

RAJALAKSHMI ENGINEERING COLLEGE

**An Autonomous Institution, Affiliated to Anna University,
Rajalakshmi Nagar, Thandalam - 602 105**



DEPARTMENT OF COMPUTER SCIENCE AND DESIGN

CS23532-COMPUTER NETWORKS

(Regulation 2023)

LAB RECORD

Name	: KEERTHIVASAN S
Register No.	: 2116231701028
Year/Branch/Section	: II/CSD/A
Semester	: IV
Academic Year	: 2024-2025



RAJALAKSHMI ENGINEERING COLLEGE

An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

BONAFIDE CERTIFICATE

NAME KEERTHIVASAN S

ACADEMIC YEAR 2024-2025 SEMESTER IV BRANCH.....B.E.CSD..

UNIVERSITY REGISTER No.

2116231701028

Certified that this is the bonafide record of work done by the above student in the

COMPUTER NETWORK Laboratory during the year 2024 - 2025

Signature of Faculty - in - Charge

Submitted for the Practical Examination held on 29.05.2025

Internal Examiner

External Examiner

Experiment -01

BASIC NETWORKING COMMANDS

AIM: - Study of various Network commands used in Linux and 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 each hop.

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.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Lenovo> arp

Displays and modifies the IP-to-Physical address translation tables used by
address resolution protocol (ARP).

ARP -s inet_addr eth_addr [if_addr]
ARP -d inet_addr [if_addr]
ARP -a [inet_addr] [-N if_addr] [-v]

-a          Displays current ARP entries by interrogating the current
           protocol data. If inet_addr is specified, the IP and Physical
           addresses for only the specified computer are displayed. If
           more than one network interface uses ARP, entries for each ARP
           table are displayed.
-g          Same as -a.
-v          Displays current ARP entries in verbose mode. All invalid
           entries and entries on the loop-back interface will be shown.
inet_addr   Specifies an internet address.
-N if_addr   Displays the ARP entries for the network interface specified
           by if_addr.
-d          Deletes the host specified by inet_addr. inet_addr may be
           wildcarded with * to delete all hosts.
-s          Adds the host and associates the Internet address inet_addr
           with the Physical address eth_addr. The Physical address is
           given as 6 hexadecimal bytes separated by hyphens. The entry
           is permanent.
eth_addr    Specifies a physical address.
if_addr     If present, this specifies the Internet address of the
           interface whose address translation table should be modified.
           If not present, the first applicable interface will be used.

Example:
> arp -s 157.55.85.212 00-AA-00-62-C6-09 .... Adds a static entry.
> arp -a                                .... Displays the arp table.
ps C:\Users\Lenovo> hostname
DESKTOP-C01BH7D
ps C:\Users\Lenovo> ipconfig

Windows IP Configuration

Ethernet adapter Ethernet:

Connection-specific DNS Suffix . :
Link-local IPv6 Address . . . . . : fe80::a098:b3f4:e84d:50c5%5
IPv4 Address . . . . . : 172.16.75.153
Subnet Mask . . . . . : 255.255.248.0
```

```
Default Gateway . . . . . : 172.16.72.1
PS C:\Users\Lenovo> route

Manipulates network routing tables.

ROUTE [-f] [-p] [-q|-6] command [destination]
      [MASK netmask] [gateway] [METRIC metric] [IF interface]

-f      Clears the routing tables of all gateway entries. If this is
       used in conjunction with one of the commands, the tables are
       cleared prior to running the command.

-p      When used with the ADD command, makes a route persistent across
       boots of the system. By default, routes are not preserved
       when the system is restarted. Ignored for all other commands,
       which always affect the appropriate persistent routes.

-6      Force using IPv6.

-4      Force using IPv4.

command One of these:
        PRINT   Prints a route
        ADD     Adds a route
        DELETE  Deletes a route
        CHANGE  Modifies an existing route

destination Specifies the host.
MASK      Specifies that the next parameter is the 'netmask' value.
netmask   Specifies a subnet mask value for this route entry.
          If not specified, it defaults to 255.255.255.255.
gateway   Specifies gateway.
interface  the interface number for the specified route.
METRIC    specifies the metric, i.e. cost for the destination.

All symbolic names used for destination are looked up in the network database
file NETWORKS. The symbolic names for gateway are looked up in the host name
database file HOSTS.

If the command is PRINT or DELETE, Destination or gateway can be a wildcard,
(wildcard is specified as a star '*'), or the gateway argument may be omitted.

If Dest contains a * or ?, it is treated as a shell pattern, and only
matching destination routes are printed. The '*' matches any string,
and '?' matches any one char. Examples: 157.*.1, 157.*, 127.*, *224*.

Pattern match is only allowed in PRINT command.

Diagnostic Notes:
  Invalid MASK generates an error, that is when (DEST & MASK) != DEST.
  Example> route ADD 157.0.0.0 MASK 155.0.0.0 157.55.60.1 IF 3
           The route addition failed: The specified mask parameter is invalid. (Destination & Mask) != Destination.
```

Examples:

```
> route PRINT  
> route PRINT -4  
> route PRINT -6  
> route PRINT 157*           .... Only prints those matching 157*  
  
> route ADD 157.0.0.0 MASK 255.0.0.0 157.55.80.1 METRIC 3 IF 2  
      destination"      "mask      "gateway"    metric"      "  
                                         Interface"  
      If IF is not given, it tries to find the best interface for a given  
      gateway.  
> route ADD 3ffe::/32 3ffe::1  
  
> route CHANGE 157.0.0.0 MASK 255.0.0.0 157.55.80.5 METRIC 2 IF 2  
      CHANGE is used to modify gateway and/or metric only.  
  
> route DELETE 157.0.0.0  
> route DELETE 3ffe::/32
```

PS C:\Users\Lenovo> ping

```
Usage: ping [-t] [-a] [-n count] [-l size] [-f] [-i TTL] [-v TOS]  
           [-r count] [-s count] [[-j host-list] | [-k host-list]]  
           [-w timeout] [-R] [-S srcaddr] [-c compartment] [-p]  
           [-4] [-6] target_name
```

Options:

-t	Ping the specified host until stopped. To see statistics and continue - type Control-Break; To stop - type Control-C.
-a	Resolve addresses to hostnames.
-n count	Number of echo requests to send.
-l size	Send buffer size.
-f	Set Don't Fragment flag in packet (IPv4-only).
-i TTL	Time To Live.
-v TOS	Type Of Service (IPv4-only. This setting has been deprecated and has no effect on the type of service field in the IP Header).
-r count	Record route for count hops (IPv4-only).
-s count	Timestamp for count hops (IPv4-only).
-j host-list	Loose source route along host-list (IPv4-only).
-k host-list	Strict source route along host-list (IPv4-only).
-w timeout	Timeout in milliseconds to wait for each reply.
-R	Use routing header to test reverse route also (IPv6-only). Per RFC 5095 the use of this routing header has been deprecated. Some systems may drop echo requests if this header is used.
-S srcaddr	Source address to use.
-c compartment	Routing compartment identifier.
-p	Ping a Hyper-V Network Virtualization provider address.

```
-4           Force using IPv4.  
-6           Force using IPv6.  
  
PS C:\Users\Lenovo> nbtstat  
  
Displays protocol statistics and current TCP/IP connections using NBT  
(NetBIOS over TCP/IP).  
  
NBTSTAT [ [-a RemoteName] [-A IP address] [-c] [-n]  
          [-r] [-R] [-RR] [-s] [-S] [interval] ]  
  
-a  (adapter status) Lists the remote machine's name table given its name.  
-A  (Adapter status) Lists the remote machine's name table given its IP address.  
-c  (cache)        Lists NBT's cache of remote [machine] names and their IP addresses.  
-n  (names)        Lists local NetBIOS names.  
-r  (resolved)    Lists names resolved by broadcast and via WINS.  
-R  (Reload)       Purges and reloads the remote cache name table.  
-S  (Sessions)    Lists sessions table with the destination IP addresses.  
-s  (sessions)    Lists sessions table converting destination IP addresses to computer NETBIOS names.  
-RR  (ReleaseRefresh) Sends Name Release packets to WINS and then, starts Refresh.  
  
RemoteName  Remote host machine name.  
IP address   Dotted decimal representation of the IP address.  
interval     Redisplays selected statistics, pausing interval seconds  
              between each display. Press Ctrl+C to stop redisplaying  
              statistics.  
  
PS C:\Users\Lenovo> 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.
```

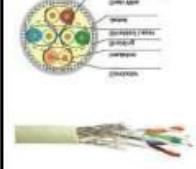
RESULT: the Study of various Network commands used in Linux and Windows has been studied successfully

Experiment No 2

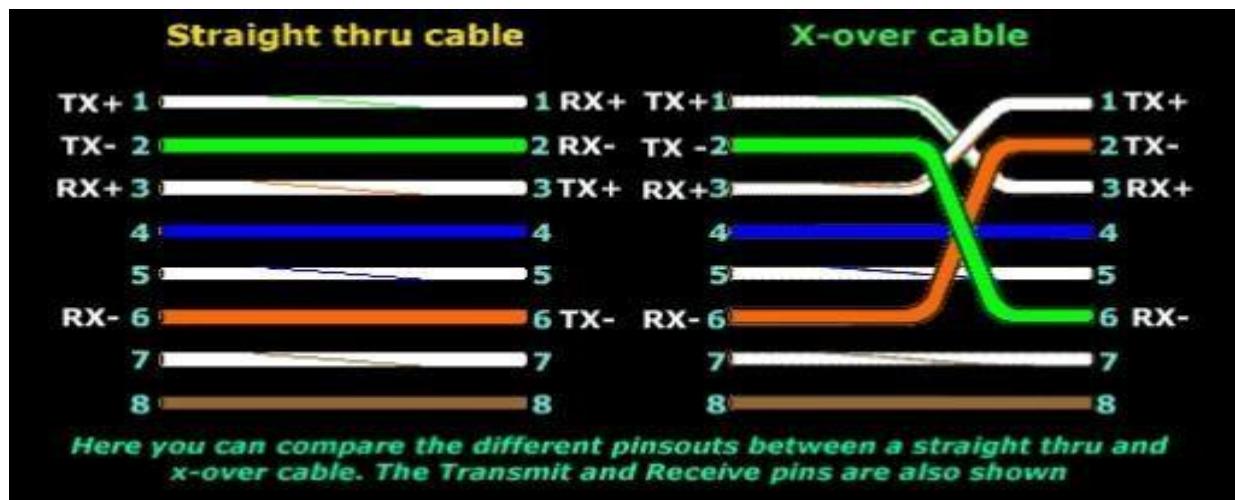
Aim: Study of different types of Network cables.

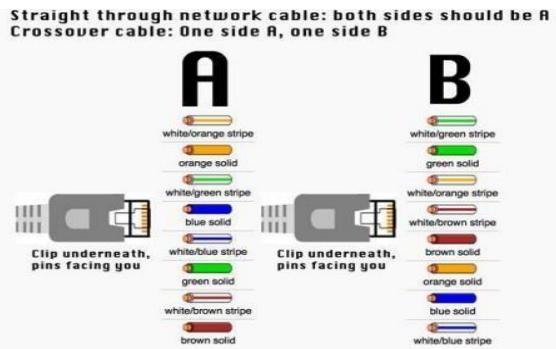
Different type of cables used in networking are:

- 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	Advantages <ul style="list-style-type: none"> • Cheaper in cost • Easy to install as they have a smaller overall diameter. Disadvantages <ul style="list-style-type: none"> • More prone to (EMI) Electromagnetic interference and noise 	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	Advantages <ul style="list-style-type: none"> • Shielded. • Faster than UTP. • Less susceptible to noise and interference Disadvantages <ul style="list-style-type: none"> • Expensive • Greater installation effort 	Gigabit Ethernet, 10G Ethernet (55m) Widely used in data centres	
			Gigabit Ethernet, 10G Ethernet (100m)		
SSTP	Category 7	10Gbps			

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





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

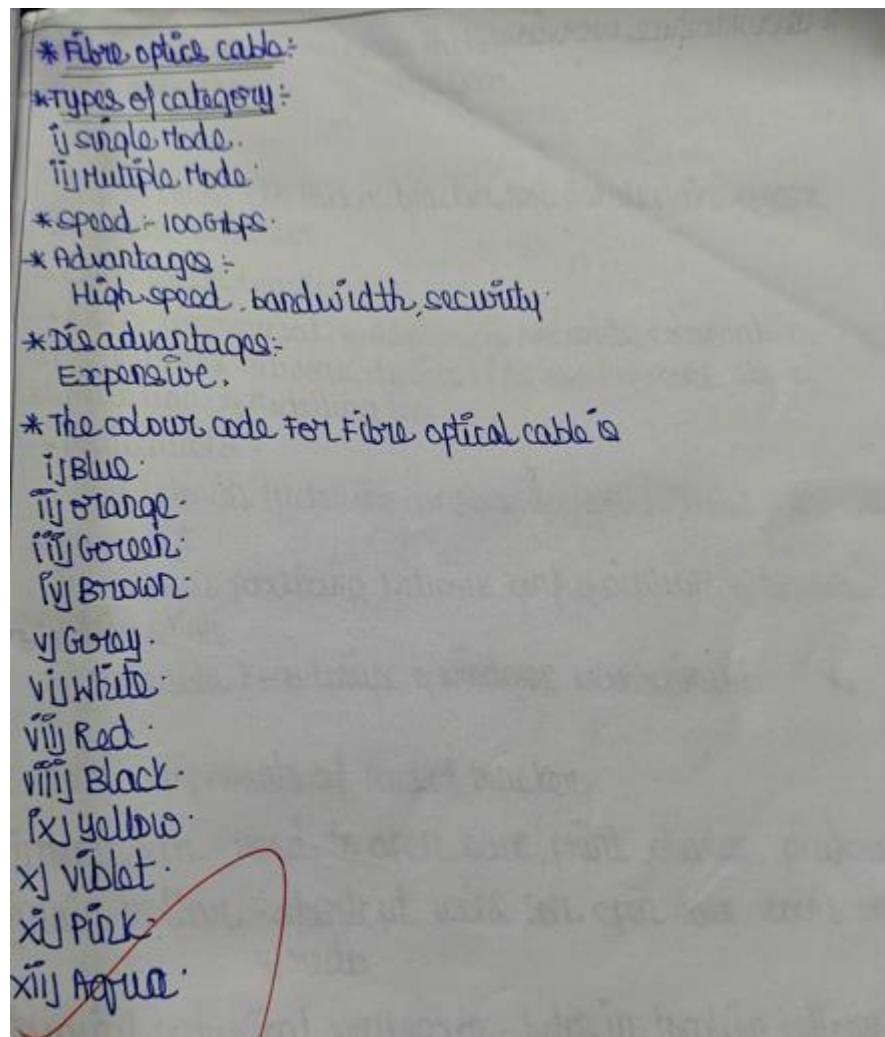
Step 6: After the wires are securely sitting inside the plug, insert it into the crimping tool

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

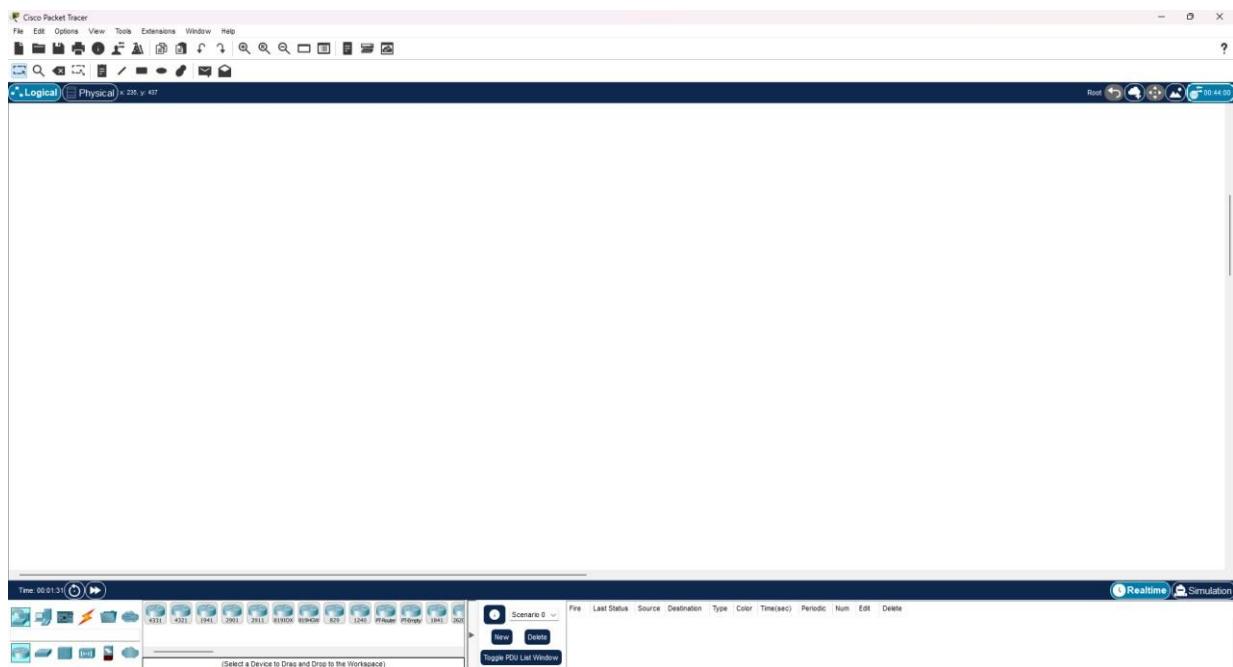
AIM: To study the Packet tracer tool Installation and User Interface

Overview

To understand environment of CISCO PACKET TRACER to design simple network.

It simulates network devices and its environment. Packet Tracer is an exciting network design, simulation and modelling tool.

1. It allows you to model complex systems without the need for dedicated equipment.
2. It helps you to practice your network configuration and troubleshooting skills via computer or an Android or iOS based mobile device.
3. It is available for both the Linux and Windows desktop environments.
4. Protocols in Packet Tracer are coded to work and behave in the same way as they would on real hardware.



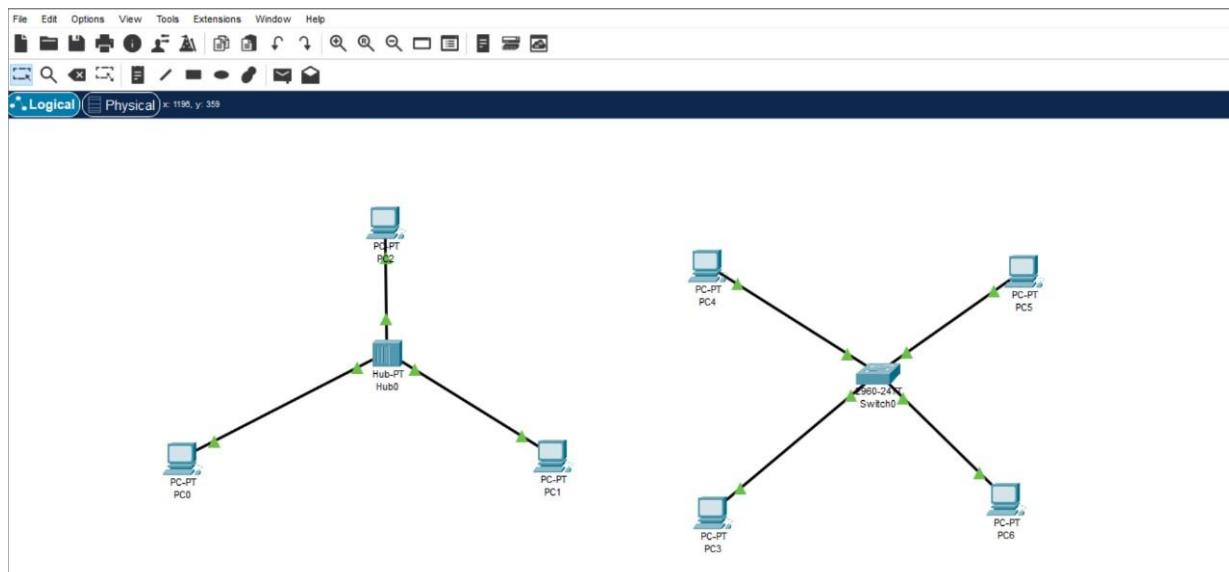
1. Menu bar - This is a common menu found in all software applications; it is used to open, save, print, change preferences, and so on.
2. Main toolbar - This bar provides shortcut icons to menu options that are commonly accessed, such as open, save, zoom, undo, and redo, and on the right-hand side is an icon for entering network information for the current network.

3. Logical/Physical workspace tabs - These tabs allow you to toggle between the Logical and Physical work areas.
4. Workspace - This is the area where topologies are created and simulations are displayed.
5. 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.
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.

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:

The study the Packet tracer tool Installation and User Interface Overview has been executed successfully

Experiment 4

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

LAN:

A Local Area Network (LAN) refers to a network that connects devices within a limited area, such as an office building, school, or home. It enables users to share resources, including data, printers, and internet access. LAN connects devices to promote collaboration and transfer information between users, such as computers, printers, servers, and switches. A local area network (LAN) switch serves as the primary connecting device, managing and directing communications within the local network. Each connected device on a LAN switch can communicate directly with each other, allowing for fast and secure data transfer.

How to set up a LAN:

Step 1. Plan and Design an appropriate network topology taking into account network requirements and equipment location.

Step 2. You can take 4 Computers, a Switch with 8, 16, or 24 ports which is sufficient for networks of these sizes, and 4 Ethernet cables.

Step3: Connect your computers to network switch via an Ethernet cable, which is as simple as plugging one end of the Ethernet cable into your computer and the other end into your network switch.

Step4: Assign IP address to your PCs **Step 5:-**

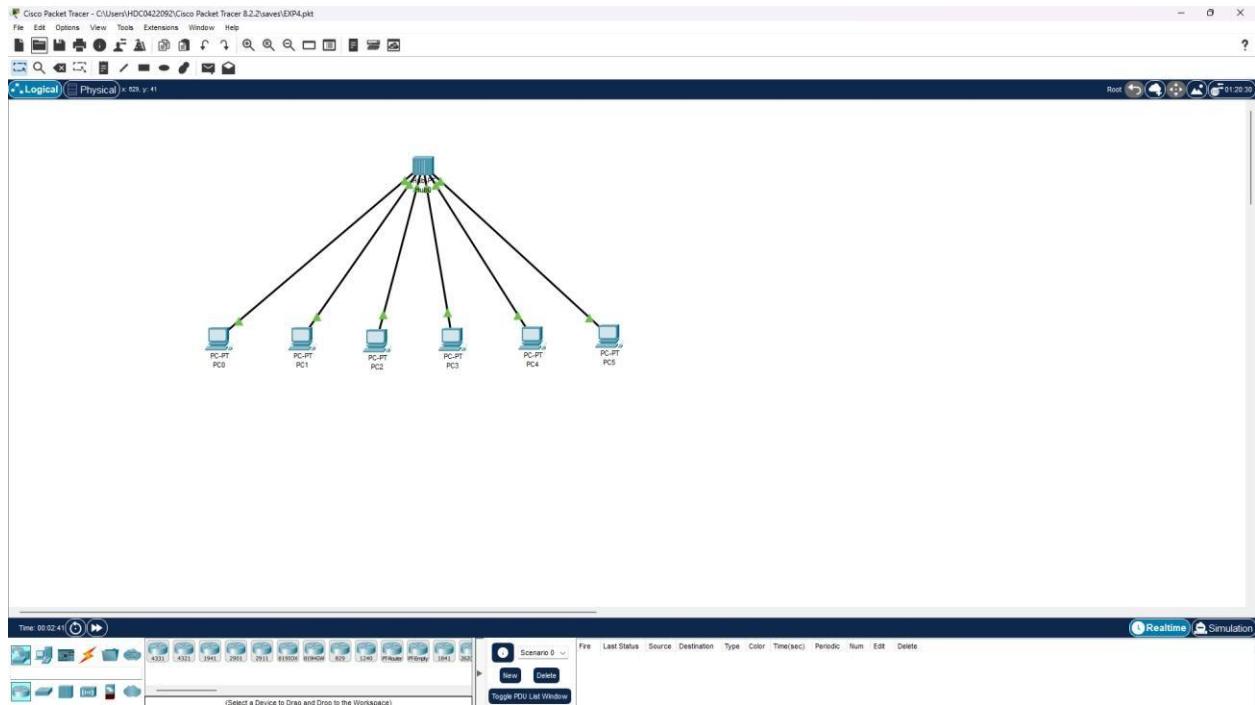
Configure a network switch:

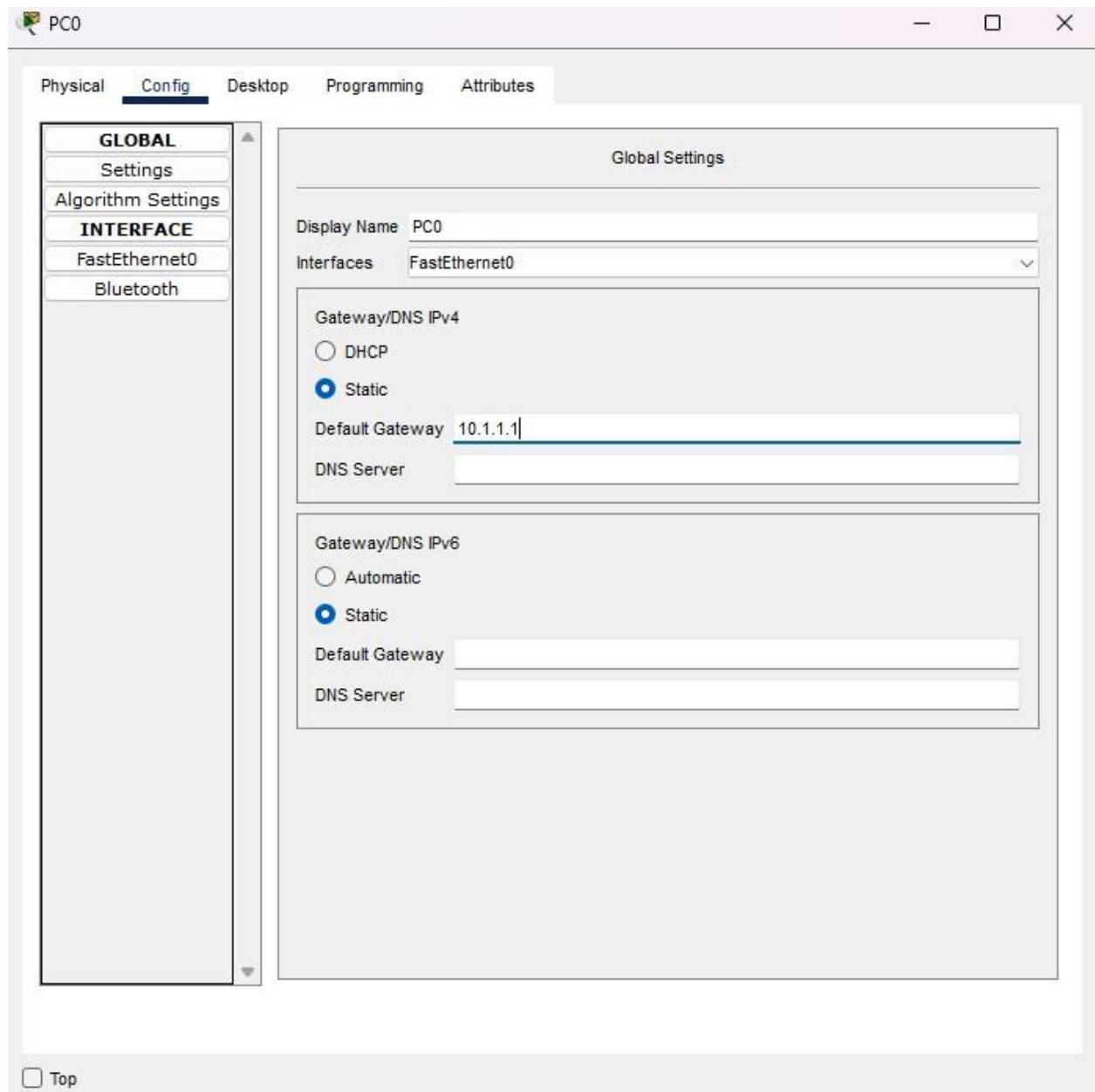
1. Connect your computer to the switch: To access the switch's web interface, you will need to connect your computer to the switch using an Ethernet cable.
2. Log in to the web interface: Open a web browser and enter the IP address of the switch in the address bar. This should bring up the login page for the switch's web interface.
3. Configure basic settings: Once you're logged in, you will be able to configure basic settings for the switch,
4. Assign IP address as: 10.1.1.5, subnet mask 255.0.0.0.

Step 6:- Check the connectivity between switch and other machine by using ping command in the command prompt of the device.

Step 7: Select a folder, ->go to properties-> click Sharing tab->share it with everyone on the same LAN.

Step 8. Try to access the shared folder from others Computers of the network.





RESULT:

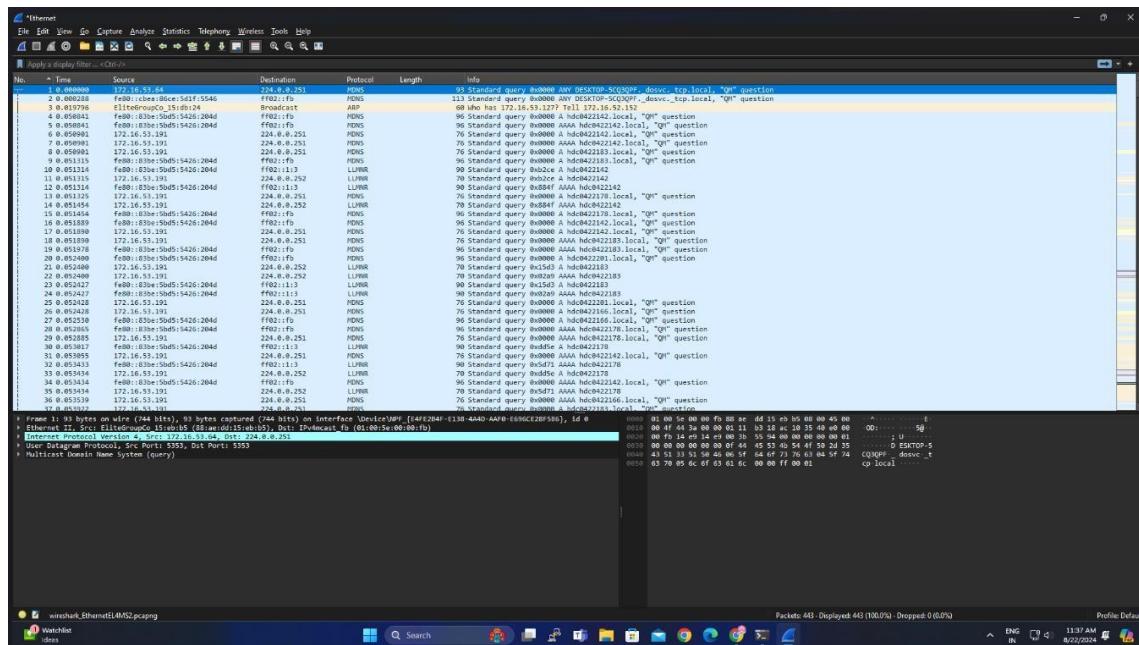
The Setup and configure a LAN (Local area network) using a Switch and Ethernet cables is executed successfully

EXPERIMENT – 5

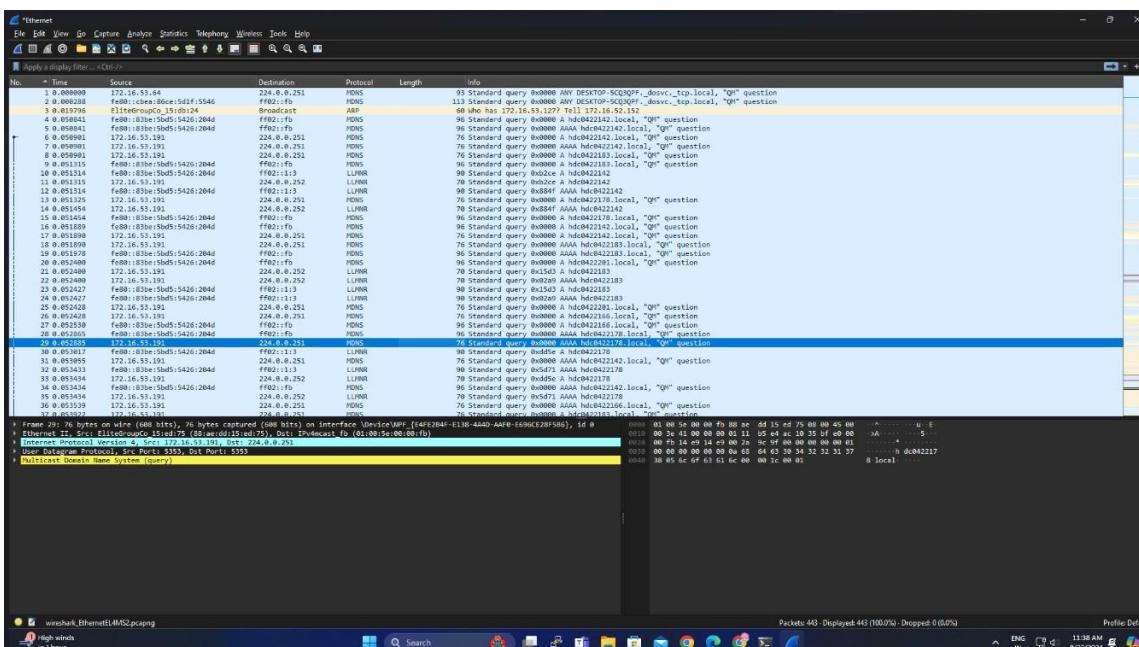
AIM: - Experiments on Packet capture tool: Wireshark

CAPTURING AND ANALYSING PACKETS USING WIRESHARK TOOL:

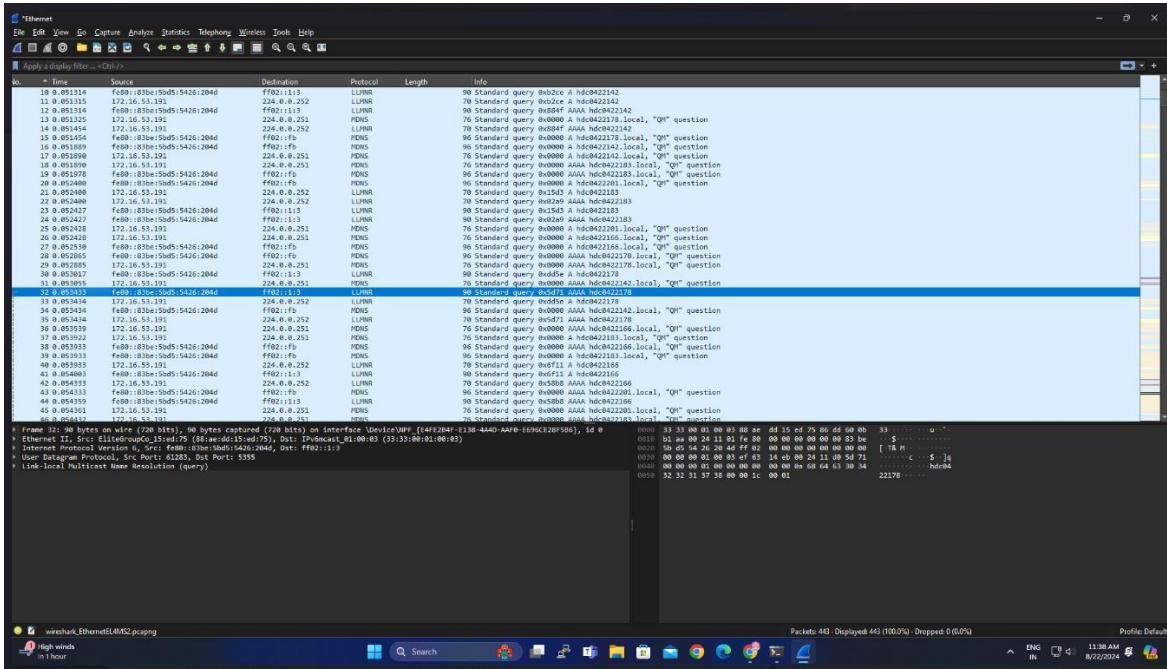
Packet 1:



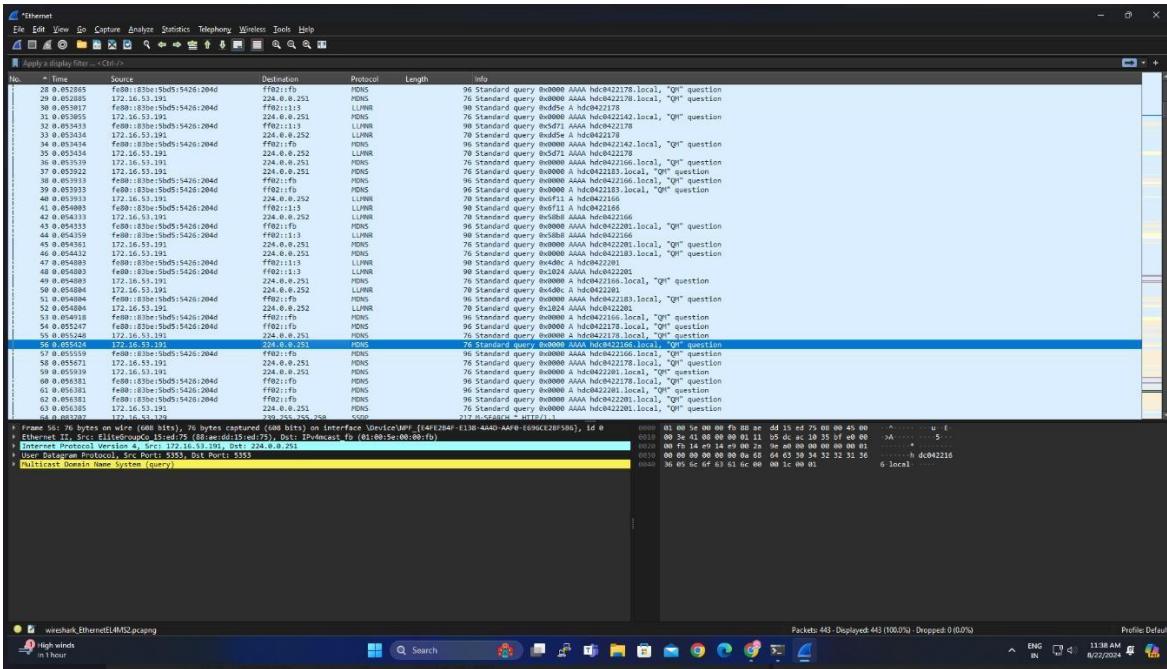
Packet 2:



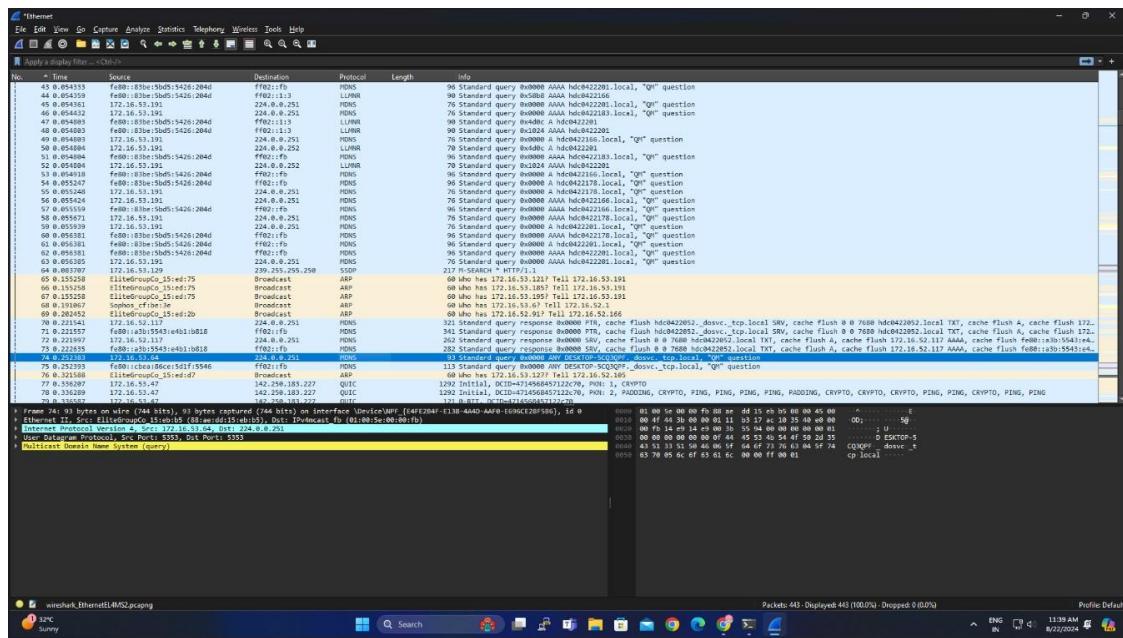
Packet 3:



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.

CODE: - def

```

calcRedundantBits(m):
    # Use the formula  $2^r \geq m + r + 1$ 
for i in range(m):           if( $2^{**i} \geq m$ 
+ 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
1- def calcRedundantBits(m):
2-     # use the formula  $2^n \geq m + r + 1$ 
3-     for i in range(m):
4-         if( $2^i \geq m + r + 1$ ):
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 ==  $2^k$ ):
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(1, 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 & ( $2^{i-1}$ ) == ( $2^{i-1}$ )):
```

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
acknowledement 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 Online Compiler interface. The code in the editor is:

```

main.c
1 #include<stdio.h>
2 int main()
3 {
4     int w,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
corruption of frames)\n\n");
13    printf("After sending %d frames at each stage sender waits for acknowledgement sent by the
receiver\n\n",w);
14    for(i=1;i<=f;i++)
15    {
16        if(i%w==0)
17        {
18            printf("%d\n",frames[i]);
19            printf("Acknowledgement of above frames sent is received by sender\n\n");
20        }
21        else
22            printf("%d ".frames[i]);
23    }
24    if(f%w!=0)
25        printf("\nAcknowledgement of above frames sent is received by sender\n");
26    return 0;
27 }

```

The output window shows the following interaction:

```

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 ---

```

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:

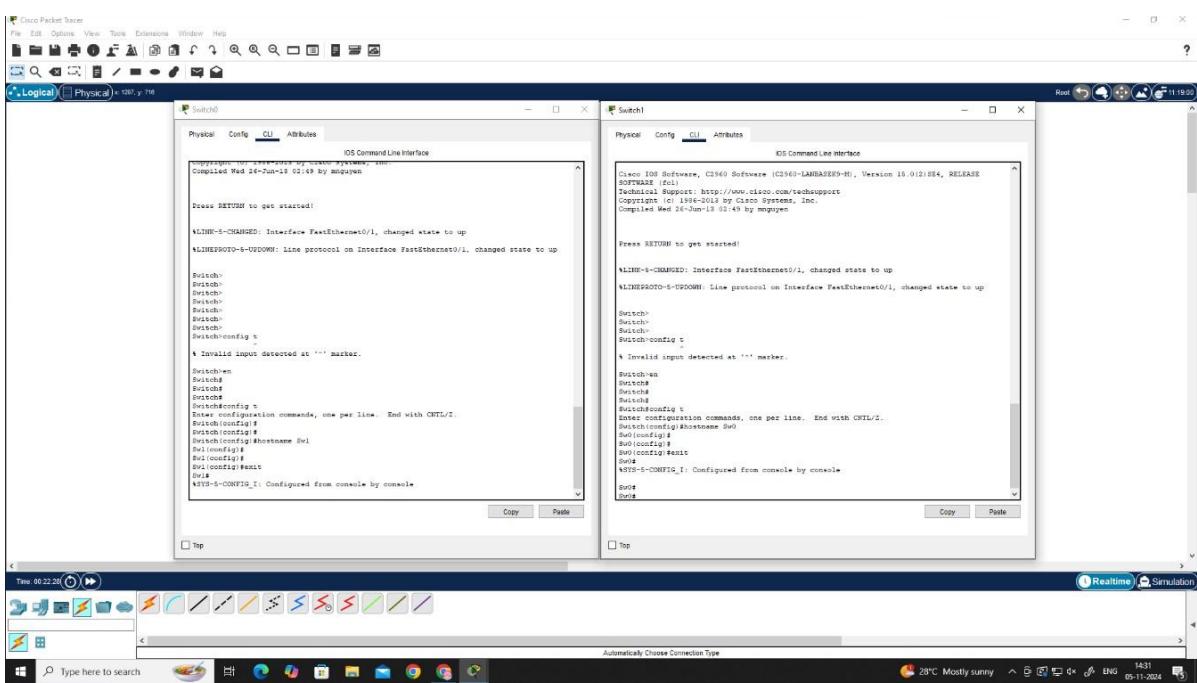
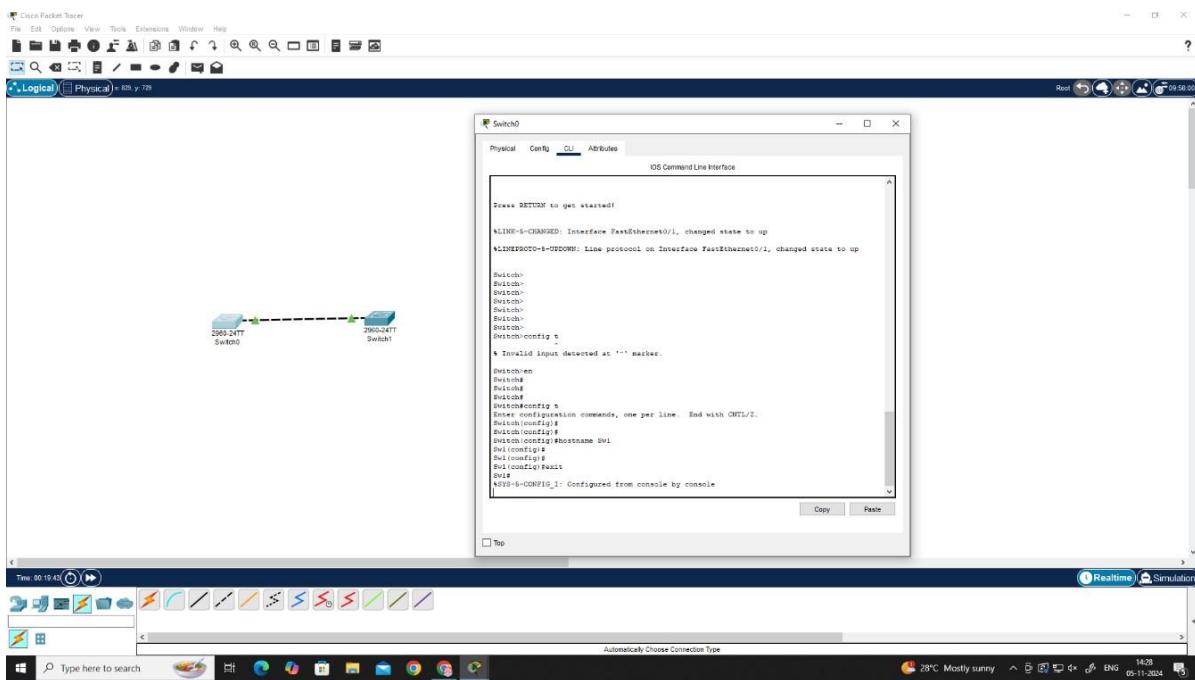
Step 1: Build the network as shown in the topology.

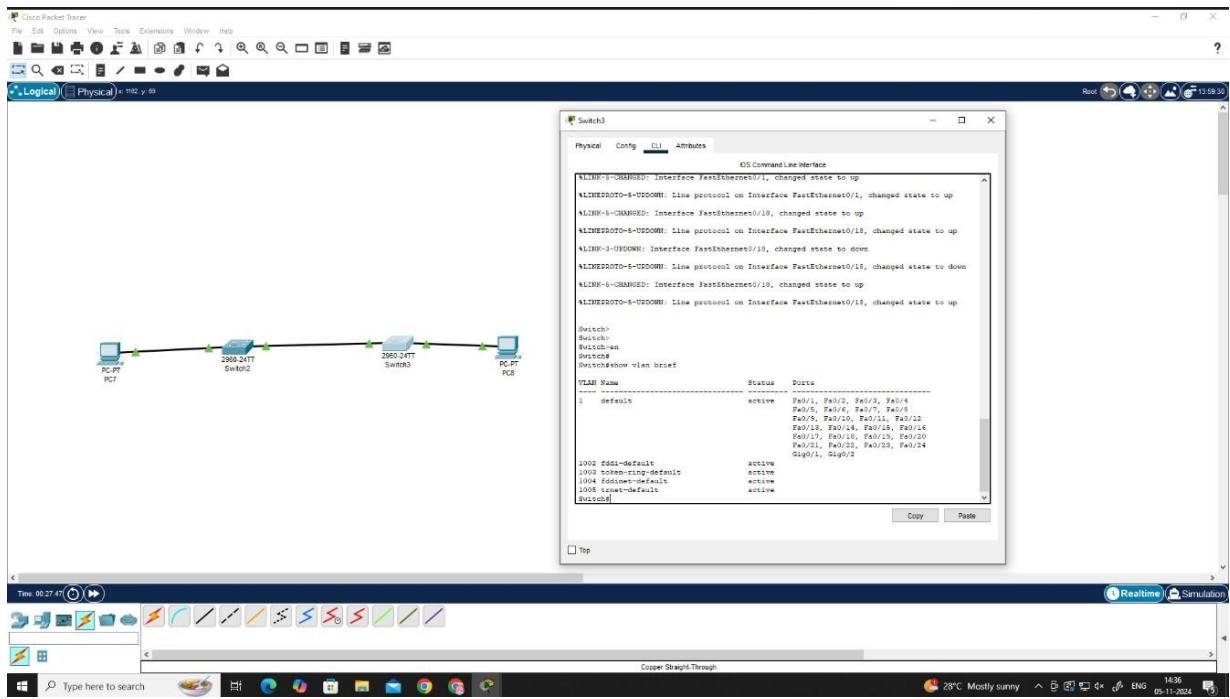
Step 2: Configure basic settings for each switch.

Step 3: Configure PC hosts.

Step 4: Test connectivity.

OUTPUT: -





RESULT: -

Simulation of virtual LAN configuration have been done successfully.

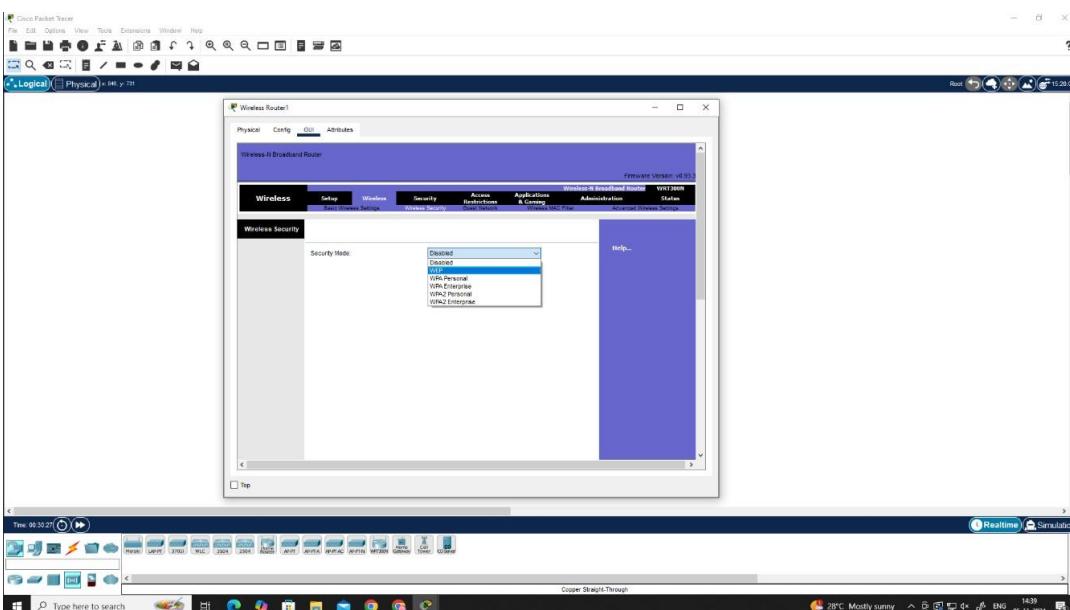
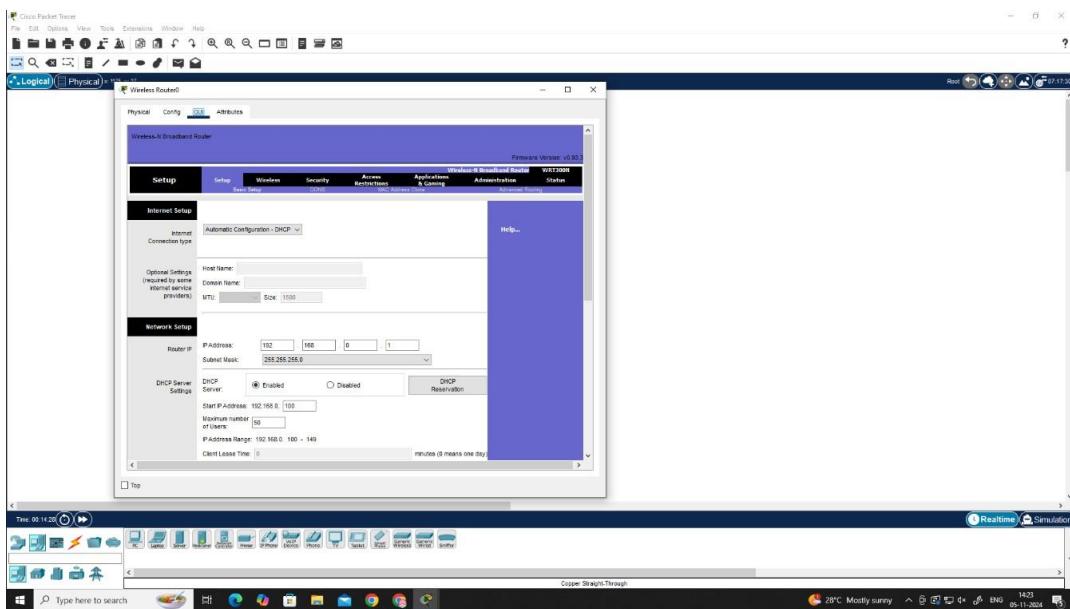
AIM: - b) Configuration of Wireless LAN using CISCO Packet Tracer.

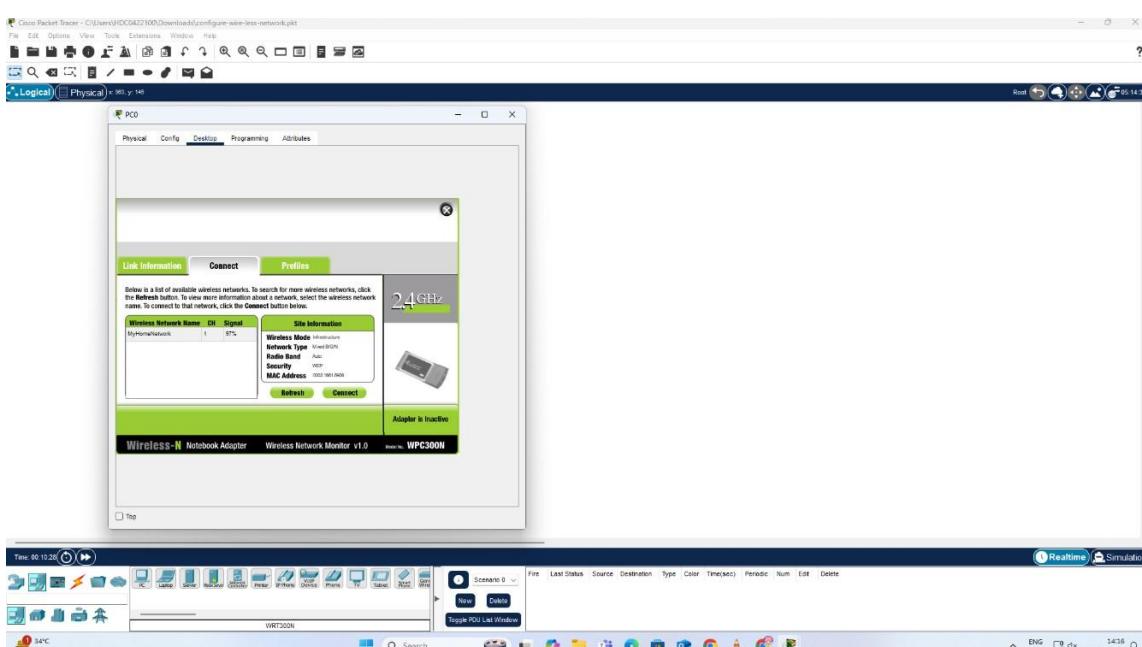
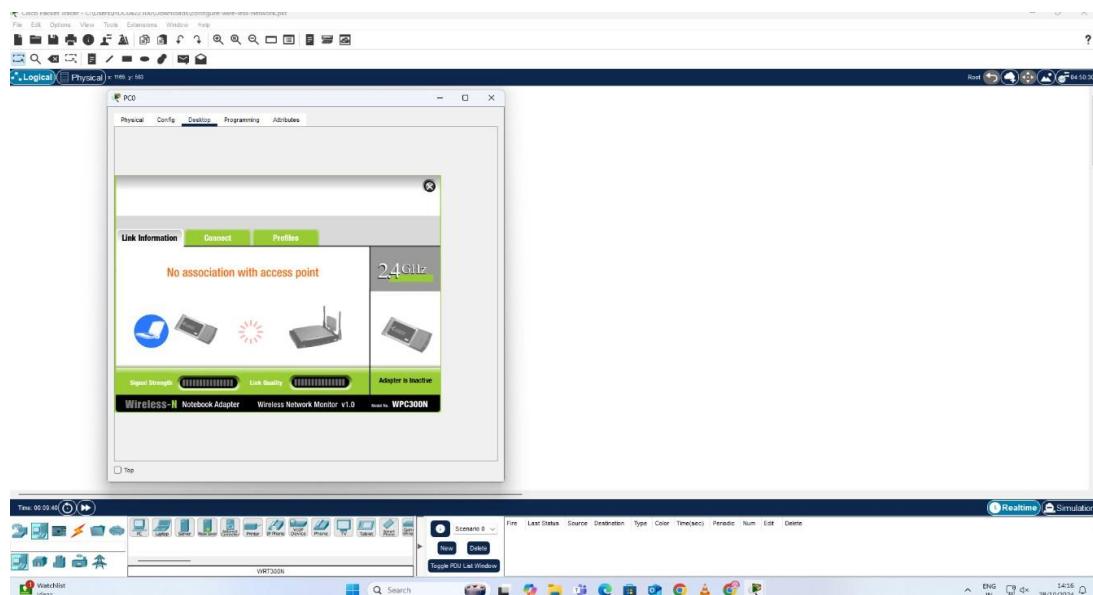
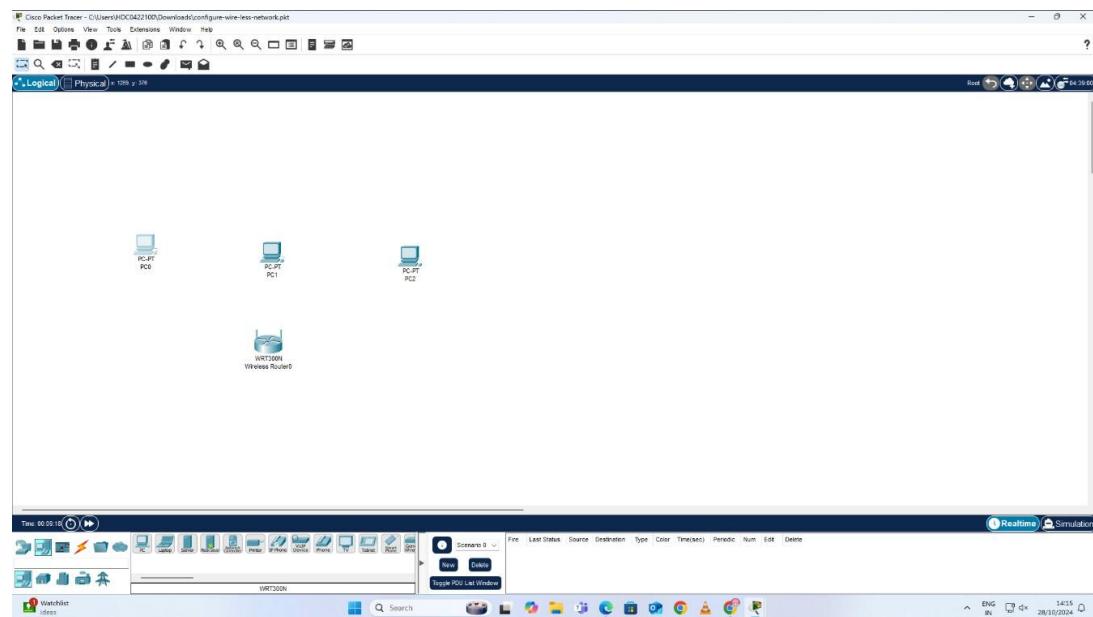
Steps:

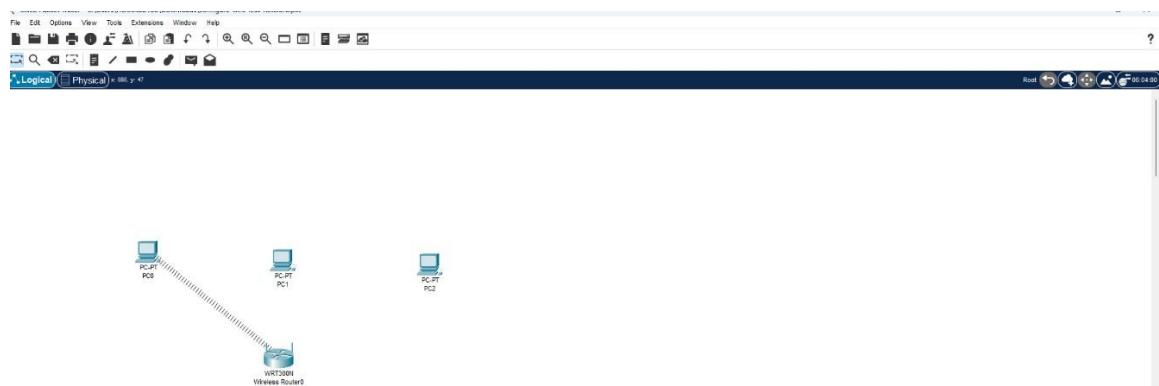
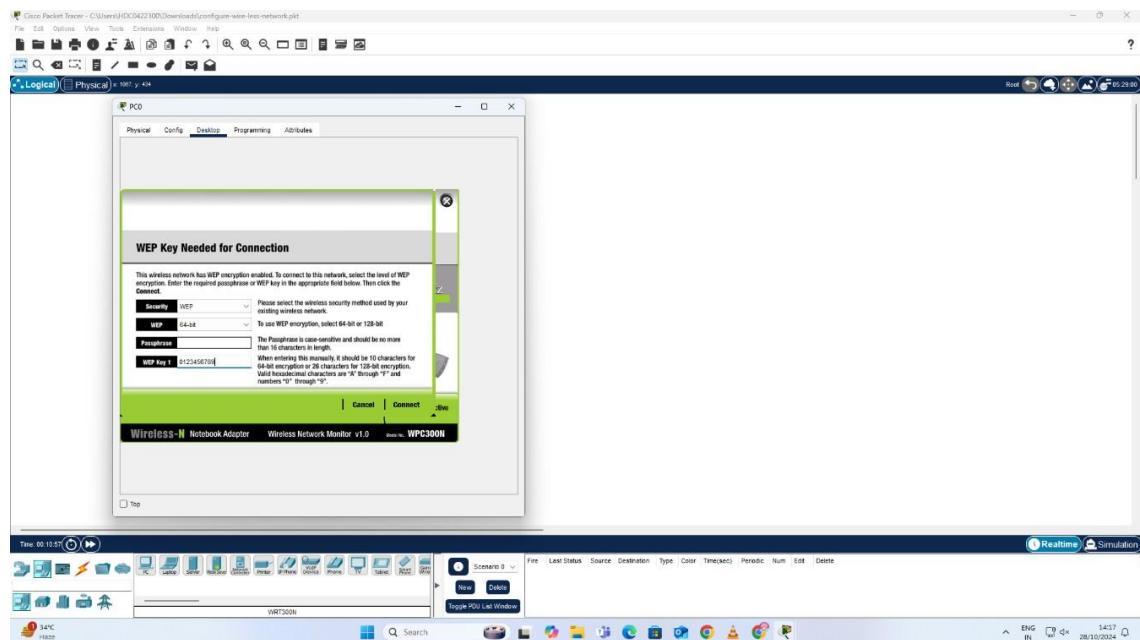
- Configure Static IP on PC and Wireless Router

- 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.

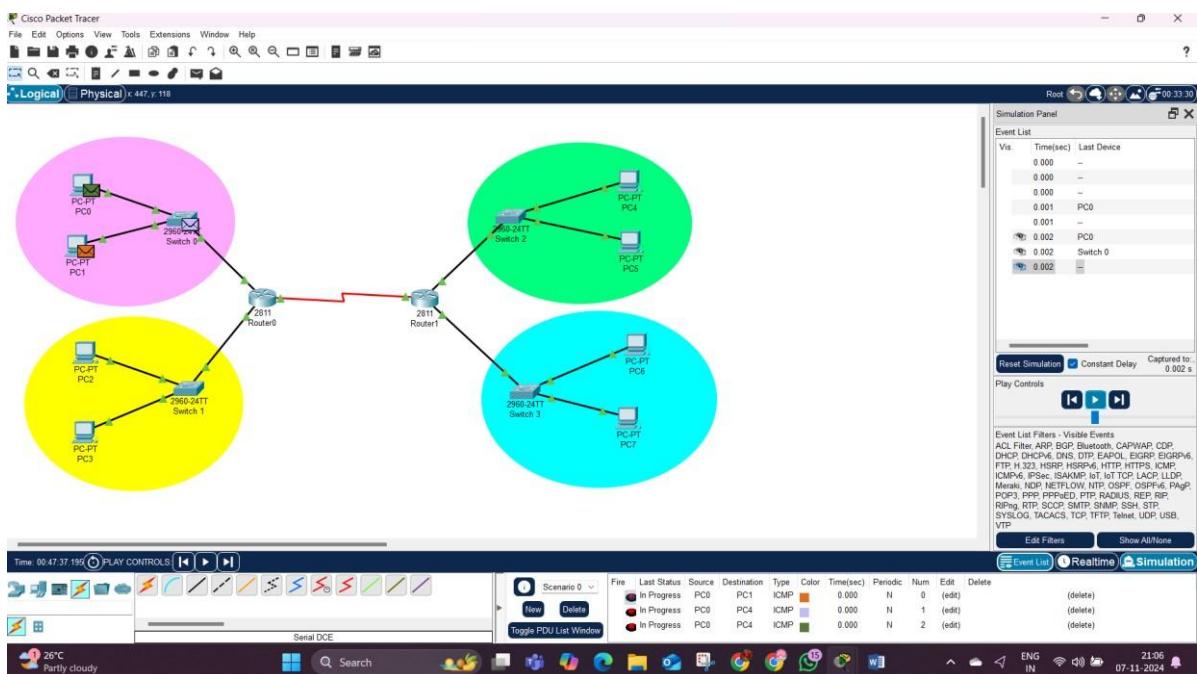
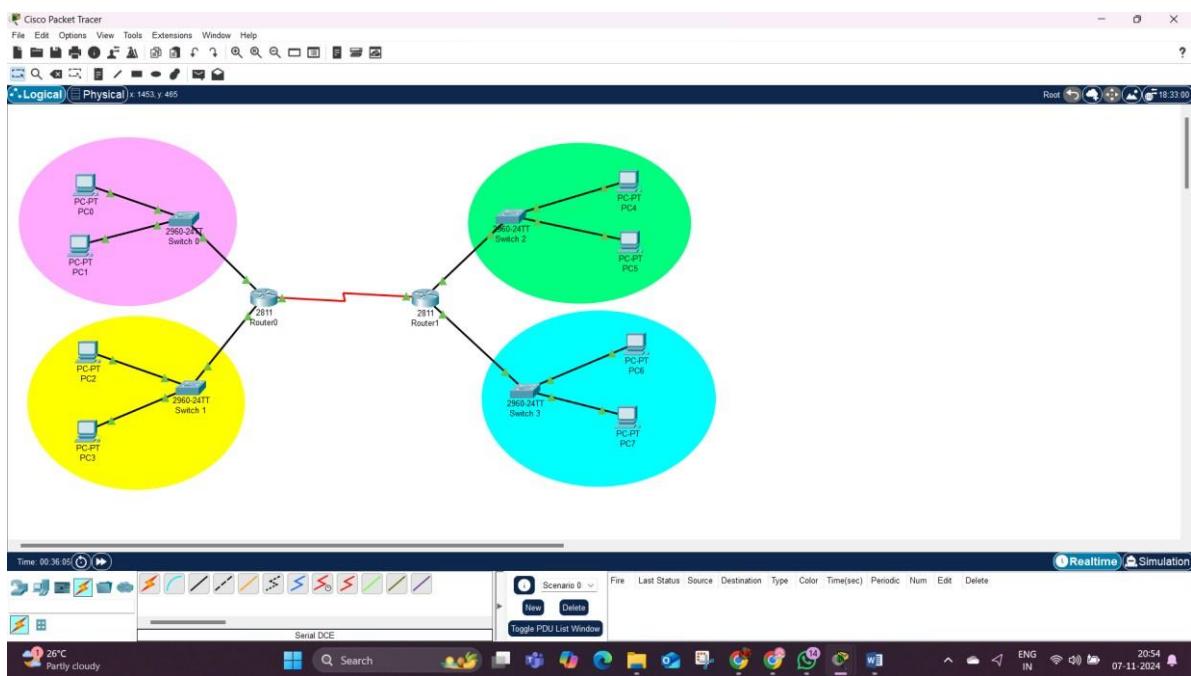
EXPERIMENT – 9

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

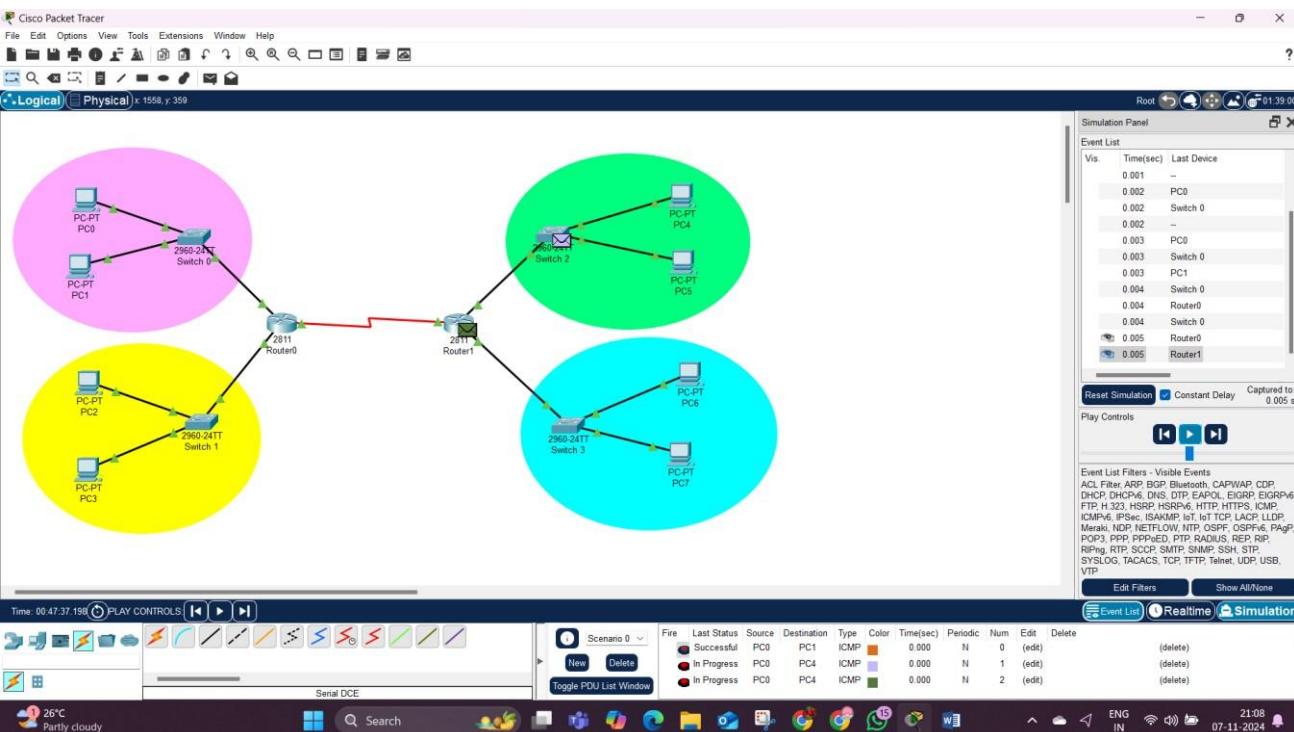
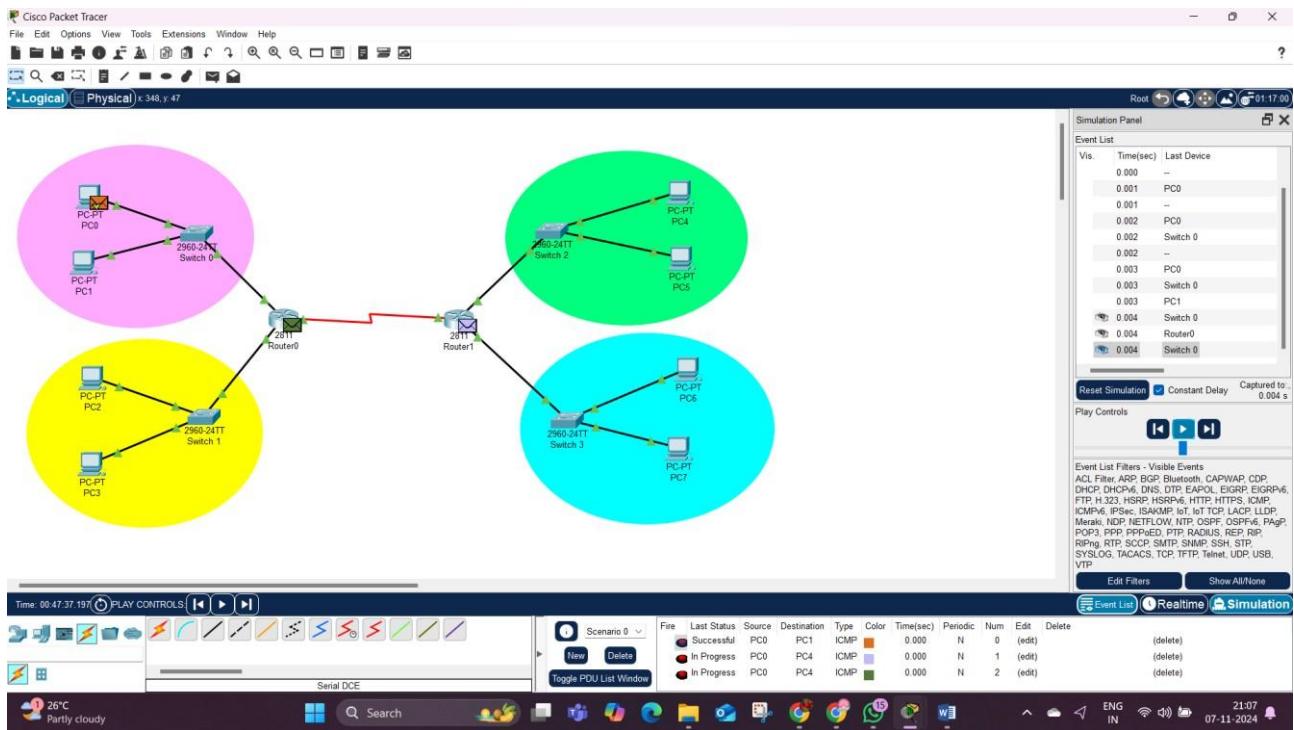
What is subnetting?

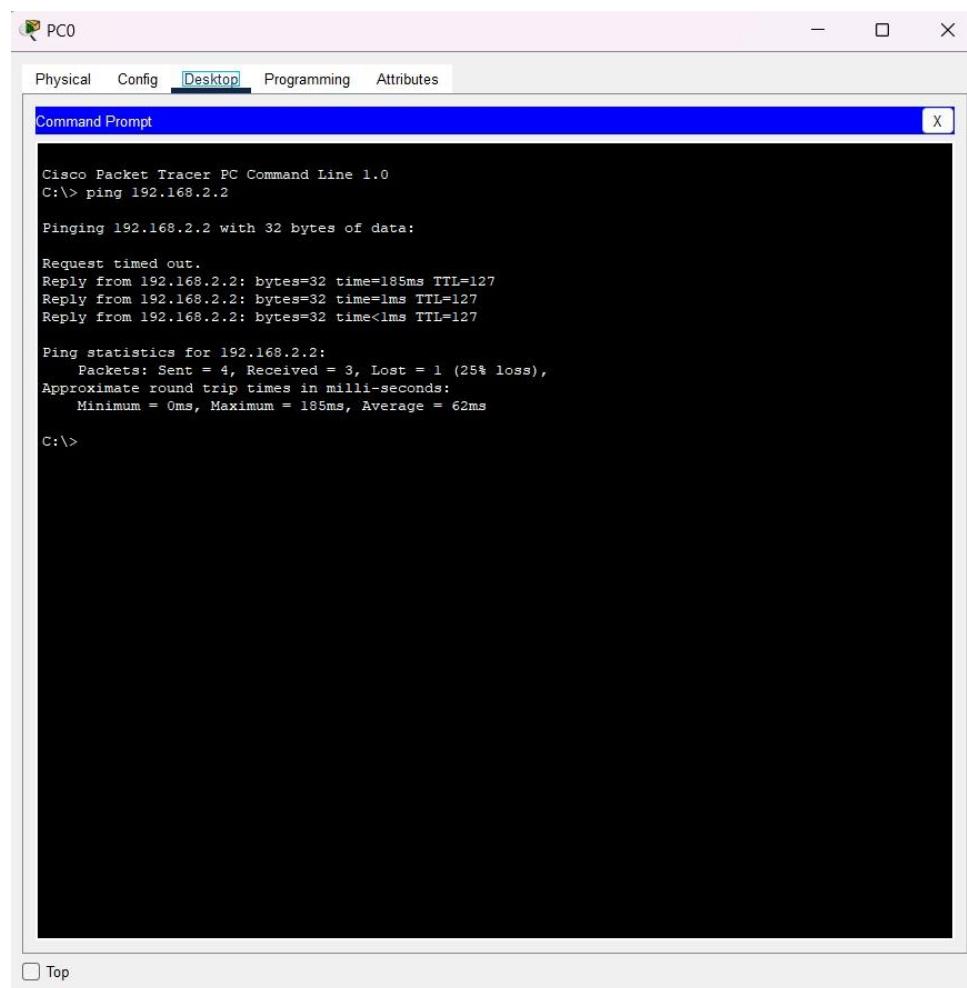
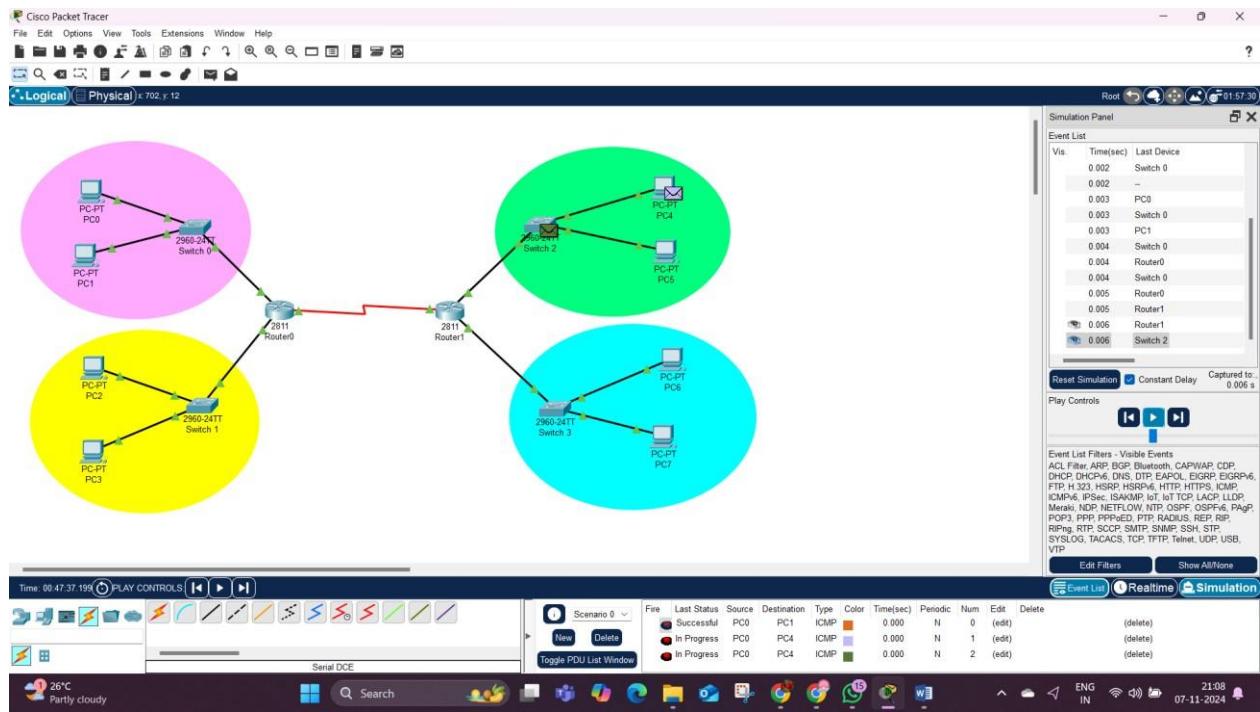
Classless IP subnetting is a technique that allows for more efficient use of IP addresses by allowing for subnet masks that are not just the default masks for each IP class. This means that we can divide our IP address space into smaller subnets, which can be useful when we have a limited number of IP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

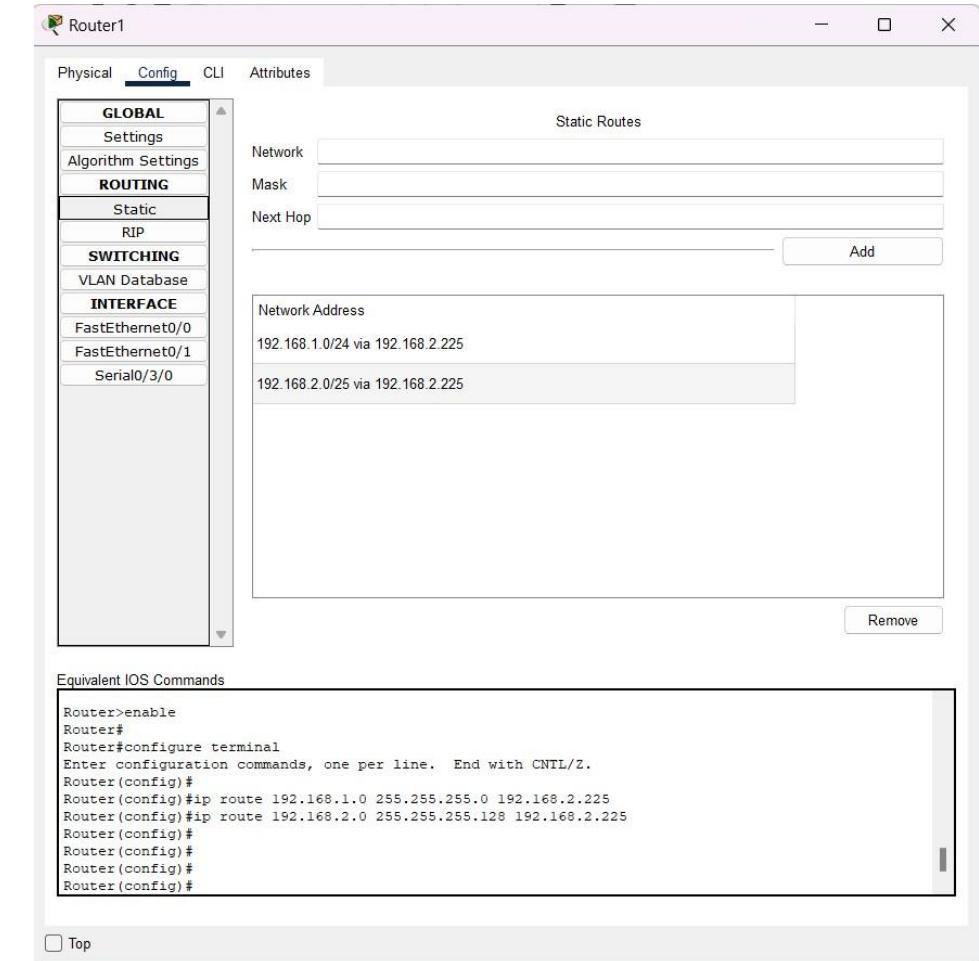
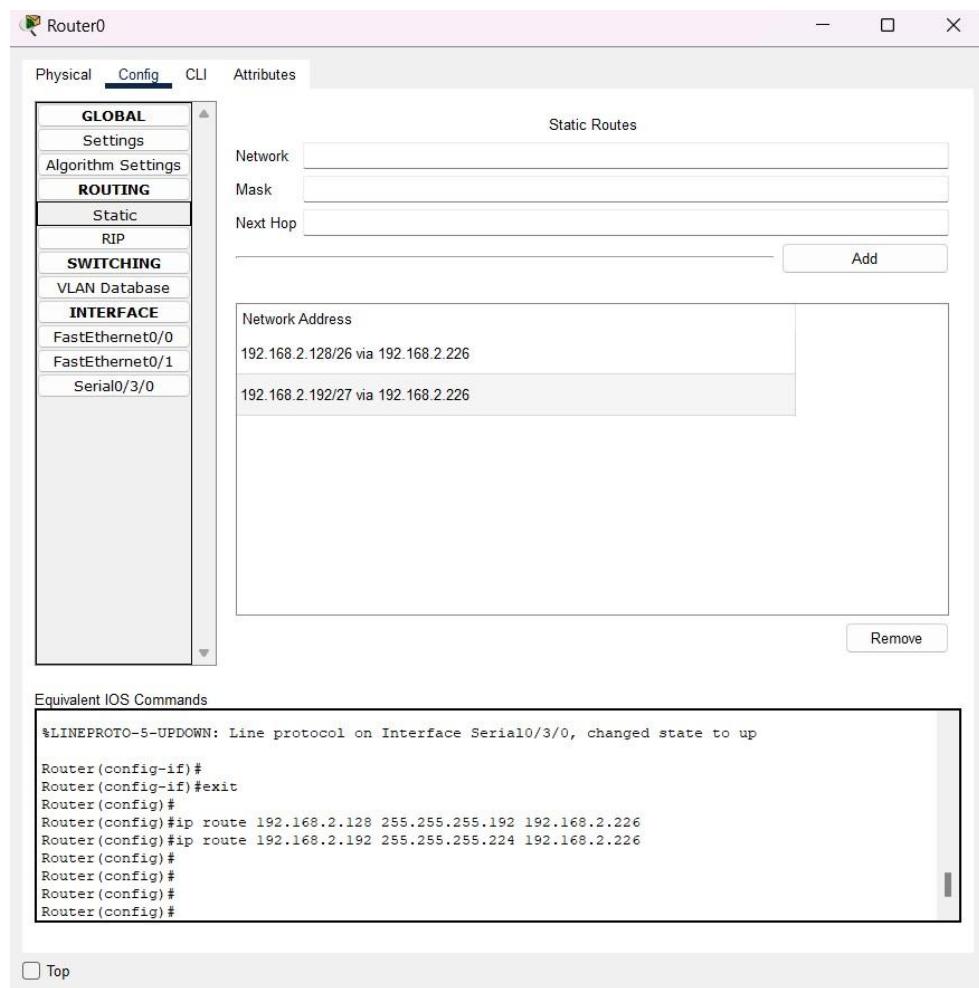
OUTPUT: -



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







The screenshot shows a Cisco Packet Tracer interface with a Command Prompt window. The window title is "Command Prompt". The content of the window shows several ping commands being run against hosts 192.168.2.129 and 192.168.2.193. The output includes statistics for each ping, such as packets sent, received, lost, and round trip times.

```
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),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

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: -

```
PC0
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
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 = 1ms, 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
Physical Config Desktop Programming Attributes
Command Prompt
Cisco Packet Tracer PC Command Line 1.0
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=1ms TTL=127
Reply from 192.168.1.1: bytes=32 time=12ms TTL=127

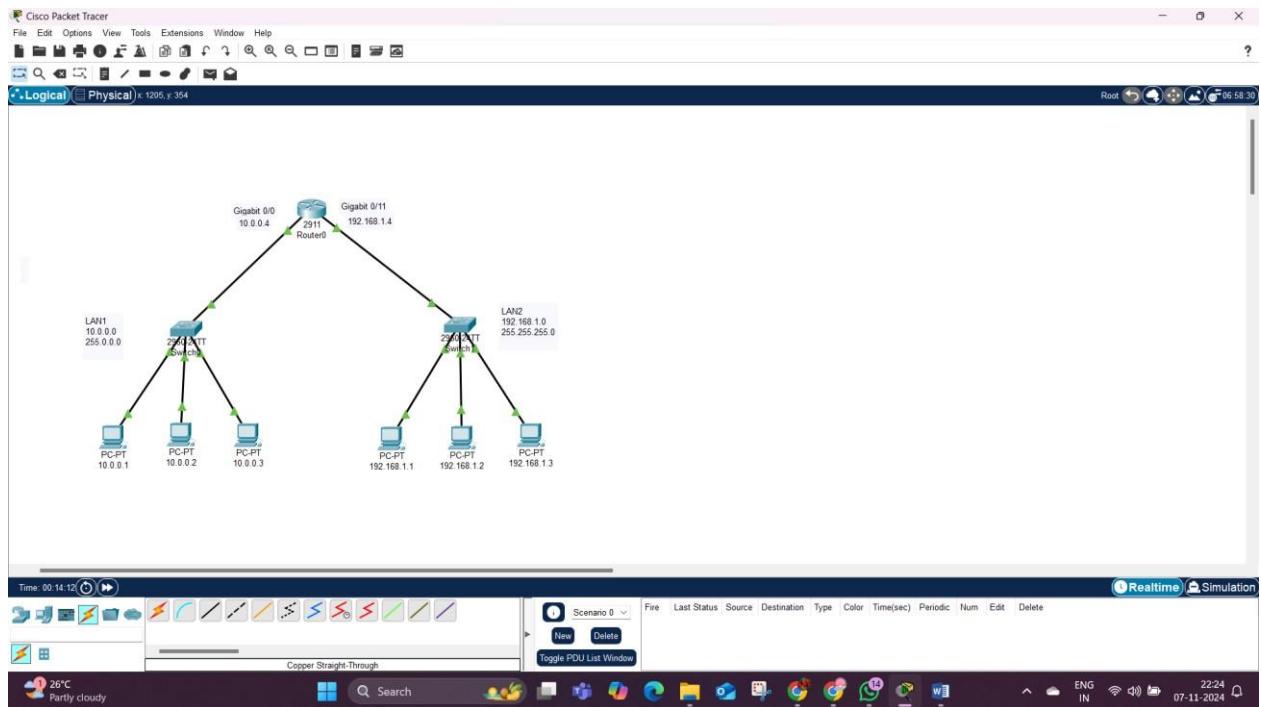
Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 10ms, Average = 3ms

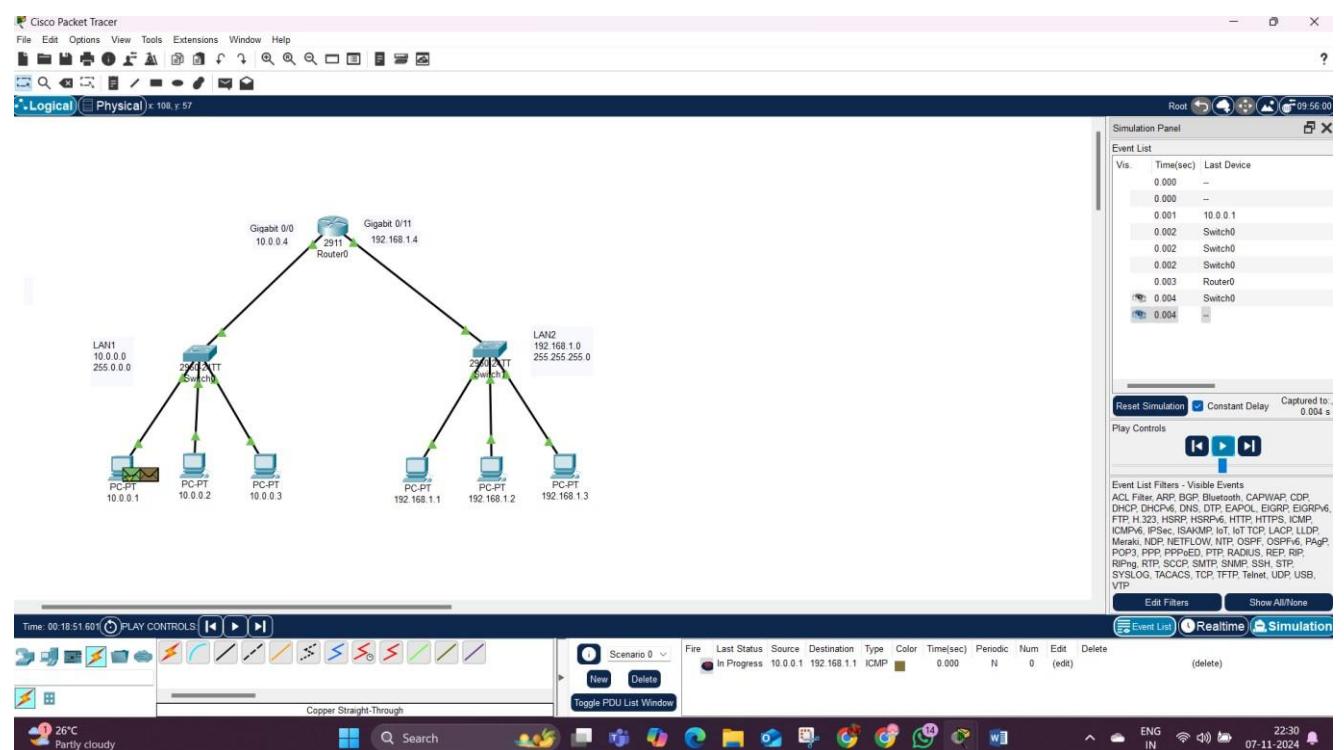
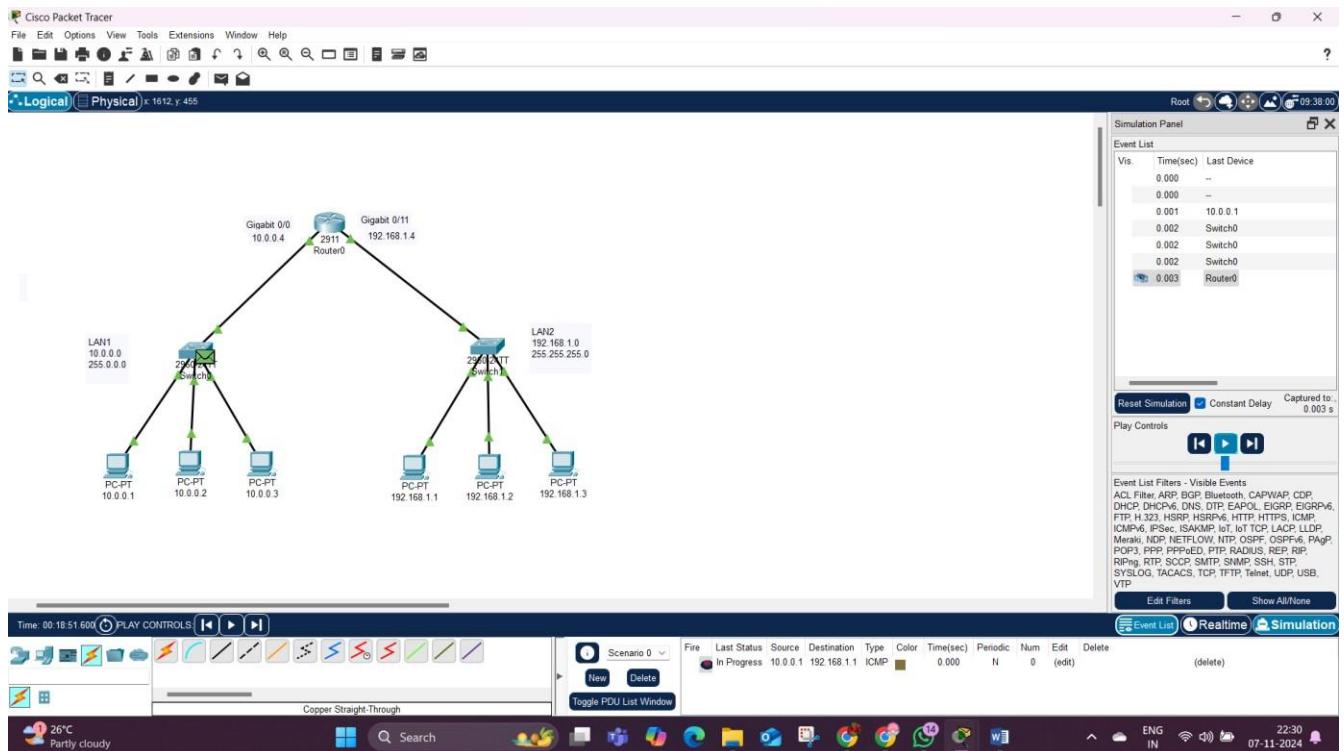
C:\>ping 192.168.1.2

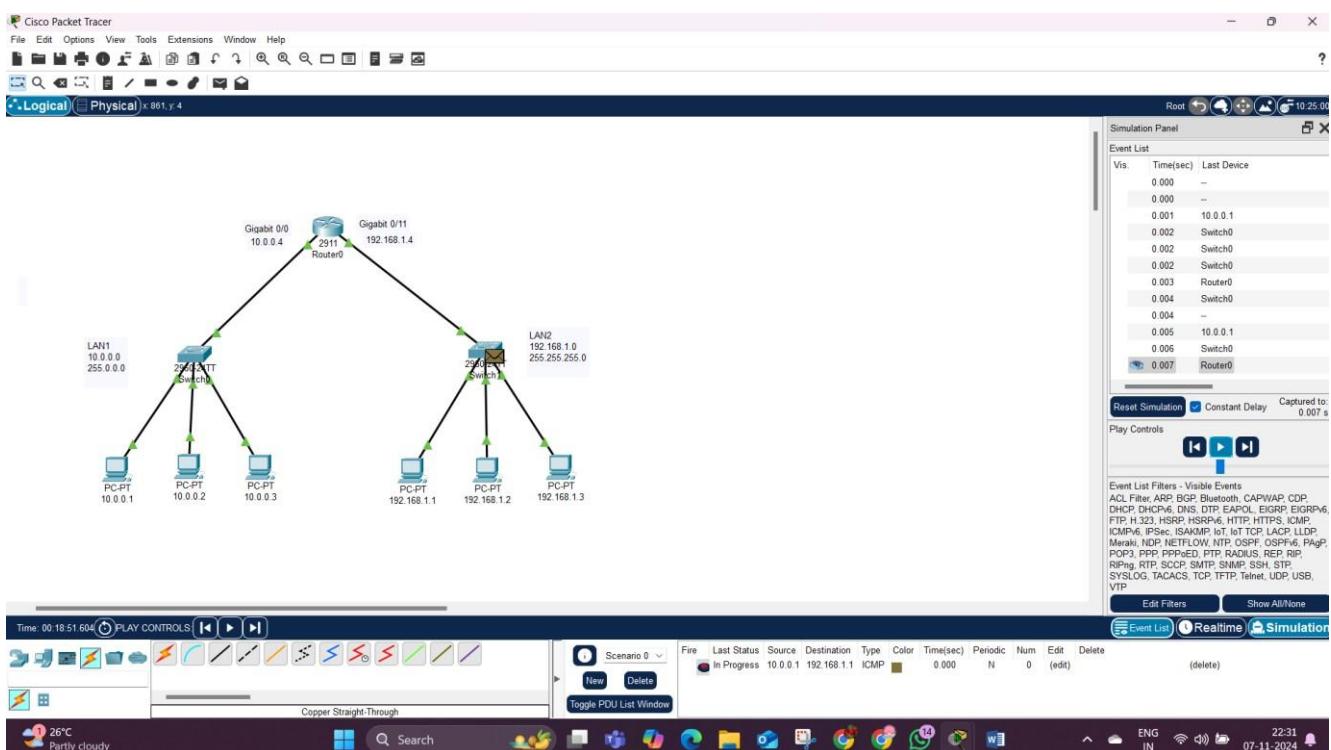
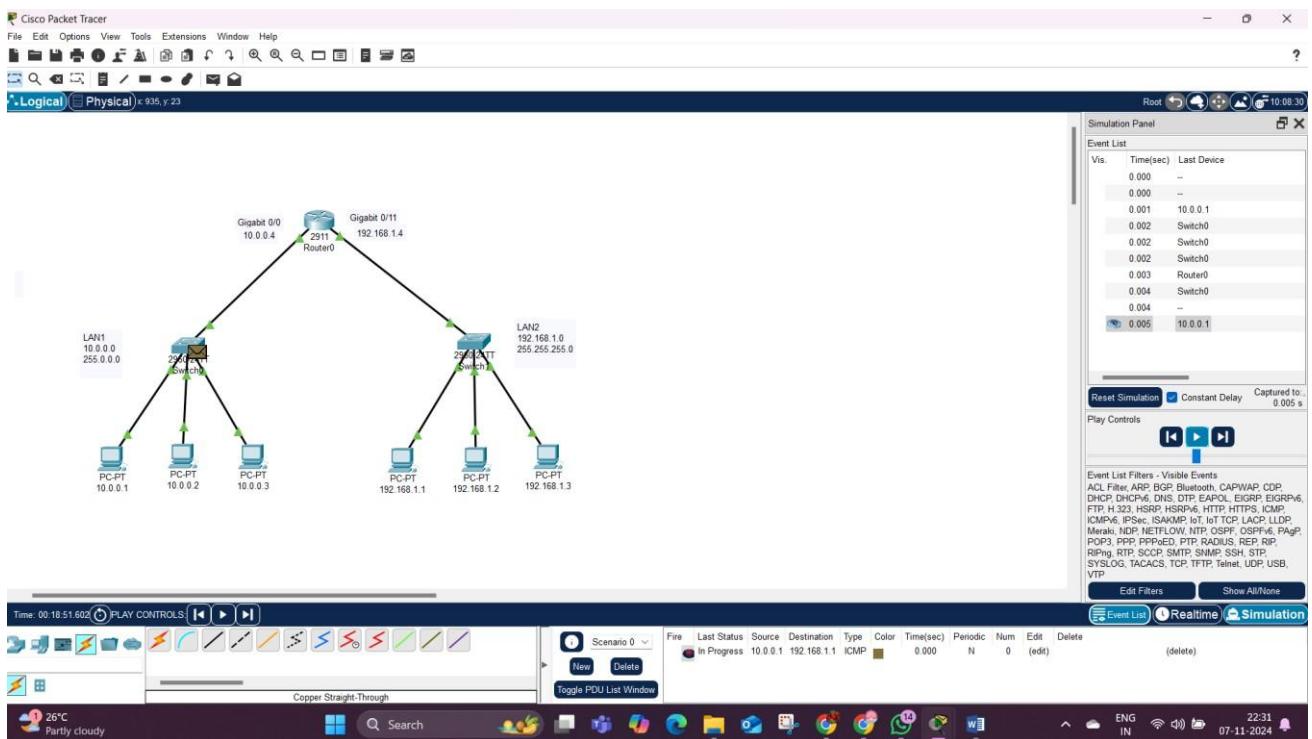
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time<1ms TTL=127
Reply from 192.168.1.2: bytes=32 time=12ms TTL=127

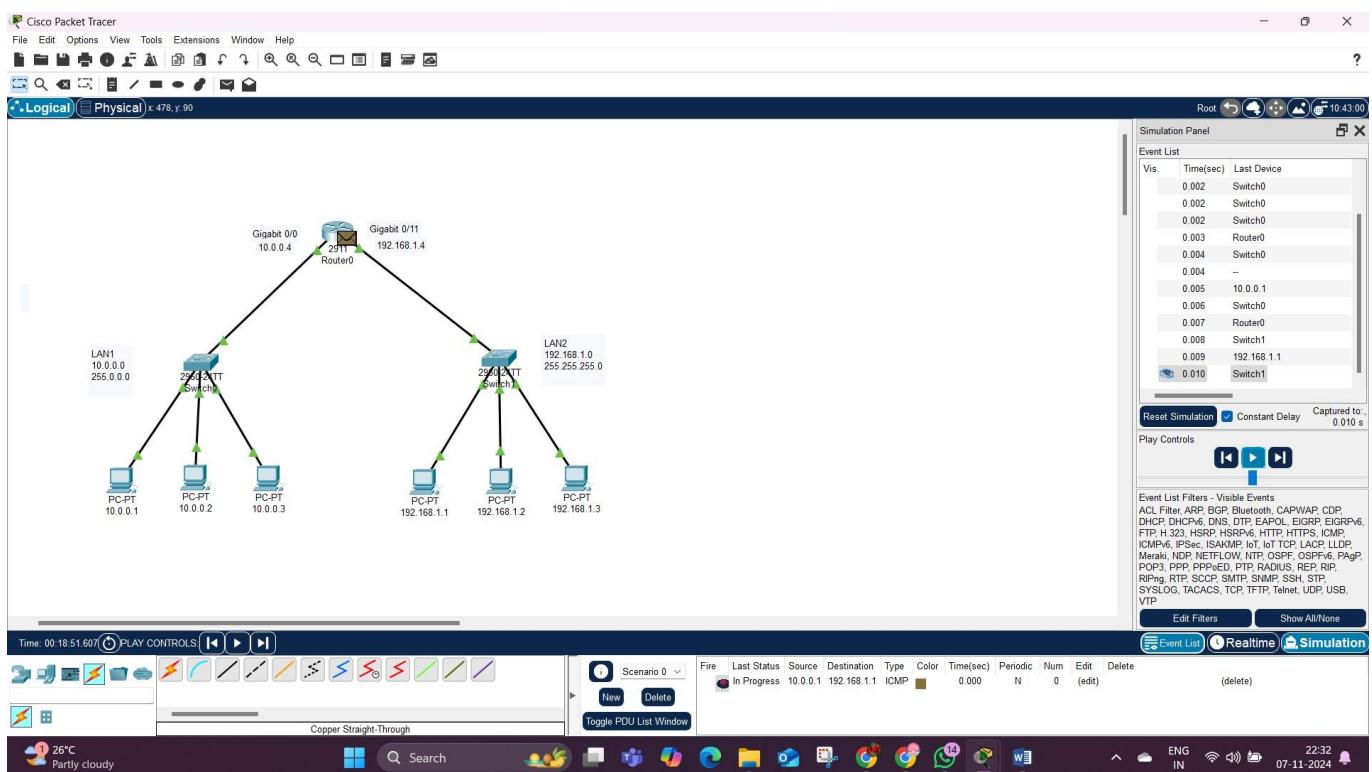
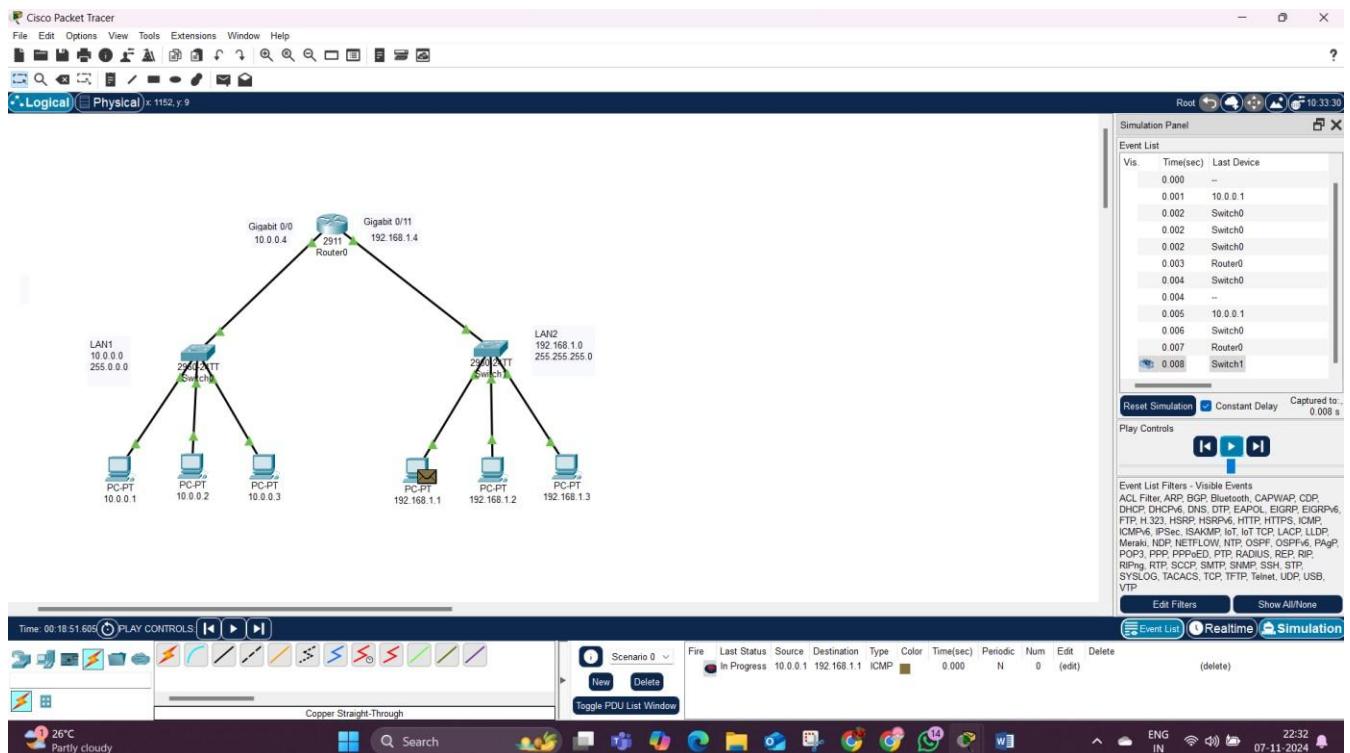
Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 12ms, Average = 4ms

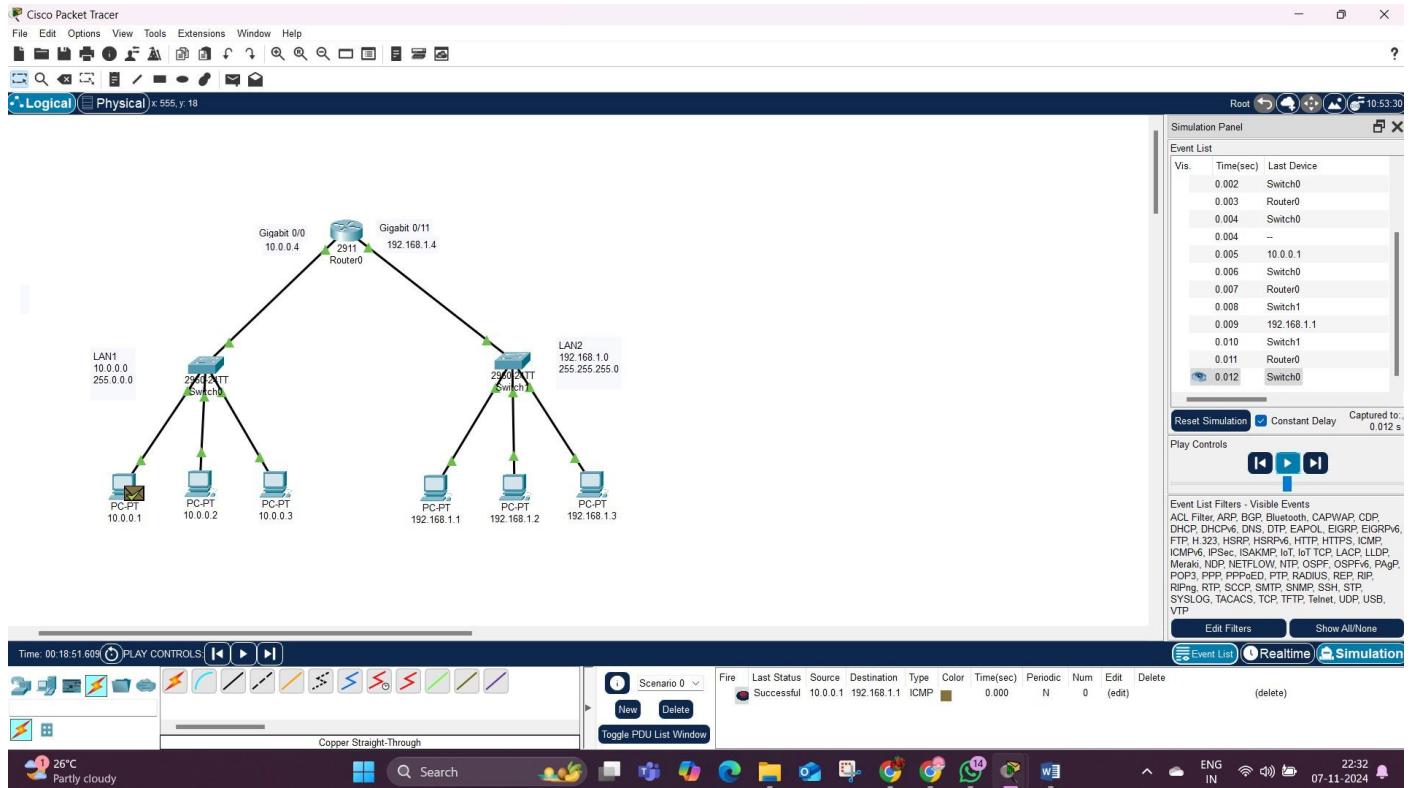
C:\>
```











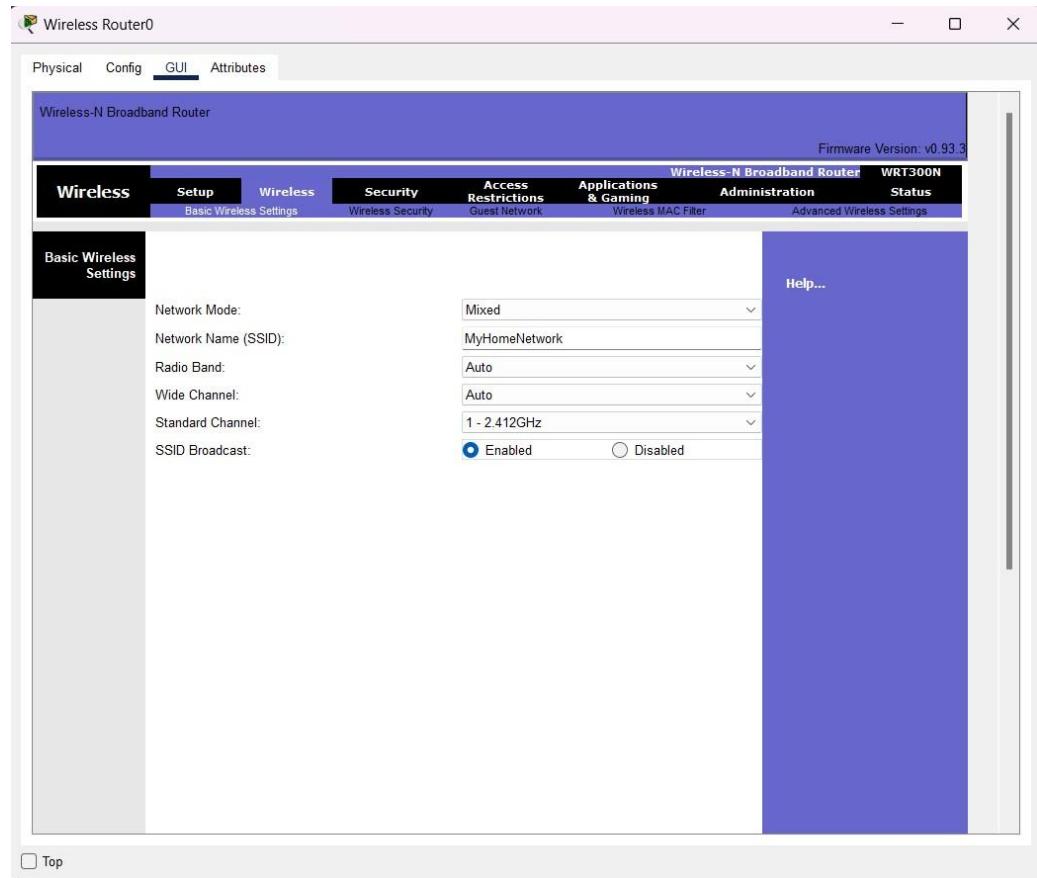
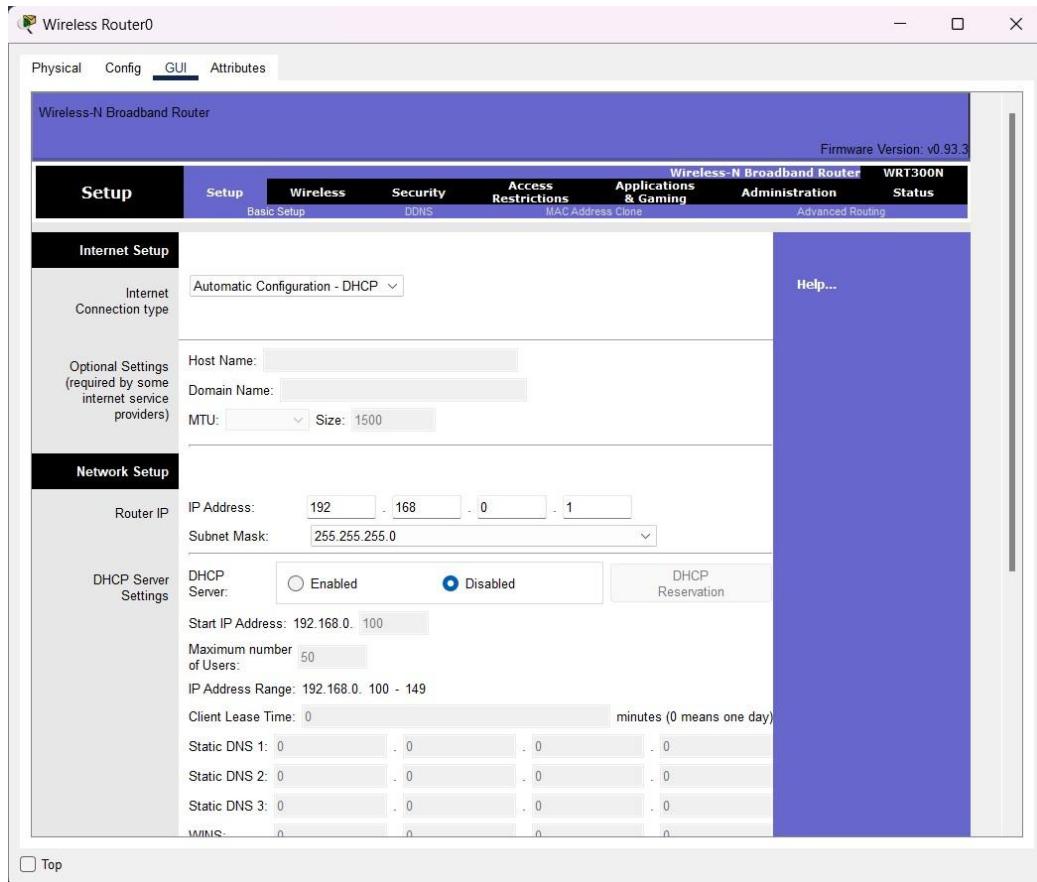
Fire	Last Status	Source	Destination	Type	Color	Time(sec)	Periodic	Num	Edit	Delete
Successful	10.0.0.1	192.168.1.1	ICMP	█	0.000	N	0	(edit)		(delete)

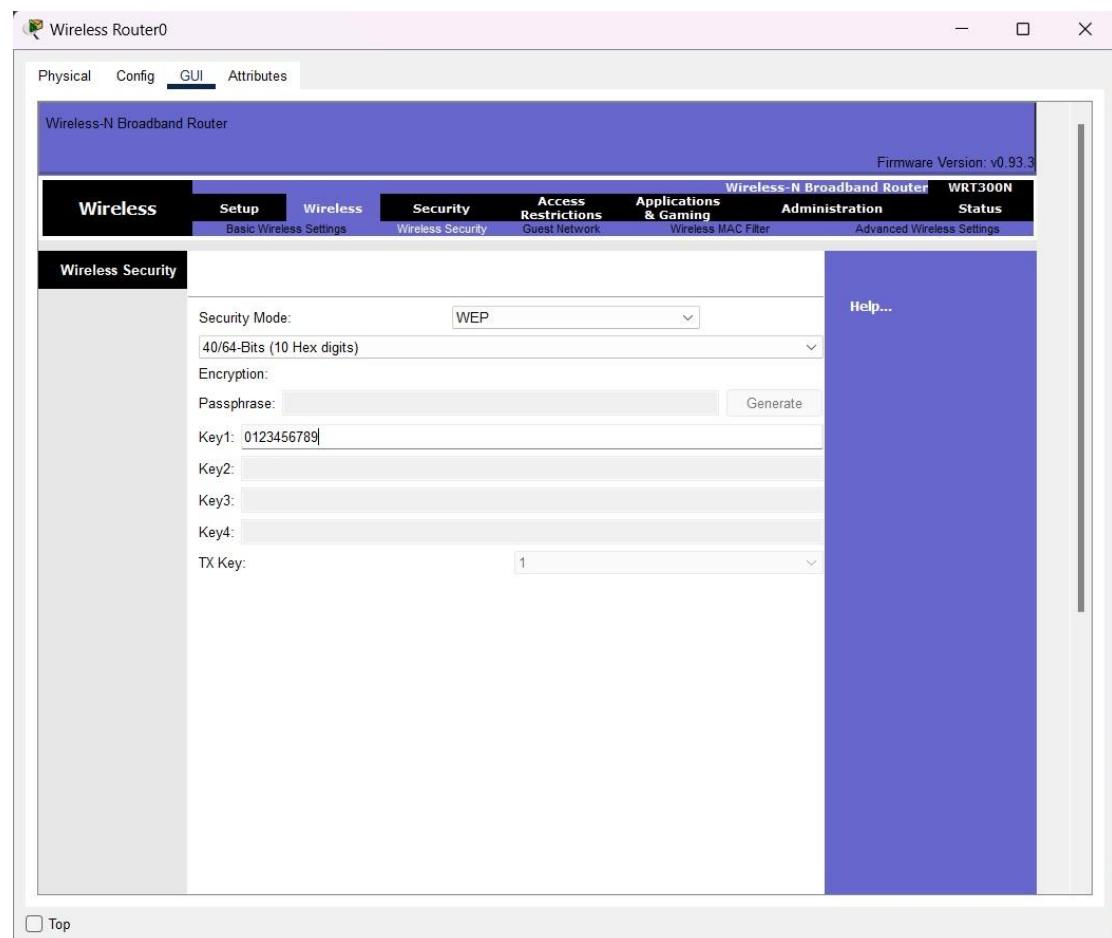
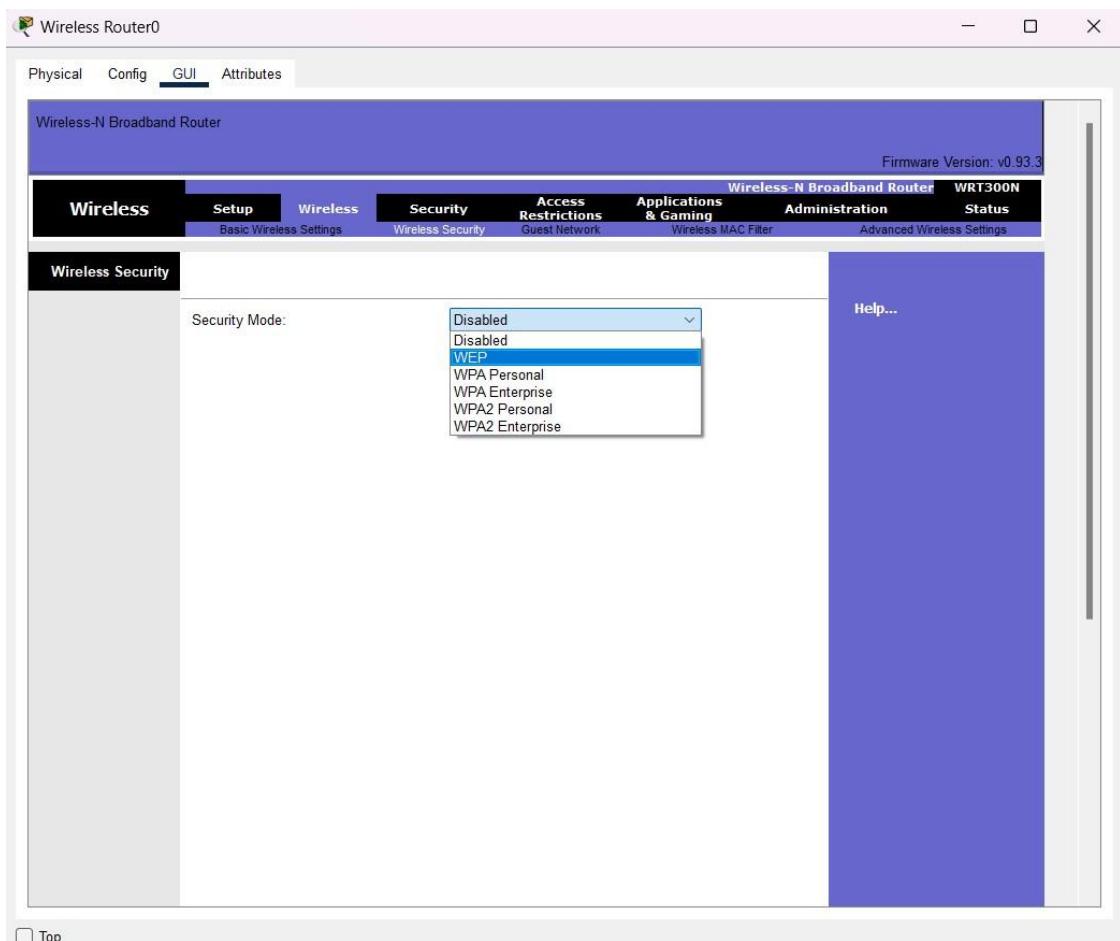
RESULT: -

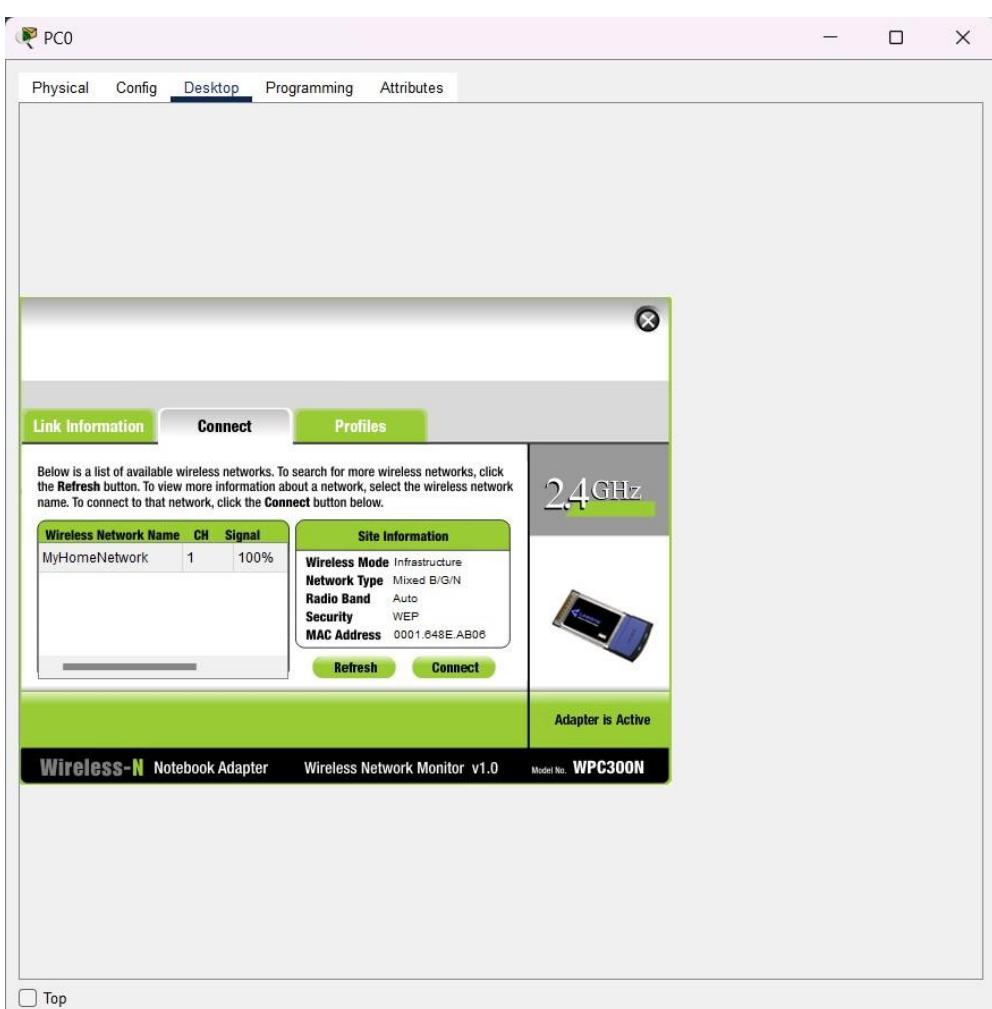
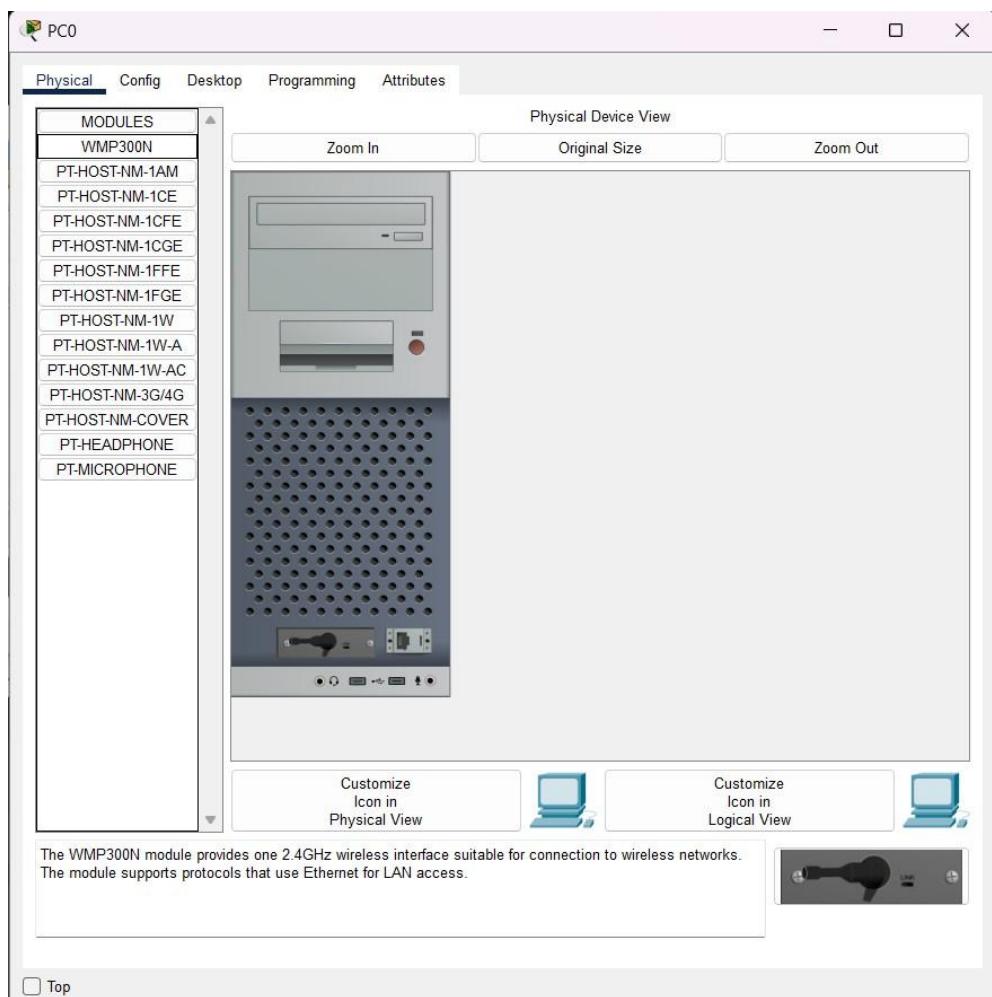
Router have been successfully done in CISCO PACKET TRACER.

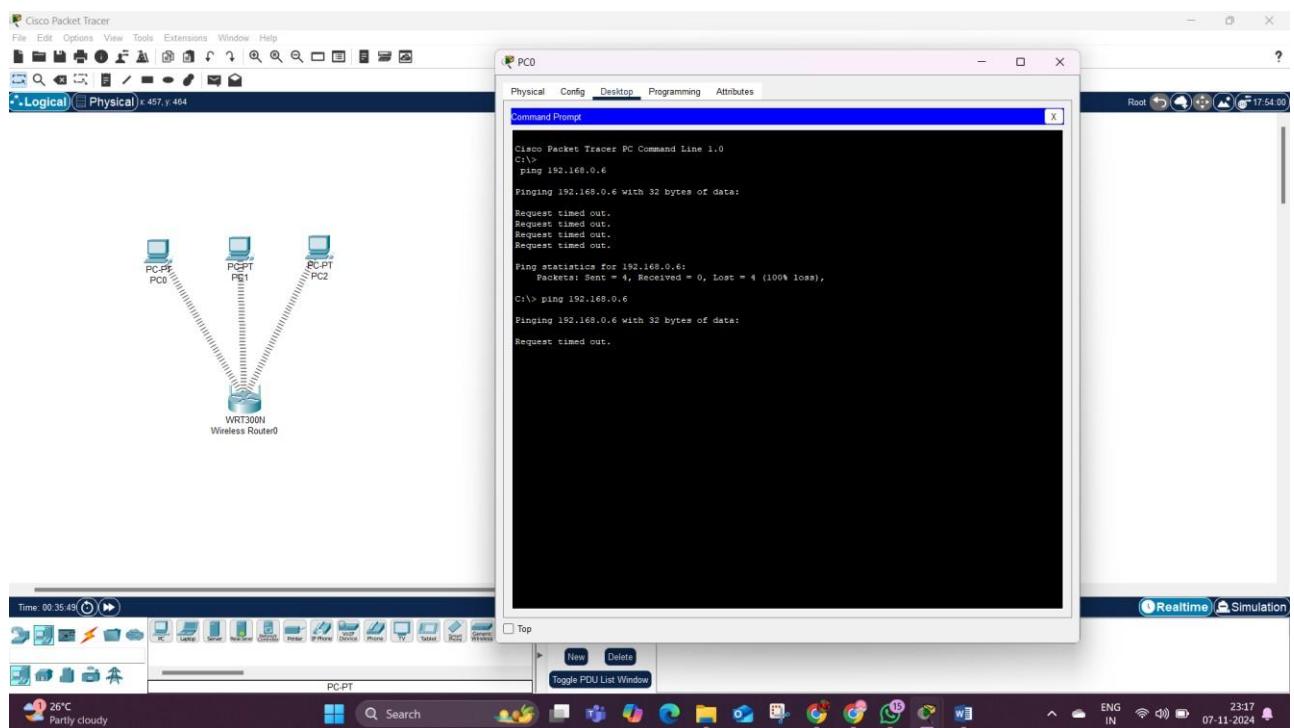
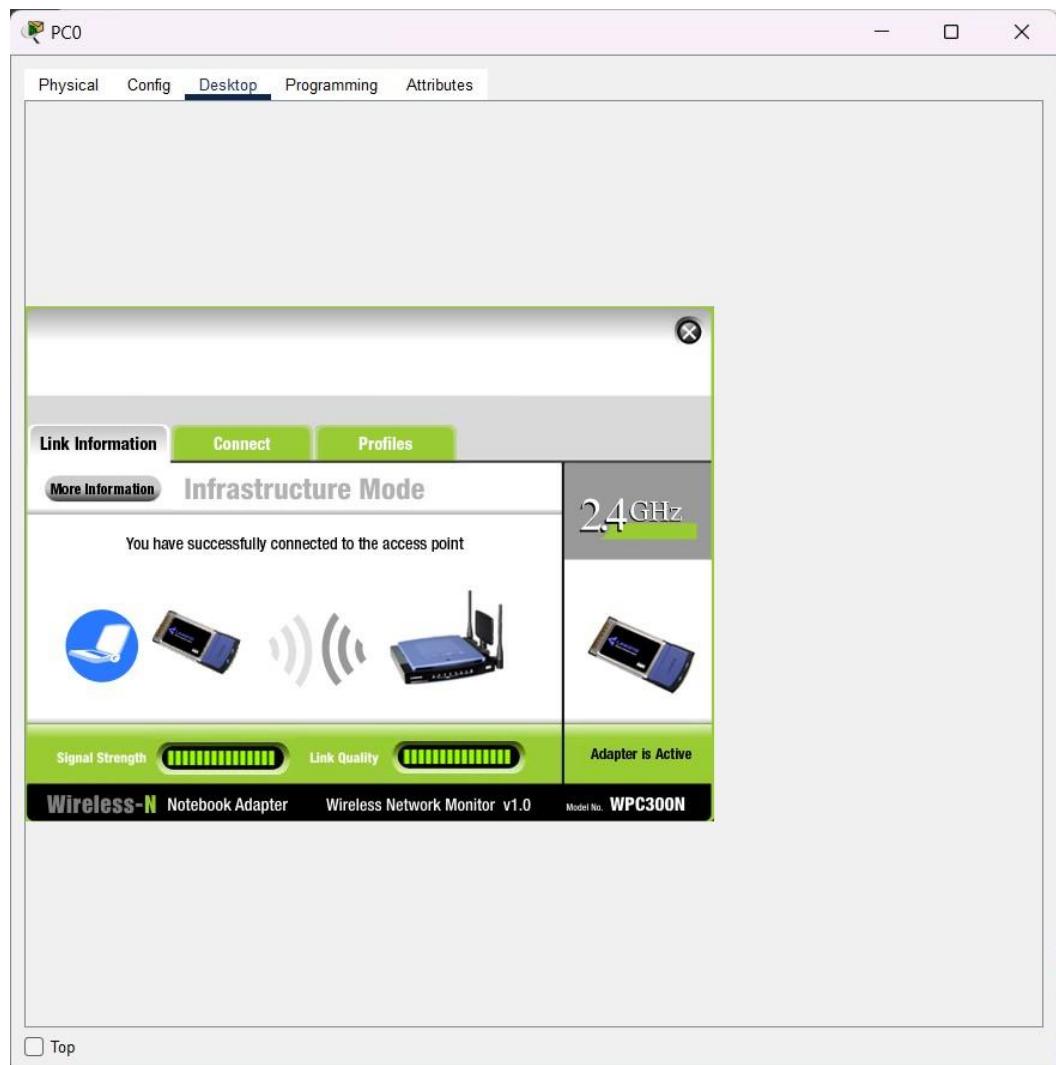
AIM: - b) Design and configure an internetwork using wireless router, DHCP server and internet cloud.

OUTPUT: -









RESULT: -

Wireless Router have been successfully done in CISCO PACKET TRACER.