# CS19541_COMPUTER NETWORKS

Experiment:1

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.

```
PS C:\Users\HDC0422076> arp -a

Interface: 172.16.53.46 --- 0xb
  Internet Address      Physical Address      Type
  172.16.52.1           7c-5a-1c-cf-be-3e     dynamic
  172.16.52.71          30-d0-42-14-73-b5     dynamic
  172.16.52.115         88-ae-dd-12-e2-2f     dynamic
  172.16.52.117         88-ae-dd-15-ef-08     dynamic
  172.16.52.118         88-ae-dd-15-ee-2e     dynamic
  172.16.52.120         88-ae-dd-15-ee-31     dynamic
  172.16.52.127         88-ae-dd-15-ef-09     dynamic
  172.16.52.128         88-ae-dd-14-6e-81     dynamic
  172.16.52.130         88-ae-dd-15-e5-c9     dynamic
  172.16.52.133         88-ae-dd-15-eb-a2     dynamic
  172.16.52.134         88-ae-dd-15-ee-0f     dynamic
  172.16.52.137         88-ae-dd-14-6e-91     dynamic
  172.16.52.143         88-ae-dd-15-ed-c6     dynamic
  172.16.52.145         88-ae-dd-15-ed-91     dynamic
  172.16.52.152         88-ae-dd-15-db-24     dynamic
  172.16.52.155         88-ae-dd-12-7e-1c     dynamic
  172.16.52.166         88-ae-dd-15-ed-2b     dynamic
  172.16.52.167         88-ae-dd-15-ed-47     dynamic
  172.16.52.171         88-ae-dd-15-ee-38     dynamic
  172.16.52.172         88-ae-dd-15-eb-cc     dynamic
  172.16.52.176         88-ae-dd-15-ed-85     dynamic
  172.16.52.181         e0-d0-45-84-5a-5d     dynamic
  172.16.53.41          88-ae-dd-15-ed-70     dynamic
  172.16.53.42          88-ae-dd-15-ec-aa     dynamic
  172.16.53.44          88-ae-dd-15-ee-6c     dynamic
  172.16.53.47          88-ae-dd-15-ed-0b     dynamic
  172.16.53.48          88-ae-dd-14-6e-0c     dynamic
  172.16.53.49          88-ae-dd-15-ee-3c     dynamic
  172.16.53.50          88-ae-dd-15-ed-1f     dynamic
  172.16.53.51          88-ae-dd-14-8a-32     dynamic
```

```
PS C:\Users\HDC0422076> hostname
DESKTOP-MEKQ8BG
PS C:\Users\HDC0422076> ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : DESKTOP-MEKQ8BG
   Primary Dns Suffix  . . . . . . . :
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter Ethernet:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek PCIe GbE Family Controller
   Physical Address. . . . . . . . . : 88-AE-DD-14-75-FA
   DHCP Enabled. . . . . . . . . . . : No
   Autoconfiguration Enabled . . . . : Yes
   Link-local IPv6 Address . . . . . : fe80::5dfc:3bef:fab1:94c3%11(Preferred)
   IPv4 Address. . . . . . . . . . . : 172.16.53.46(Preferred)
   Subnet Mask . . . . . . . . . . . : 255.255.254.0
   Default Gateway . . . . . . . . . : 172.16.52.1
   DHCPv6 IAID . . . . . . . . . . . : 109620957
   DHCPv6 Client DUID. . . . . . . . : 00-01-00-01-2D-8E-14-A7-88-AE-DD-14-75-FA
   DNS Servers . . . . . . . . . . . : 172.16.52.1
   NetBIOS over Tcpip. . . . . . . . : Enabled
PS C:\Users\HDC0422076> nbtstat -a

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
```

```
PS C:\Users\HDC0422076> 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.
PS C:\Users\HDC0422076> 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.
```

Result: The Study of various Network commands used in Linux and Windows has been successfully done.

# Experiment-5

## Wireshark

### AIM: Experiments on Packet capture tool

**Packet Sniffer**

- Sniffs messages being sent/received from/by your computer
- Store and display the contents of the various protocol fields in the messages.

**DESCRIPTION:**

**WIRESHARK**

Wireshark, a network analysis tool formerly known as Ethereal, captures packets in real time and display them in human-readable format. Wireshark includes filters, color coding, and other features that let you dig deep into network traffic and inspect individual packets. You can use Wireshark to inspect a suspicious program's network traffic, analyze the traffic flow on your network, or troubleshoot network problems.

**What we can do with Wireshark:**

- Capture network traffic
- Decode packet protocols using dissectors
- Define fil ters – capture and display
- Watch smart statistics
- Analyze problems
- Interactively browse that traffic

**Wireshark used for**:

- Network administrators: troubleshoot network problems
- Network security engineers: examine security problems
- Developers: debug protocol implementations
- People: l earn network protocol internals Getting Wireshark

Wireshark can be downloaded for Windows or macOS from its official website. For Linux or another UNIX-like system, Wireshark will be found in its package repositories. For Ubuntu,Wireshark will be found in the Ubuntu Software Center.

**Capturing Packets**

After downloading and installing Wireshark, launch it and double-click the name of a network interface under Capture to start capturing packets on that interface As soon as you click the interface's name, you'll see the packets start to appear in real time.Wireshark captures each packet sent to or from your system. If you have promiscuous mode enabled—it's enabled by default—you'll also see all the other packets on the network instead of only packets addressed to your network adapter. To check if promiscuous mode is enabled, click Capture > Options and verify the ―Enable promiscuous mode on all interfaces‖ checkbox is activated at the bottom of this window. Click the red ―Stop‖ button near the top left corner of the window when you want to stop capturing traffic.

The ―Packet List‖ Pane

The packet list pane displays all the packets in the current capture file. The ―Packet List‖ pane Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the ―Packet Details‖ and ―Packet Bytes‖ panes.

The ―Packet Details‖ Pane

The packet details pane shows the current packet (selected in the ―Packet List‖ pane) in a more detailed form. This pane shows the protocols and protocol fields of the packet selected in the ―Packet List‖ pane. The protocols and fields of the packet shown in a tree which can be expanded And collapsed.

The ―Packet Bytes‖ Pane

The packet bytes pane shows the data of the current packet (selected in the ―Packet List‖ pane) in a hexdump style.

**Color Coding**

You'll probably see packets highlighted in a variety of different colors. Wireshark uses colors to help you identify the types of traffic at a glance. By default, light purple is TCP traffic, light blue is UDP traffic, and black identifies packets with errors—for example, they could have been delivered out of order.

To view exactly what the color codes mean, click View > Coloring Rules. You can also customize and modify the coloring rules from here, if you like.

**Sample Captures:**

If there's nothing interesting on your own network to inspect, Wireshark's wiki has you covered. The wiki contains a page of sample capture files that you can load and inspect. Click File > Open in Wireshark and browse for your downloaded file to open one. You can also save your own captures in Wireshark and open them later. Click File > Save to save your captured packets.

**Filtering Packets**

If you’re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you’ll likely have a large amount of packets to sift through. That’s where Wireshark’s filters come in.

**CAPTURING AND ANALYSING PACKETS USING WIRESHARK TOOL**

To filter, capture, view, packets in Wireshark Tool. Capture 100 packets from the Ethernet: IEEE 802.3 LAN Interface and save it.

Procedure1. Create a Filter to display only TCP/UDP packets, inspect the packets and provide the flow graph

2. Create a Filter to display only ARP packets and inspect the packets.

Procedure

3. Create a Filter to display only DNS packets and provide the flow graph.

4. Create a Filter to display only HTTP packets and inspect the packets

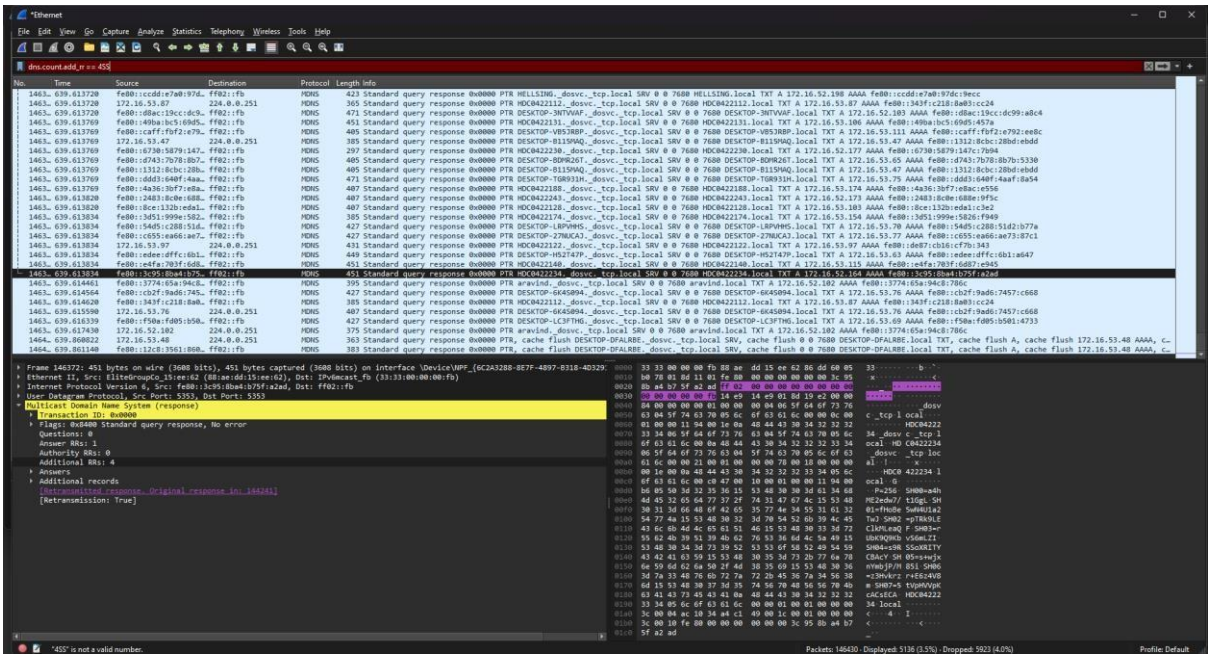
Procedure

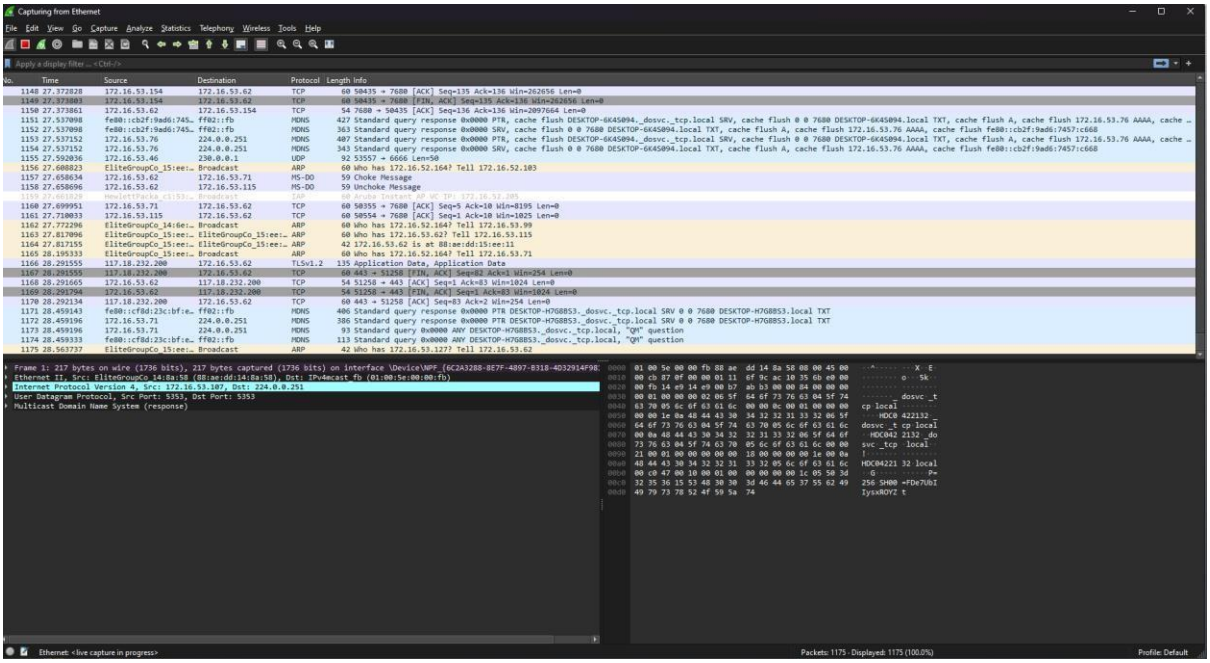
5. Create a Filter to display only IP/ICMP packets and inspect the packets.

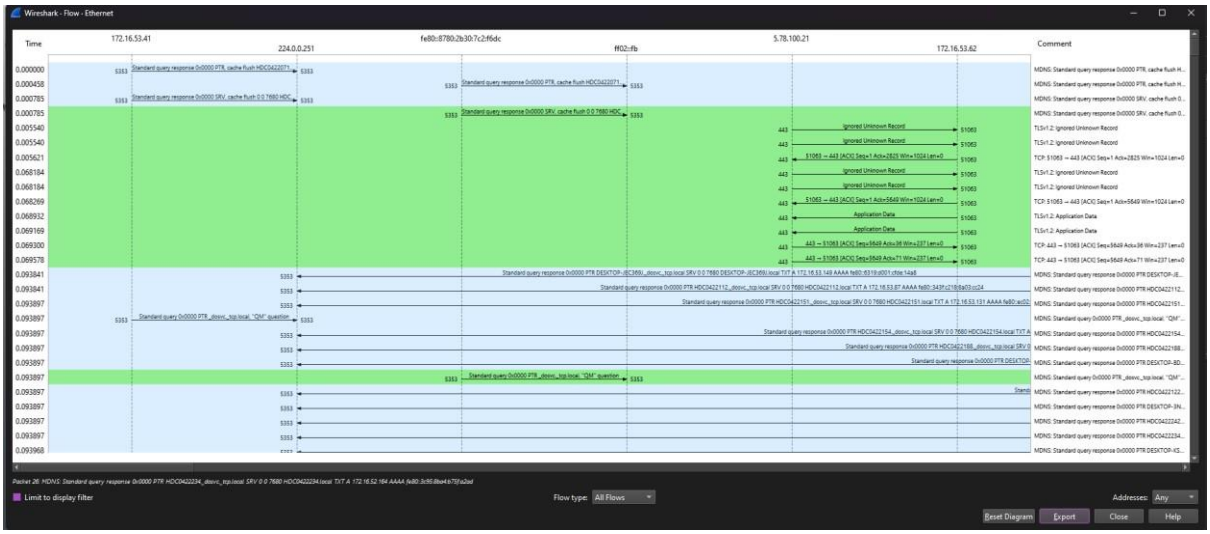
Procedure

6. Create a Filter to display only DHCP packets and inspect the packets.

**Procedure :**


*Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dns.count.add_r == 455
No. Time Source Destination Protocol Length Info
Frame 146372: 451 bytes on wire (3608 bits), 451 bytes captured (3608 bits) on interface \Device\NPF_{6C2A3288-8E7F-4897-B318-4D329
Ethernet II, Src: EliteGroupCo_33:ee:62 (88:ae:dd:15:ee:62), Dst: IPv4mcast_fb (33:33:00:00:00:fb)
Internet Protocol Version 6, Src: fe80::3c95:8ba4:b75f:a2ad, Dst: ff02::fb
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (response)
Transaction ID: 0x0000
Flags: 0x8400 Standard query response, No error
Questions: 0
Answer RRs: 1
Authority RRs: 0
Additional RRs: 4
Answers
Additional records
[Retransmitted response. Original response in: 144241]
[Retransmission: True]
'455' is not a valid number.
Packets: 146430 · Displayed: 5136 (3.5%) · Dropped: 5923 (4.0%)
Profile: Default


Capturing from Ethernet
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Info
1148 27.372828 172.16.53.154 172.16.53.62 TCP 60 50435 → 7680 [ACK] Seq=135 Ack=136 Win=262656 Len=0
1149 27.373003 172.16.53.154 172.16.53.62 TCP 60 50435 → 7680 [FIN, ACK] Seq=135 Ack=136 Win=262656 Len=0
1150 27.373861 172.16.53.62 172.16.53.154 TCP 54 7680 → 50435 [ACK] Seq=136 Ack=136 Win=2097664 Len=0
1151 27.537098 fe80::cb2f:9ad6:745… ff02::fb MDNS 427 Standard query response 0x0000 PTR, cache flush DESKTOP-6K4S094._dosvc._tcp.local SRV, cache flush 0 0 7680 DESKTOP-6K4S094.local TXT, cache flush A, cache flush 172.16.53.76 AAAA, cache …
1152 27.537098 fe80::cb2f:9ad6:745… ff02::fb MDNS 363 Standard query response 0x0000 SRV, cache flush 0 0 7680 DESKTOP-6K4S094.local TXT, cache flush A, cache flush 172.16.53.76 AAAA, cache flush fe80::cb2f:9ad6:7457:c668
1153 27.537152 172.16.53.76 224.0.0.251 MDNS 407 Standard query response 0x0000 PTR, cache flush DESKTOP-6K4S094._dosvc._tcp.local SRV, cache flush A, cache flush 0 0 7680 DESKTOP-6K4S094.local TXT, cache flush A, cache flush 172.16.53.76 AAAA, cache …
1154 27.537152 172.16.53.76 224.0.0.251 MDNS 343 Standard query response 0x0000 SRV, cache flush 0 0 7680 DESKTOP-6K4S094.local TXT, cache flush A, cache flush 172.16.53.76 AAAA, cache flush fe80::cb2f:9ad6:7457:c668
1155 27.592036 172.16.53.46 230.0.0.1 UDP 92 53557 → 6666 Len=50
1156 27.608823 EliteGroupCo_15:ee:… Broadcast ARP 60 Who has 172.16.52.164? Tell 172.16.52.103
1157 27.658634 172.16.53.62 172.16.53.71 MS-DO 59 Choke Message
1158 27.658696 172.16.53.62 172.16.53.115 MS-DO 59 Unchoke Message
1159 27.681828 HewlettPacka_c1:53:… Broadcast TAP 60 Aruba Instant_AP_VC_TP: 172.16.52.205
1160 27.699951 172.16.53.71 172.16.53.62 TCP 60 50355 → 7680 [ACK] Seq=5 Ack=10 Win=8195 Len=0
1161 27.710033 172.16.53.115 172.16.53.62 TCP 60 50554 → 7680 [ACK] Seq=1 Ack=10 Win=1025 Len=0
1162 27.772296 EliteGroupCo_14:6e:… Broadcast ARP 60 Who has 172.16.52.164? Tell 172.16.53.99
1163 27.817096 EliteGroupCo_15:ee:… EliteGroupCo_15:ee:… ARP 60 Who has 172.16.53.62? Tell 172.16.53.115
1164 27.817155 EliteGroupCo_15:ee:… EliteGroupCo_15:ee:… ARP 42 172.16.53.62 is at 88:ae:dd:15:ee:11
1165 28.195333 EliteGroupCo_15:ee:… Broadcast ARP 60 Who has 172.16.52.164? Tell 172.16.53.71
1166 28.291555 117.18.232.200 172.16.53.62 TLSv1.2 135 Application Data, Application Data
1167 28.291555 117.18.232.200 172.16.53.62 TCP 60 443 → 51250 [FIN, ACK] Seq=82 Ack=1 Win=254 Len=0
1168 28.291665 172.16.53.62 117.18.232.200 TCP 54 51250 → 443 [ACK] Seq=1 Ack=83 Win=1024 Len=0
1169 28.291794 172.16.53.62 117.18.232.200 TCP 54 51250 → 443 [FIN, ACK] Seq=1 Ack=83 Win=1024 Len=0
1170 28.292134 117.18.232.200 172.16.53.62 TCP 60 443 → 51250 [ACK] Seq=83 Ack=2 Win=254 Len=0
1171 28.459143 fe80::cf8d:23c:bf1e… ff02::fb MDNS 406 Standard query response 0x0000 PTR DESKTOP-H7G8853._dosvc._tcp.local SRV 0 0 7680 DESKTOP-H7G8853.local TXT
1172 28.459196 172.16.53.71 224.0.0.251 MDNS 386 Standard query response 0x0000 PTR DESKTOP-H7G8853._dosvc._tcp.local SRV 0 0 7680 DESKTOP-H7G8853.local TXT
1173 28.459196 172.16.53.71 224.0.0.251 MDNS 93 Standard query 0x0000 ANY DESKTOP-H7G8853._dosvc._tcp.local, "QM" question
1174 28.459333 fe80::cf8d:23c:bf1e… ff02::fb MDNS 113 Standard query 0x0000 ANY DESKTOP-H7G8853._dosvc._tcp.local, "QM" question
1175 28.563737 EliteGroupCo_15:ee:… Broadcast ARP 42 Who has 172.16.53.122? Tell 172.16.53.62
Frame 1: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{6C2A3288-8E7F-4897-B318-4D3291
Ethernet II, Src: EliteGroupCo_14:8a:58 (88:ae:dd:14:8a:58), Dst: IPv4mcast_fb (01:00:5e:00:00:fb)
Internet Protocol Version 4, Src: 172.16.53.107, Dst: 224.0.0.251
User Datagram Protocol, Src Port: 5353, Dst Port: 5353
Multicast Domain Name System (response)
Ethernet: <live capture in progress>
Packets: 1175 · Displayed: 1175 (100.0%)
Profile: Default




## RESULT:

The Experiments on Packet capture tool 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.

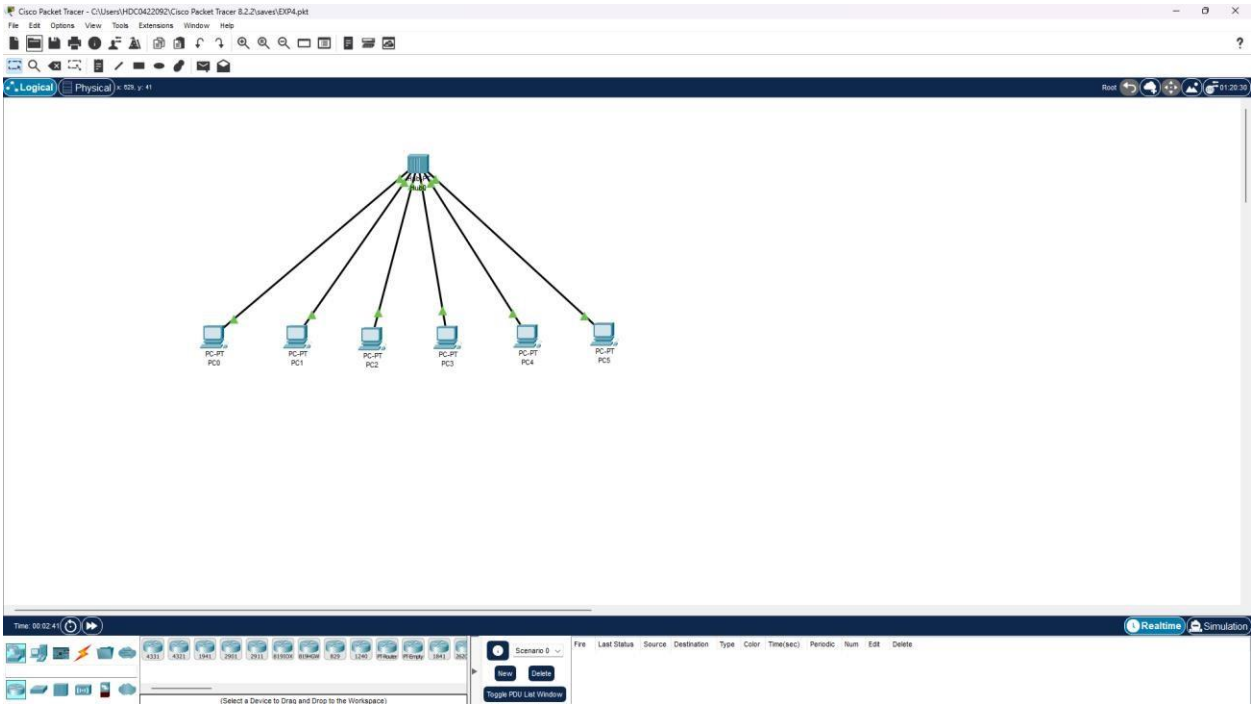
3. Configure basic settings: Once you're logged in, you will be able to configure basic settings for the switch,

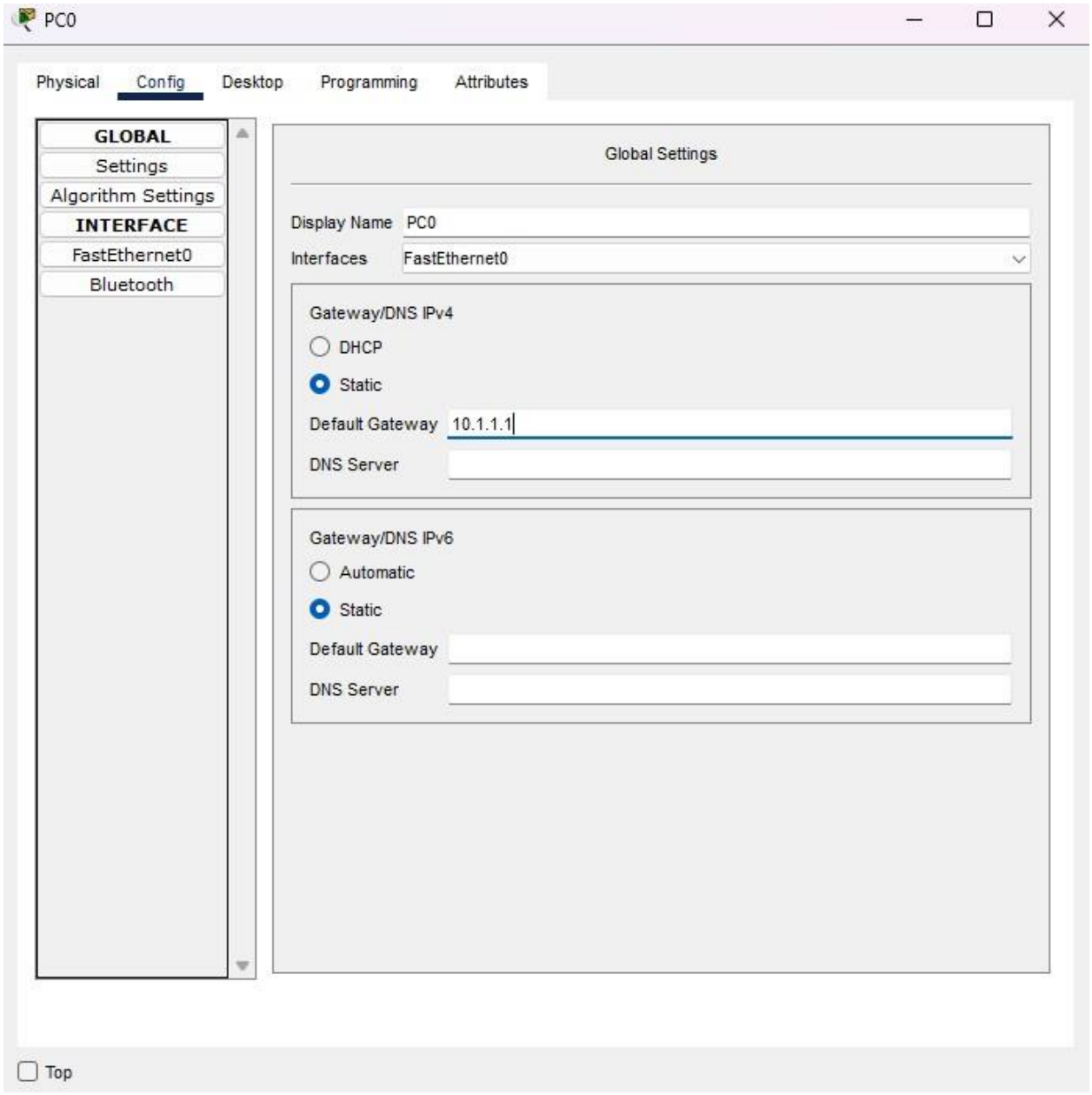
4. Assign IP address as: 10.1.1.5, subnet mask 255.0.0.0.

**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-9

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

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.

## CREATING A NETWORK TOPOLOGY:

The first step in implementing classless IP subnetting is to create a network topology in Packet Tracer. To create a network topology in Packet Tracer, select the "New" button in the top left corner, then select "Network" and "Generic". This will create a blank network topology that we can use to add devices.

## ADDING THE DEVICES:

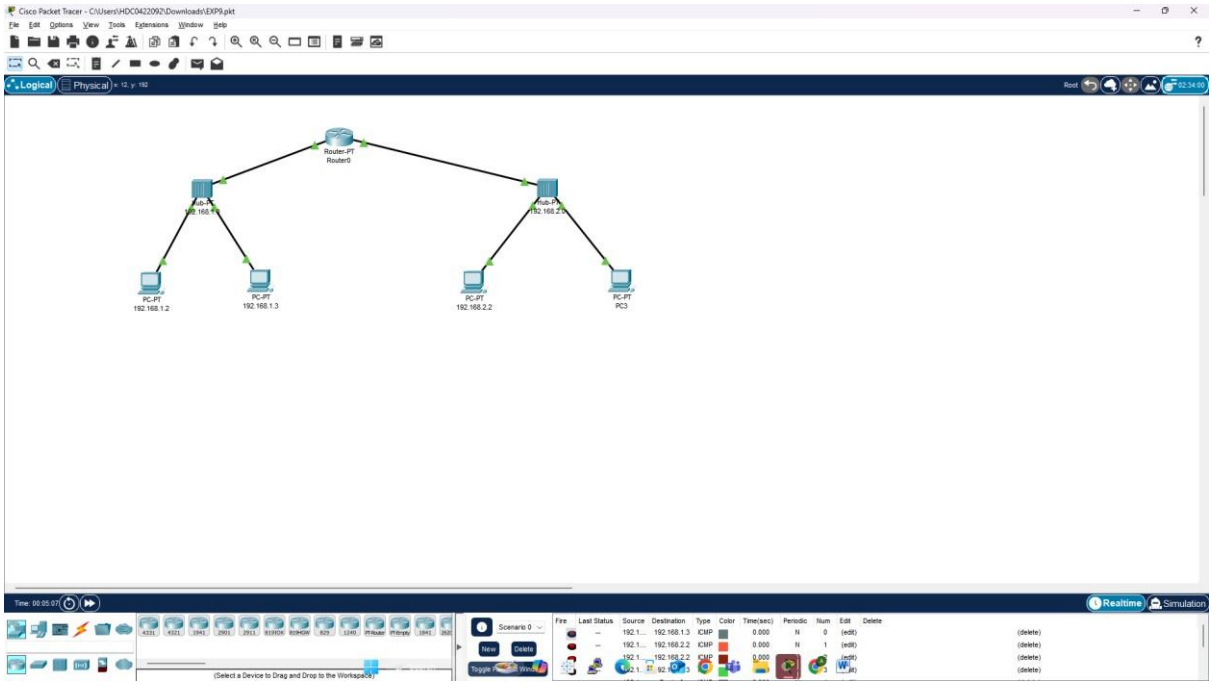
Once we have created our network topology, we can add devices to it. Here, we will be adding routers, witches, and PCs. To add a device, select the device from the bottom left corner and drag it onto the network topology. Then, connect the devices by dragging a cable from one device's port to another device's port.

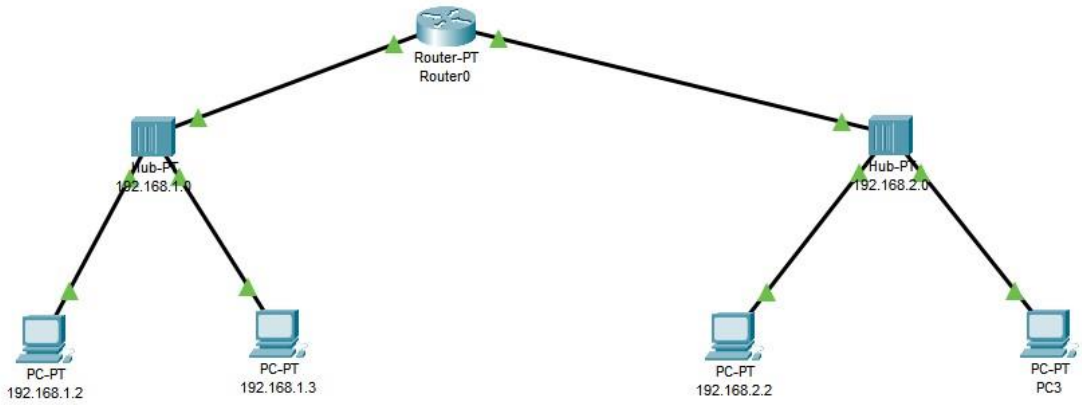
## SUBNETTING:

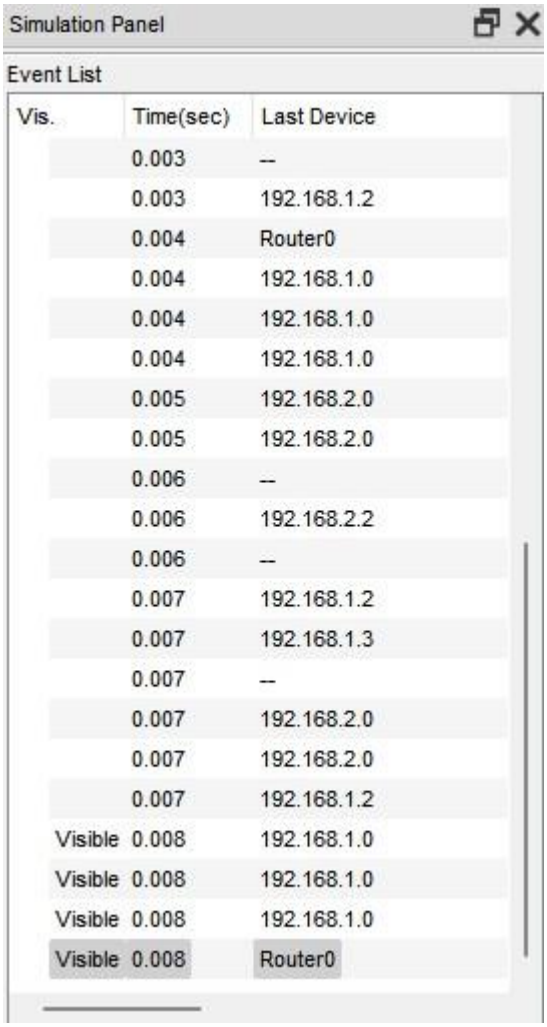
To subnet the network address of 192.168.1.0/24 to provide enough space for at least 5 addresses for end devices, the switch, and the router, we can use a /27 subnet mask. This will give us 8 subnets with 30 host addresses each.

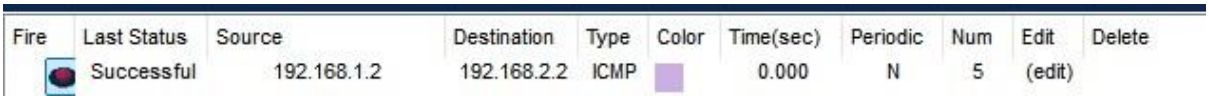
## TESTING THE NETWORK:

Now that our network topology is configured, we can test the network. Open a commandprompt on each PC and try to ping the other PC. If the ping is successful, then the network isfunctioning properly. We can also use the"ping" command to test connectivity between the router and the PCs.


Cisco Packet Tracer - C:\Users\HDC0422092\Downloads\EXP9.pkt
File Edit Options View Tools Extensions Window Help
Logical
Physical
Router-PT
Router0
PC-PT
192.168.1.2
PC-PT
192.168.1.3
PC-PT
192.168.2.2
PC-PT
PC3
Realtime
Simulation
(Select a Device to Drag and Drop to the Workspace)


Router-PT
Router0
Hub-PT
192.168.1.0
Hub-PT
192.168.2.0
PC-PT
192.168.1.2
PC-PT
192.168.1.3
PC-PT
192.168.2.2
PC-PT
PC3




**RESULT:**

The Implementation of SUBNETTING in CISCO PACKET TRACER is been successfully executed.

# Experiment-8

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

**Packet Tracer - Configure VLANs and Trunking - Physical Mode Topology**

Part 1: Build the Network and Configure Basic Device Settings

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.

Part 2: Create VLANs and Assign Switch Ports

Step 1: Create VLANs on the switches. Step 2: Assign VLANs to the correct switch interfaces.

Part 3: Maintain VLAN Port Assignments and the VLAN Database

Step 1: Assign a VLAN to multiple interfaces.

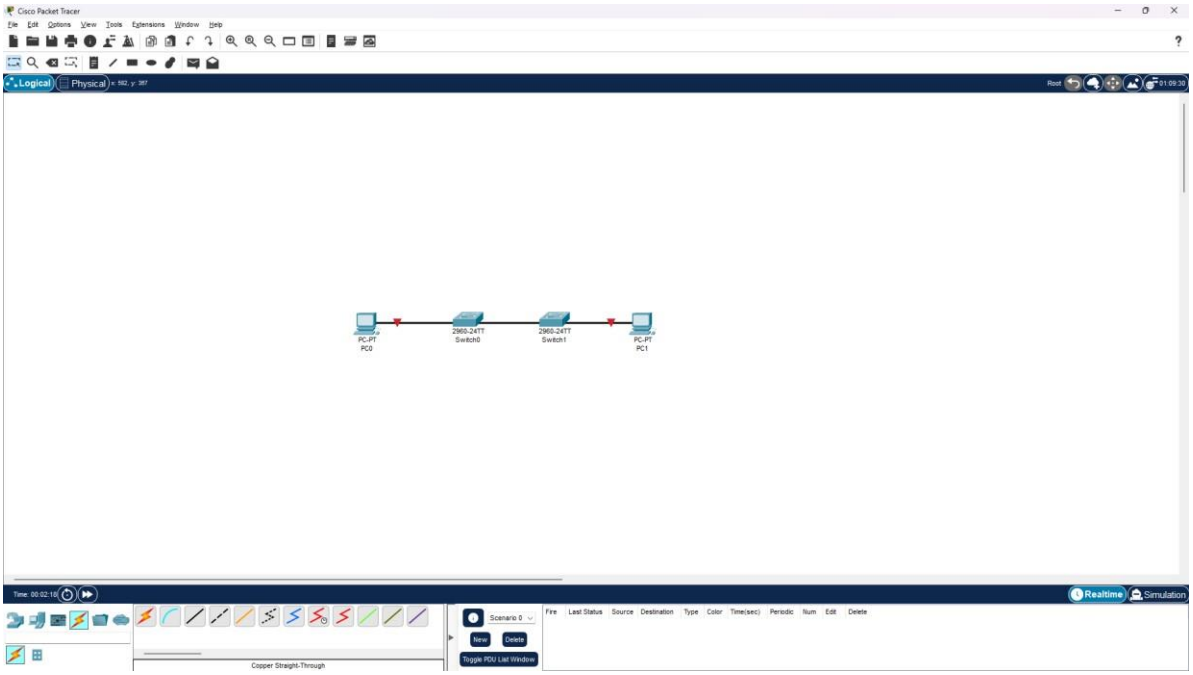
Step 2: Remove a VLAN assignment from an interface.

Step 3: Remove aVLAN ID from the VLAN database.

Part 4: Configure an 802.1Q Trunk Between the Switches

Step 1: Use DTP to initiate trunking on F0/1.

Step 2: Manually configure trunk interface F0/1.




| Device | Interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| S1 | VLAN 1 | 192.168.1.11 | 255.255.255.0 | N/A |
| S2 | VLAN 1 | 192.168.1.12 | 255.255.255.0 | N/A |
| PC-A | NIC | 192.168.10.3 | 255.255.255.0 | 192.168.10.1 |
| PC-B | NIC | 192.168.10.4 | 255.255.255.0 | 192.168.10.1 |

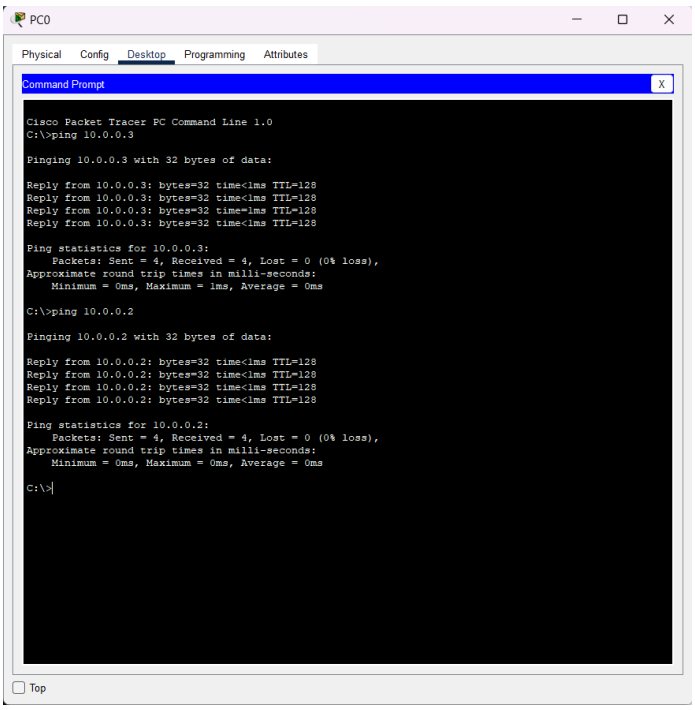
**Result:**

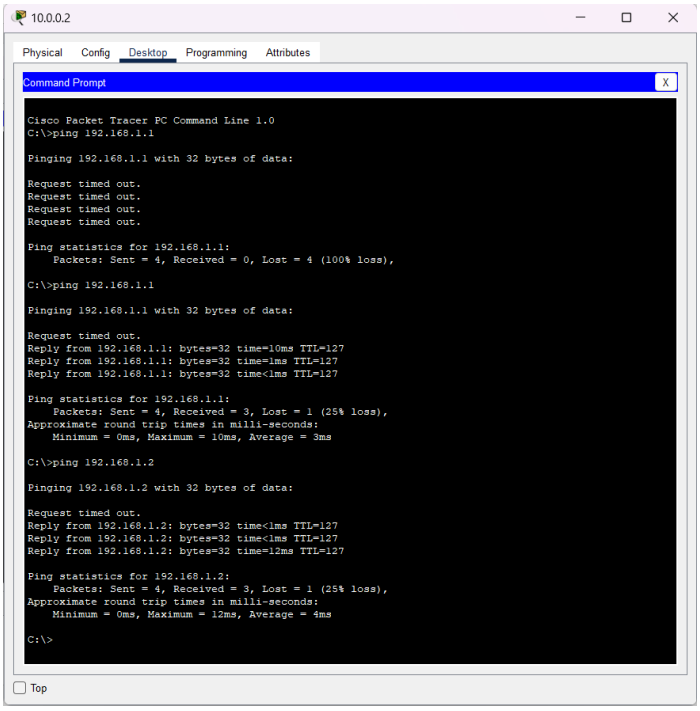
The Simulate Virtual LAN configuration using CISCO Packet Tracer
Simulation has been suceessfully executed

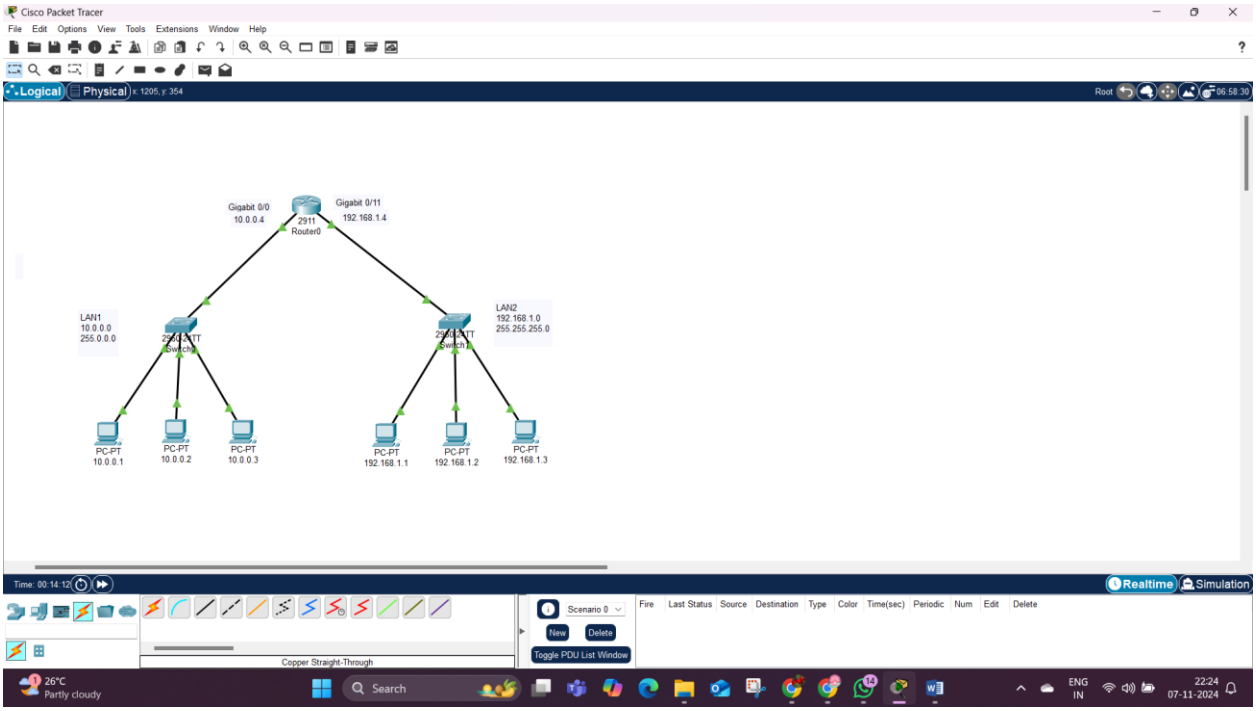
# EXPERIMENT – 10

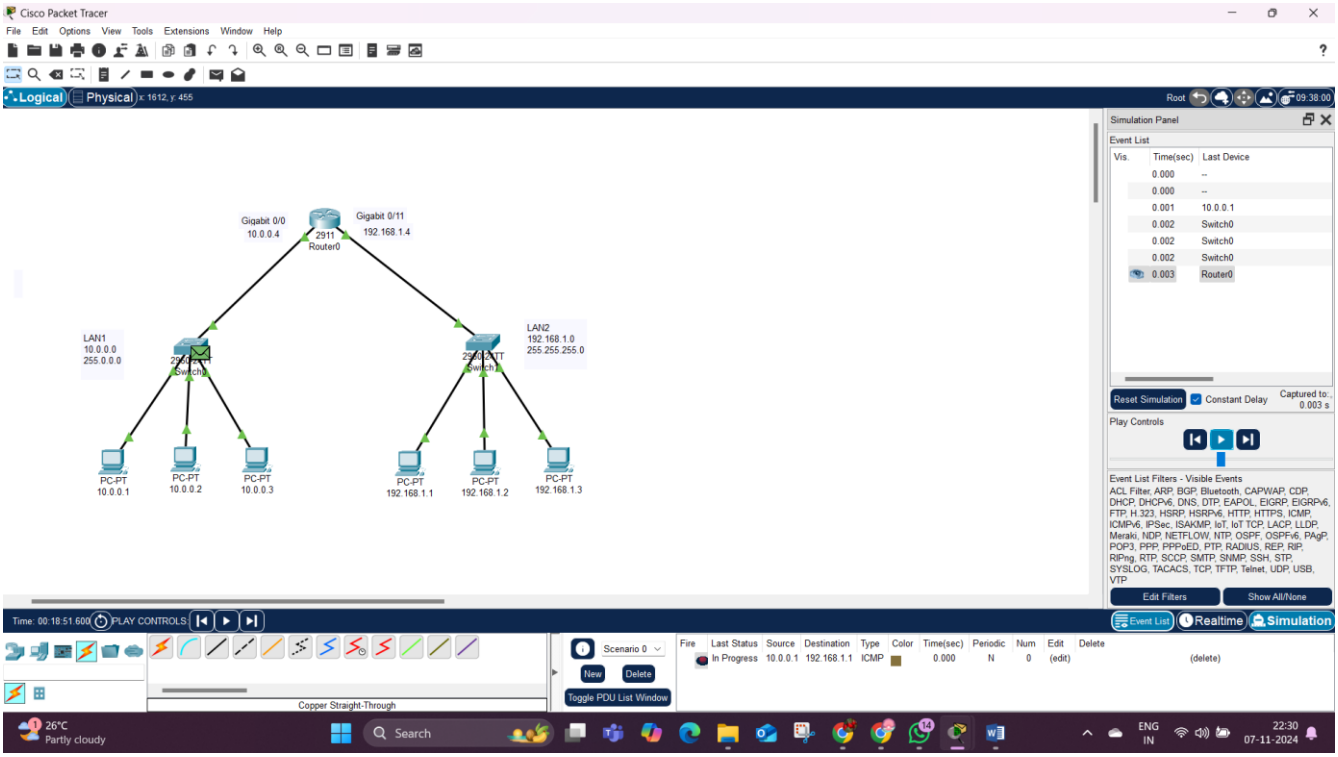
**AIM:** - a) Internetworking with routers in CISCO PACKET TRACER simulator.

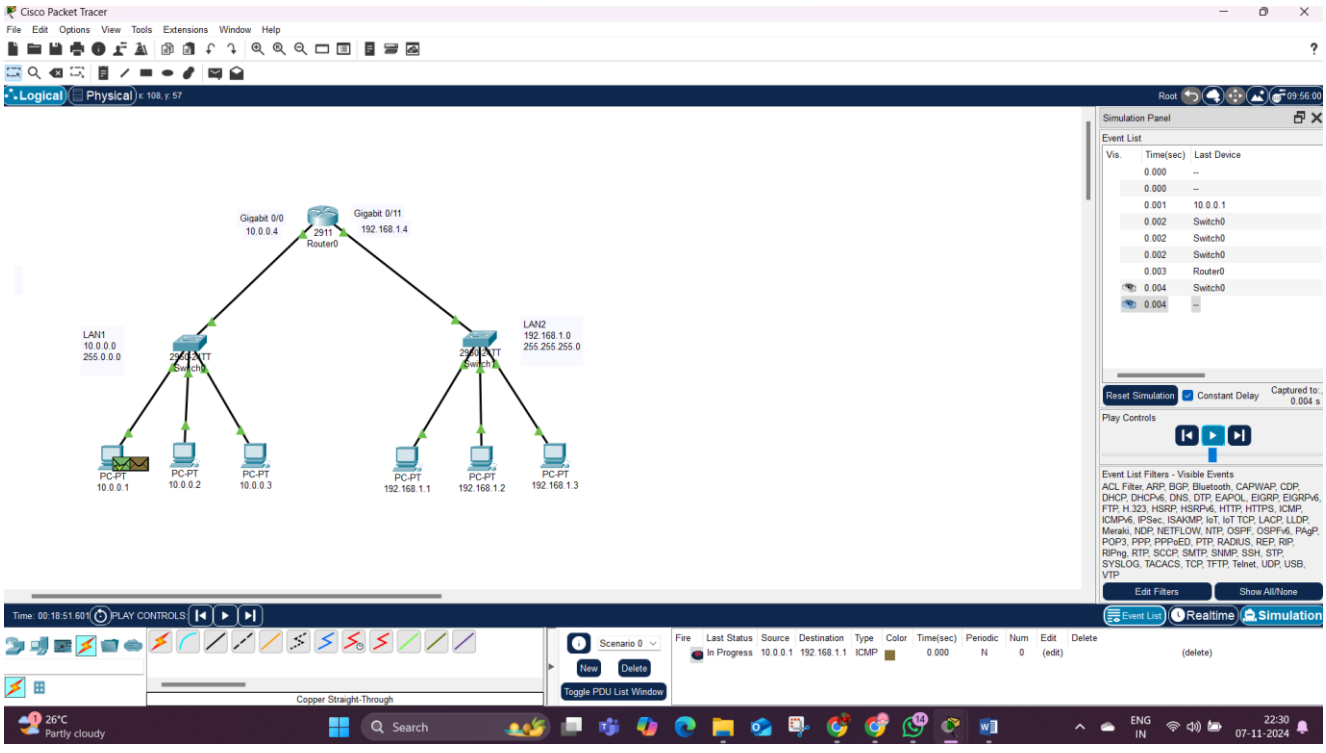
## OUTPUT: -

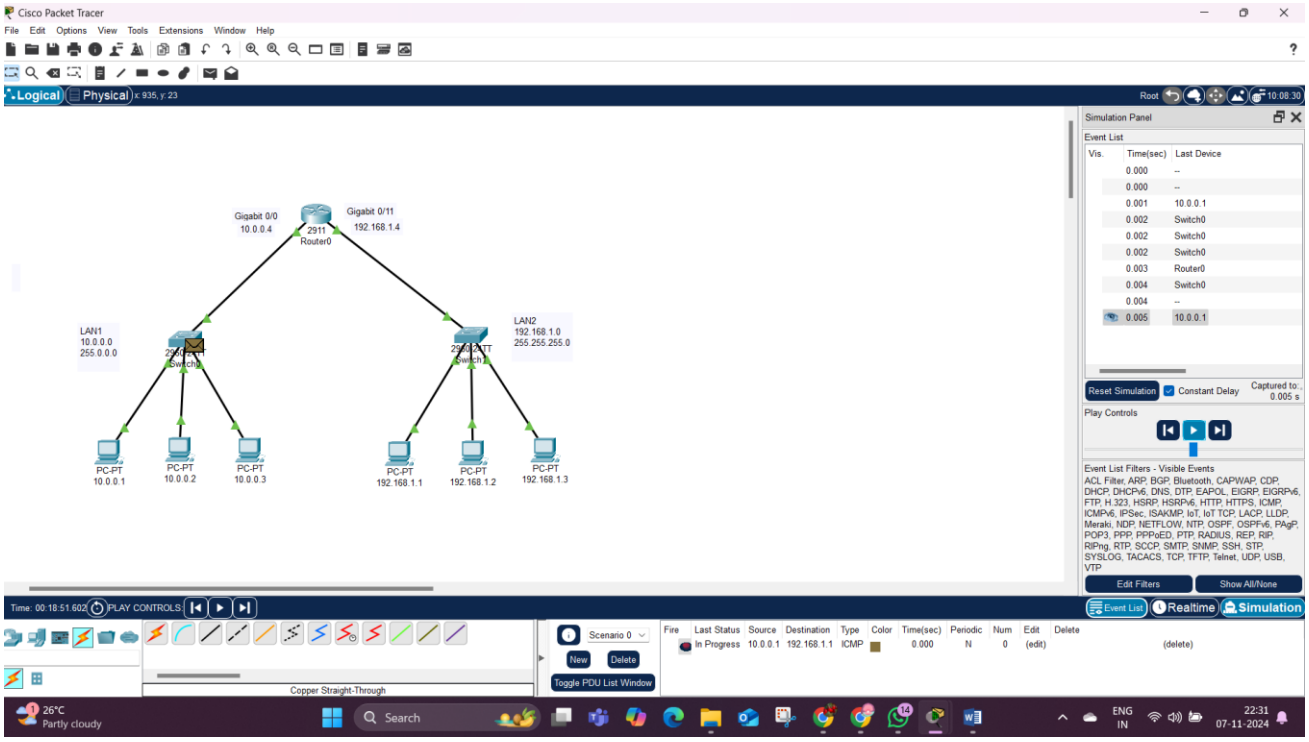


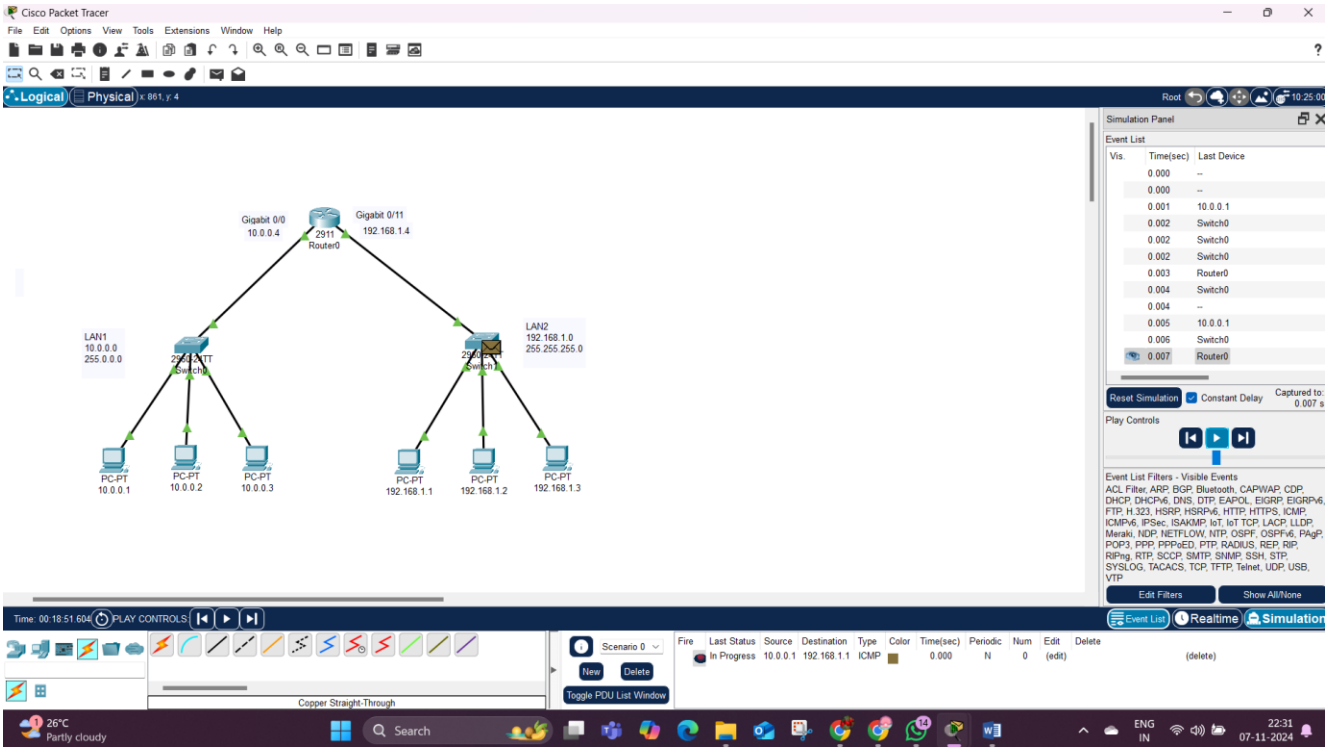



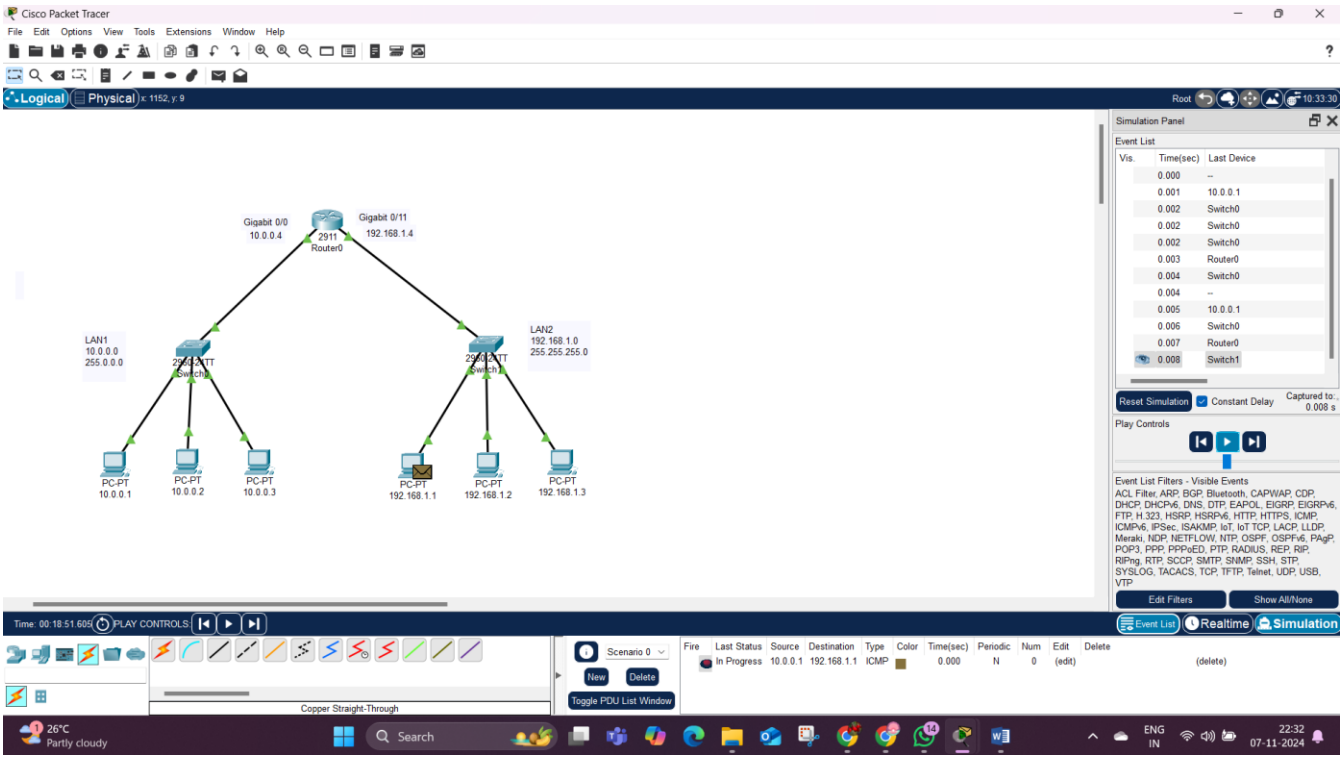



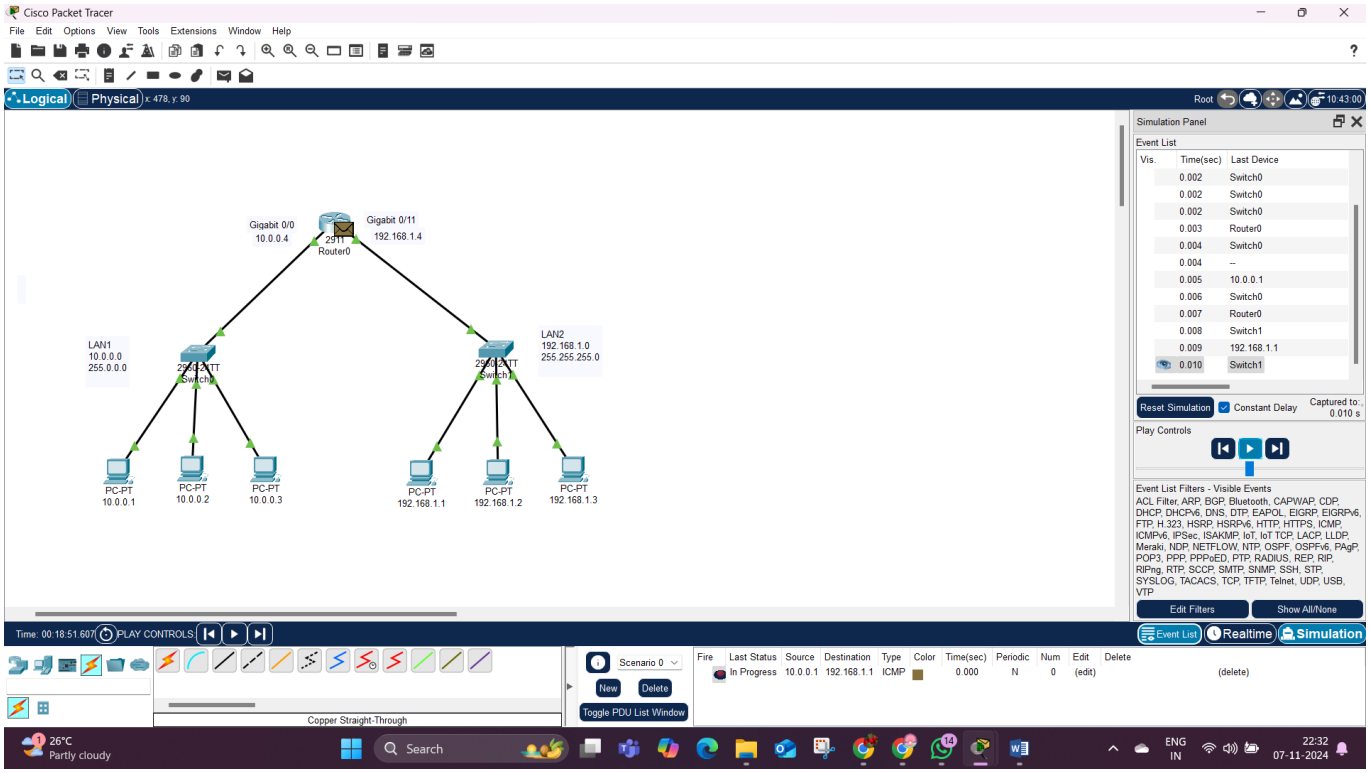

Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Root
Gigabit 0/0
10.0.0.4
2911
Router0
Gigabit 0/11
192.168.1.4
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.000 --
0.000 --
0.001 10.0.0.1
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
Reset Simulation
Constant Delay
Captured to:
0.003 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:18:51.600
PLAY CONTROLS:
Event List
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
Copper Straight-Through
26°C
Partly cloudy
Search
ENG
IN
22:30
07-11-2024

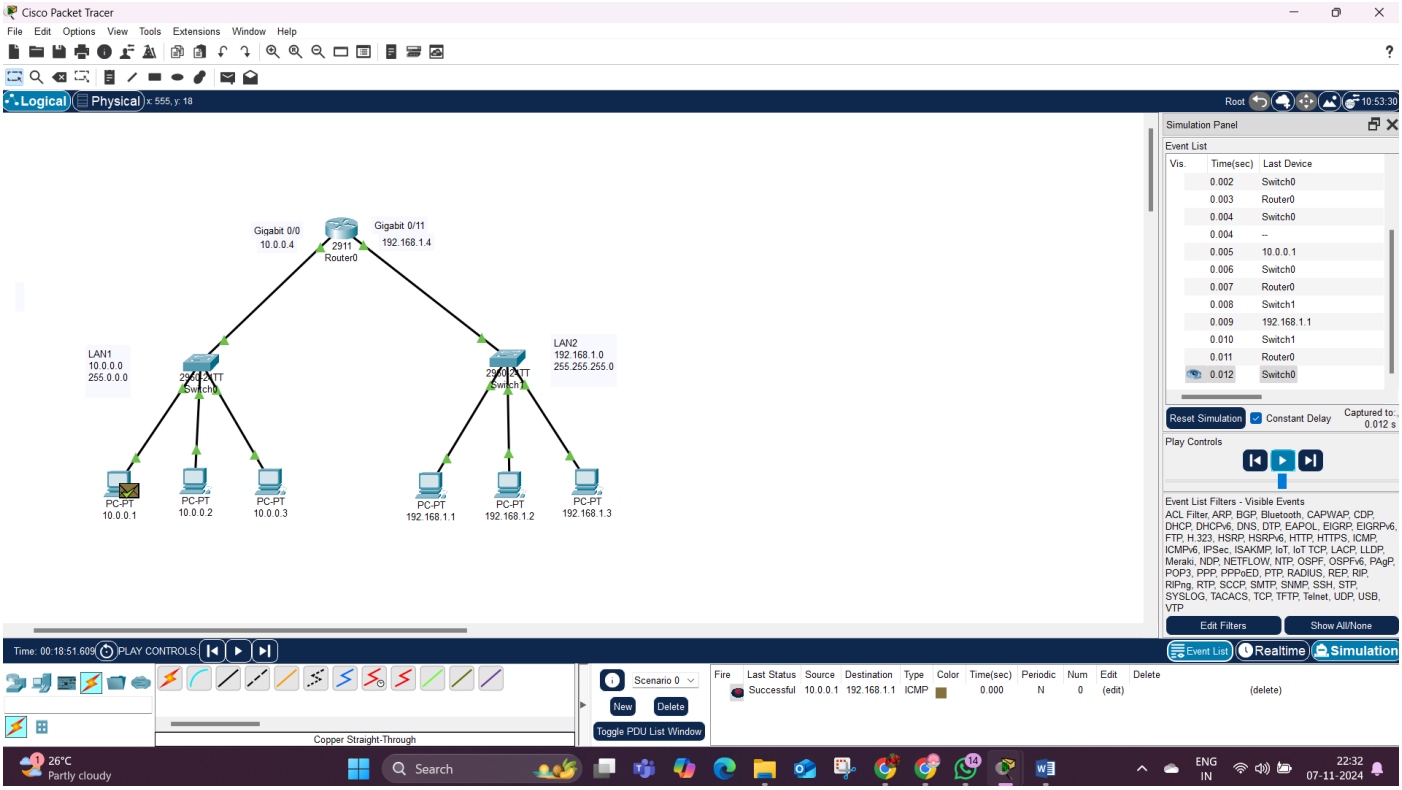

Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Root
Gigabit 0/0
10.0.0.4
2911
Router0
Gigabit 0/11
192.168.1.4
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.000 --
0.000 --
0.001 10.0.0.1
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
0.004 Switch0
0.004 --
Reset Simulation
Constant Delay
Captured to:
0.004 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:18:51.601
PLAY CONTROLS:
Event List
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
Copper Straight-Through
26°C
Partly cloudy
Search
ENG
IN
22:30
07-11-2024


Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Gigabit 0/0
10.0.0.4
2911
Router0
Gigabit 0/11
192.168.1.4
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.000 --
0.000 --
0.001 10.0.0.1
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
0.004 Switch0
0.004 --
0.005 10.0.0.1
Reset Simulation
Constant Delay
Play Controls
Event List Filters - Visible Events
Edit Filters
Show All/None
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Copper Straight-Through
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
26°C
Partly cloudy
Search
ENG
IN
22:31
07-11-2024


Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Gigabit 0/0
10.0.0.4
2911
Router0
Gigabit 0/11
192.168.1.4
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.000 --
0.000 --
0.001 10.0.0.1
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
0.004 Switch0
0.004 --
0.005 10.0.0.1
0.006 Switch0
0.007 Router0
Reset Simulation
Constant Delay
Play Controls
Event List Filters - Visible Events
Edit Filters
Show All/None
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Copper Straight-Through
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
26°C
Partly cloudy
Search
ENG
IN
22:31
07-11-2024


Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Gigabit 0/0
10.0.0.4
Gigabit 0/11
192.168.1.4
2911
Router0
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.000 --
0.001 10.0.0.1
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
0.004 Switch0
0.004 --
0.005 10.0.0.1
0.006 Switch0
0.007 Router0
0.008 Switch1
Reset Simulation
Constant Delay
Captured to: 0.008 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:18:51.605
PLAY CONTROLS
Event List
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
Copper Straight-Through
26°C
Partly cloudy
Search
ENG
IN
22:32
07-11-2024


Cisco Packet Tracer
File Edit Options View Tools Extensions Window Help
Logical Physical
Gigabit 0/0
10.0.0.4
Gigabit 0/11
192.168.1.4
2911
Router0
LAN1
10.0.0.0
255.0.0.0
LAN2
192.168.1.0
255.255.255.0
PC-PT
10.0.0.1
PC-PT
10.0.0.2
PC-PT
10.0.0.3
PC-PT
192.168.1.1
PC-PT
192.168.1.2
PC-PT
192.168.1.3
Simulation Panel
Event List
Vis. Time(sec) Last Device
0.002 Switch0
0.002 Switch0
0.002 Switch0
0.003 Router0
0.004 Switch0
0.004 --
0.005 10.0.0.1
0.006 Switch0
0.007 Router0
0.008 Switch1
0.009 192.168.1.1
0.010 Switch1
Reset Simulation
Constant Delay
Captured to: 0.010 s
Play Controls
Event List Filters - Visible Events
ACL Filter, ARP, BGP, Bluetooth, CAPWAP, CDP, DHCP, DHCPv6, DNS, DTP, EAPOL, EIGRP, EIGRPv6, FTP, H.323, HSRP, HSRPv6, HTTP, HTTPS, ICMP, ICMPv6, IPSec, ISAKMP, IoT, IoT TCP, LACP, LLDP, Meraki, NDP, NETFLOW, NTP, OSPF, OSPFv6, PAgP, POP3, PPP, PPPoED, PTP, RADIUS, REP, RIP, RIPng, RTP, SCCP, SMTP, SNMP, SSH, STP, SYSLOG, TACACS, TCP, TFTP, Telnet, UDP, USB, VTP
Edit Filters
Show All/None
Time: 00:18:51.607
PLAY CONTROLS
Event List
Realtime
Simulation
Scenario 0
New
Delete
Toggle PDU List Window
Fire Last Status Source Destination Type Color Time(sec) Periodic Num Edit Delete
In Progress 10.0.0.1 192.168.1.1 ICMP 0.000 N 0 (edit) (delete)
Copper Straight-Through
26°C
Partly cloudy
Search
ENG
IN
22:32
07-11-2024



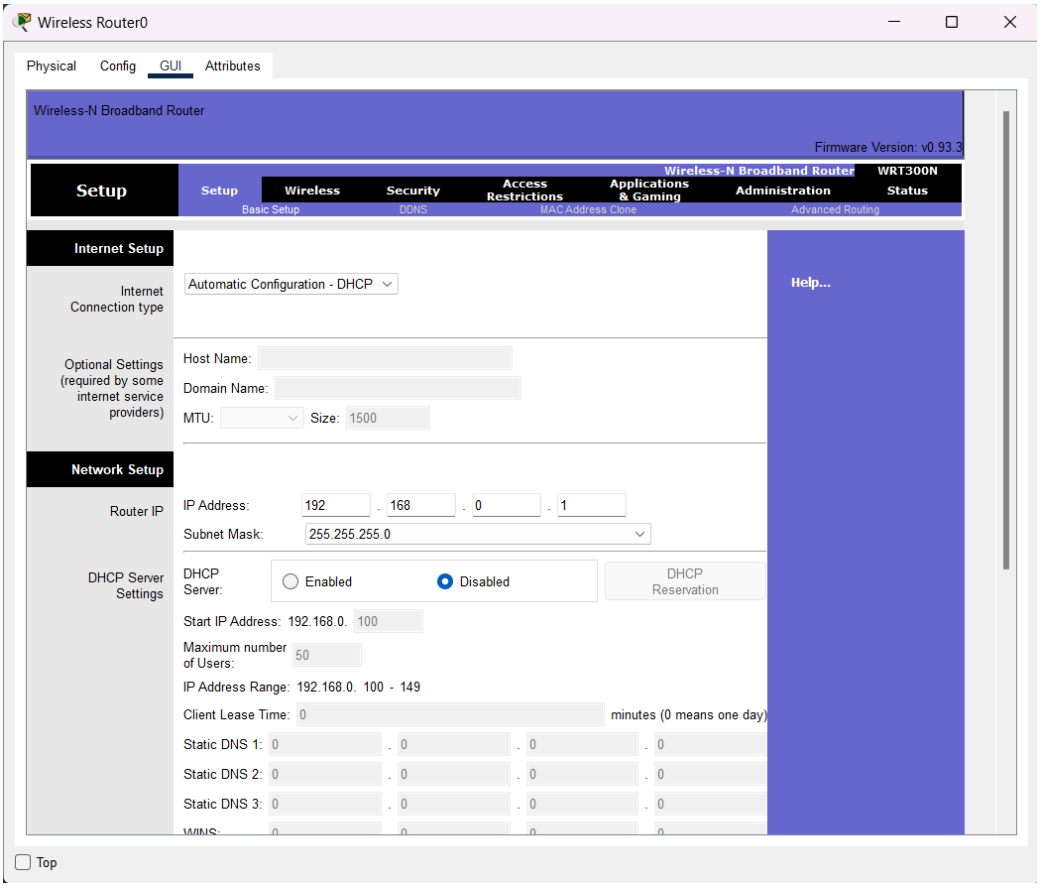

| Fire | Last Status | Source | Destination | Type | Color | Time(sec) | Periodic | Num | Edit | Delete |
|---|---|---|---|---|---|---|---|---|---|---|
| | Successful | 10.0.0.1 | 192.168.1.1 | ICMP | | 0.000 | N | 0 | (edit) | (delete) |

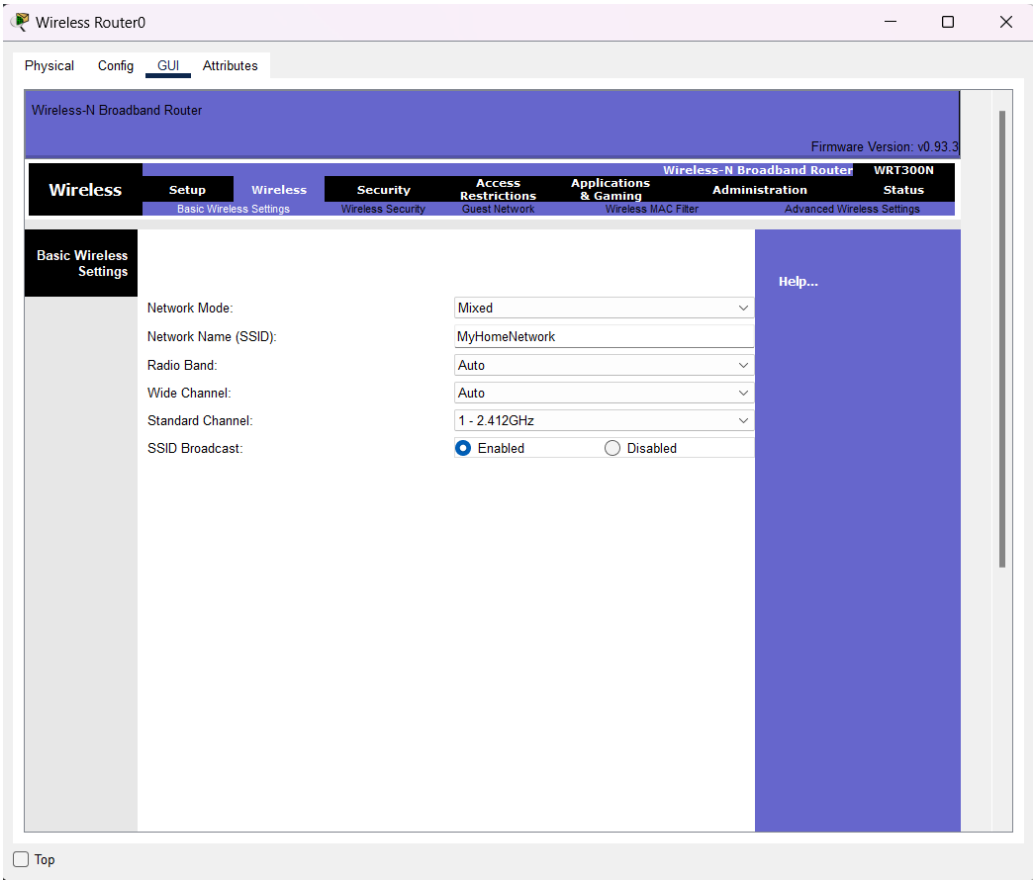
**RESULT: -**

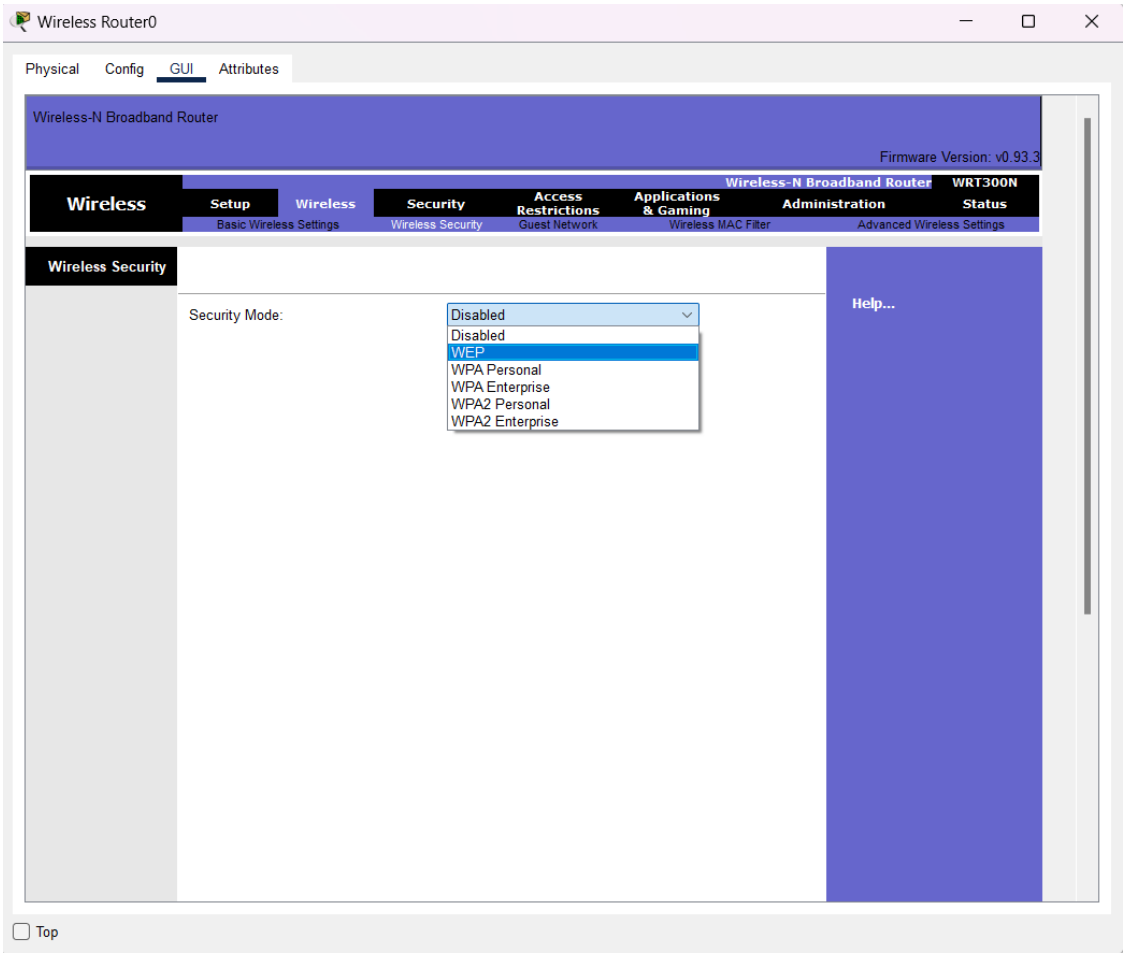
Router have been successfully done in CISCO PACKET TRACER.

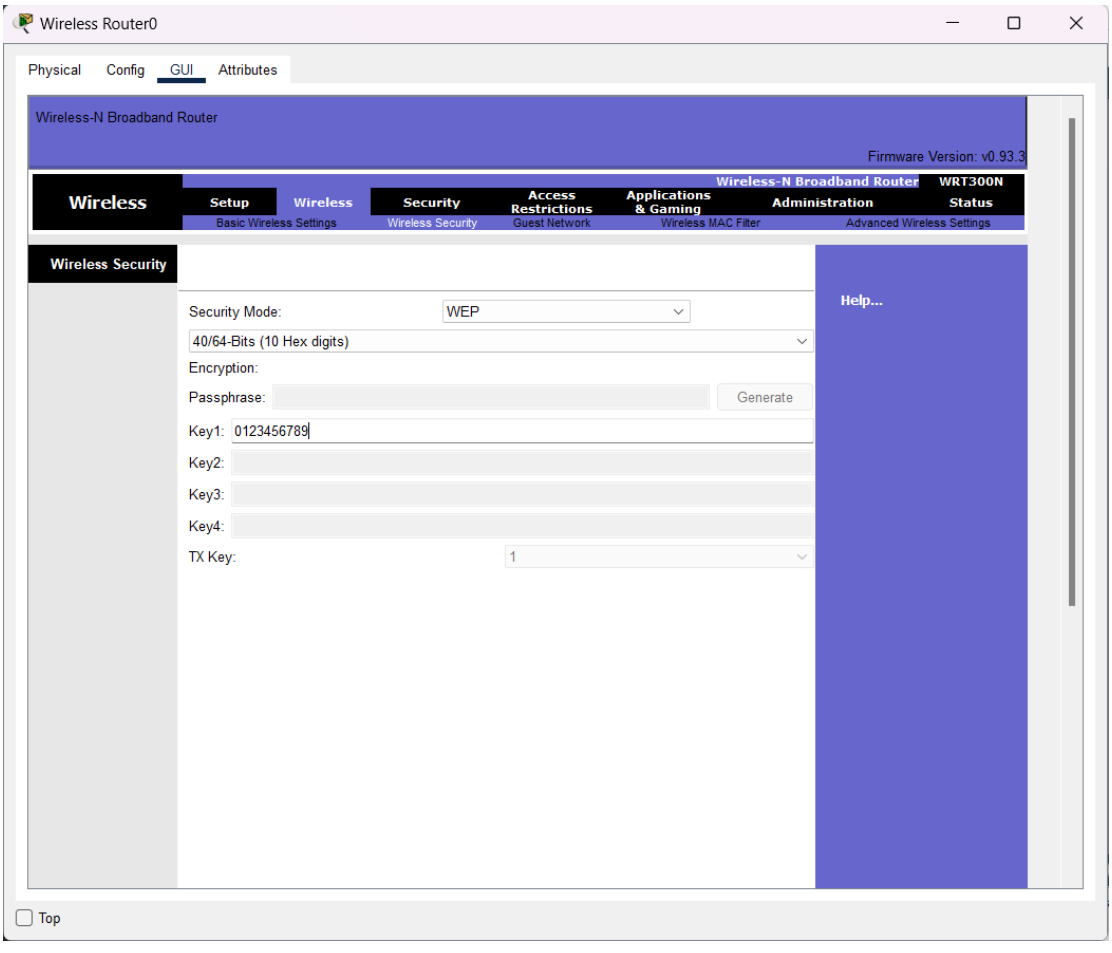
**AIM:** - b) Design and configure an internetwork using wireless router, DHCP server and internet cloud.

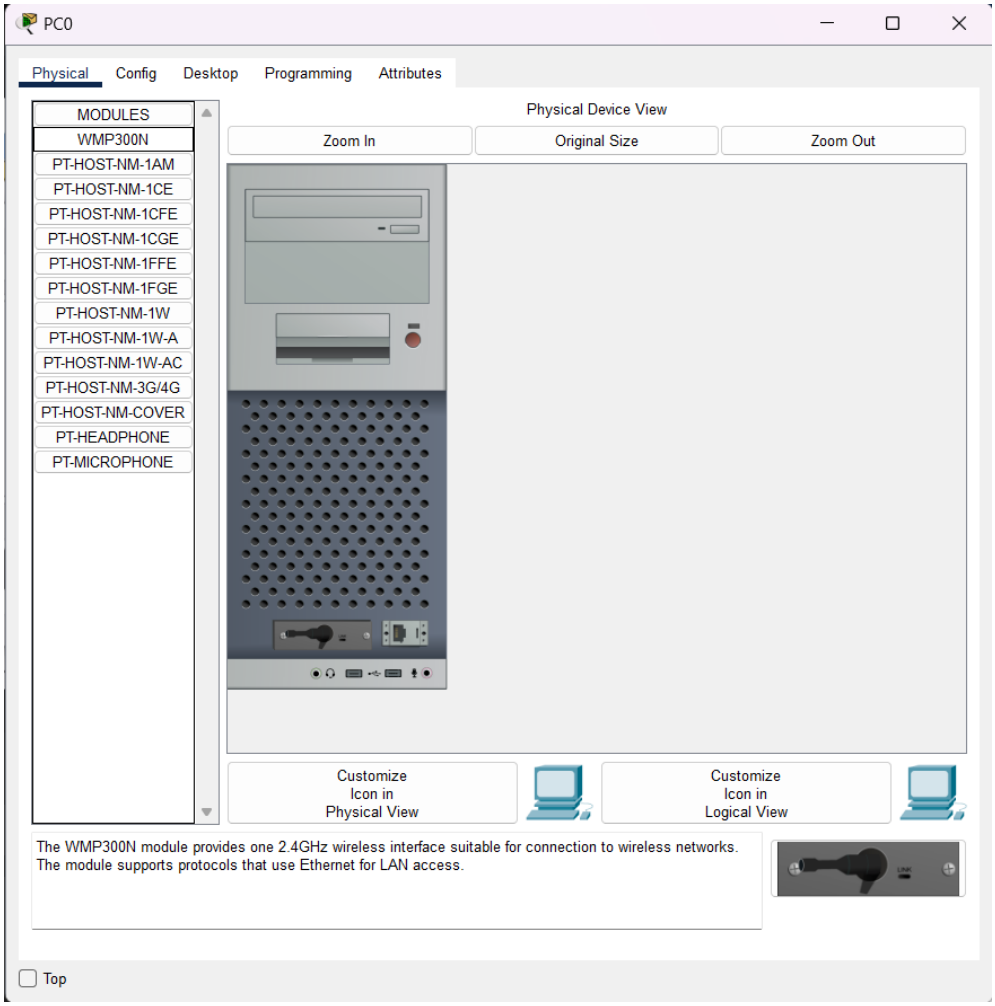
**OUTPUT:** -

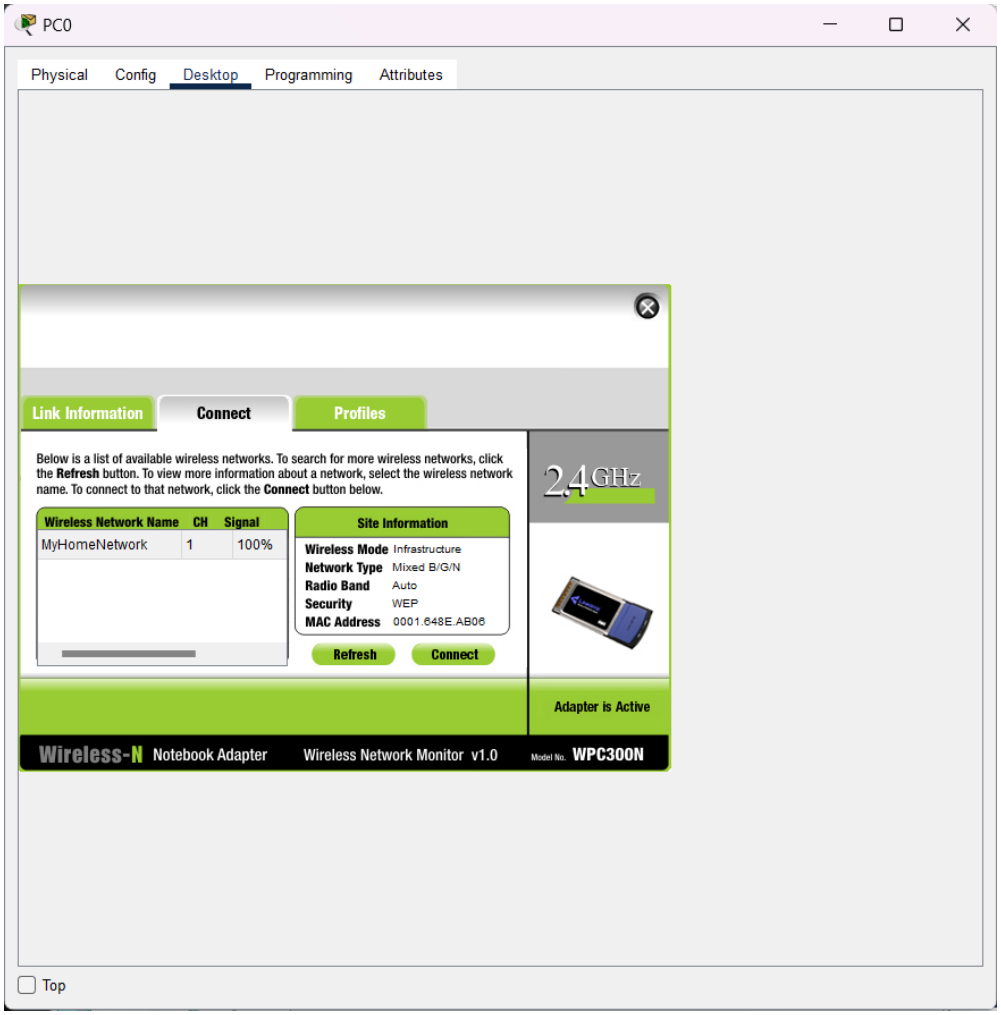


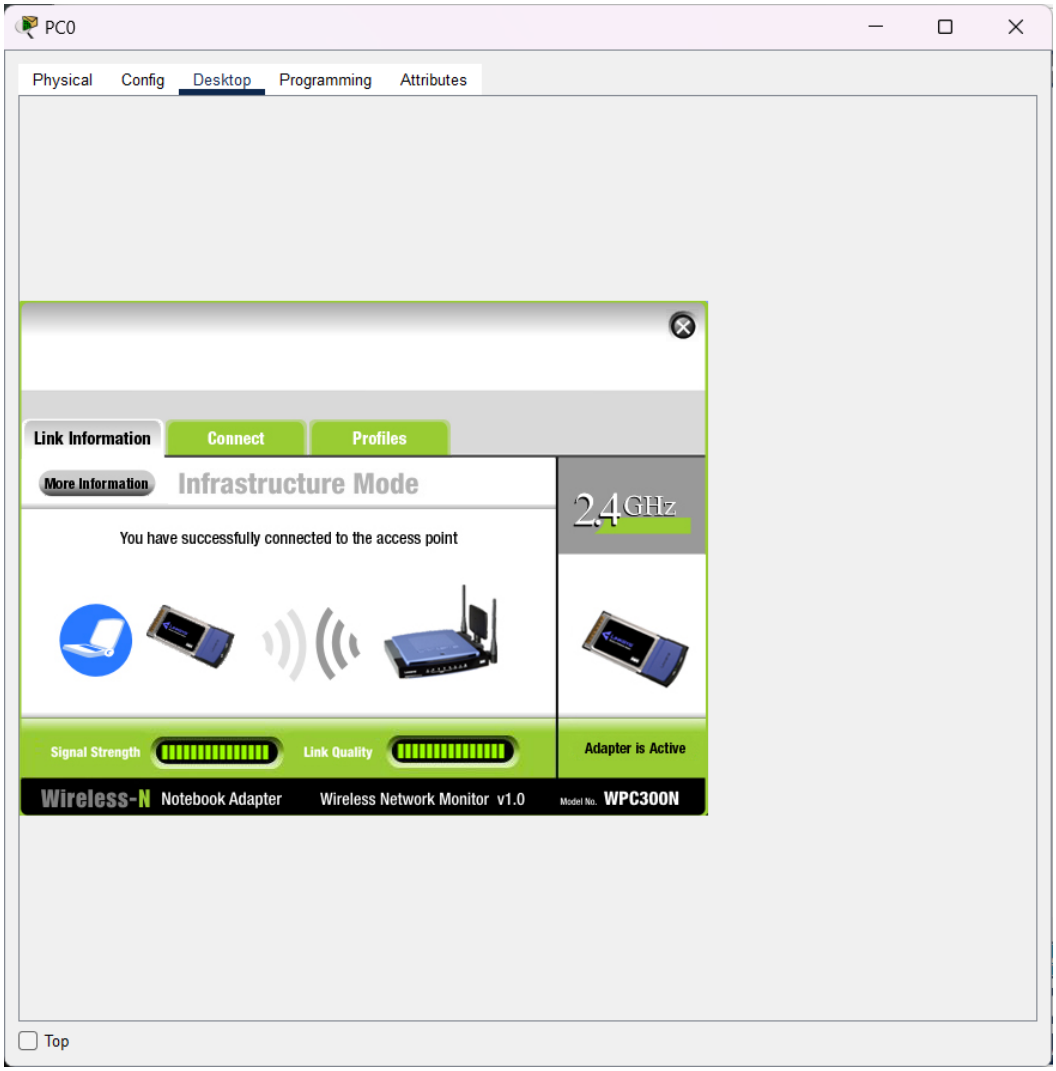



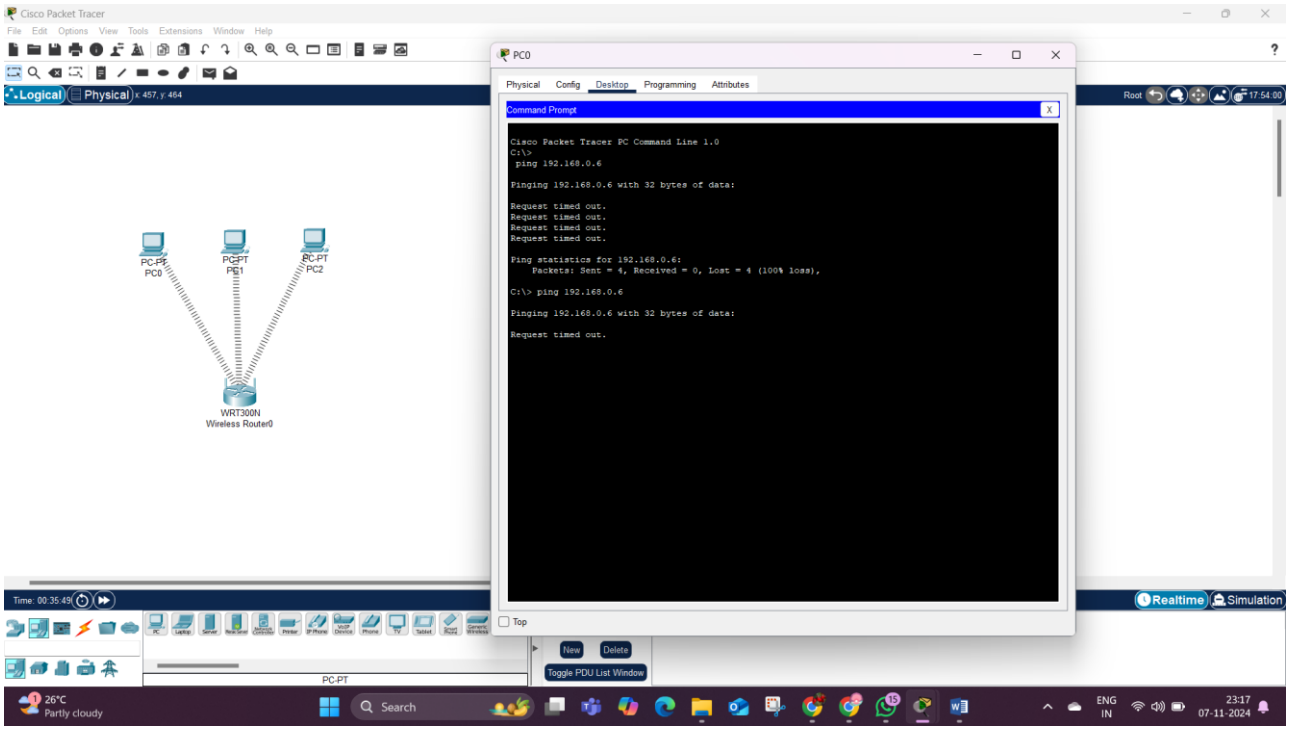

Wireless Router0
Physical
Config
GUI
Attributes
Wireless-N Broadband Router
Firmware Version: v0.93.3
Wireless
Setup
Wireless
Security
Access Restrictions
Applications & Gaming
Administration
Status
Wireless-N Broadband Router WRT300N
Basic Wireless Settings
Wireless Security
Guest Network
Wireless MAC Filter
Advanced Wireless Settings
Wireless Security
Security Mode:
Disabled
Disabled
WEP
WPA Personal
WPA Enterprise
WPA2 Personal
WPA2 Enterprise
Help...
Top


Wireless Router0
Physical
Config
GUI
Attributes
Wireless-N Broadband Router
Firmware Version: v0.93.3
Wireless
Setup
Wireless
Security
Access Restrictions
Applications & Gaming
Administration
Status
Wireless-N Broadband Router WRT300N
Basic Wireless Settings
Wireless Security
Guest Network
Wireless MAC Filter
Advanced Wireless Settings
Wireless Security
Security Mode:
WEP
40/64-Bits (10 Hex digits)
Encryption:
Passphrase:
Generate
Key1: 0123456789
Key2:
Key3:
Key4:
TX Key:
1
Help...
Top


PC0
Physical
Config
Desktop
Programming
Attributes
MODULES
WMP300N
PT-HOST-NM-1AM
PT-HOST-NM-1CE
PT-HOST-NM-1CFE
PT-HOST-NM-1CGE
PT-HOST-NM-1FFE
PT-HOST-NM-1FGE
PT-HOST-NM-1W
PT-HOST-NM-1W-A
PT-HOST-NM-1W-AC
PT-HOST-NM-3G/4G
PT-HOST-NM-COVER
PT-HEADPHONE
PT-MICROPHONE
Physical Device View
Zoom In
Original Size
Zoom Out
Customize Icon in Physical View
Customize Icon in Logical View
The WMP300N module provides one 2.4GHz wireless interface suitable for connection to wireless networks.
The module supports protocols that use Ethernet for LAN access.
Top


PC0
Physical
Config
Desktop
Programming
Attributes
Link Information
Connect
Profiles
Below is a list of available wireless networks. To search for more wireless networks, click the Refresh button. To view more information about a network, select the wireless network name. To connect to that network, click the Connect button below.
2.4GHz
Wireless Network Name
CH
Signal
MyHomeNetwork
1
100%
Site Information
Wireless Mode Infrastructure
Network Type Mixed B/G/N
Radio Band Auto
Security WEP
MAC Address 0001.648E.AB06
Refresh
Connect
Adapter is Active
Wireless-N Notebook Adapter
Wireless Network Monitor v1.0
Model No. WPC300N
Top




**RESULT: -**

Wireless Router have been successfully done in CISCO PACKET TRACER.

# Experiment-6

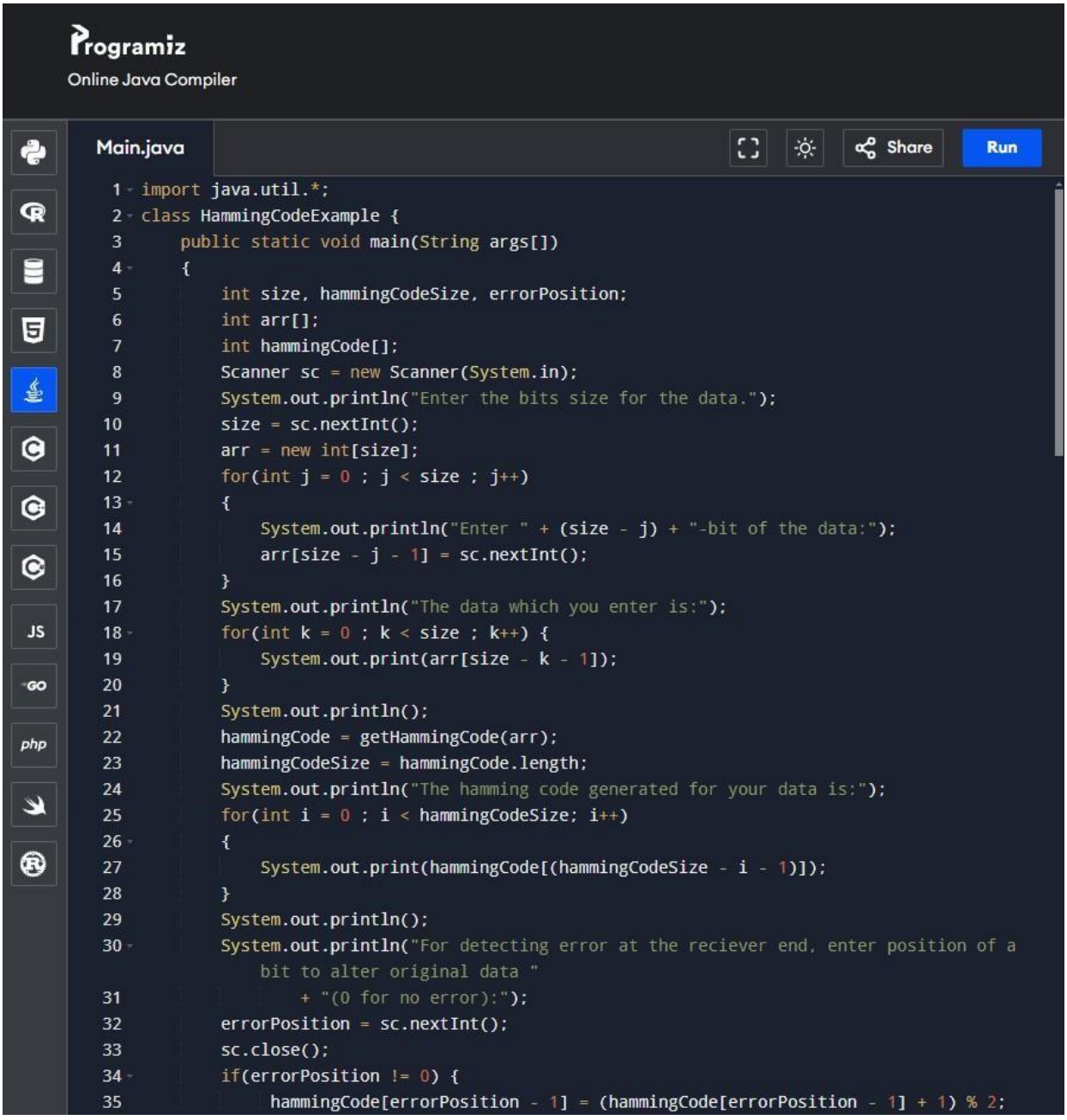
## Hamming Code

**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.

**Error Correction at Data Link Layer:**

Hamming code is a set of error-correction codes that can be used to detect and correct the errors that can occur when the data is transmitted from the sender to the receiver. It is a technique developed by R.W. Hamming for error correction.

**CODE:**




```java
import java.util.*;
class HammingCodeExample {
    public static void main(String args[])
    {
        int size, hammingCodeSize, errorPosition;
        int arr[];
        int hammingCode[];
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the bits size for the data.");
        size = sc.nextInt();
        arr = new int[size];
        for(int j = 0 ; j < size ; j++)
        {
            System.out.println("Enter " + (size - j) + "-bit of the data:");
            arr[size - j - 1] = sc.nextInt();
        }
        System.out.println("The data which you enter is:");
        for(int k = 0 ; k < size ; k++) {
            System.out.print(arr[size - k - 1]);
        }
        System.out.println();
        hammingCode = getHammingCode(arr);
        hammingCodeSize = hammingCode.length;
        System.out.println("The hamming code generated for your data is:");
        for(int i = 0 ; i < hammingCodeSize; i++)
        {
            System.out.print(hammingCode[(hammingCodeSize - i - 1)]);
        }
        System.out.println();
        System.out.println("For detecting error at the reciever end, enter position of a
            bit to alter original data "
                + "(0 for no error):");
        errorPosition = sc.nextInt();
        sc.close();
        if(errorPosition != 0) {
            hammingCode[errorPosition - 1] = (hammingCode[errorPosition - 1] + 1) % 2;
```

Main.java

Share Run

```
36          }
37          System.out.println("Sent Data is:");
38          for(int k = 0; k < hammingCodeSize; k++) {
39              System.out.print(hammingCode[hammingCodeSize - k - 1]);
40          }
41          System.out.println();
42          receiveData(hammingCode, hammingCodeSize - arr.length);
43      }
44      static int[] getHammingCode(int data[]) {
45          int returnData[];
46          int size;
47          int i = 0, parityBits = 0 ,j = 0, k = 0;
48          size = data.length;
49          while(i < size) {
50              if(Math.pow(2, parityBits) == (i + parityBits + 1)) {
51                  parityBits++;
52              }
53              else {
54                  i++;
55              }
56          }
57          returnData = new int[size + parityBits];
58          for(i = 1; i <= returnData.length; i++) {
59              if(Math.pow(2, j) == i) {
60
61                  returnData[(i - 1)] = 2;
62                  j++;
63              }
64              else {
65                  returnData[(k + j)] = data[k++];
66              }
67          }
68          for(i = 0; i < parityBits; i++) {
69
70              returnData[((int) Math.pow(2, i)) - 1] = getParityBit(returnData, i);
71          }
```

Main.java

Share Run

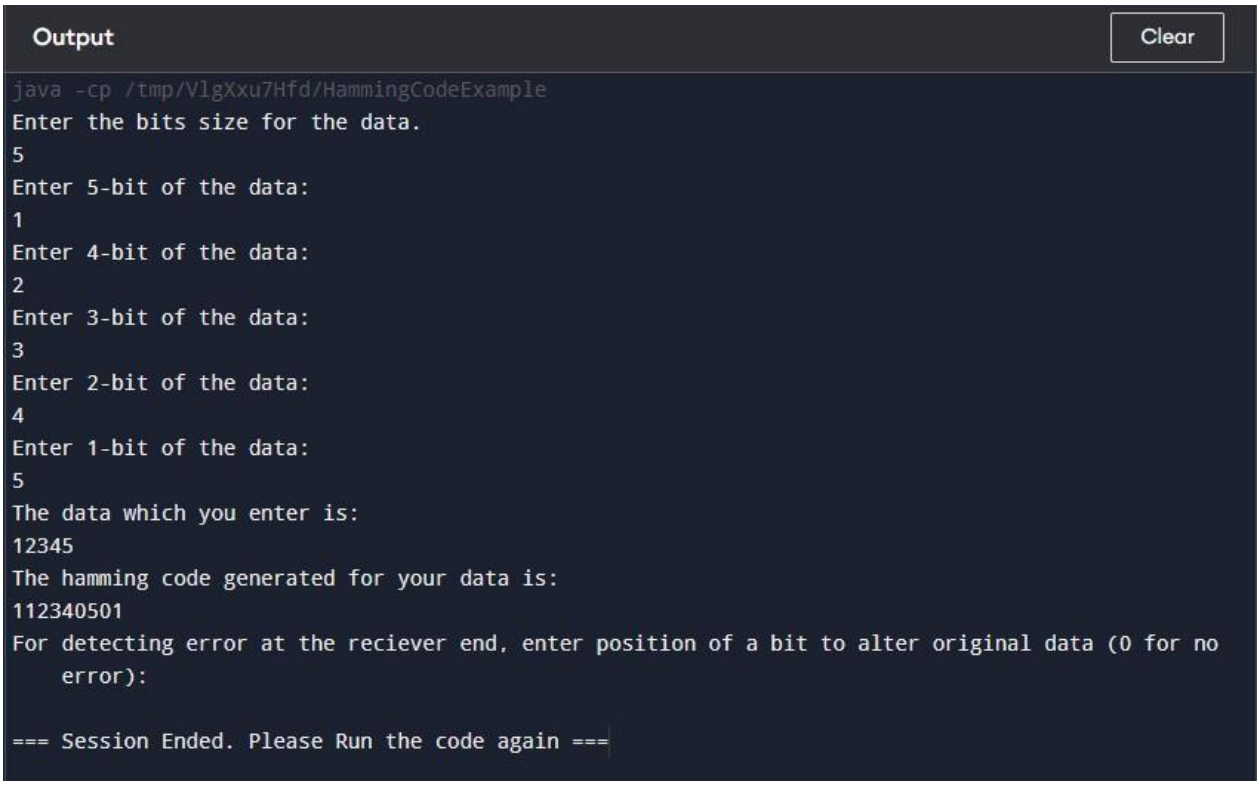
```
73          return returnData;
74      }
75      static int getParityBit(int returnData[], int pow) {
76          int parityBit = 0;
77          int size = returnData.length;
78
79          for(int i = 0; i < size; i++) {
80              if(returnData[i] != 2) {
81                  int k = (i + 1);
82                  String str = Integer.toBinaryString(k);
83                  int temp = ((Integer.parseInt(str)) / ((int) Math.pow(10, pow))) % 10;
84                  if(temp == 1) {
85                      if(returnData[i] == 1) {
86                          parityBit = (parityBit + 1) % 2;
87                      }
88                  }
89              }
90          }
91          return parityBit;
92      }
93
94      static void receiveData(int data[], int parityBits) {
95          int pow;
96          int size = data.length;
97          int parityArray[] = new int[parityBits];
98          String errorLoc = new String();
99          for(pow = 0; pow < parityBits; pow++) {
100             for(int i = 0; i < size; i++) {
101                 int j = i + 1;
102
103                 String str = Integer.toBinaryString(j);
104
105                 int bit = ((Integer.parseInt(str)) / ((int) Math.pow(10, pow))) % 10;
106                 if(bit == 1) {
107                     if(data[i] == 1) {
108                         parityArray[pow] = (parityArray[pow] + 1) % 2;
```

Main.java

Share Run

```
                    parityArray[pow] = (parityArray[pow] + 1) % 2;
                }
            }
        }
        errorLoc = parityArray[pow] + errorLoc;
    }
    int finalLoc = Integer.parseInt(errorLoc, 2);
    if(finalLoc != 0) {
        System.out.println("Error is found at location " + finalLoc + ".");
        data[finalLoc - 1] = (data[finalLoc - 1] + 1) % 2;
        System.out.println("After correcting the error, the code is:");
        for(int i = 0; i < size; i++) {
            System.out.print(data[size - i - 1]);
        }
        System.out.println();
    }
    else {
        System.out.println("There is no error in the received data.");
    }
    System.out.println("The data sent from the sender:");
    pow = parityBits - 1;
    for(int k = size; k > 0; k--) {
        if(Math.pow(2, pow) != k) {
            System.out.print(data[k - 1]);
        }
        else {
            pow--;
        }
    }
    System.out.println();
  }
}
```

**Output:**




## Result:

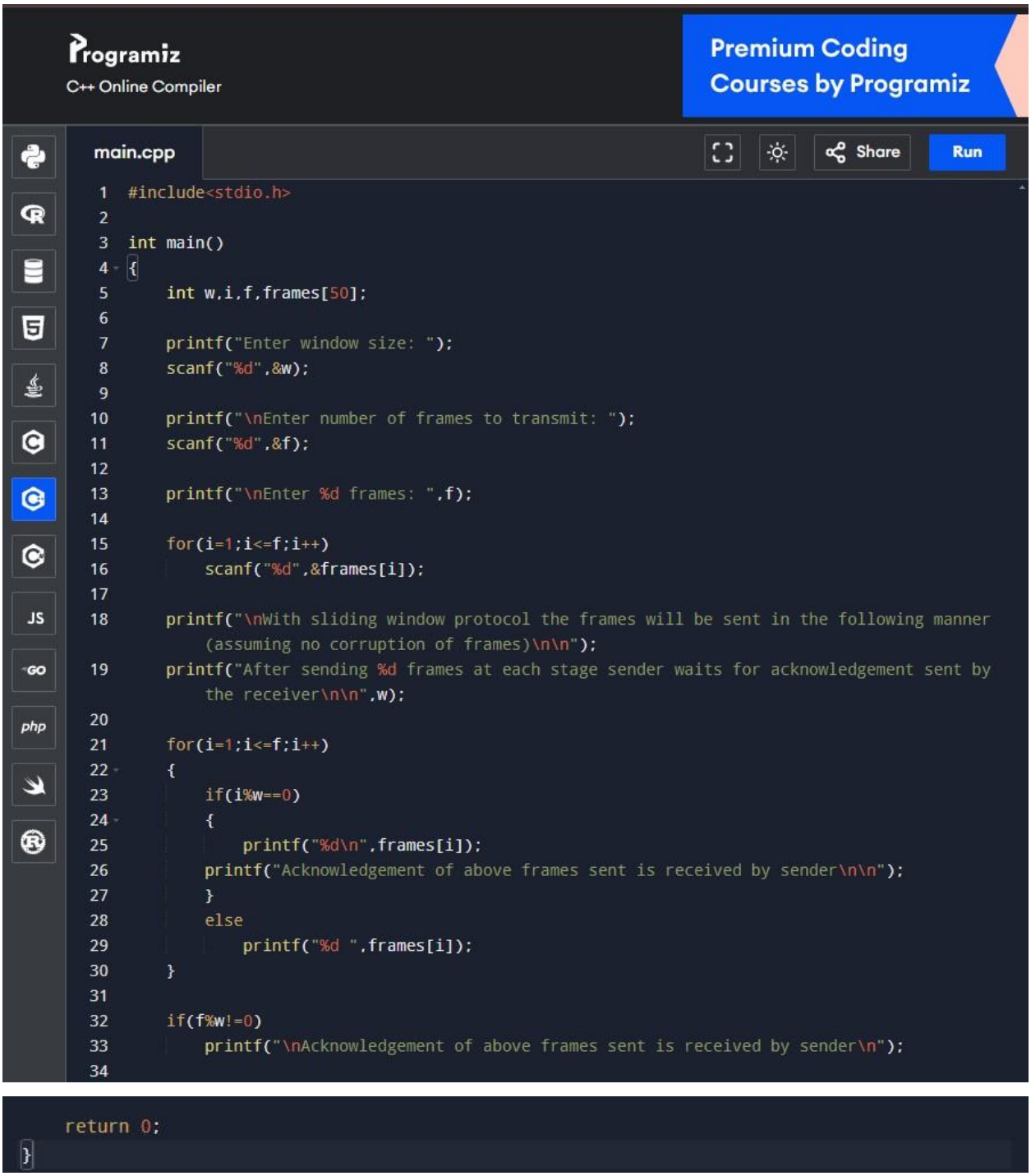
The program to implement error detection and correction using Hamming code concept is executed successfully.

# 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.

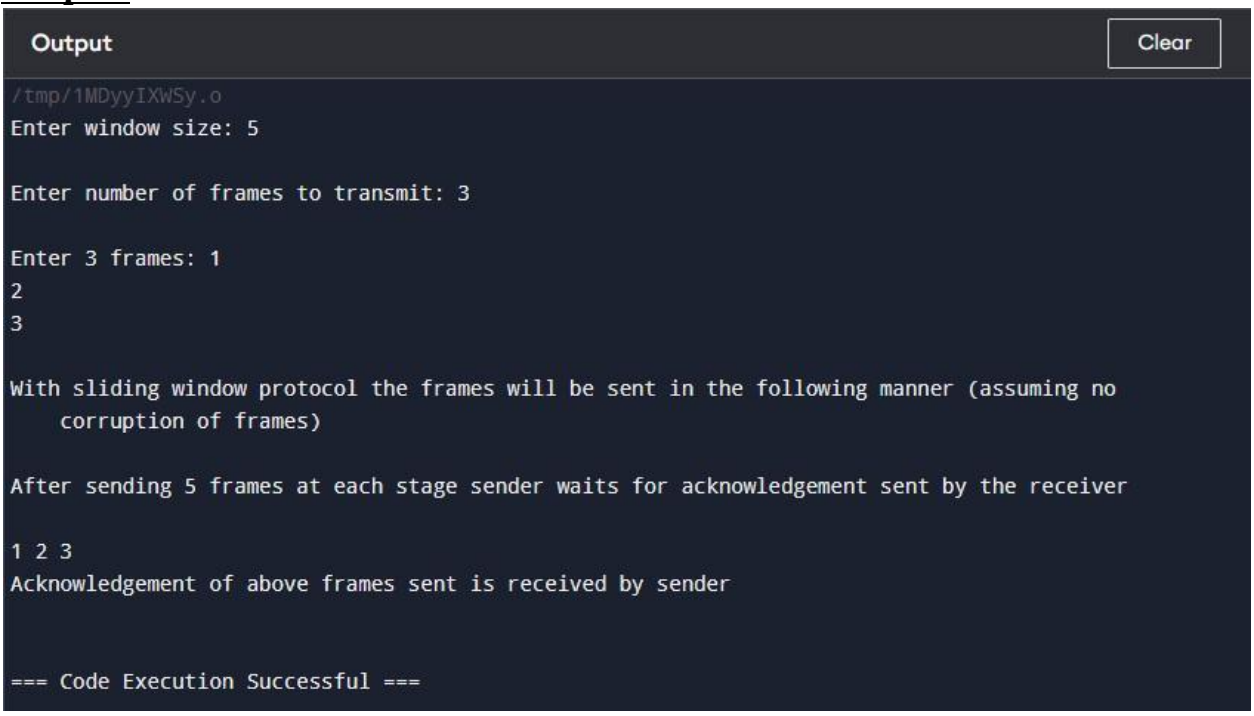
Program should achieve at least below given requirements. You can make it a bidirectional program wherein receiver is sending its data frames with acknowledgement (Piggybacking).

**Code:**




```
#include<stdio.h>

int main()
{
    int w,i,f,frames[50];

    printf("Enter window size: ");
    scanf("%d",&w);

    printf("\nEnter number of frames to transmit: ");
    scanf("%d",&f);

    printf("\nEnter %d frames: ",f);

    for(i=1;i<=f;i++)
        scanf("%d",&frames[i]);

    printf("\nWith 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 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]);
        printf("Acknowledgement of above frames sent is received by sender\n\n");
        }
        else
            printf("%d ",frames[i]);
    }

    if(f%w!=0)
        printf("\nAcknowledgement of above frames sent is received by sender\n");

    return 0;
}
```

**Output:**




**Result:**

Write a program to implement flow control at data link layer using Sliding window protocol has been executed successfully

# CS19541-COMPUTER NETWORK

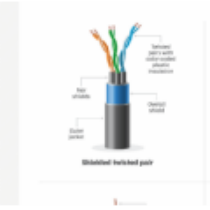
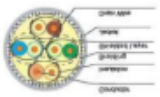
Experiment:2

Aim: Study of different types of Network cables.

a) Understand different types of network cable.

Different type of cables used in networking are:

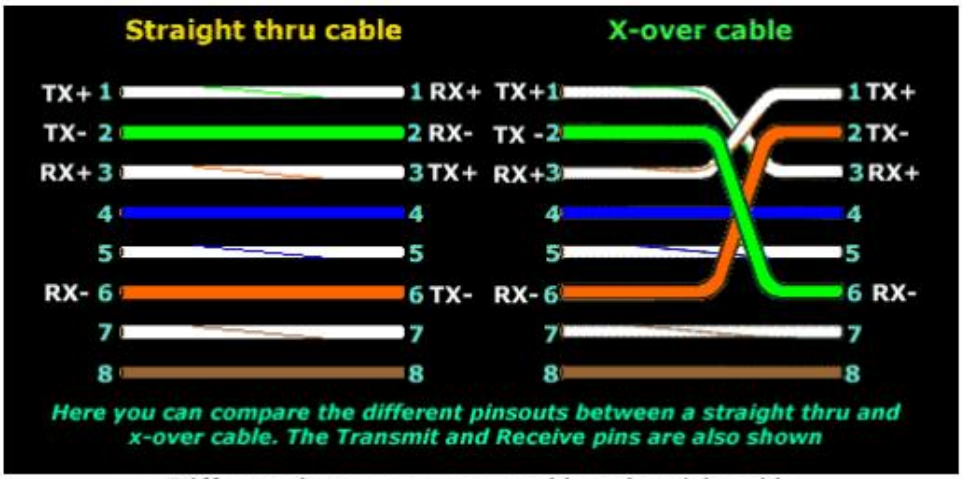
1. Unshielded Twisted Pair (UTP) Cable
2. Shielded Twisted Pair (STP) Cable
3. Coaxial Cable
4. Fibre Optic Cable

| Cable type | Category | Maximum Data Transmission | Advantages/ Disadvantages | Application/Use | Image |
|---|---|---|---|---|---|
| UTP | Category 3 | 10 bps | **Advantages**<br>• Cheaper in cost<br>• Easy to install as they have a smaller overall diameter.<br>**Disadvantages**<br>• 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 | Category6,6a | 10Gbps | **Advantages**<br>• Shielded.<br>• Faster than UTP.<br>• Less susceptible to noise and interference<br>**Disadvantages**<br>• Expensive<br>• Greater installation effort | Gigabit Ethernet, 10G Ethernet (55m)<br>Widely used in data centres |  |
| SSTP | Category 7 | 10Gbps | | Gigabit Ethernet, 10G Ethernet (100m) |  <br>  |

b) **Make Your Own Ethernet Cross-Over Cable/ Straight cable**

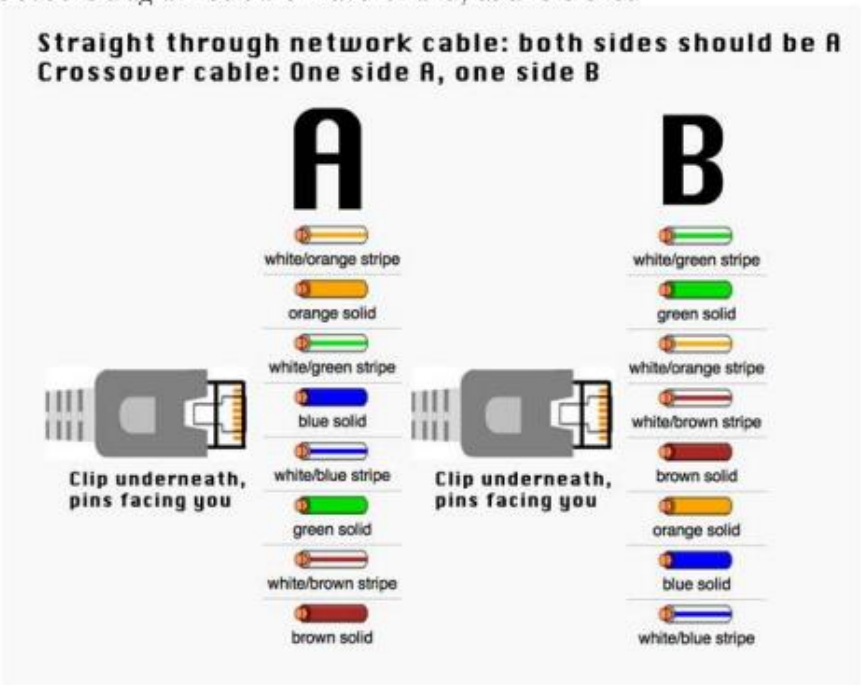
Tools and parts needed:

- Ethernet cabling. CAT5e is certified for gigabit support, but CAT5 cabling works as well, just over shorter distances.
- A crimping tool. This is an all-in-one networking tool shaped to push down the pins in the plug and strip and cut the shielding off the cables.
- Two RJ45 plugs.
- Optional two plug shields.




Difference between crossover cable and straight cable

Take a print out the diagram below or have it handy as a reference



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

Step 2: Next, strip approximately 1.5 cm of cable shielding from both ends. 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.

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)

**Result:**The study of  different types of cable has been successfully done.

# EXPERIMENT-8B

## AIM:-

b) Configuration of Wireless LAN using CISCO Packet Tracer.

Design a topology with three PCs connected from Linksys Wireless routers.


Perform following configuration:-

- 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

## RESULT:

The Configuration of Wireless LAN using CISCO Packet Tracer has been executed successfully.