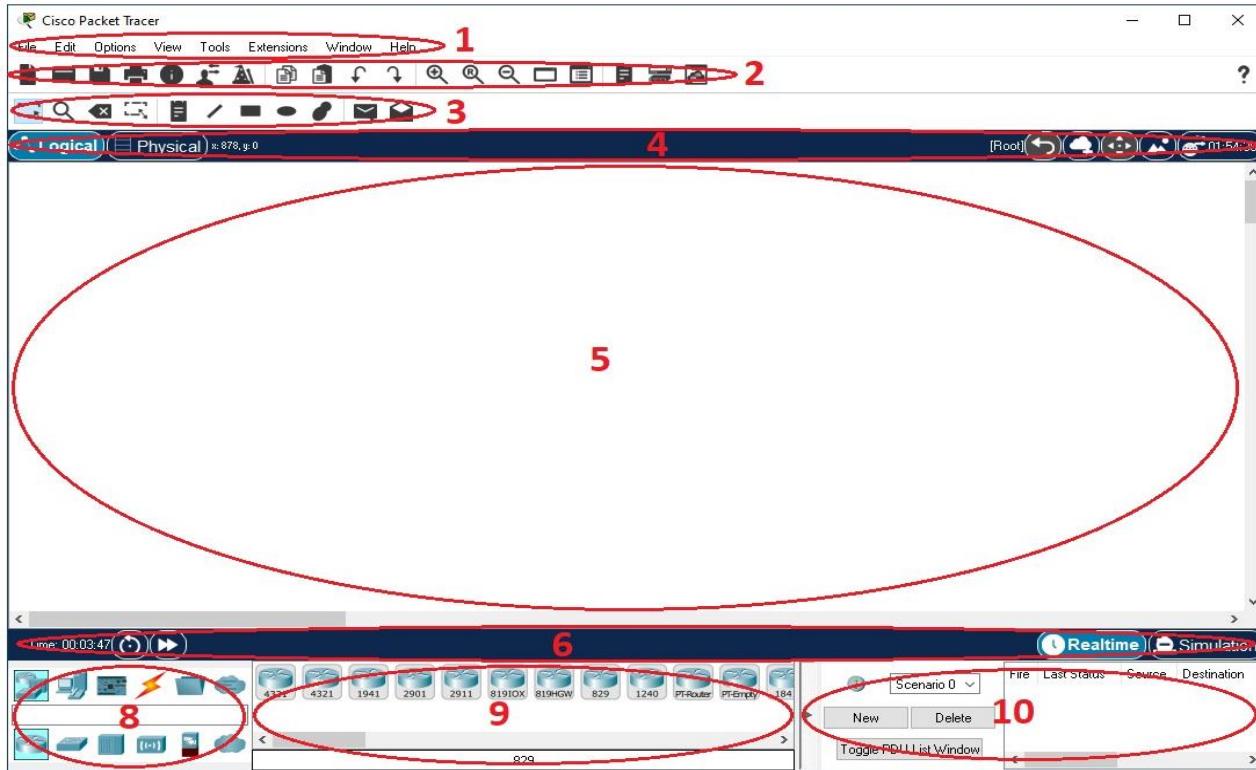
### Result :

**The Study of different types of Network cables has been successfully executed.**

## EXPERIMENT – 3

**AIM:** - To study the Packet tracer tool Installation and User Interface Overview

**To understand environment of CISCO PACKET TRACER to design simple network.**  
**USER INTERFACE OVERVIEW:**

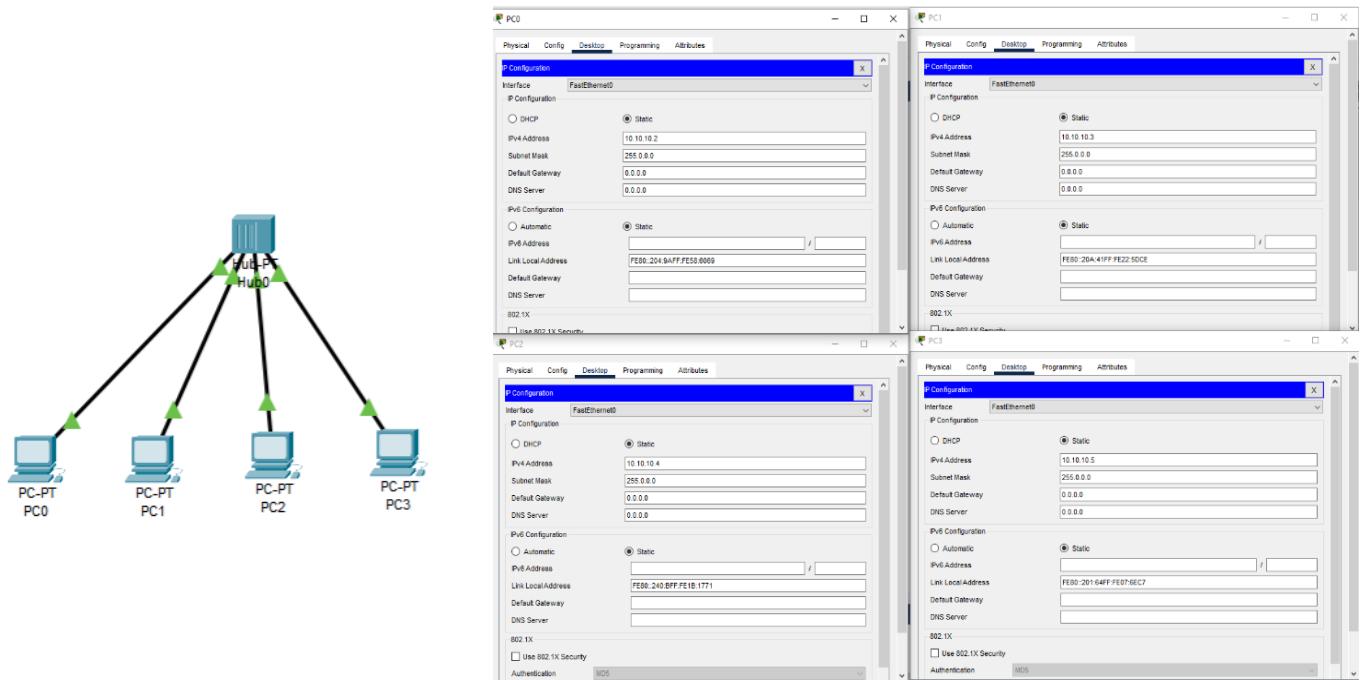


The layout of Packet Tracer is divided into several components. The components of the Packet Tracer interface are as follows:

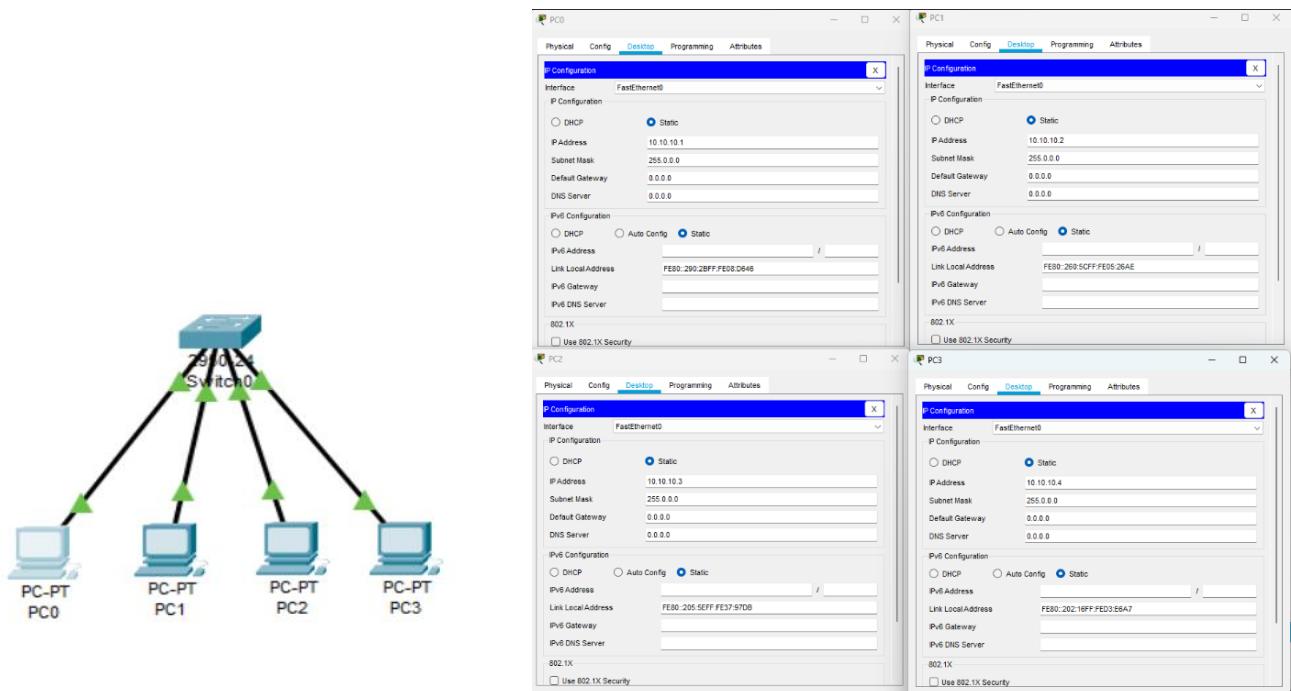
- 1. Menu bar** – This is a common menu found in all software applications; it is used to open, save, print, change preferences, and so on.
- 2. Main toolbar** – This bar provides shortcut icons to menu options that are commonly accessed, such as open, save, zoom, undo, and redo, and on the right-hand side is an icon for entering network information for the current network.
- 3. Common tools bar** – This toolbar provides controls for manipulating topologies, such as select, move layout, place note, delete, inspect, resize shape, and add simple/complex PDU.
- 4. Logical/Physical workspace tabs** – These tabs allow you to toggle between the Logical and Physical work areas.
- 5. Workspace** – This is the area where topologies are created and simulations are displayed.
- 6. Real-time/Simulation tabs** – These tabs are used to toggle between the real and simulation modes. Buttons are also provided to control the time, and to capture the packets.
- 7. Network component box** – This component contains all of the network and end devices available with Packet Tracer, and is further divided into two areas:
- 8. Device-type selection box** – This area contains device categories
- 9. Device-specific selection box** – When a device category is selected, this selection box displays the different device models within that category
- 10. User-created packet box** – Users can create highly-customized packets to test them topology from this area, and the results are displayed as a list.

# Analyse the behaviour of network devices using CISCO PACKET TRACER simulator.

a. 4 Generic PCs and One HUB:



b. 4 Generic PCs and One Switch:



## RESULT: -

Cisco packet have been installed successfully and the interface is studied.

## EXPERIMENT – 4

**AIM:** - Setup and configure a LAN (Local area network) using a Switch and Ethernet cables in your lab.

### How to set up a LAN:

- **Plan and Design Network Topology:** Decide on network layout and equipment location.
- **Gather Equipment:** Use 4 computers, an 8/16/24-port switch, and 4 Ethernet cables.
- **Connect Devices to Switch:** Plug Ethernet cables from each computer into the switch.
- **Assign IP Addresses:**

Log in as Administrator on each PC.

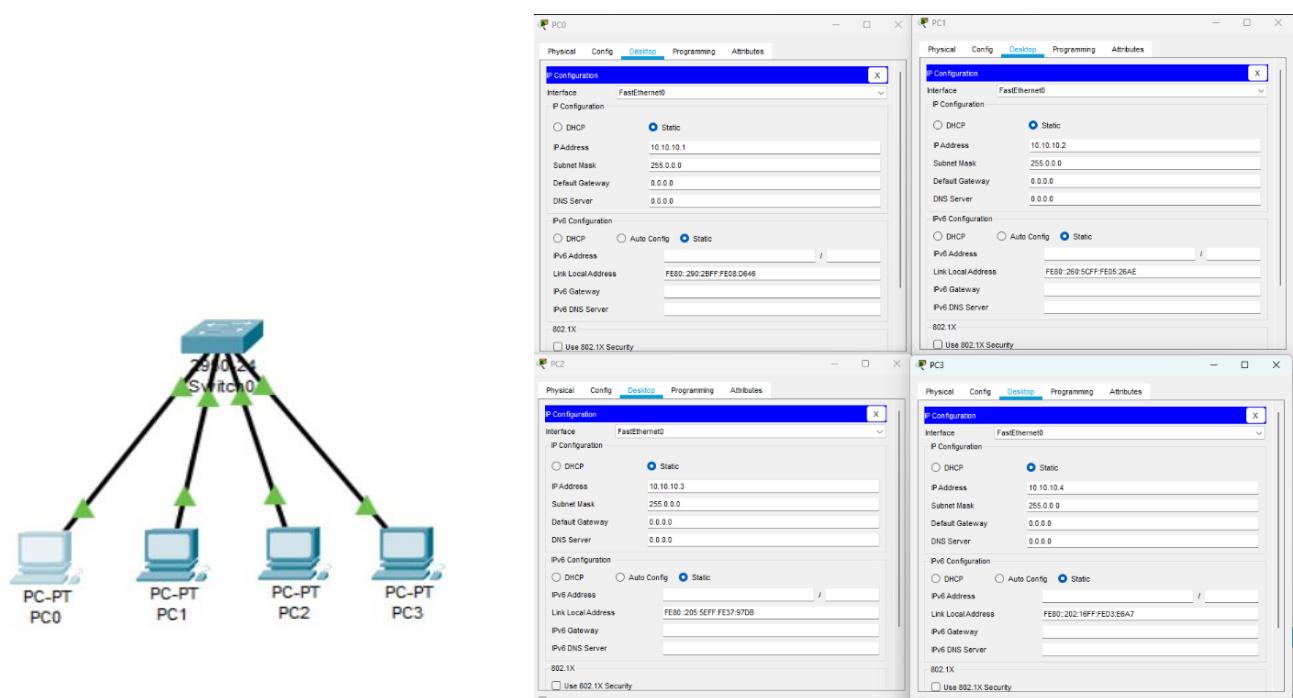
Go to **Network and Internet Connections > Local Area Connection/Ethernet**.

Select **Properties > TCP/IPv4 > Use the following IP address**, then assign:

- PC1: 10.1.1.1, Subnet Mask: 255.0.0.0
- PC2: 10.1.1.2, Subnet Mask: 255.0.0.0
- PC3: 10.1.1.3, Subnet Mask: 255.0.0.0
- PC4: 10.1.1.4, Subnet Mask: 255.0.0.0

- **Configure the Switch:** Connect a computer to the switch and log in to its web interface. Set the switch's IP address to 10.1.1.5, Subnet Mask: 255.0.0.0.
- **Verify Connectivity:** Use the **ping** command to test connection between devices.
- **Share Folder:** Select a folder, go to **Properties > Sharing tab**, and share it with everyone on the LAN.
- **Access Shared Folder:** Try accessing the shared folder from other network computers.

### OUTPUT: -



PC0

Physical Config Desktop Custom Interface

Command Prompt X

```
PC>ipconfig

FastEthernet0 Connection: (default port)

Link-local IPv6 Address.....: FE80::2D0:97FF:FE42:E319
IP Address.....: 10.10.10.1
Subnet Mask.....: 255.0.0.0
Default Gateway.....: 0.0.0.0

PC>ping 10.10.10.3

Pinging 10.10.10.3 with 32 bytes of data:

Reply from 10.10.10.3: bytes=32 time=0ms TTL=128

Ping statistics for 10.10.10.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

PC>nslookup

Server: [255.255.255.255]
Address: 255.255.255.255

>pathping
```

PC1

Physical Config Desktop Custom Interface

Command Prompt X

```
Packet Tracer PC Command Line 1.0
PC>ping 10.10.10.4

Pinging 10.10.10.4 with 32 bytes of data:

Reply from 10.10.10.4: bytes=32 time=1ms TTL=128
Reply from 10.10.10.4: bytes=32 time=0ms TTL=128
Reply from 10.10.10.4: bytes=32 time=1ms TTL=128
Reply from 10.10.10.4: bytes=32 time=0ms TTL=128

Ping statistics for 10.10.10.4:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms

PC>ipconfig

FastEthernet0 Connection: (default port)

Link-local IPv6 Address.....: FE80::230:A3FF:FE8A:16D9
IP Address.....: 10.10.10.2
Subnet Mask.....: 255.0.0.0
Default Gateway.....: 0.0.0.0

PC>
```

## RESULT: -

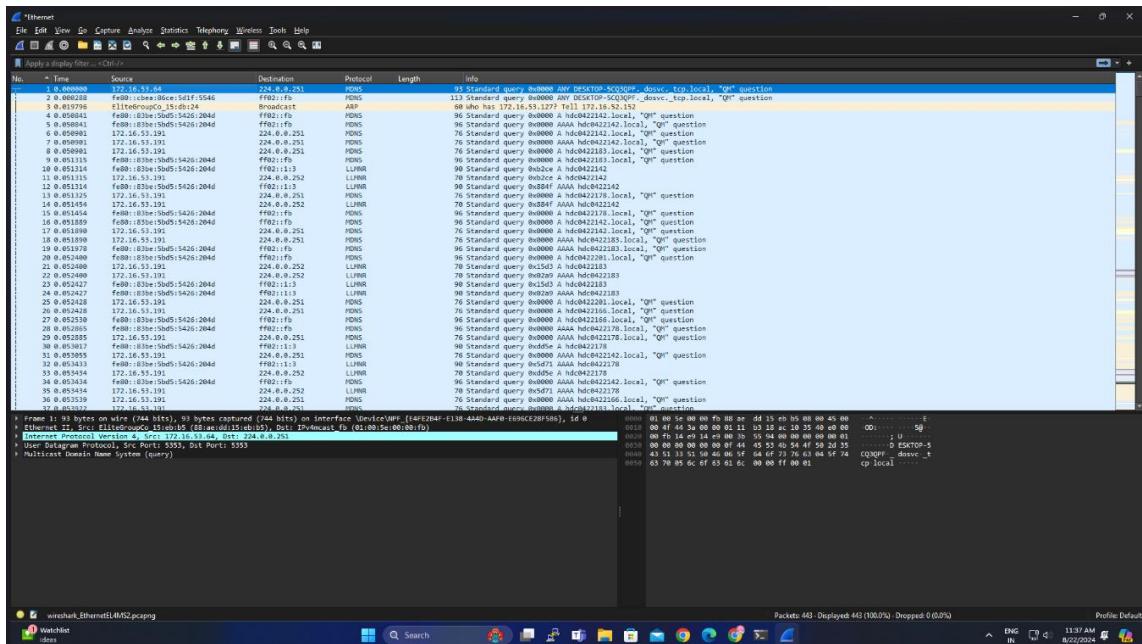
LAN Configuration using Switch has been done and studied successfully.

## EXPERIMENT – 5

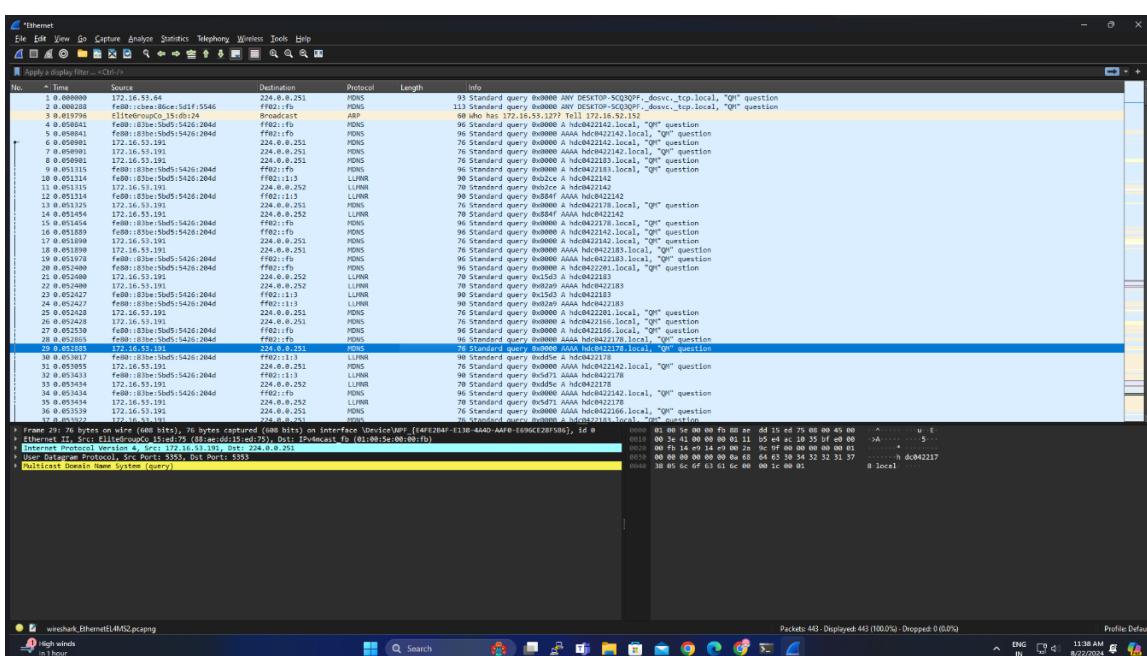
**AIM:** - Experiments on Packet capture tool: Wireshark

### CAPTURING AND ANALYSING PACKETS USING WIRESHARK TOOL:

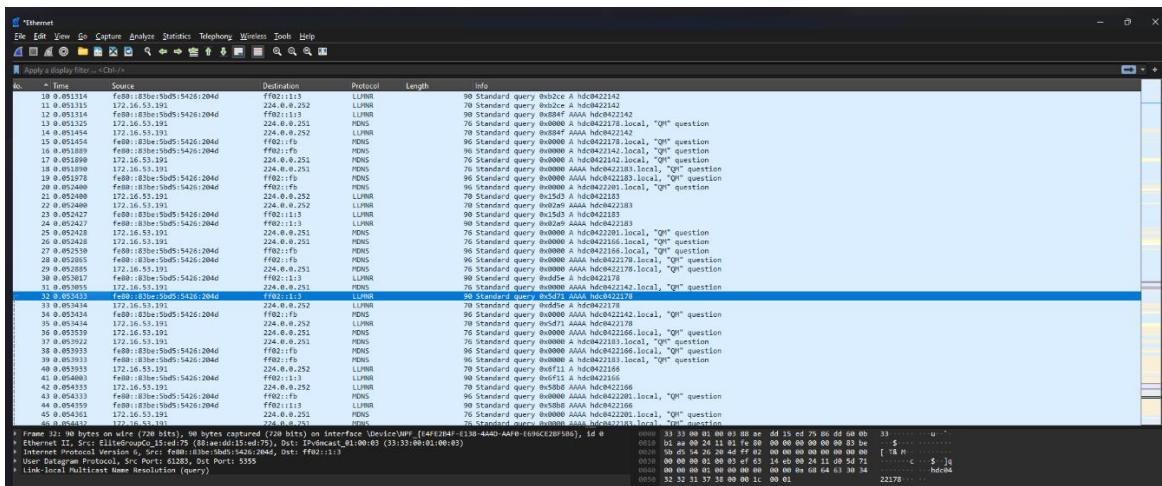
#### Packet 1:



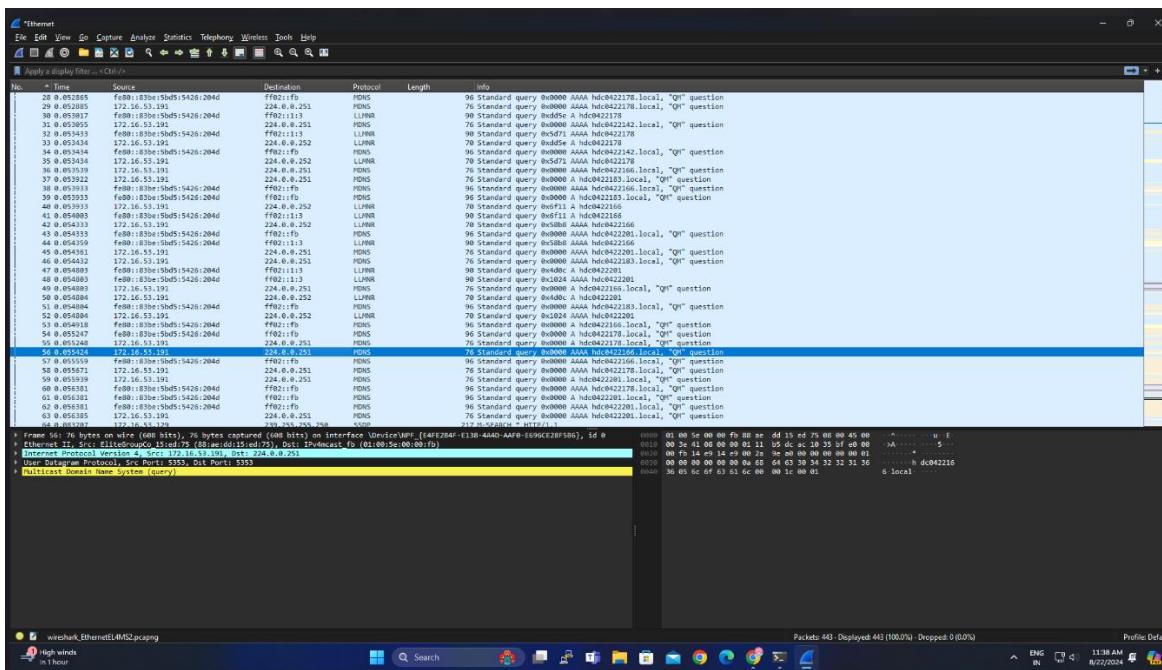
#### Packet 2:



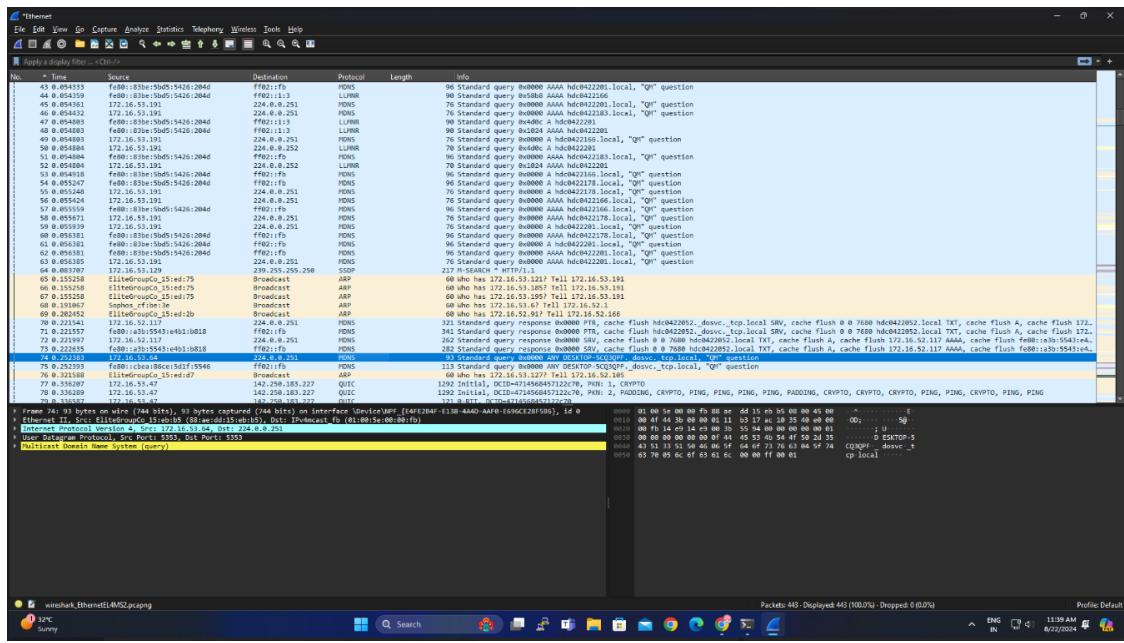
## Packet 3:



## Packet 4:



## Packet 5:



## RESULT: -

Capturing and analysing the packets have been done successfully using Wireshark.

## **EXPERIMENT – 6**

**AIM:** - Write a program to implement error detection and correction using HAMMING code concept. Make a test run to input data stream and verify error correction feature.

1001101'					
i'	power of 2	action	res'	j'	k'
1	1=2**0=yes	insert '0'	0'	1	1
2	2=2**1=yes	insert '0'	00'	2	1
3	3=2**2(NO)	insert data	001'	2	2
4	4=2**2(YES)	insert 0	0010'	3	2
5	5=2**3	insert data	00100'	3	3
6	6=2**3	insert data	001001'	3	4
7	7=2**3	insert data	0010011'	3	5
8	8=2**3(YES)	insert '0'	00100110'	4	5
9	9=2**3(no)				

**CODE:** -

```

def calcRedundantBits(m):
    # Use the formula 2 ^ r >= m + r + 1
    for i in range(m):
        if(2**i >= m + i + 1):
            return i

def posRedundantBits(data, r):
    # Redundancy bits are placed at the positions
    j = 0
    k = 1
    m = len(data)
    res = ''
    # If position is power of 2 then insert '0' Else append the data
    for i in range(1, m+r+1):
        if(i == 2**j):
            res = res + '0'
            j += 1
        else:
            res = res + data[-1 * k]
            k += 1
    # The result is reversed since positions are counted backwards. (m + r+1
... 1)
    return res[::-1]

def calcParityBits(arr, r):
    n = len(arr)
    # For finding rth parity bit, iterate over

```

```

# 0 to r - 1
for i in range(r):
    val = 0
    for j in range(1, n + 1):

        # If position has 1 in ith significant
        # position then Bitwise OR the array value
        # to find parity bit value.
        if(j & (2**i) == (2**i)):
            val = val ^ int(arr[-1 * j])
            # -1 * j is given since array is reversed

    # String Concatenation
    # (0 to n - 2^r) + parity bit + (n - 2^r + 1 to n)
    arr = arr[:n-(2**i)] + str(val) + arr[n-(2**i)+1:]
return arr

def detectError(arr, nr):
    n = len(arr)
    res = 0

    # Calculate parity bits again
    for i in range(nr):
        val = 0
        for j in range(1, n + 1):
            if(j & (2**i) == (2**i)):
                val = val ^ int(arr[-1 * j])

    # Create a binary no by appending
    # parity bits together.

    res = res + val*(10**i)

    # Convert binary to decimal
    return int(str(res), 2)

# Enter the data to be transmitted
data = '1011001'

# Calculate the no of Redundant Bits Required
m = len(data)
r = calcRedundantBits(m)

# Determine the positions of Redundant Bits
arr = posRedundantBits(data, r)

# Determine the parity bits
arr = calcParityBits(arr, r)

```

```

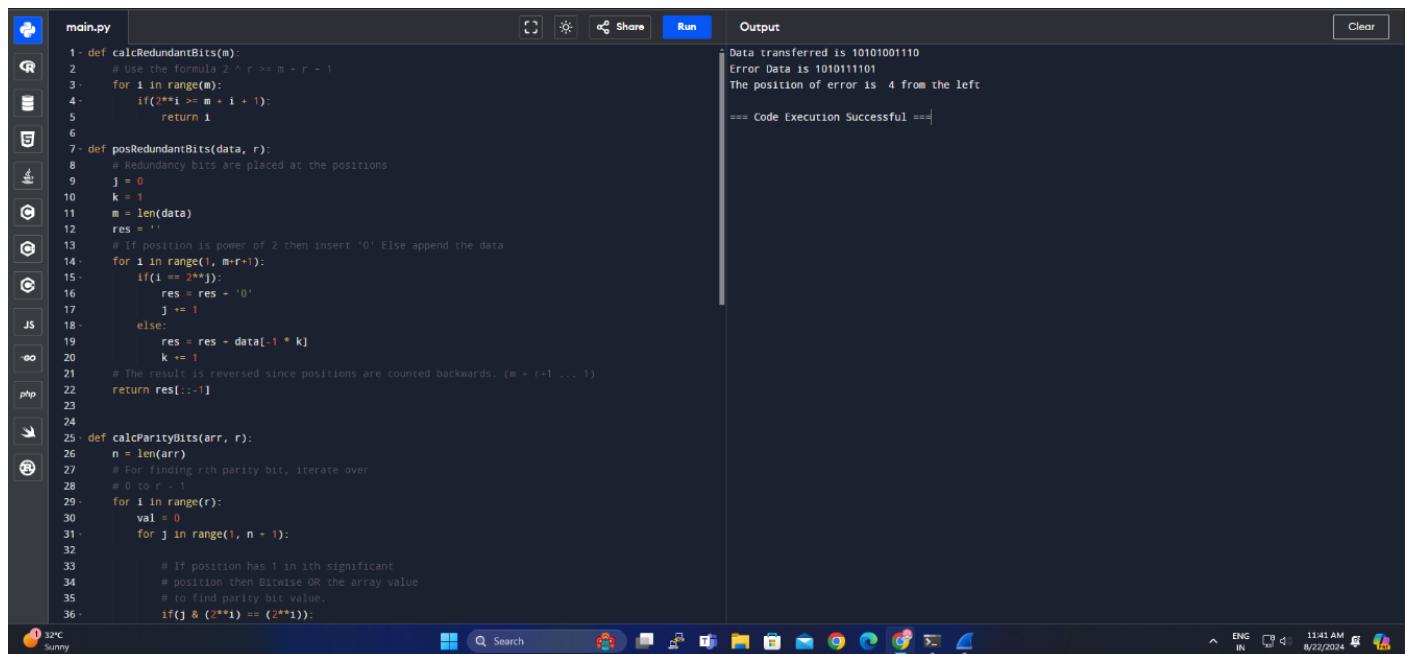
# Data to be transferred
print("Data transferred is " + arr)

# Stimulate error in transmission by changing
# a bit value.
# 10101001110 -> 11101001110, error in 10th position.

arr = '10101001110'
print("Error Data is " + arr)
correction = detectError(arr, r)
if(correction==0):
    print("There is no error in the received message.")
else:
    print("The position of error is ",len(arr)-correction+1,"from the left")

```

## OUTPUT: -



```

main.py | Run | Output | Clear
1- def calcRedundantBits(m):
2     # Use the formula  $2^r \geq m + r + 1$ 
3     for i in range(m):
4         if(2**i >= m + i + 1):
5             return i
6
7- def posRedundantBits(data, r):
8     # Redundancy bits are placed at the positions
9     j = 0
10    k = 1
11    m = len(data)
12    res = ''
13    # If position is power of 2 then insert '0' Else append the data
14    for i in range(1, m+r+1):
15        if(i == 2**j):
16            res = res + '0'
17            j += 1
18        else:
19            res = res + data[-1 * k]
20            k -= 1
21    # The result is reversed since positions are counted backwards. (m + r+1 ... 1)
22    return res[::-1]
23
24
25- def calcParityBits(arr, r):
26    n = len(arr)
27    # For finding rth parity bit, iterate over
28    # 0 to r - 1
29    for i in range(r):
30        val = 0
31        for j in range(i, n + 1):
32
33            # If position has 1 in ith significant
34            # position then Bitwise OR the array value
35            # to find parity bit value.
36            if(j & (2**i) == (2**i)):

```

Output:

```

Data transferred is 10101001110
Error Data is 11101001110
The position of error is 4 from the left
==== Code Execution Successful ====

```

## RESULT: -

The code for HAMMING CODE have been executed successfully and the output is verified.

## EXPERIMENT – 7

**AIM:** - Write a program to implement flow control at data link layer using SLIDING WINDOW PROTOCOL. Simulate the flow of frames from one node to another.

**CODE:** -

```
# include <stdio.h>
int main()
{
    int w,i,f,frames[50];
    printf("Enter window size");
    scanf("%d", &w);
    printf("\n Enter %d frames:", f);
    scanf("%d", &f);
    printf("\n Enter %d frames:", f);

    for (i=1; i<=f; i++)
        scanf("%d", &frames[i]);
    printf("\n With sliding window protocol the frames will be sent
in the following manner (assuming no corruption of frames)\n\n");
    printf("After sending %d frames at each frames at each stage
sender waits for acknowledgement sent by the receiver \n\n", w);

    for(i=1; i<=f;i++)
    {
        if(i%w==0)
        {
            printf("%d\n", frames[i]);
        }
        else
            printf("%d\n", frames[i]);
    }
    if (f%w!=0)
    printf("\n Acknowledgement of above frames sent is received by sender
\n");
    return 0;
}
```

## OUTPUT: -

The screenshot shows the Programiz C Online Compiler interface. On the left, the code file 'main.c' is displayed:

```
1 #include<stdio.h>
2 int main()
3 {
4     int w,i,f,frames[50];
5     printf("Enter window size: ");
6     scanf("%d",&w);
7     printf("\nEnter number of frames to transmit: ");
8     scanf("%d",&f);
9     printf("\nEnter %d frames: ",f);
10    for(i=1;i<=f;i++)
11        scanf("%d",&frames[i]);
12    printf("\nWith sliding window protocol the frames will be sent in the following manner (assuming no
13    corruption of frames)\n\n");
14    printf("After sending %d frames at each stage sender waits for acknowledgement sent by the
15    receiver\n\n",w);
16    for(i=1;i<=f;i++)
17    {
18        if(i%w==0)
19        {
20            printf("%d\n",frames[i]);
21            printf("Acknowledgement of above frames sent is received by sender\n\n");
22        }
23        else
24            printf("%d ",frames[i]);
25    }
26    if(f%w!=0)
27        printf("\nAcknowledgement of above frames sent is received by sender\n");
28    return 0;
29 }
```

In the center, there are several icons for file operations like Open, Save, Share, Run, and Output. The 'Run' button is highlighted. To the right of the Run button is a 'Premium Coding Courses by Programiz' banner featuring a person working on a laptop and various icons. Below the banner is a 'Programiz PRO' button.

The 'Output' section on the right shows the program's execution:

```
/tmp/C6dvbbKouu.o
Enter window size: 5
Enter number of frames to transmit: 6
Enter 6 frames: 15 16 17 18 19 20
With sliding window protocol the frames will be sent in the following manner (assuming no corruption of
frames)
After sending 5 frames at each stage sender waits for acknowledgement sent by the receiver
15 16 17 18 19
Acknowledgement of above frames sent is received by sender
20
Acknowledgement of above frames sent is received by sender
--- Code Execution Successful ---
```

At the bottom of the window, there is a toolbar with icons for search, file operations, and other utilities. The status bar at the bottom right shows the date and time: 8/29/2024 11:08 AM.

## RESULT: -

The code for SLIDING WINDOW have been executed successfully and the output is verified.

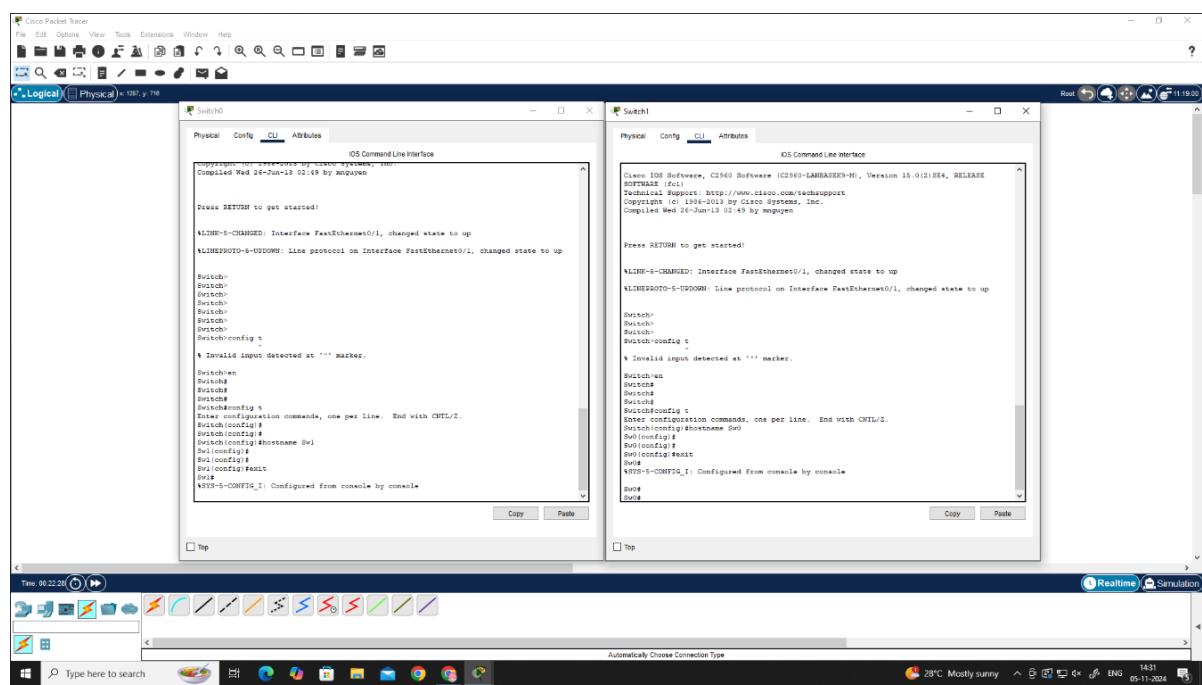
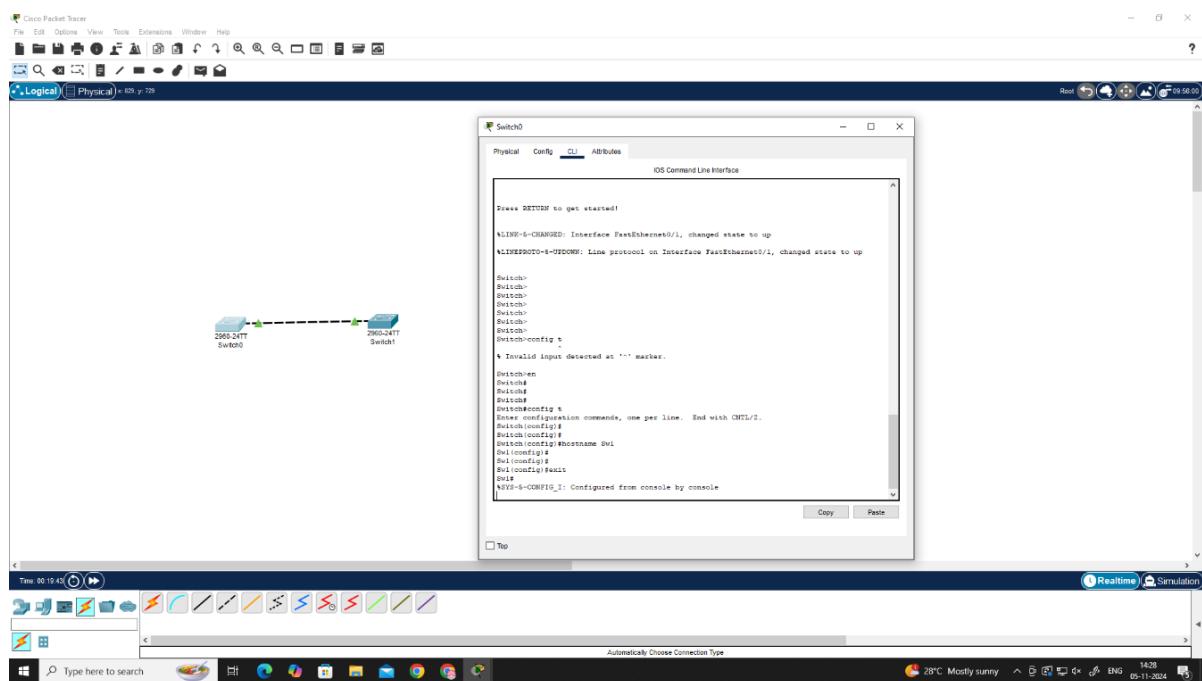
## EXPERIMENT – 8

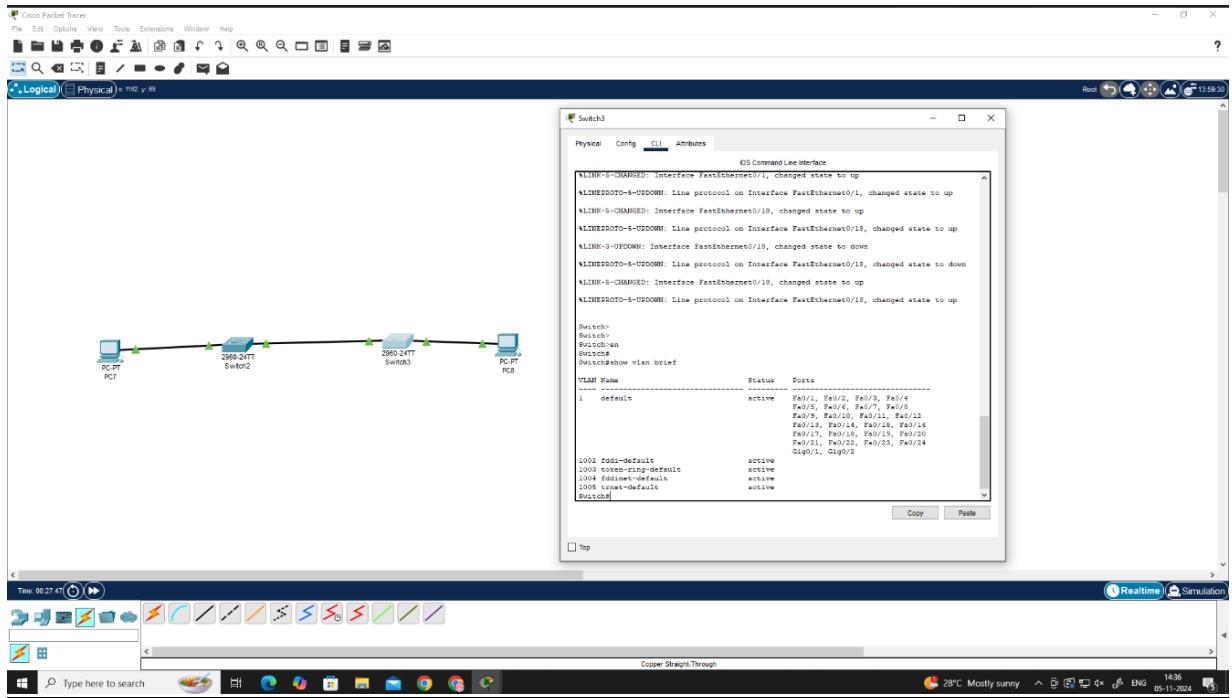
AIM: - a) Simulate Virtual LAN configuration using CISCO Packet Tracer Simulation.

### Steps:

- Step 1: Build the network as shown in the topology.
- Step 2: Configure basic settings for each switch.
- Step 3: Configure PC hosts.
- Step 4: Test connectivity.

### OUTPUT: -





## RESULT: -

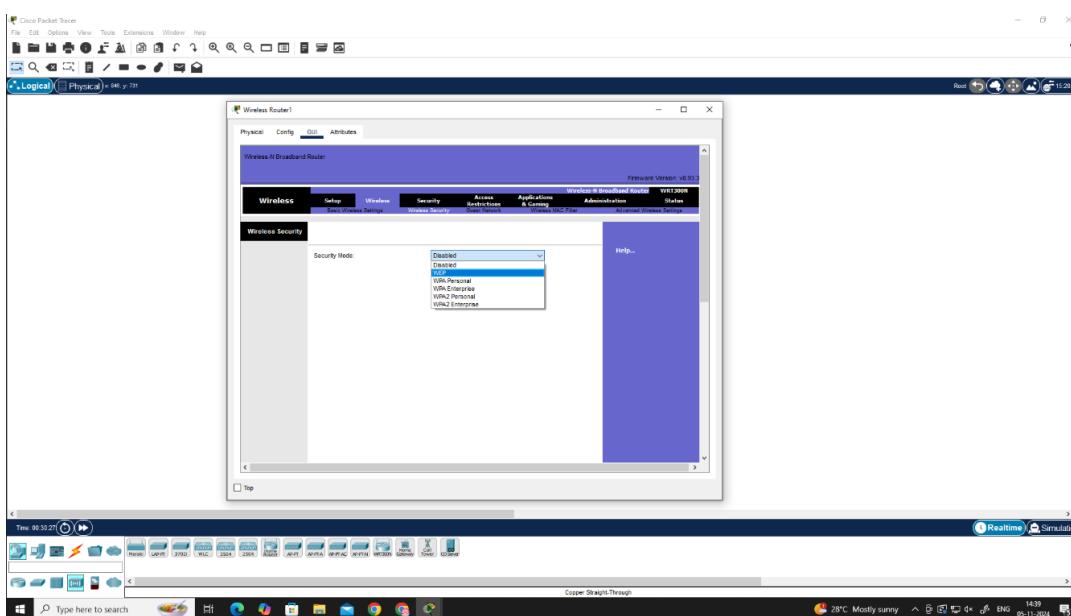
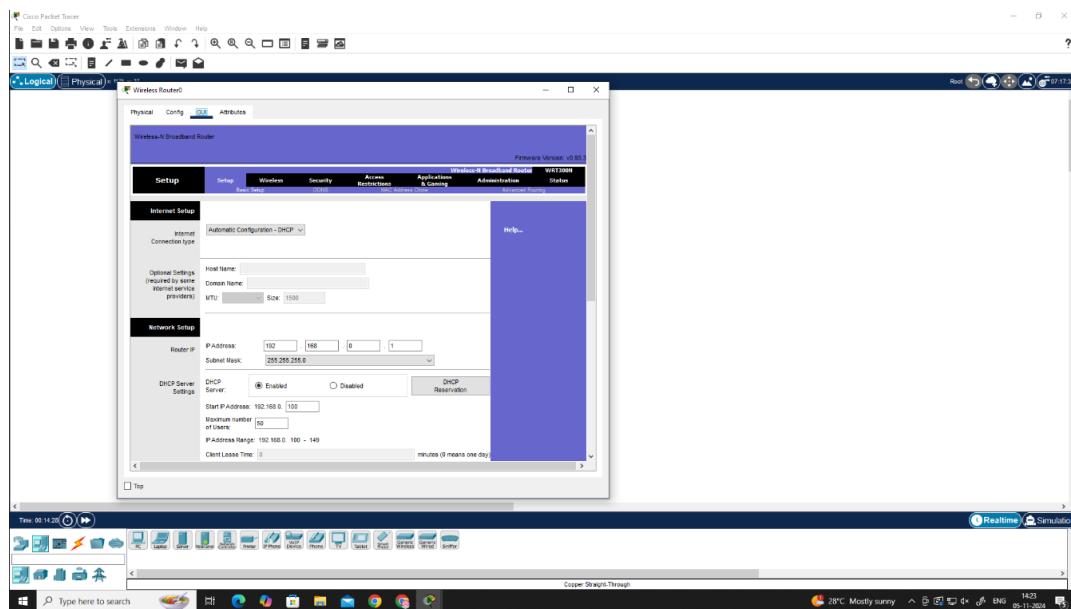
Simulation of virtual LAN configuration have been done successfully.

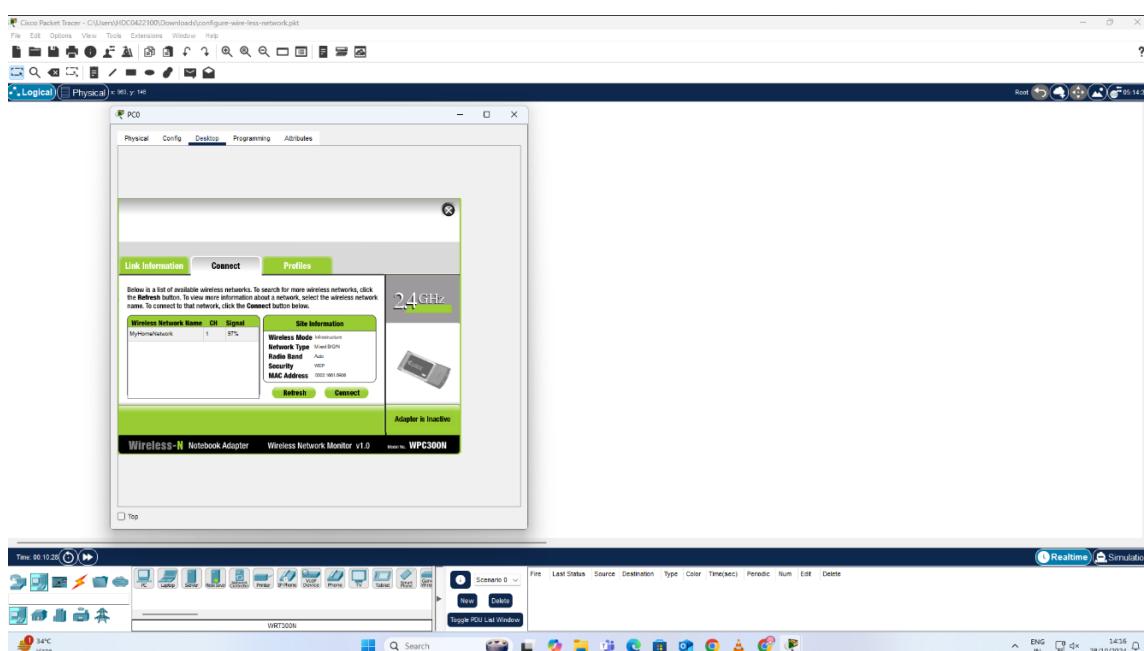
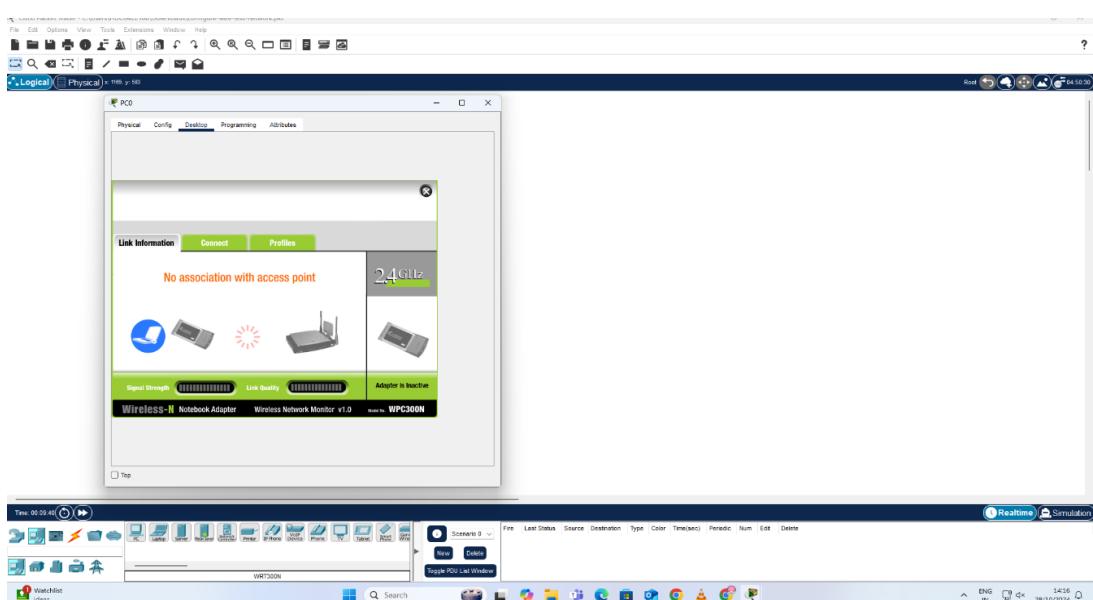
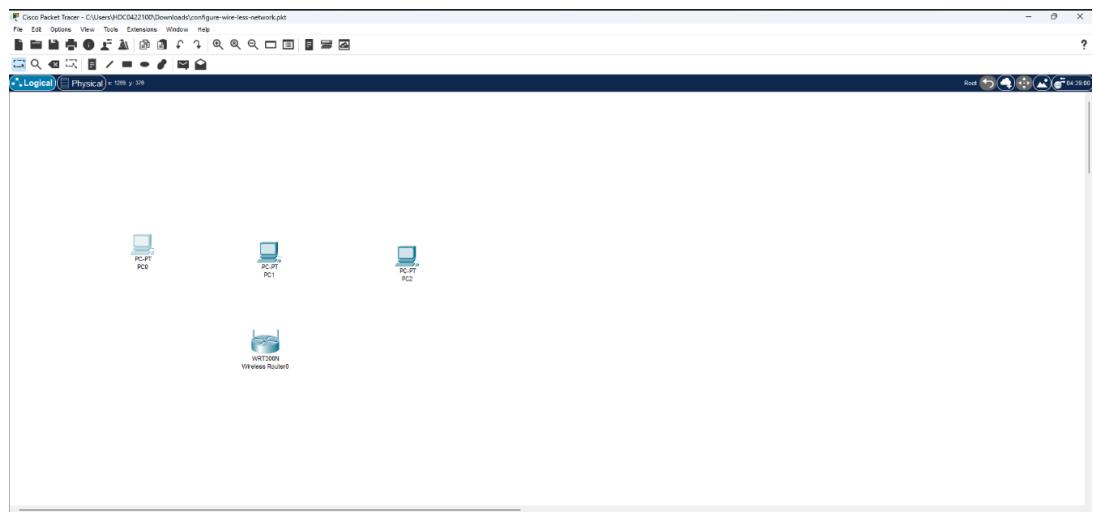
## AIM: - b) Configuration of Wireless LAN using CISCO Packet Tracer.

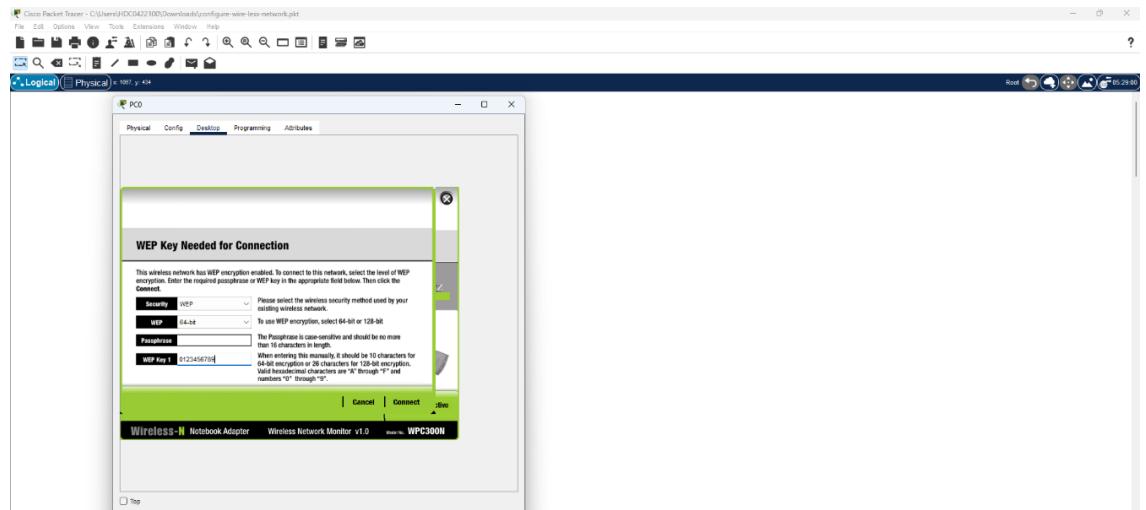
### Steps:

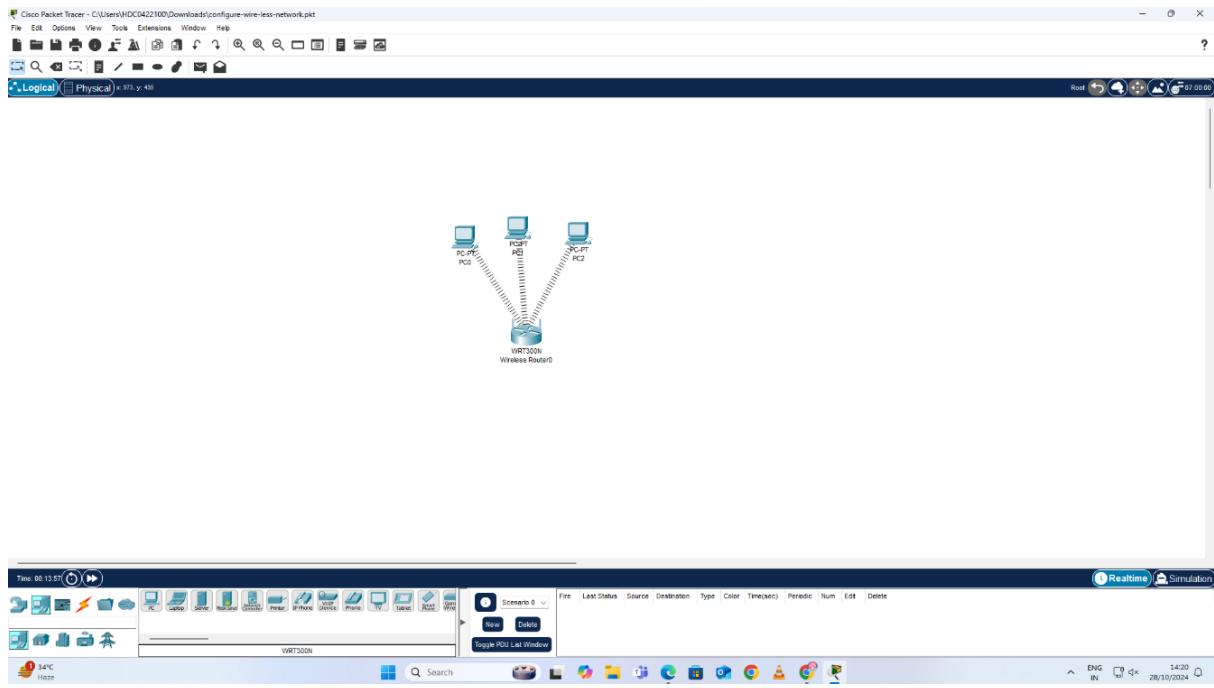
- Configure Static IP on PC and Wireless Router
- Set SSID to MotherNetwork
- Set IP address of router to 192.168.0.1, PC0 to 192.168.0.2, PC1 to 192.168.0.3 and PC2 to 192.168.0.4.
- Secure your network by configuring WAP key on Router
- Connect PC by using WAP key

### OUTPUT: -









## **RESULT: -**

Configuration of Wireless LAN using CISCO Packet Tracer have been done successfully and verified.

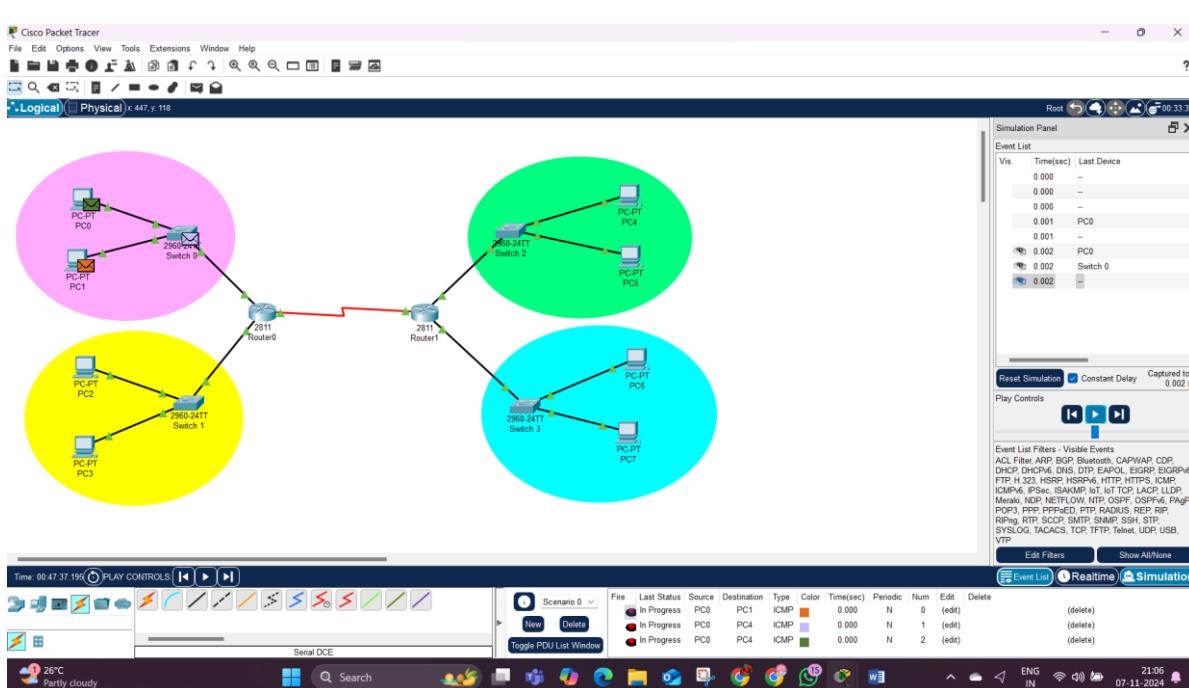
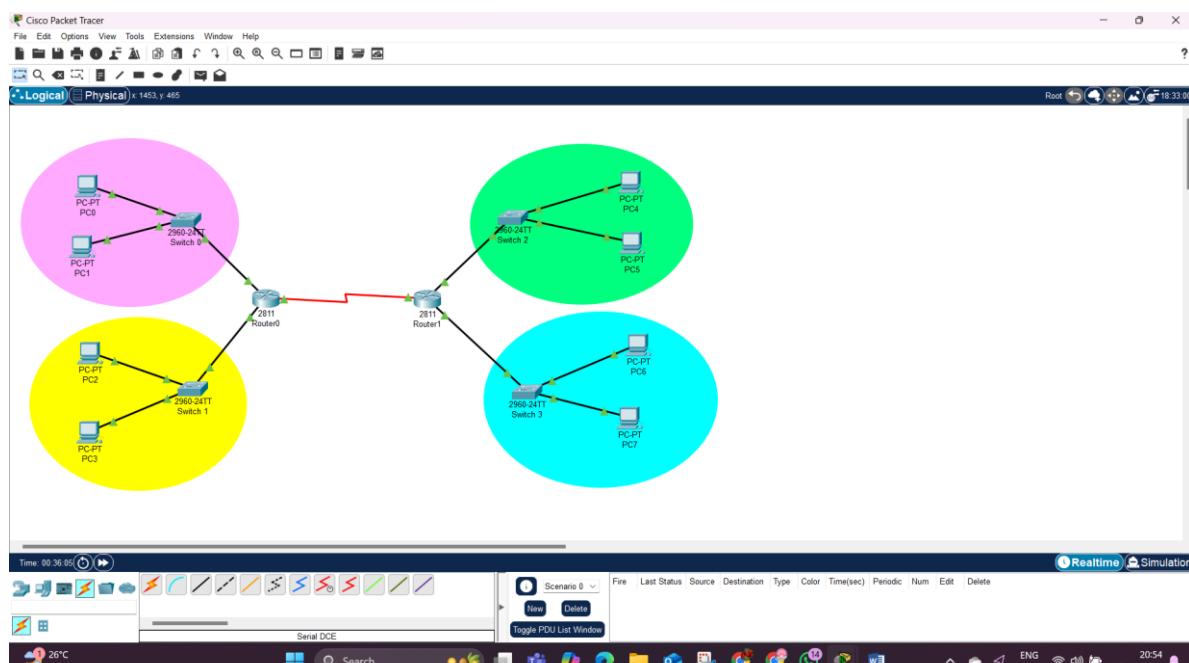
## EXPERIMENT – 9

**AIM:** - Implementation of SUBNETTING in CISCO PACKET TRACER simulator.

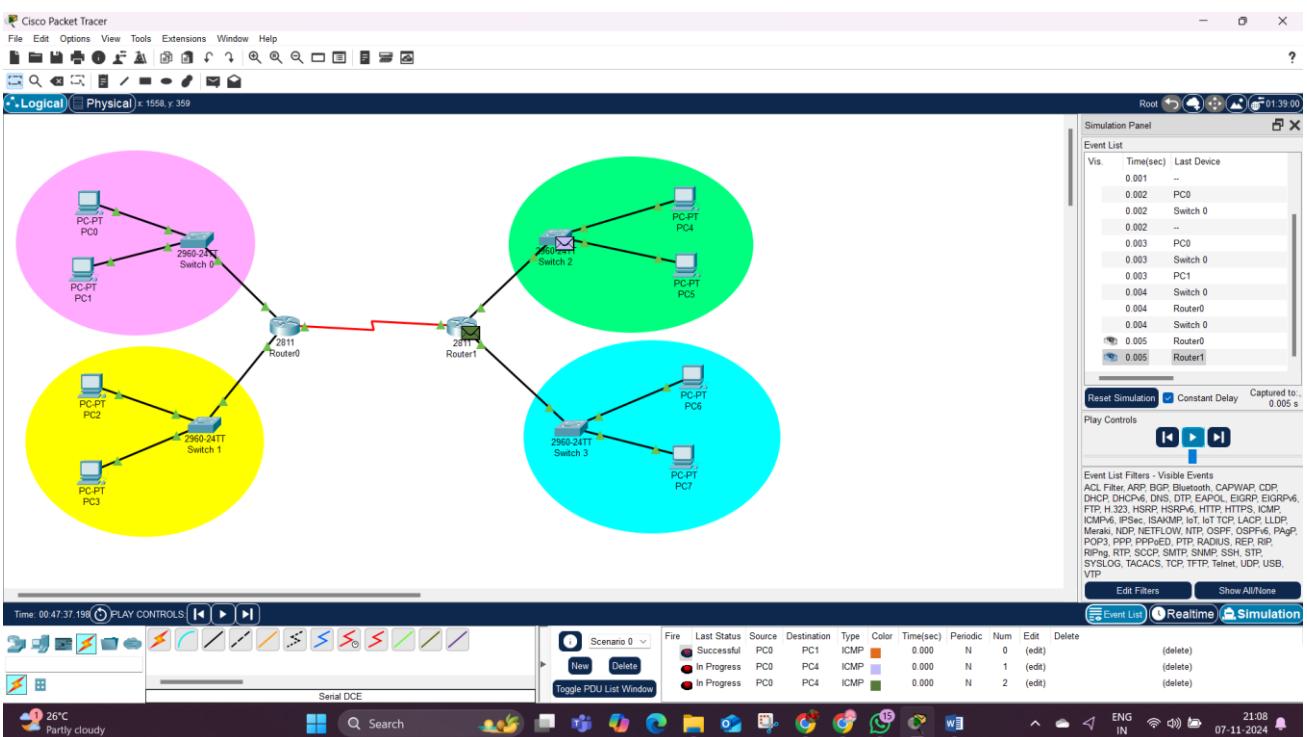
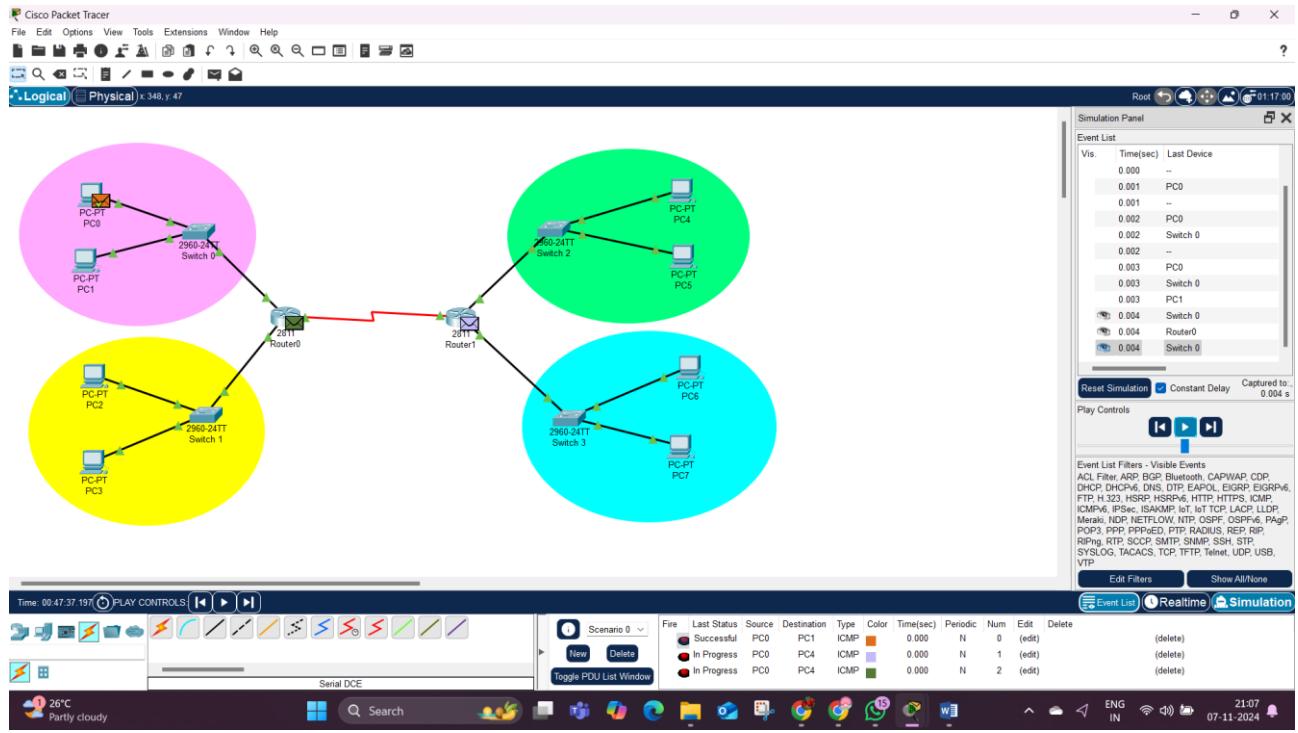
### What is subnetting?

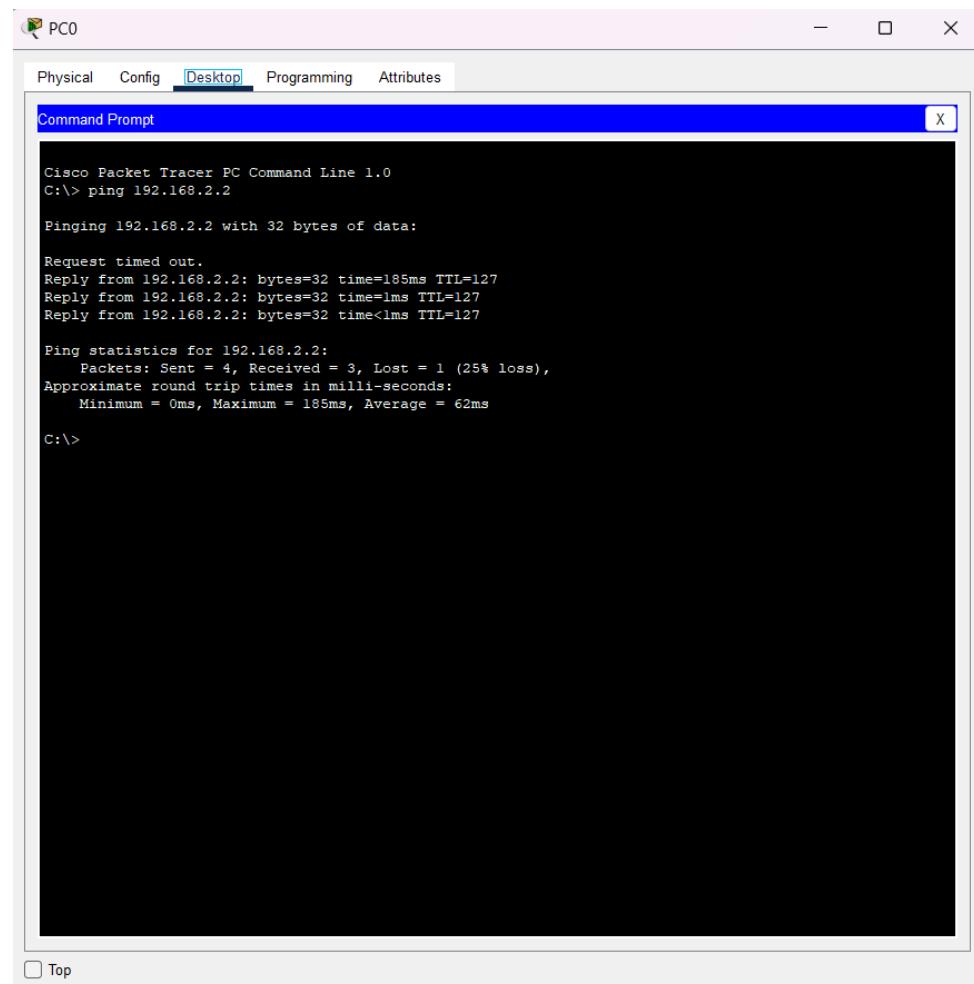
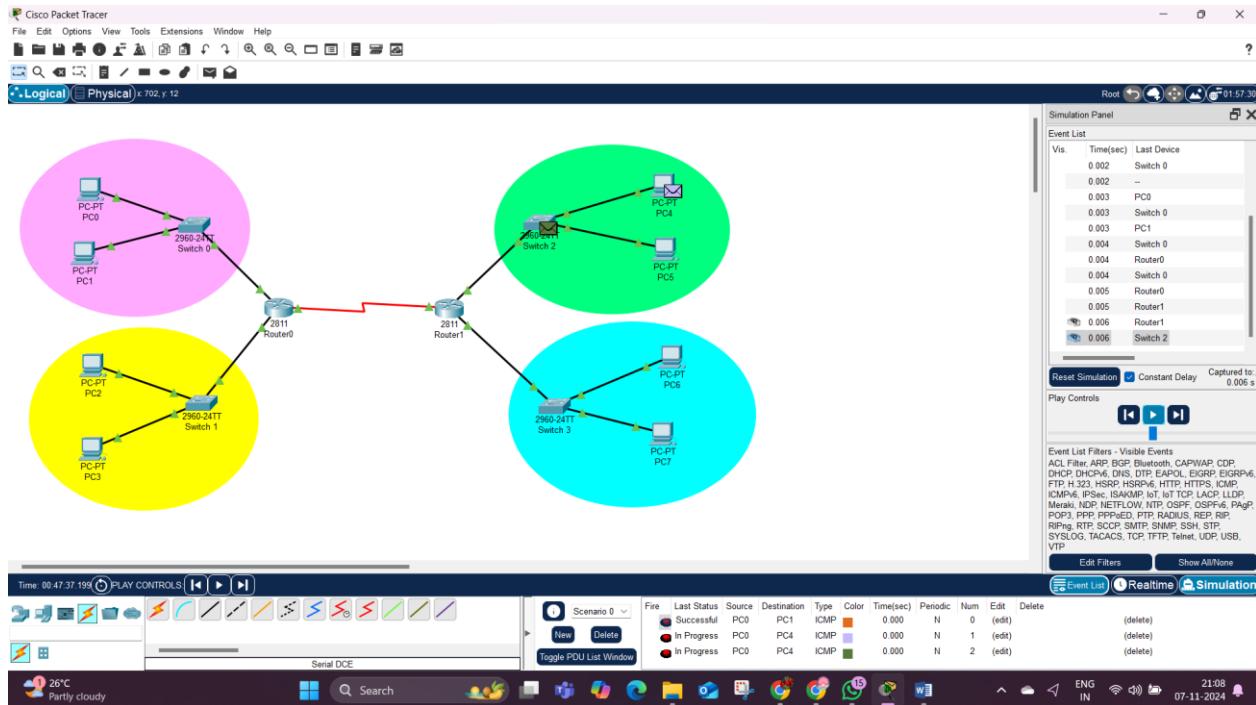
Classless IP subnetting is a technique that allows for more efficient use of IP addresses by allowing for subnet masks that are not just the default masks for each IP class. This means that we can divide our IP address space into smaller subnets, which can be useful when we have a limited number of IP addresses but need to create multiple networks.

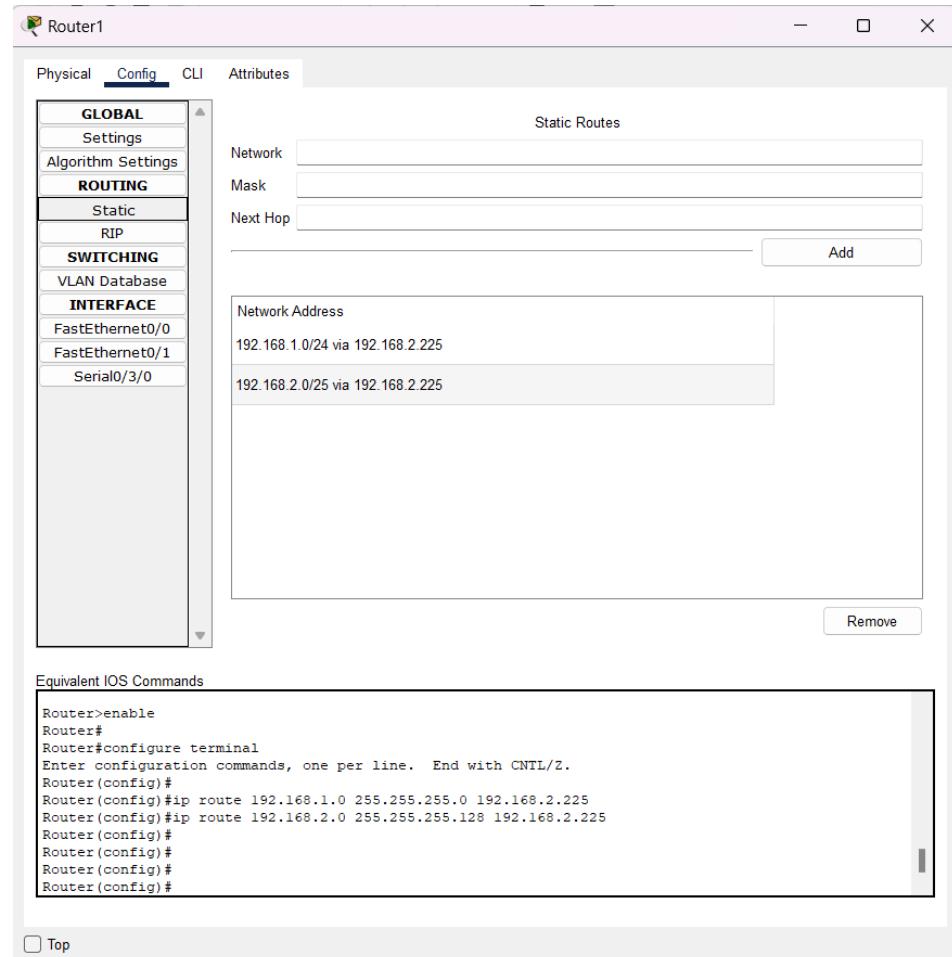
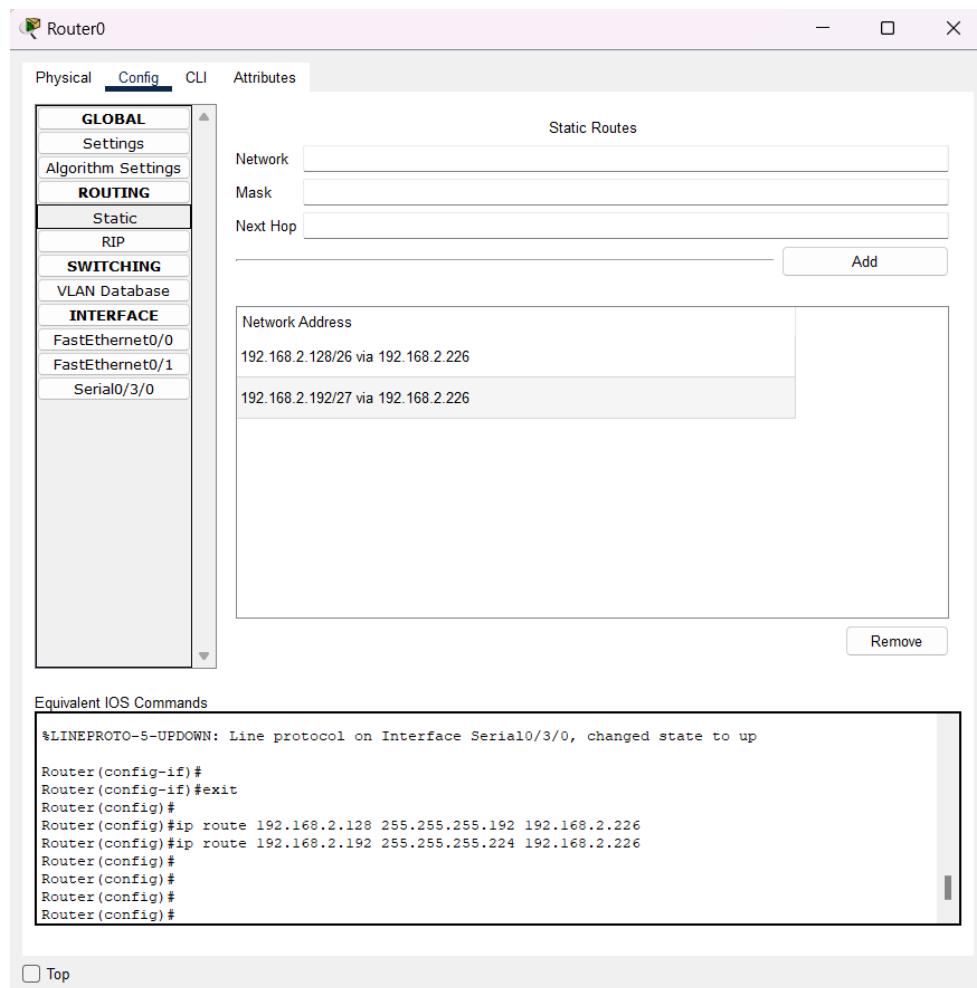
### OUTPUT: -



Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
Successful	PC0	PC1	ICMP	Orange	0.000	N	0	(edit)	(delete)	
Failed	PC0	PC4	ICMP	Light Blue	0.000	N	1	(edit)	(delete)	
Successful	PC0	PC4	ICMP	Dark Green	0.000	N	2	(edit)	(delete)	







The screenshot shows a Cisco Packet Tracer interface with a window titled "Command Prompt". The window displays several "ping" commands and their results. The results show varying levels of network reachability and latency.

```
Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 0ms, Maximum = 185ms, Average = 62ms  
  
C:\> ping 192.168.2.129  
  
Pinging 192.168.2.129 with 32 bytes of data:  
  
Reply from 192.168.1.100: Destination host unreachable.  
  
Ping statistics for 192.168.2.129:  
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),  
  
C:\> ping 192.168.2.193  
  
Pinging 192.168.2.193 with 32 bytes of data:  
  
Request timed out.  
Reply from 192.168.2.193: bytes=32 time=10ms TTL=126  
Reply from 192.168.2.193: bytes=32 time=16ms TTL=126  
Reply from 192.168.2.193: bytes=32 time=3ms TTL=126  
  
Ping statistics for 192.168.2.193:  
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 3ms, Maximum = 16ms, Average = 9ms  
  
C:\> ping 192.168.2.193  
  
Pinging 192.168.2.193 with 32 bytes of data:  
  
Reply from 192.168.2.193: bytes=32 time=24ms TTL=126  
Reply from 192.168.2.193: bytes=32 time=10ms TTL=126  
Reply from 192.168.2.193: bytes=32 time=10ms TTL=126  
Reply from 192.168.2.193: bytes=32 time=1ms TTL=126  
  
Ping statistics for 192.168.2.193:  
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),  
Approximate round trip times in milli-seconds:  
    Minimum = 1ms, Maximum = 24ms, Average = 11ms  
  
C:\>
```

## **RESULT: -**

Implementation of SUBNETTING in CISCO PACKET TRACER simulator have been done successfully

## EXPERIMENT – 10

AIM: - a) Internetworking with routers in CISCO PACKET TRACER simulator.

OUTPUT: -

The image shows two side-by-side Command Prompt windows from the Cisco Packet Tracer software. Both windows have tabs for Physical, Config, Desktop, Programming, and Attributes, with Desktop selected. The left window is titled 'PC0' and the right is '10.0.0.2'. Each window has a 'Command Prompt' tab where ping commands are run. The output of the pings is displayed in the windows.

**PC0 Command Prompt Output:**

```
C:\>ping 10.0.0.3
Pinging 10.0.0.3 with 32 bytes of data:
Reply from 10.0.0.3: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 10.0.0.2
Pinging 10.0.0.2 with 32 bytes of data:
Reply from 10.0.0.2: bytes=32 time<1ms TTL=128

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```

**10.0.0.2 Command Prompt Output:**

```
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

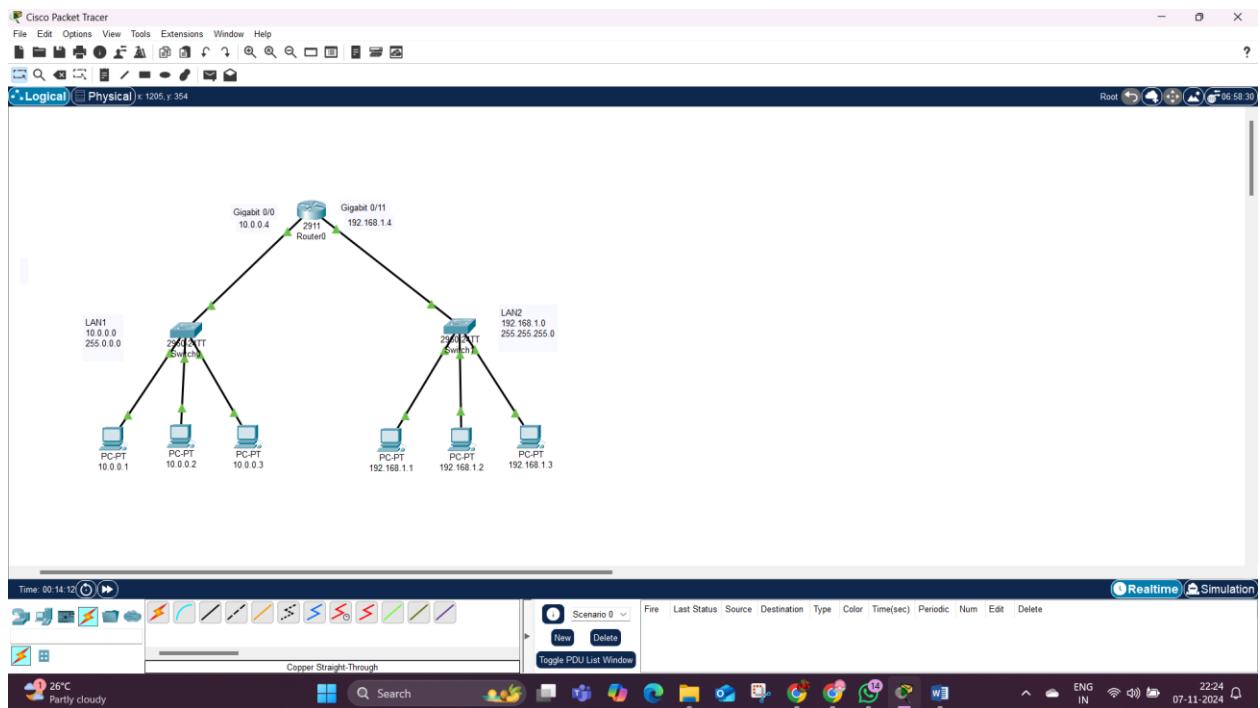
C:\>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.1: bytes=32 time=10ms TTL=127
Reply from 192.168.1.1: bytes=32 time=10ms TTL=127
Reply from 192.168.1.1: bytes=32 time=10ms TTL=127

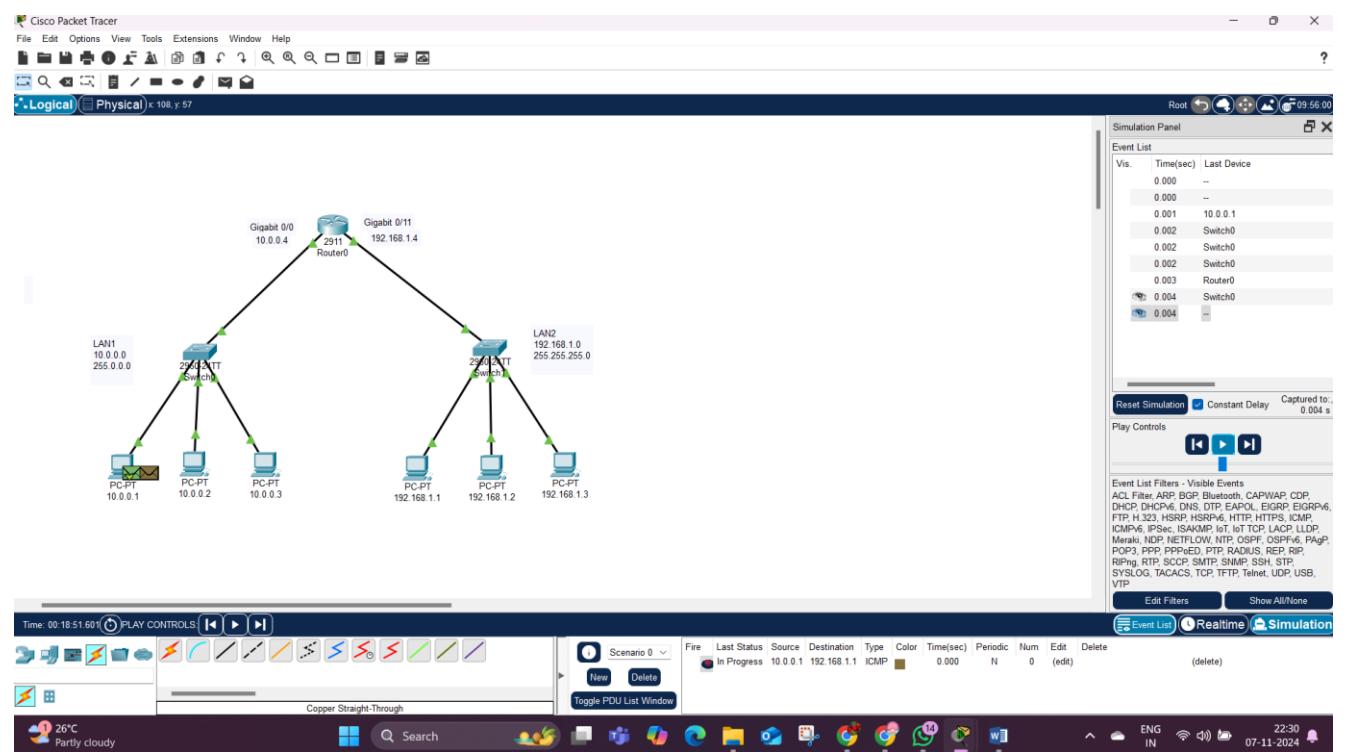
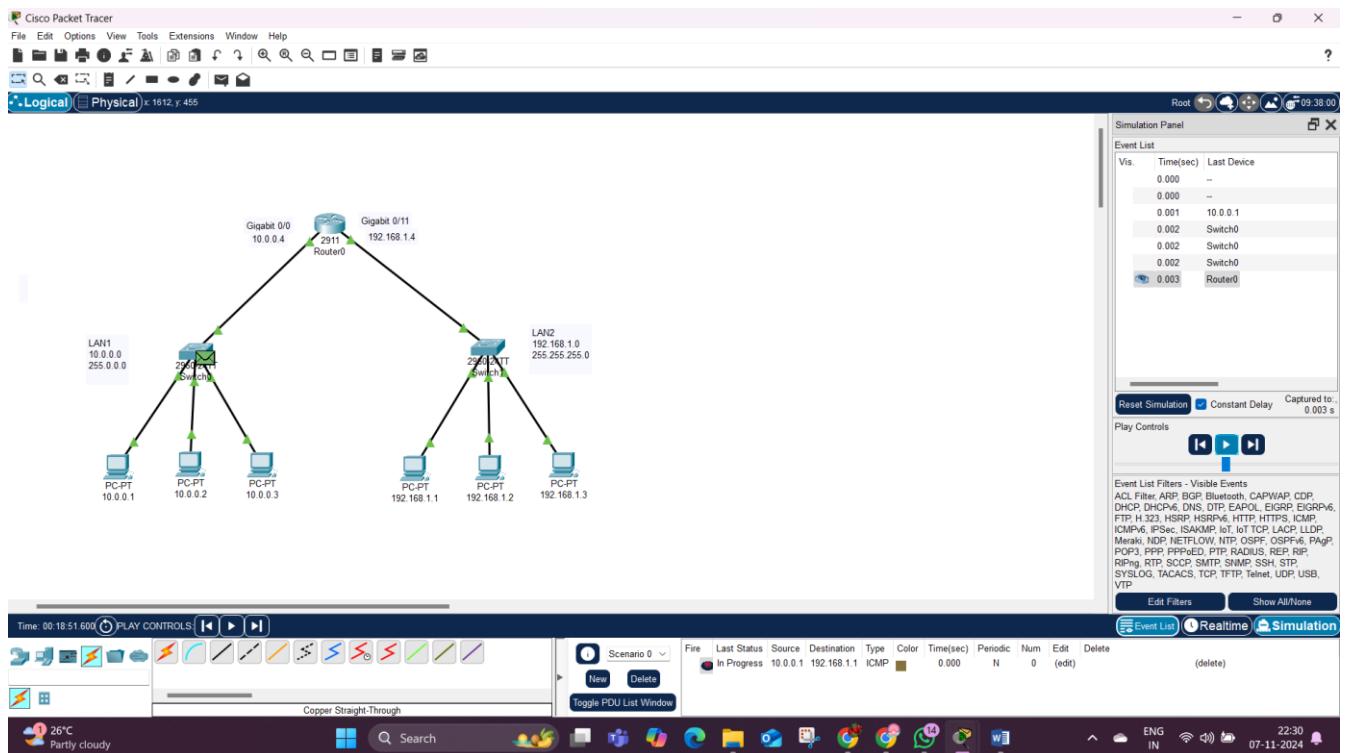
Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 3ms

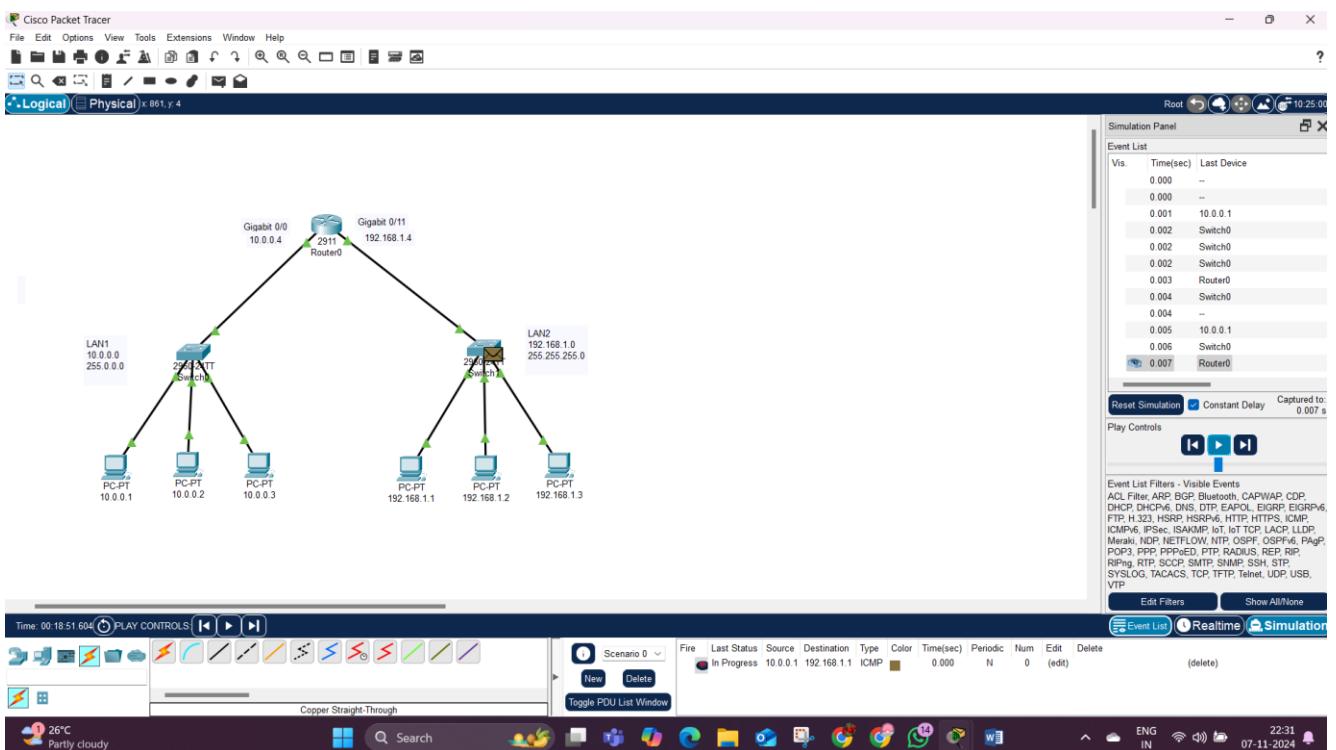
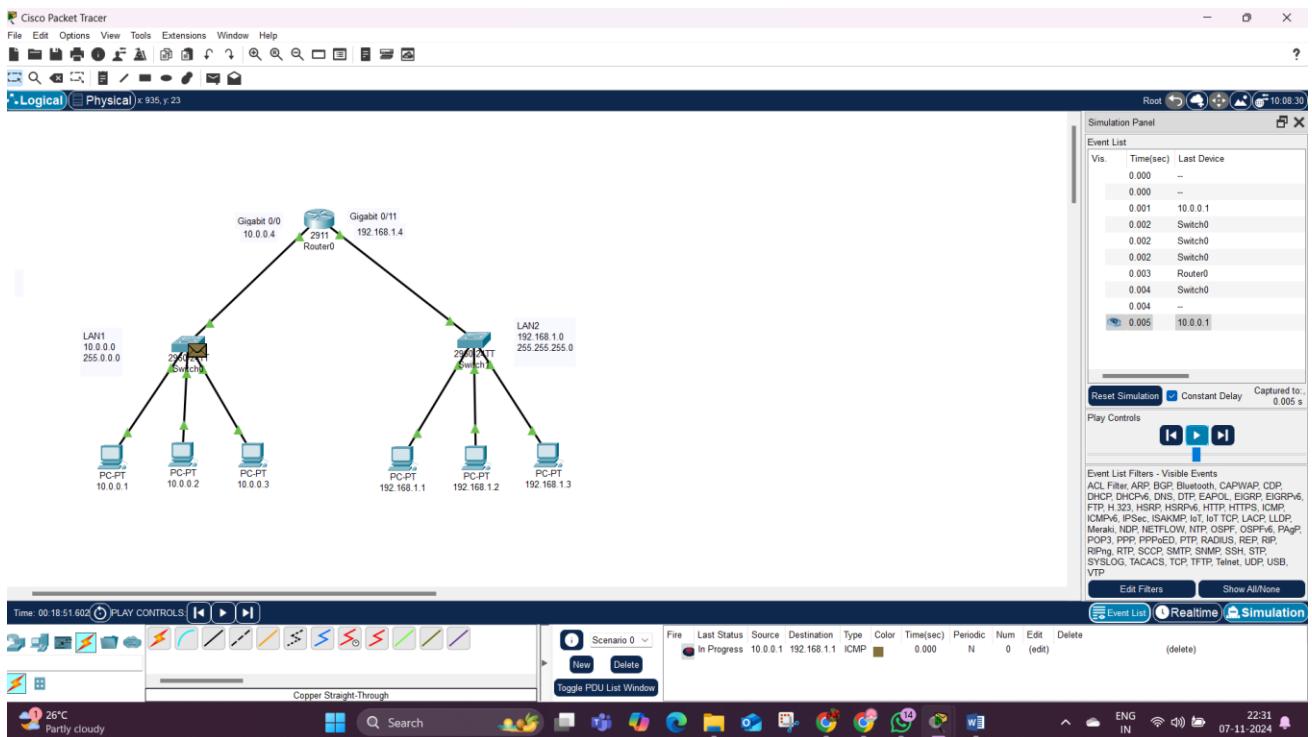
C:\>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.2: bytes=32 time=12ms TTL=127
Reply from 192.168.1.2: bytes=32 time=12ms TTL=127
Reply from 192.168.1.2: bytes=32 time=12ms TTL=127

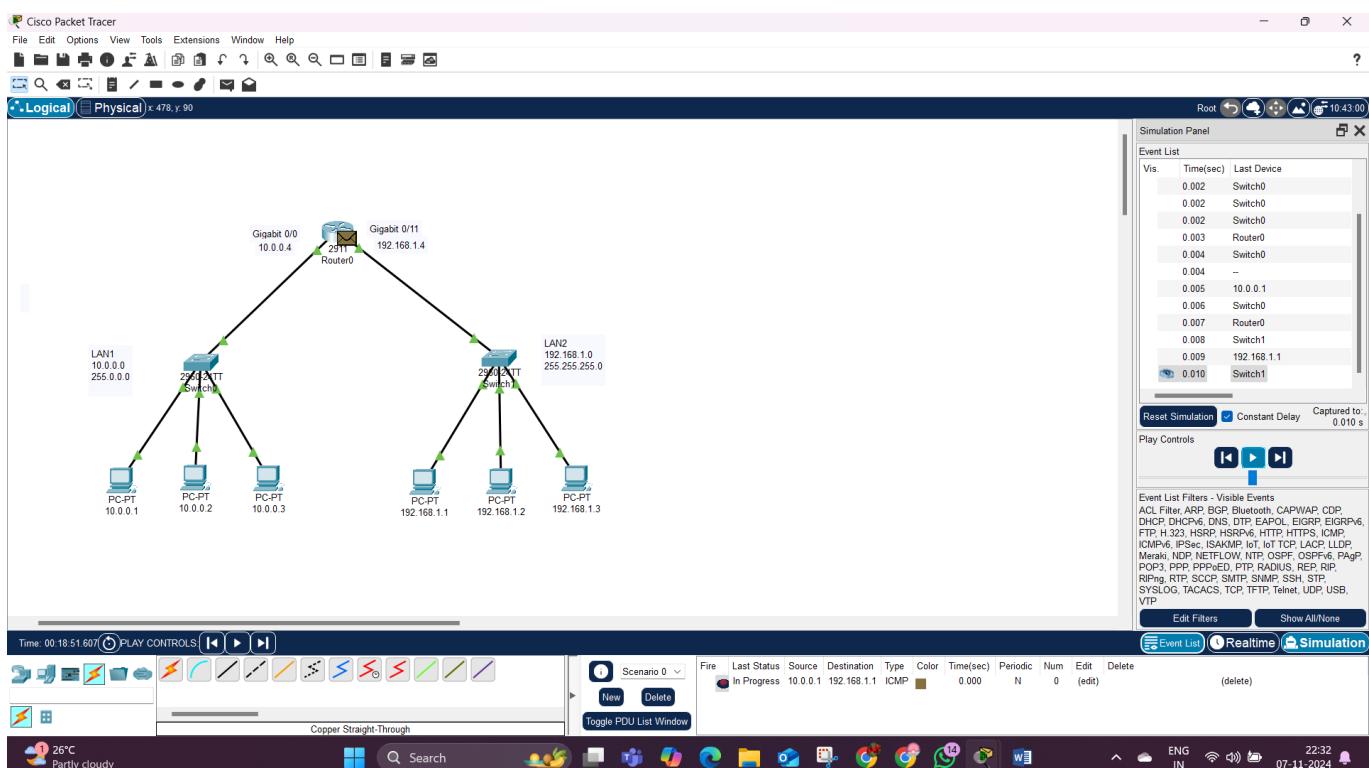
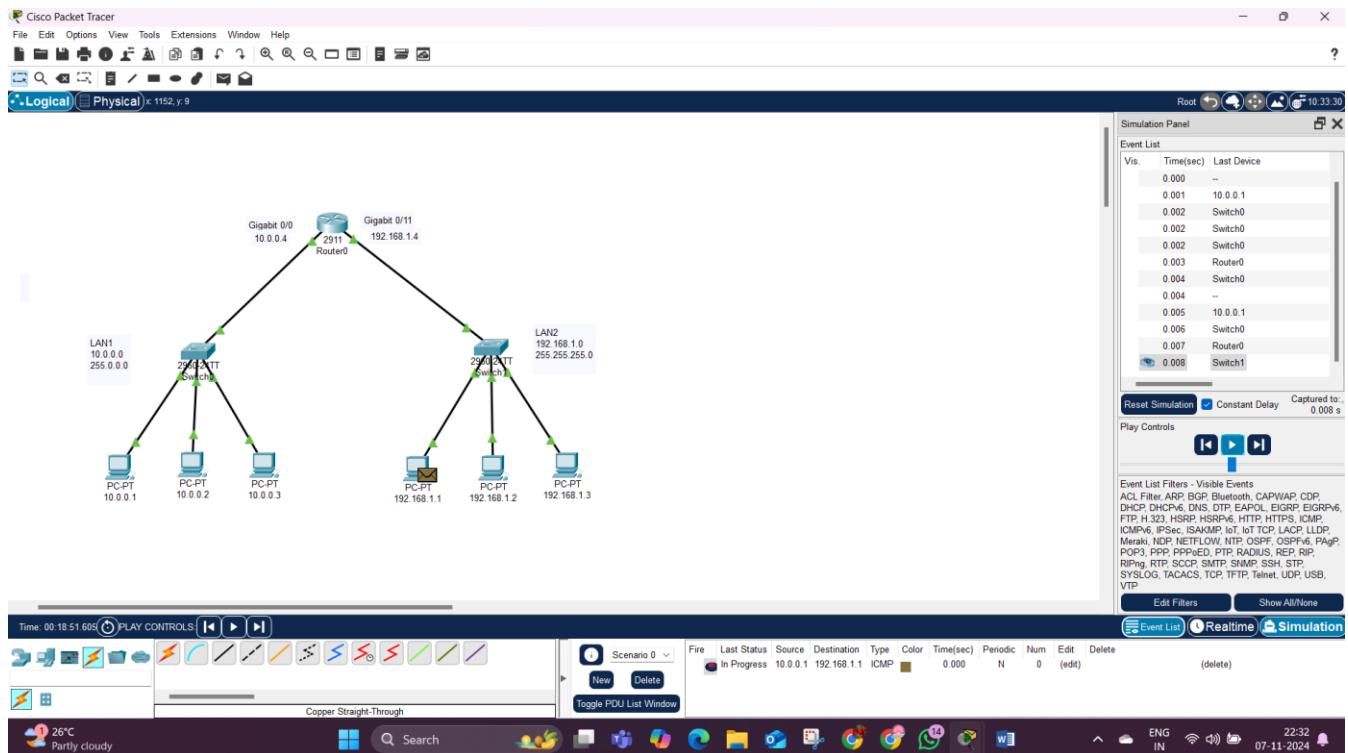
Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 12ms, Average = 4ms

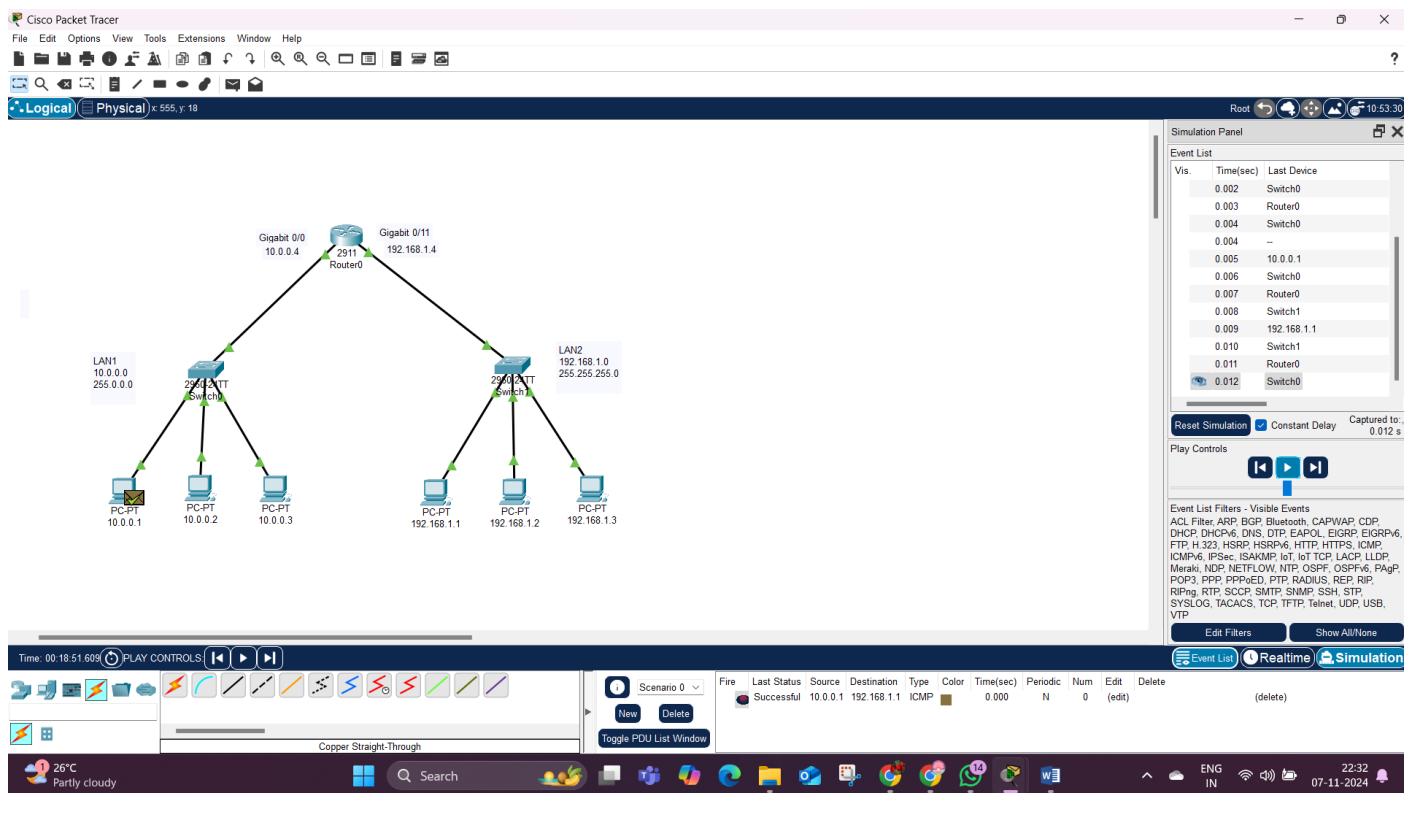
C:\>
```











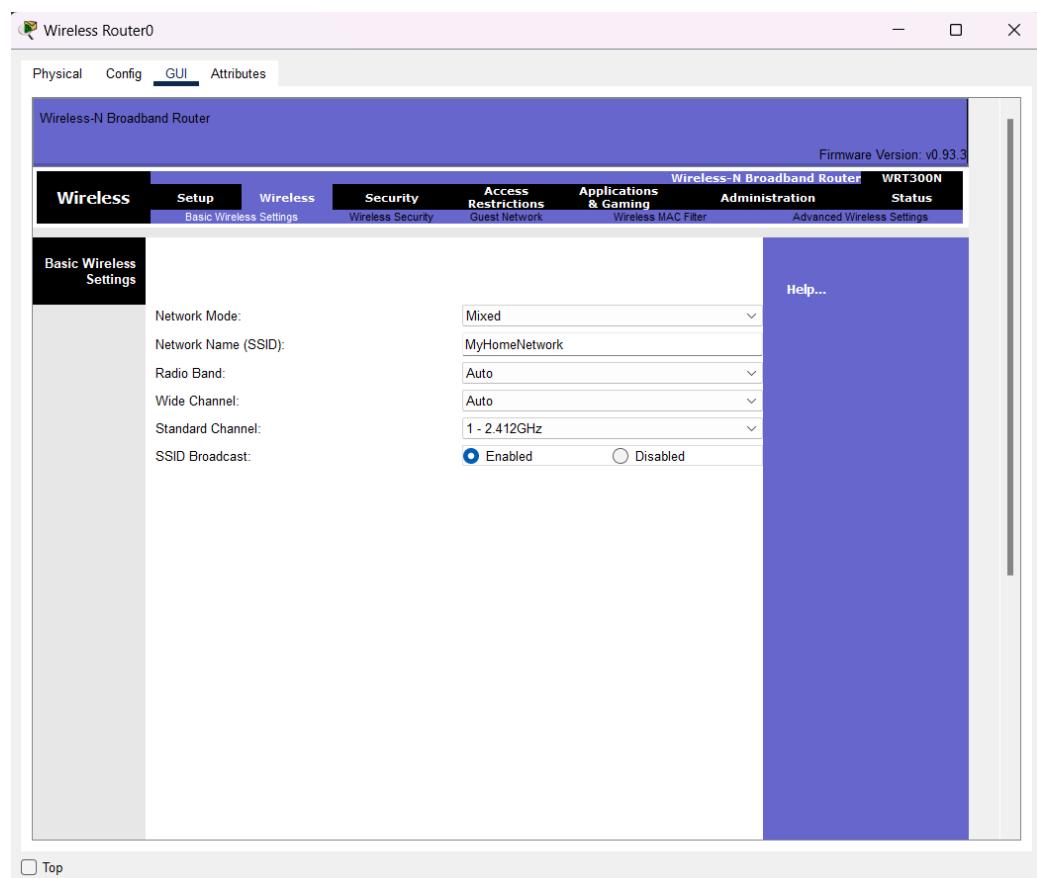
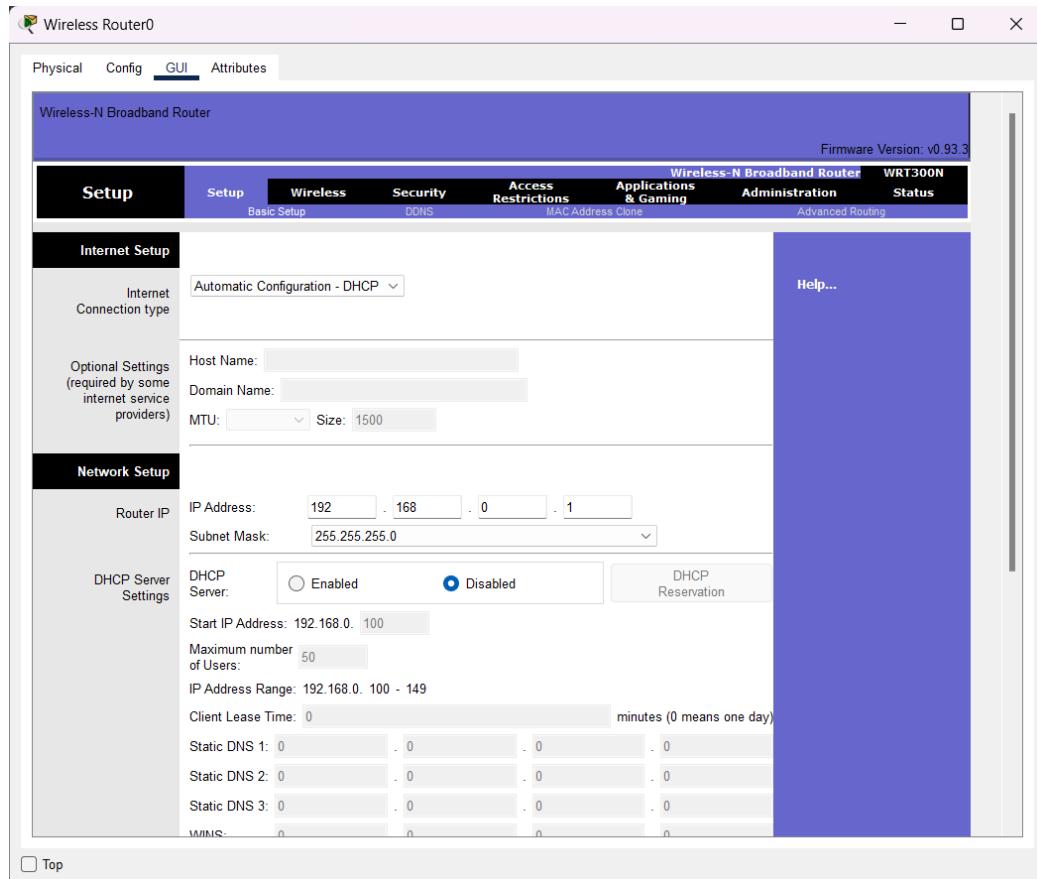
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
	Successful	10.0.0.1	192.168.1.1	ICMP	<span style="background-color: #8B4513; border: 1px solid black; padding: 2px 5px;"></span>	0.000	N	0	(edit)	<a href="#">(delete)</a>

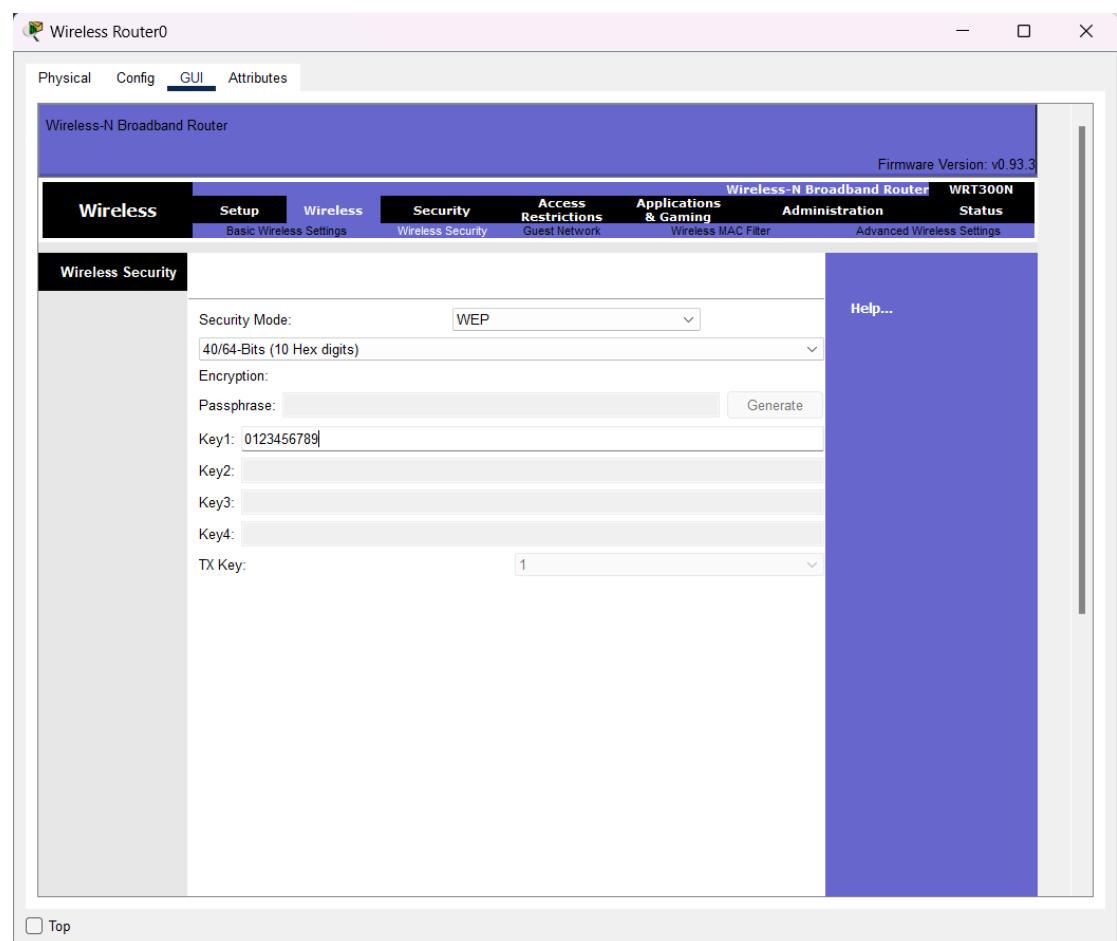
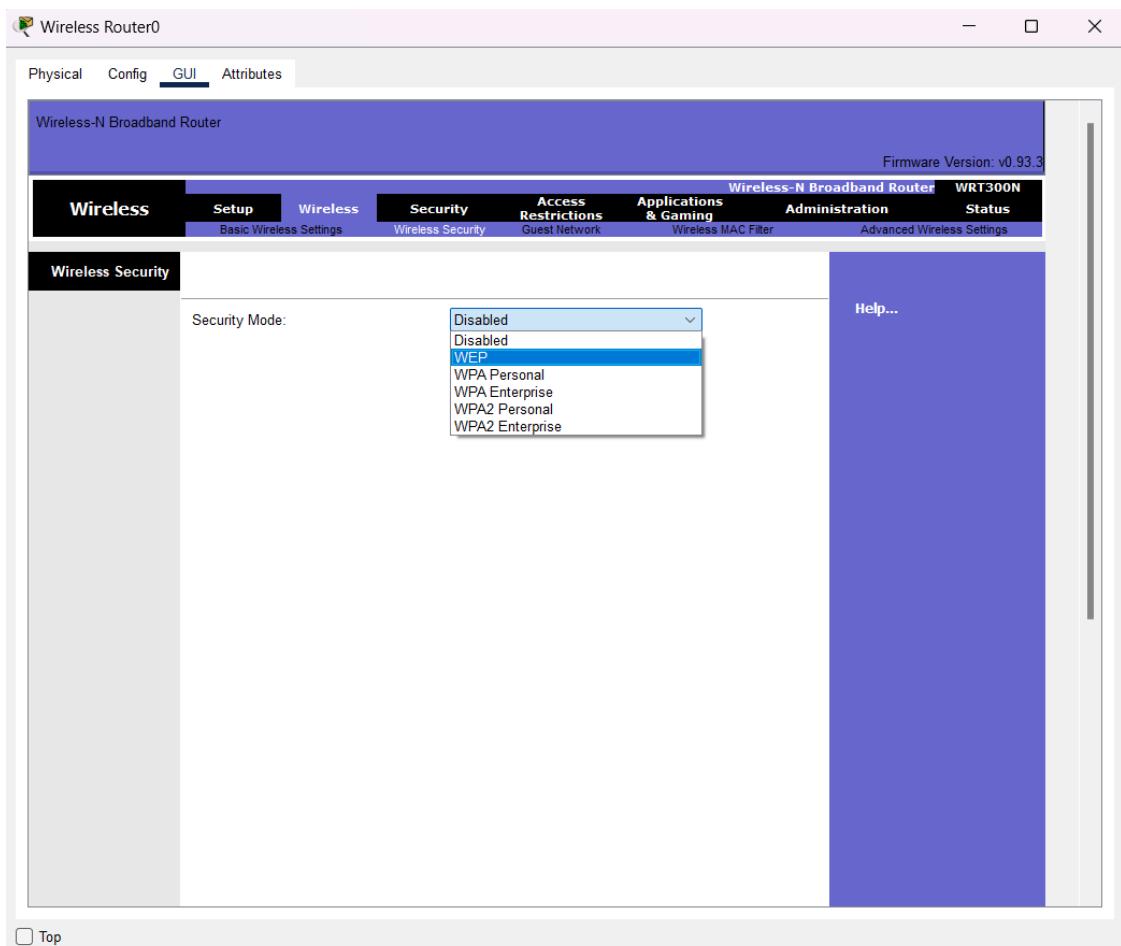
## RESULT: -

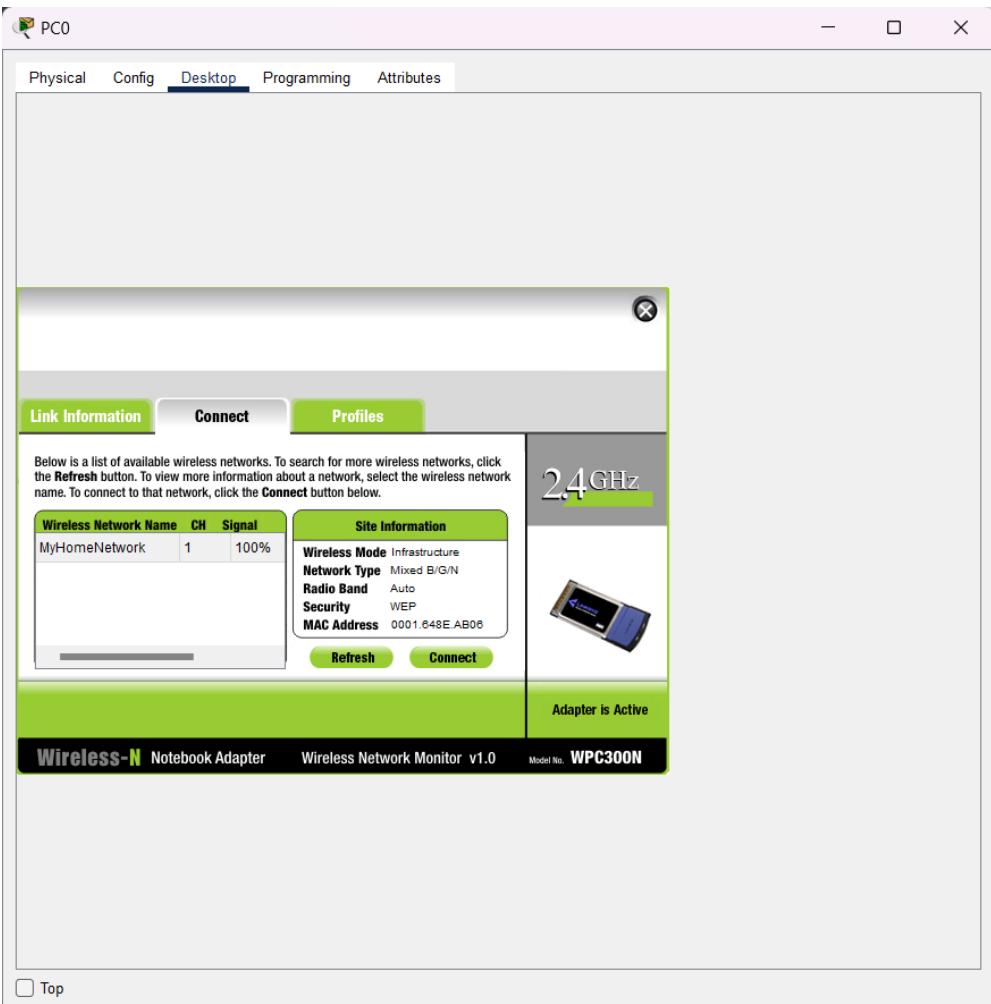
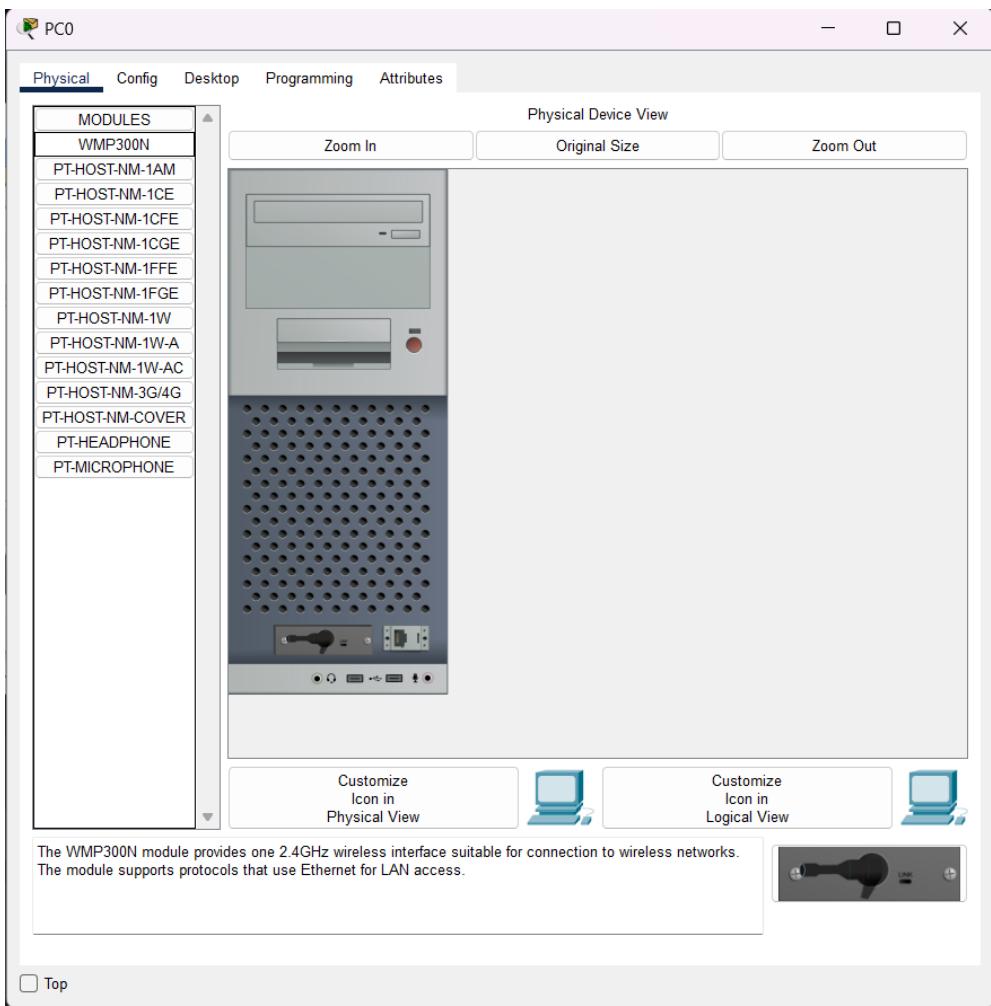
Router have been successfully done in CISCO PACKET TRACER.

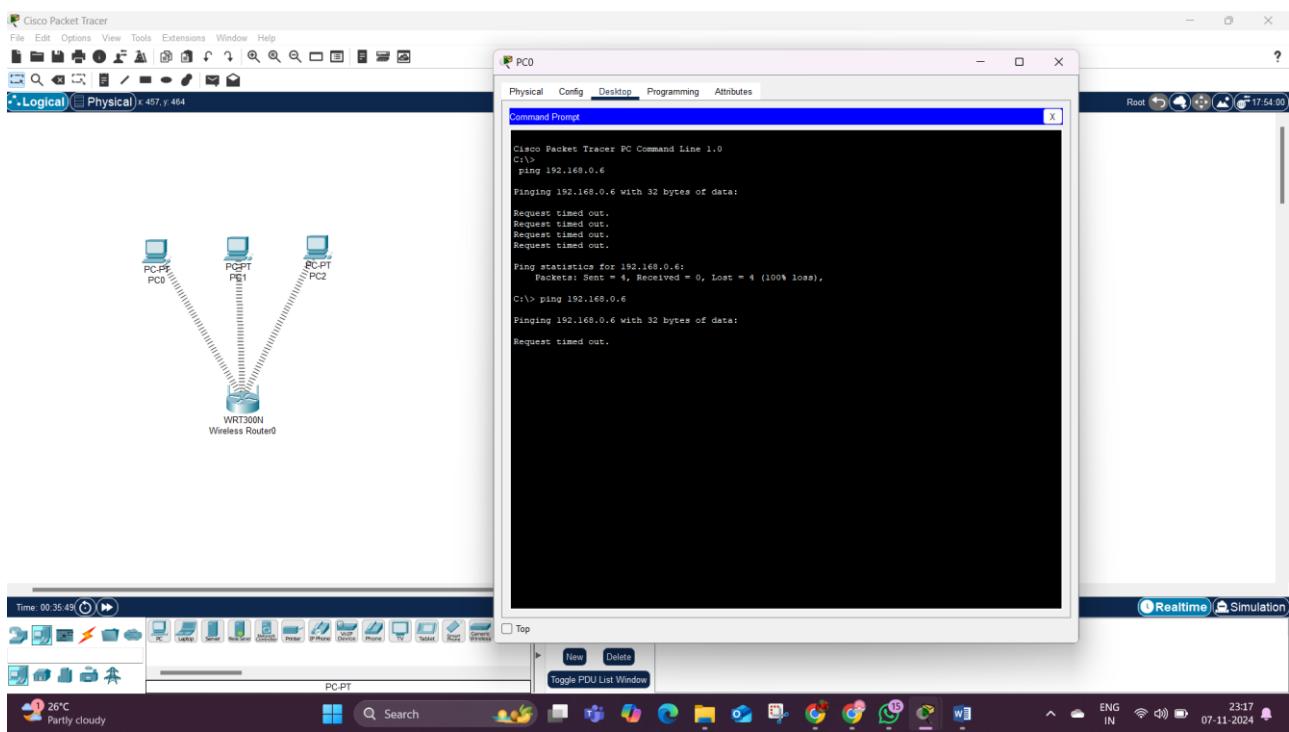
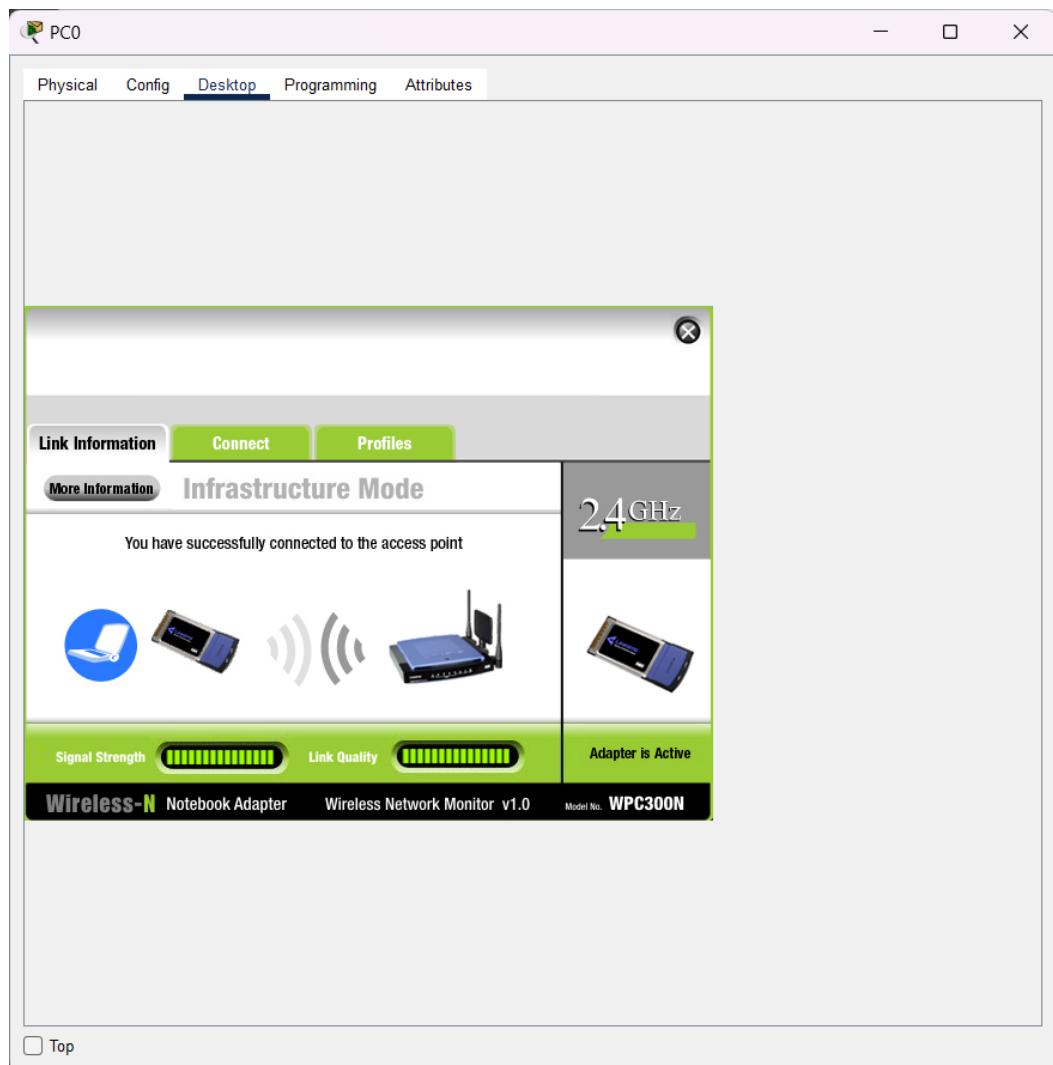
**AIM:** - b) Design and configure an internetwork using wireless router, DHCP server and internet cloud.

**OUTPUT:** -









## **RESULT: -**

Wireless Router have been successfully done in CISCO PACKET TRACER.