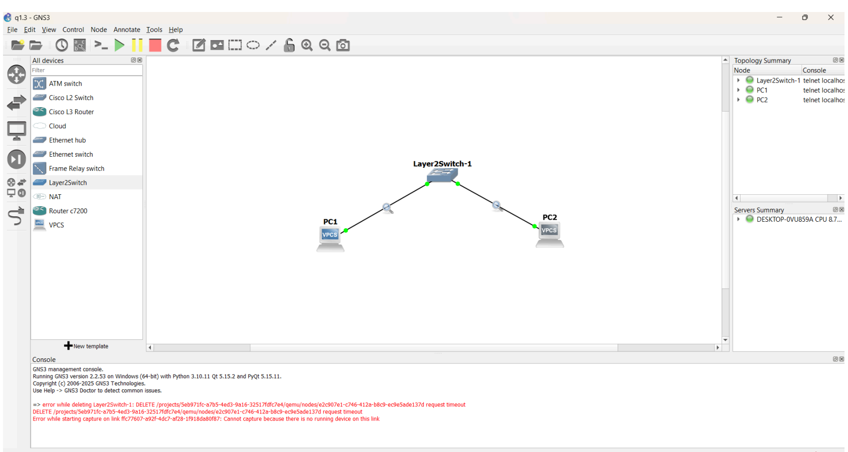
Networking Assessment 3 - Module 5

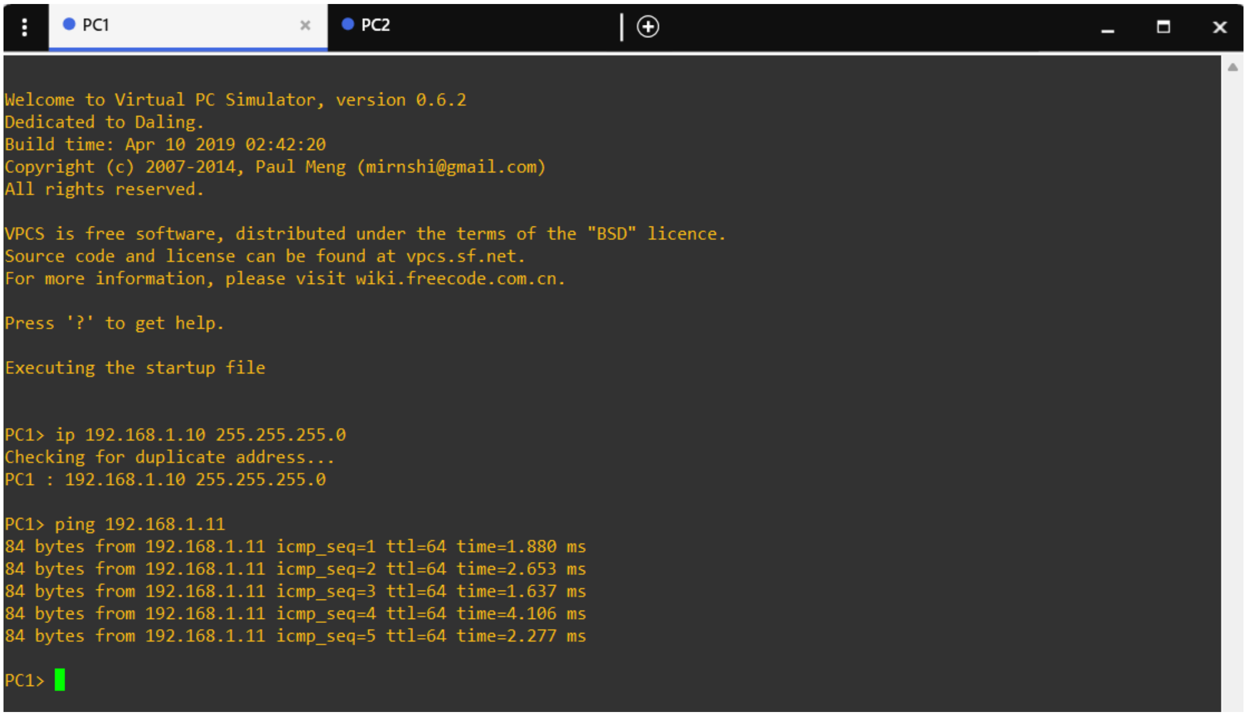
Name	T K Gowtham
Email ID	gowthamkamalasekar@gmail.com
College	VIT Chennai

1. Capture and analyze ARP packets using Wireshark. Inspect the ARP request and reply frames, and discuss the role of the sender's IP and MAC address in these packets.

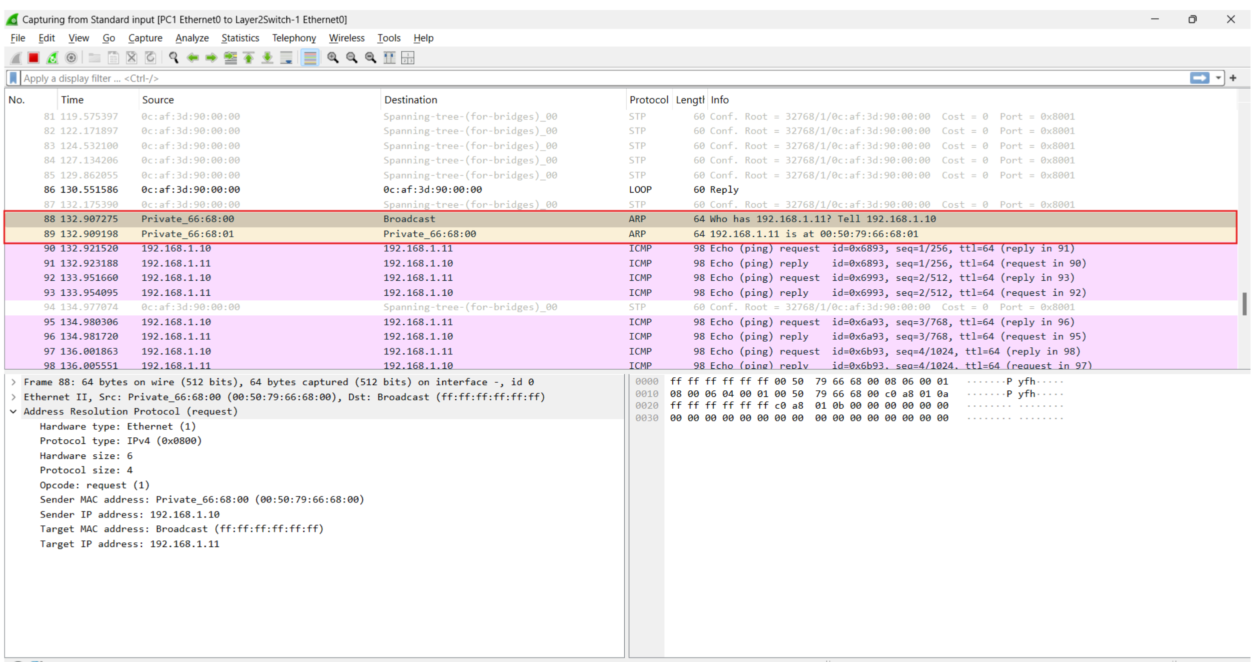
ARP is Address Resolution Protocol used for discovering the MAC address associated with given IP address in a local network. When a device wants to communicates with another device in a network, it uses ARP to identify the MAC address of the target device.

The device sends an ARP request broadcast address to all device on the same network. The device matching the IP address will give a ARP reply containing its MAC address, with which it can build a Ethernet frame.

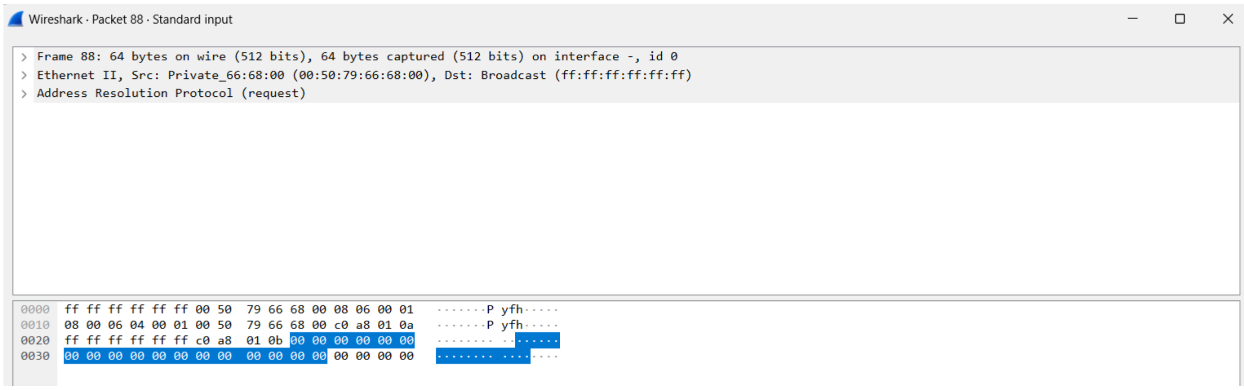




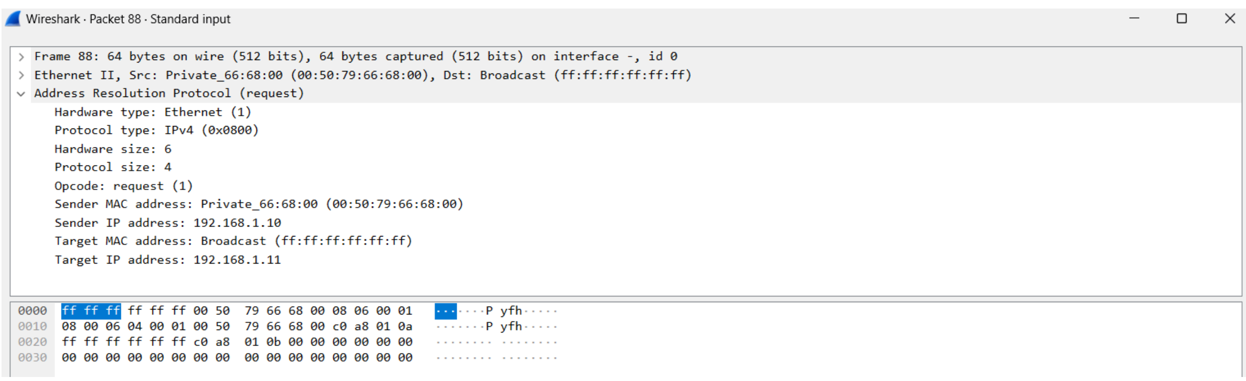
Captured an ARP request and reply packets in wireshark.



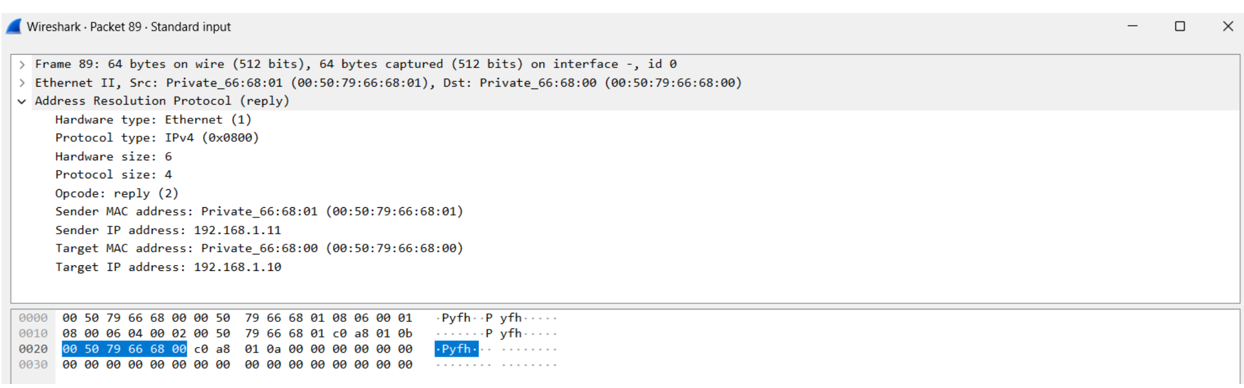
Let's examine the Request ARP packet, The frame is of 43 bytes and ethernet frame consist of the source and destination IP address.



In the ARP request type, it consist of the Hardware type which is ethernet, the protocol type which IPv4, hardware and protocol size, Opcode which states whether request or reply. We can the sender IP address, MAC address and target IP address is as specified but the target MAC address is a broadcast address as it is sent to every device in the network.

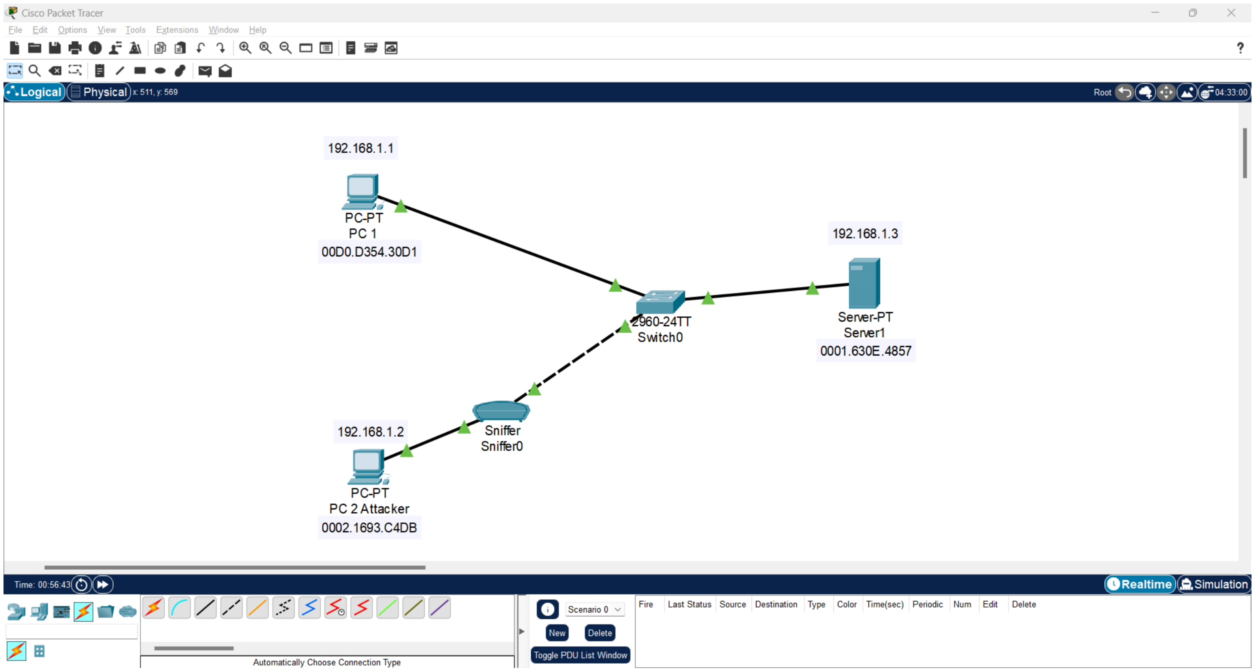


Only the intended device with the target IP address will reply, with its own IP and MAC as the sender address, and the the previously sender IP and MAC address as the current target IP and MAC Address.

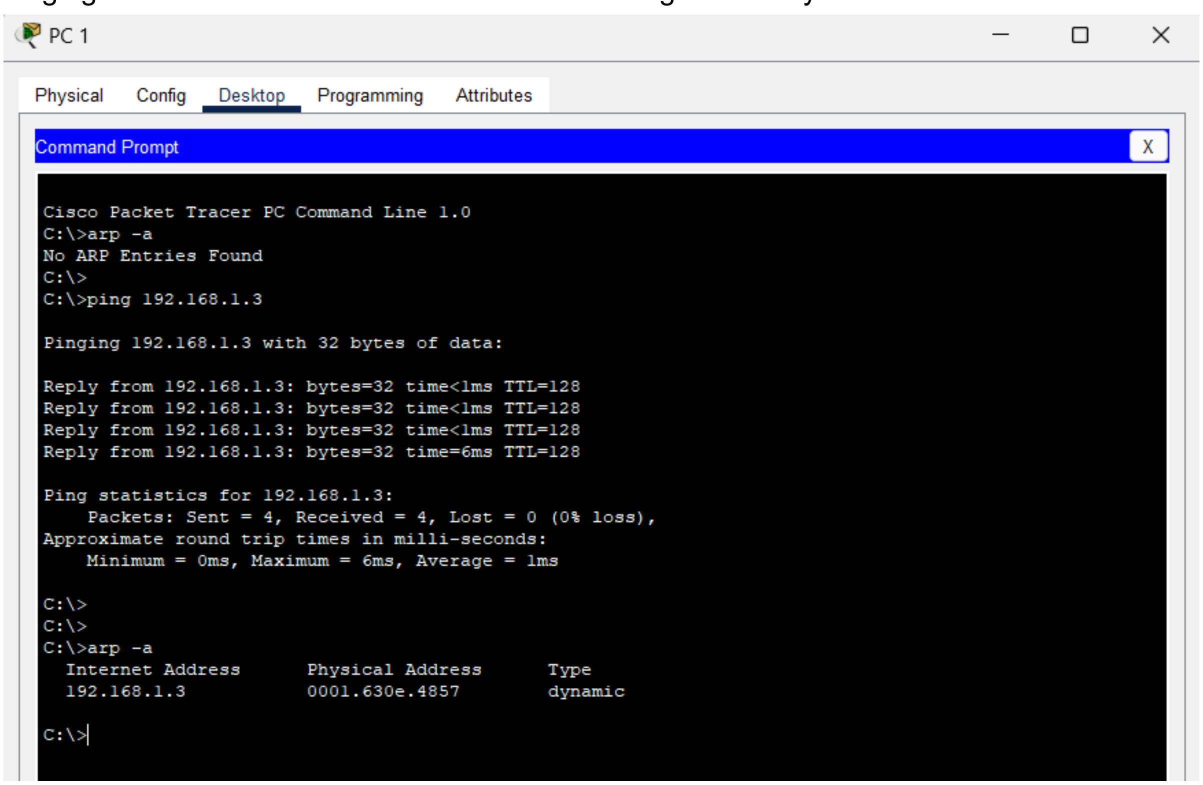


2. Using Packet Tracer, simulate an ARP spoofing attack. Analyze the behavior of devices on the network when they receive a malicious ARP response.

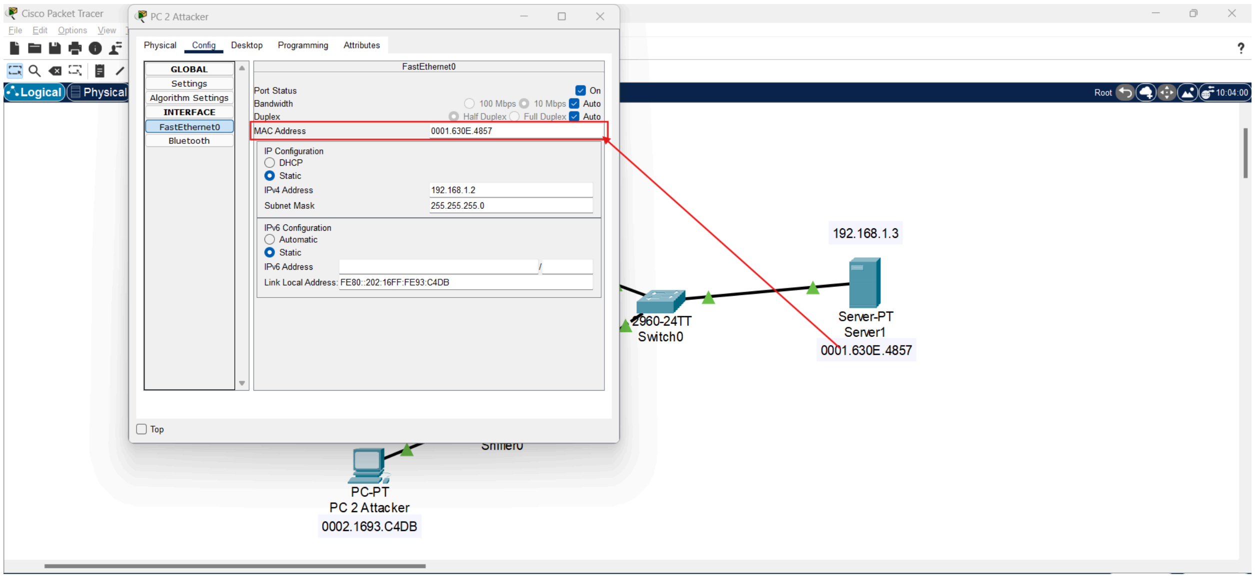
Creating a network in Cisco Packet Tracer with 2 PCs and a Web Server. The IP address and MAC address are noted by each device :



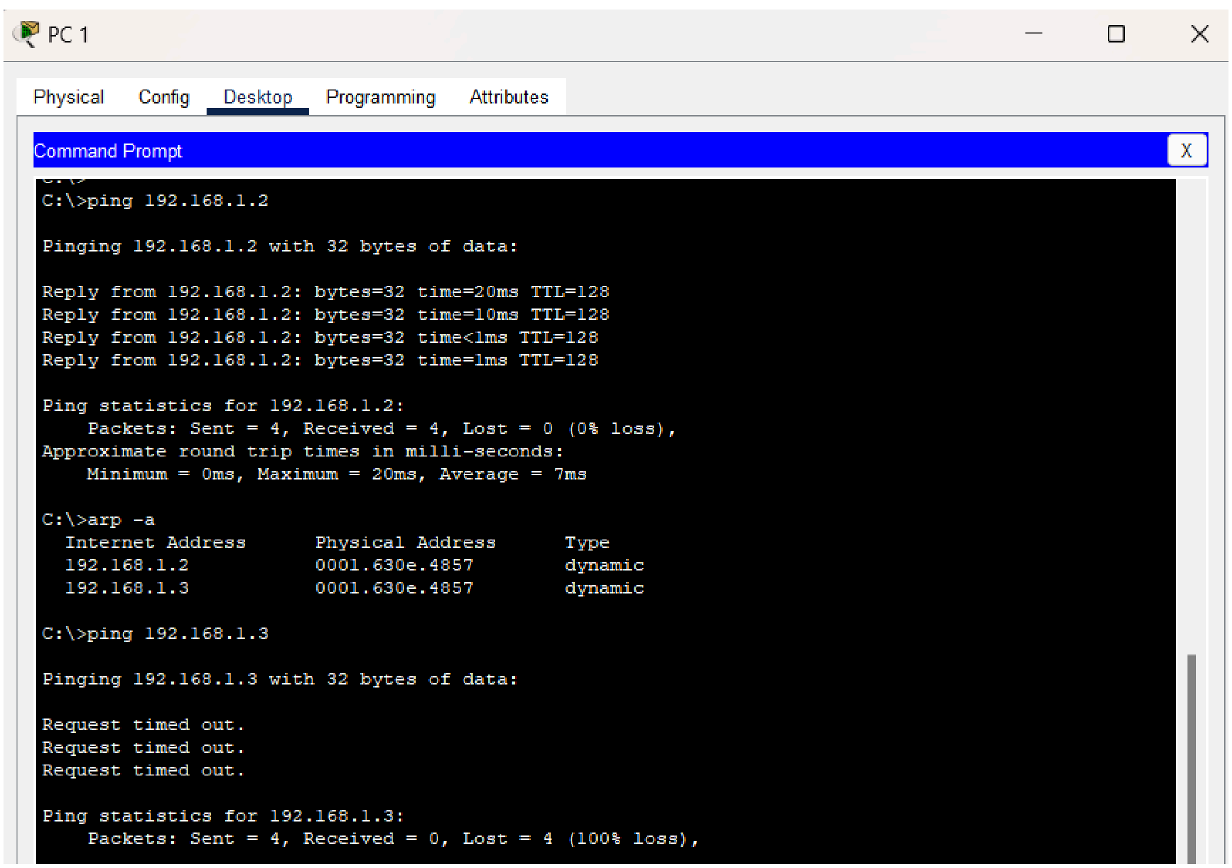
Pinging from PC1 to the Server to create an existing ARP entry :



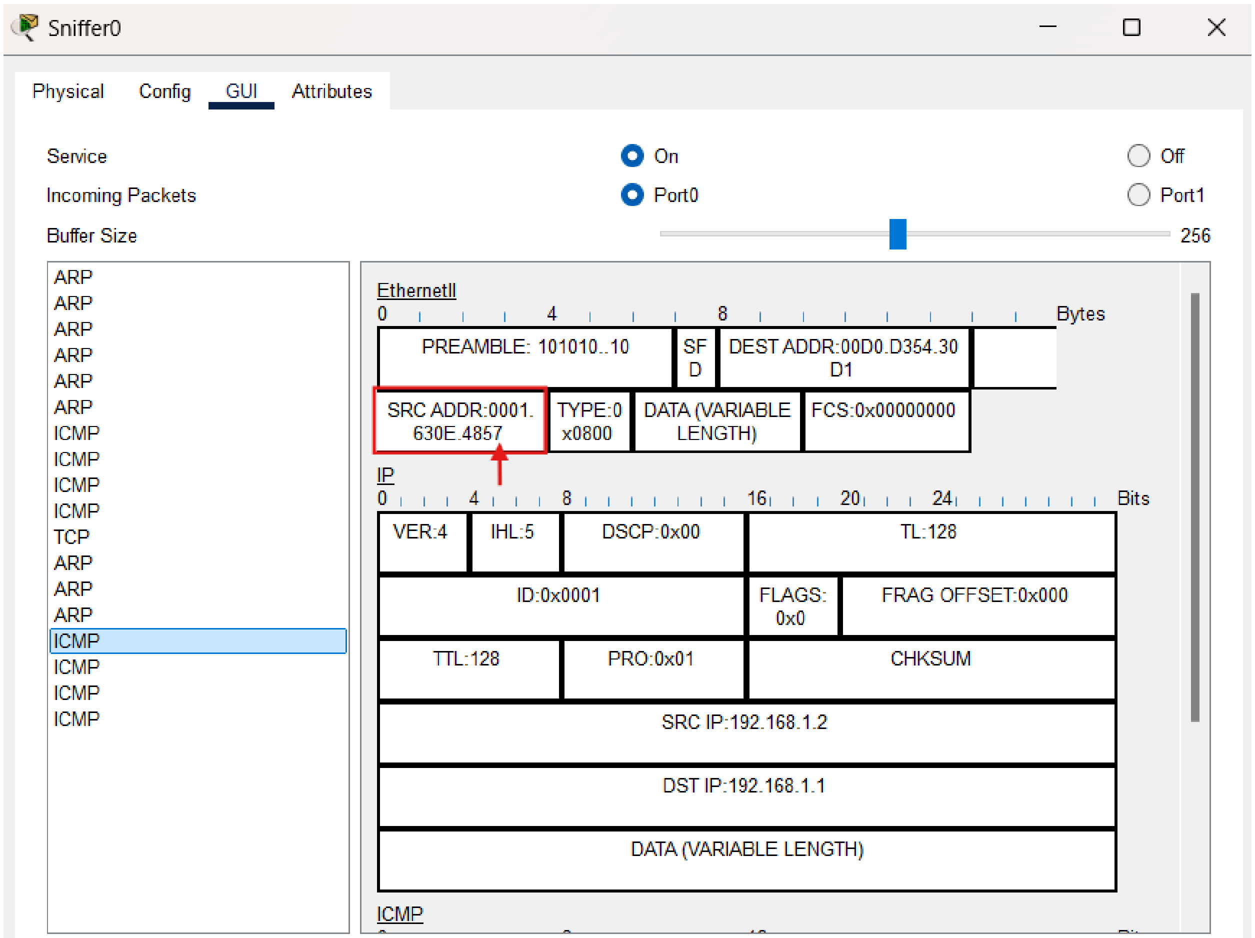
Now, let's change the MAC address of the PC 2 to the MAC address of the server to spoof the MAC address.



Now again pinging from PC1 to the Server it is being timed out, so lets ping to PC2 and while checking the ARP table we see that both PC2 and Server has the same MAC address, which states the MAC Address Spoofing.



Now let's check the sniffer also, to confirm the MAC spoofing worked or not, and while examining the ICMP packet received we can the see the target MAC address of PC2 is the MAC address of the Server.



So, we can also tell the behaviour of the actual device that the request is being dropped by the switch and request is being timed out at the PC1.

3. Manually configure static IPs on the client devices(like Pc or your mobile phone) and verify connectivity using ping.

Current IP address of the machine :

```
mint@mint: ~
File  Edit  View  Search  Terminal  Help
mint@mint:~$ ifconfig
enp0s3: flags=4099<UP,BROADCAST,MULTICAST>  mtu 1500
        inet 192.168.56.106  netmask 255.255.255.0  broadcast 192.168.56.255
        inet6 fe80::f19:a46f:74ef:b645  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:7e:42:67  txqueuelen 1000  (Ethernet)
        RX packets 32  bytes 8446 (8.4 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 95  bytes 12802 (12.8 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 138  bytes 11972 (11.9 KB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 138  bytes 11972 (11.9 KB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

To view the static IP address : (It is 192.168.1.11/24 in enp0s3)

```
mint@mint:/etc/netplan$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default
 qlen 1000
    link/ether 08:00:27:7e:42:67 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.11/24 brd 192.168.1.255 scope global dynamic noprefixroute enp0s3
       valid_lft 84477sec preferred_lft 84477sec
    inet6 2401:4900:1cc8:89e4:4755:e4a1:a201:7ca0/64 scope global temporary dynamic
       valid_lft 86184sec preferred_lft 84049sec
    inet6 2401:4900:1cc8:89e4:f2b3:2574:7801:12da/64 scope global dynamic mngtmpaddr nopref
ixroute
       valid_lft 86184sec preferred_lft 86184sec
    inet6 fe80::f19:a46f:74ef:b645/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
mint@mint:/etc/netplan$
```

Move to the /etc/netplan folder where we have to make modification to the yaml file to set the static IP address :

```
mint@mint:~$ cd /etc/netplan
mint@mint:/etc/netplan$ ls
1-network-manager-all.yaml
```

Open the file with sudo and add the ethernets to the file and save it :

```
mint@mint: /etc/netplan
File  Edit  View  Search  Terminal  Help
  GNU nano 6.2                     1-network-manager-all.yaml *
# Let NetworkManager manage all devices on this system
network:
  version: 2
  renderer: NetworkManager
  ethernets:
    enp0s3:
      dhcp4: no
      addresses: [192.168.1.100/24]
      routes:
        - to: default
          via: 192.168.1.1
      nameservers:
        addresses: [192.168.1.1]
```

To see if the changes don't have any errors and to apply it use the netplan try command followed by enter or just do netplan apply :

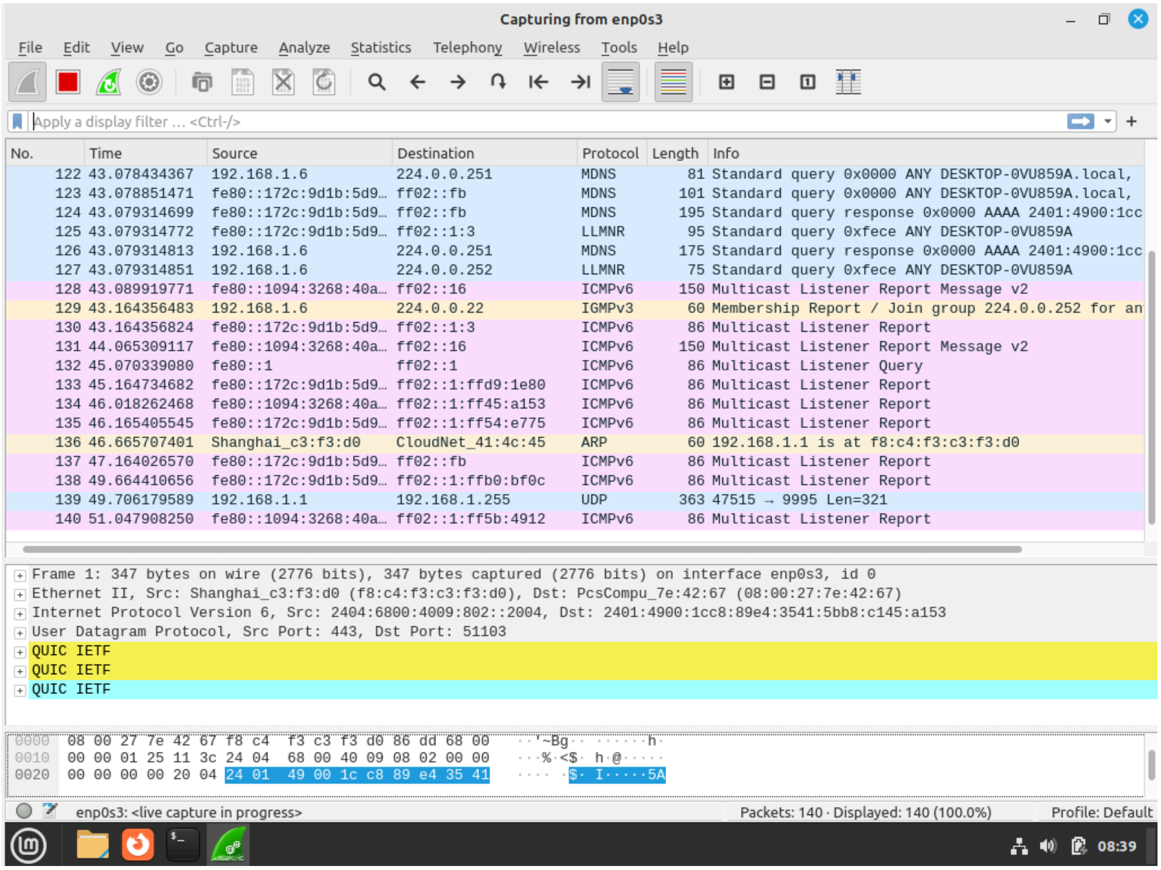
```
mint@mint:/etc/netplan$ sudo netplan try
Do you want to keep these settings?


Press ENTER before the timeout to accept the new configuration


Changes will revert in  93 seconds
Configuration accepted.
mint@mint:/etc/netplan$
```

Now the changes have been applied you can check it using ip a, and we see that the static IP has been updated with 192.168.1.100/24 :

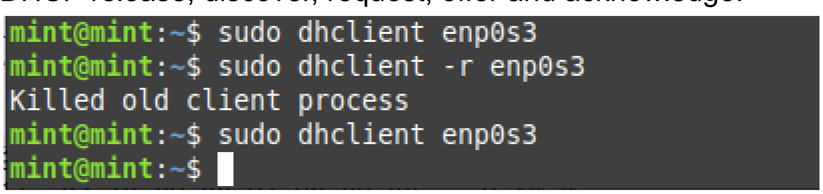
```
mint@mint:/etc/netplan$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:7e:42:67 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.100/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::a00:27ff:fe7e:4267/64 scope link
       valid_lft forever preferred_lft forever
mint@mint:/etc/netplan$
```

4. Use Wireshark to capture DHCP Discover, Offer, Request, and Acknowledge messages and explain the process.

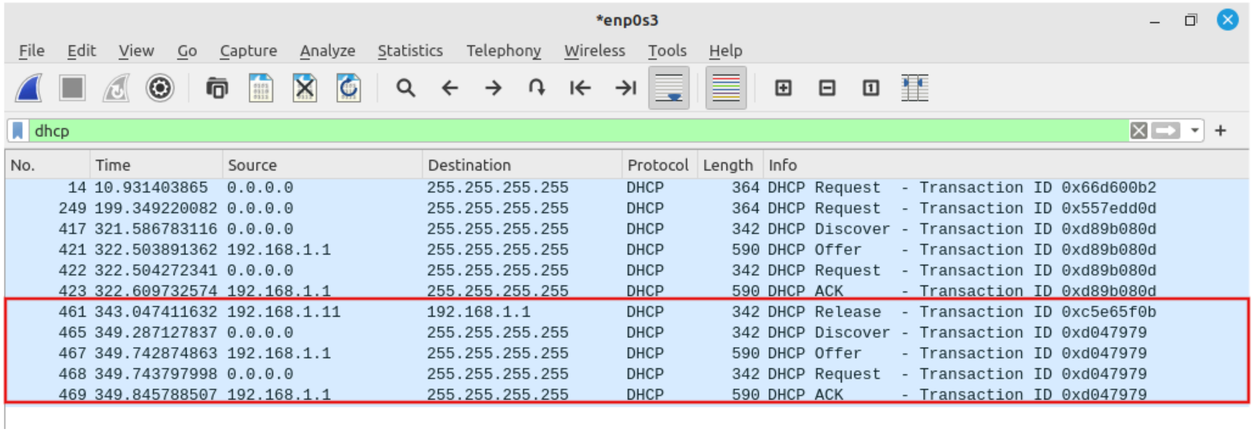
Open wireshark in your linux machine :



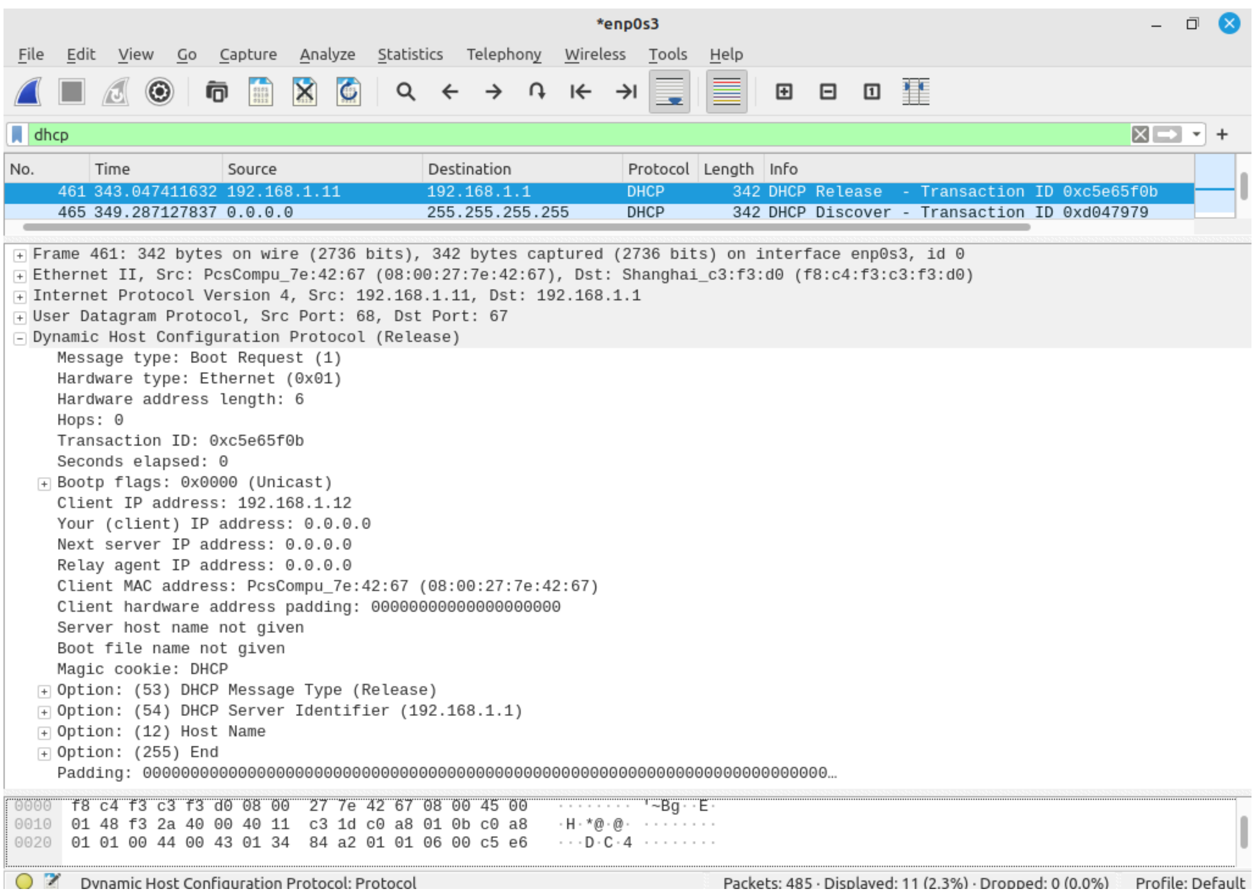
Use dhclient to renew and with option -r to release the IP address, so we can simulate the DHCP release, discover, request, offer and acknowledge.



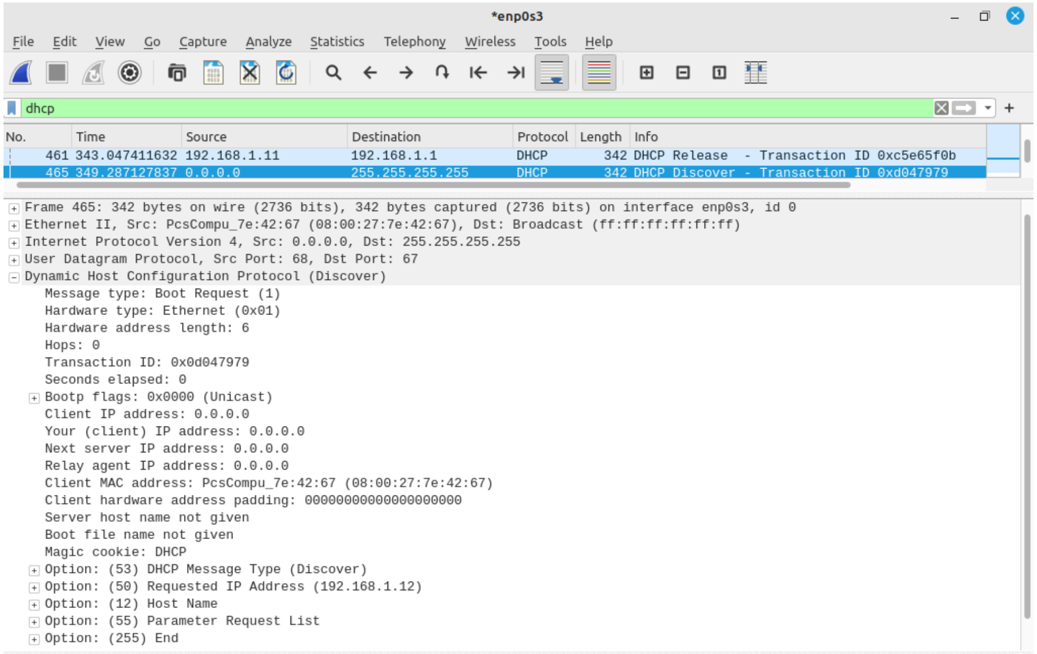
Here, we see the DHCP Release, Discover, Offer, Request and Acknowledgment are captured in the wireshark.



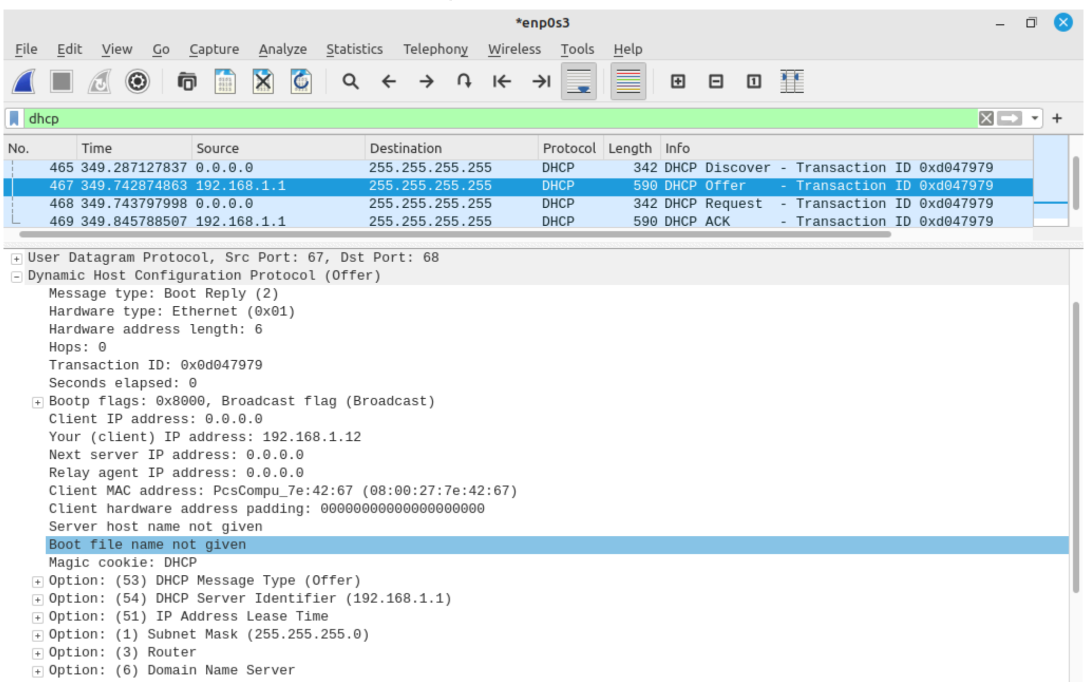
The DHCP release is used for a DHCP client which is the device, disconnecting or shutting down from using the IP address and sending it to the DHCP server that is not going to use it, so it will be made available to the others to use it.



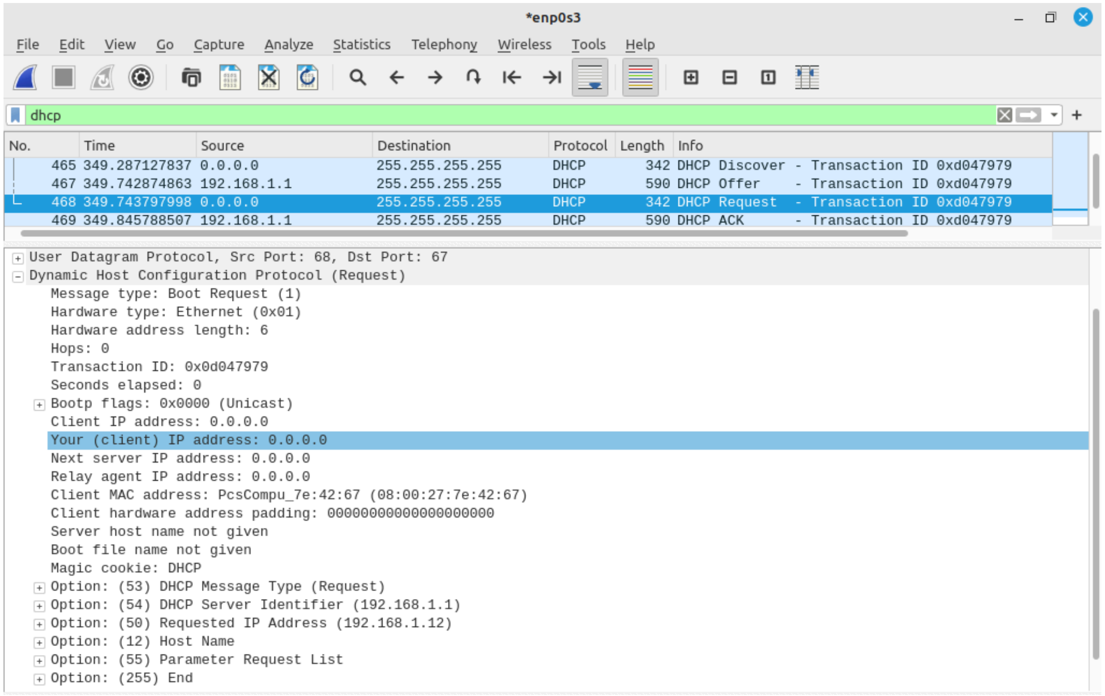
The DHCP discover is sent by a client to the all the devices in the network to the DHCP servers, as a broadcast, since the device doesn’t know the DHCP server.



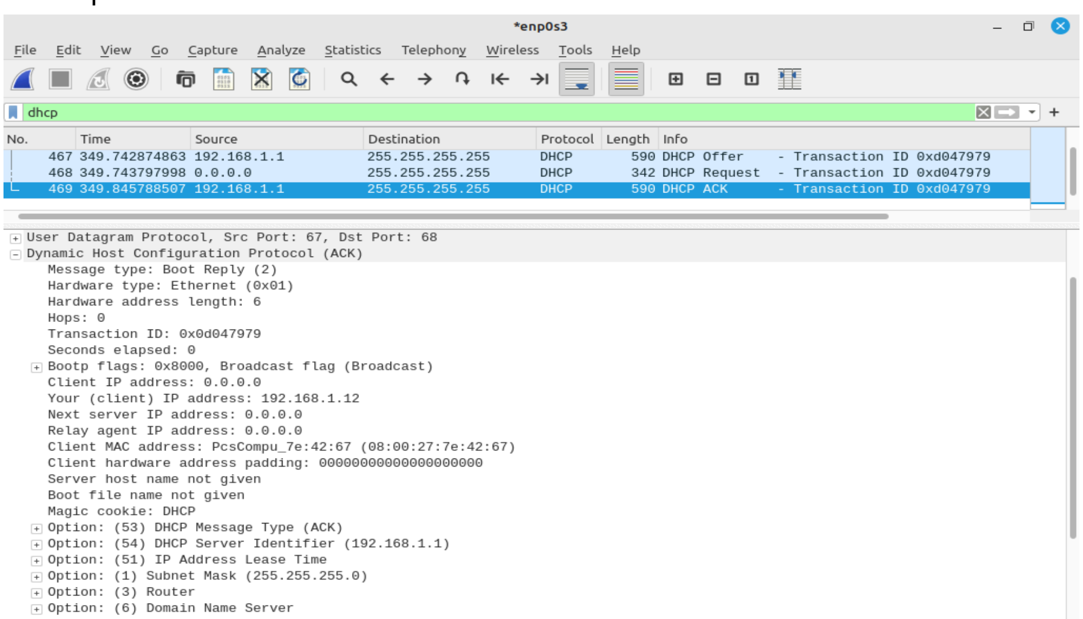
The DHCP offer is sent as a response by the DHCP server to the client. It sends its own information as well as the IP address offered by the DHCP. If there are multiple DHCP servers in the network, all the servers will respond.



The DHCP request is when the offers made by the servers are received by the client and it gets to choose one of them. It then sends the request message as a broadcast to indicate to all the devices in the network that it is going to use the IP address given. The broadcast also informs the DHCP servers too, so the same IP won't be used again.



The DHCP Acknowledgement is the acknowledgement message given by the DHCP that the IP address and configuration it gave is accepted by the client and hence will be used by the client henceforth. This is a confirmation process of the IP address assignment and completes the DHCP process.



5. Given an IP address range of 192.168.1.0/24, divide the network into 4 subnets. Task: Manually calculate the new subnet mask and the range of valid IP addresses for each subnet. Assign IP addresses from these subnets to devices in Cisco Packet Tracer and verify connectivity using ping between them.

Given IP 192.168.1.0/24, we have to device it into 4 subnets. Let's take the last octet and divide into 4 parts by converting into the binary form. We will take first two bits from the left to form the subnets :

00000000 → 00111111 = 0 → 63
01000000 → 01111111 = 64 → 127
10000000 → 10111111 = 128 → 191
11000000 → 11111111 = 192 → 255

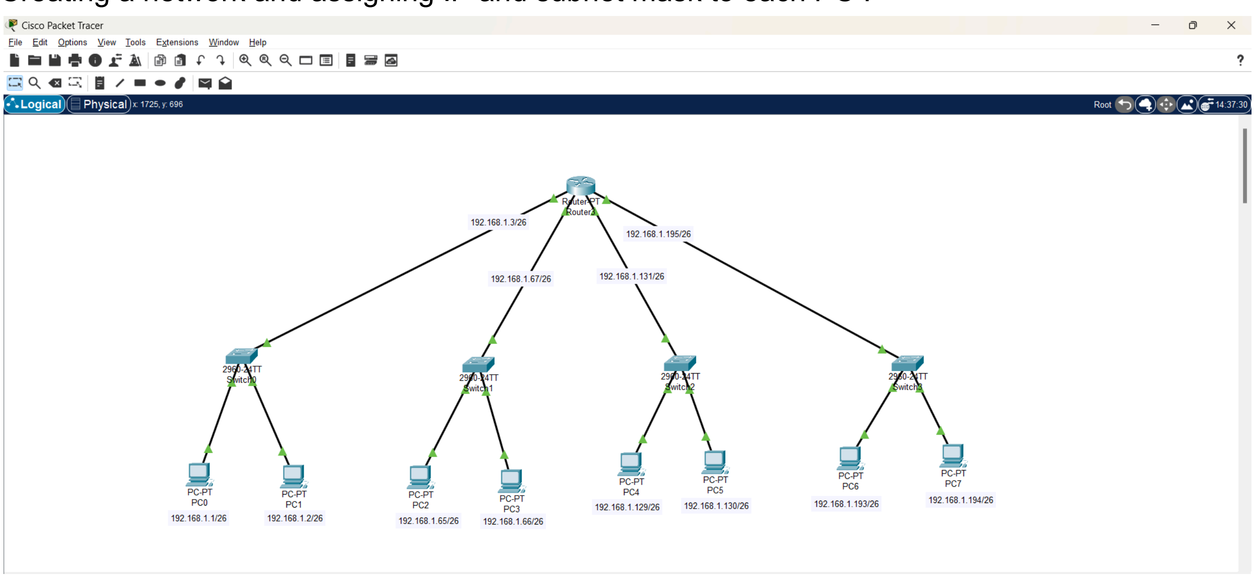
So, the 4 subnets will be :

192.168.1.0/26
192.168.1.64/26
192.168.1.128/26
192.168.1.192/26

The subnet mask, we have to take the last octet, and the network ID should be fully 1 and host ID should be fully 0.

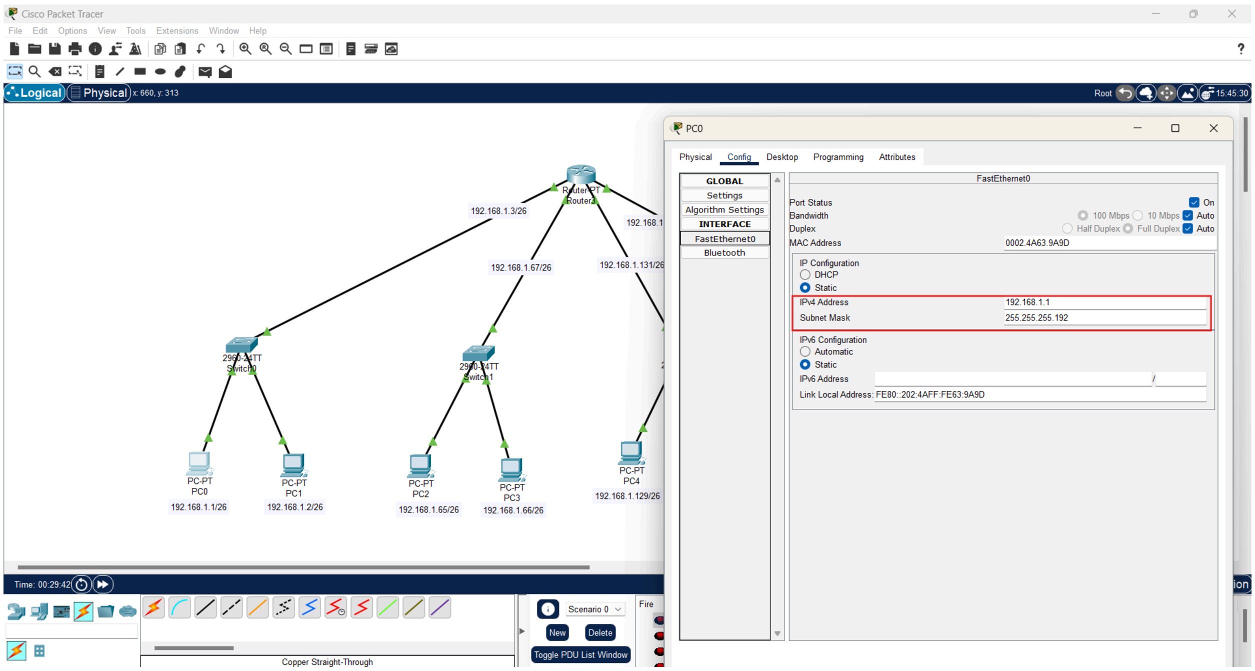
11111111.11111111.11111111.11000000 = 255.255.255.192 which is the subnet mask

Creating a network and assigning IP and subnet mask to each PC :



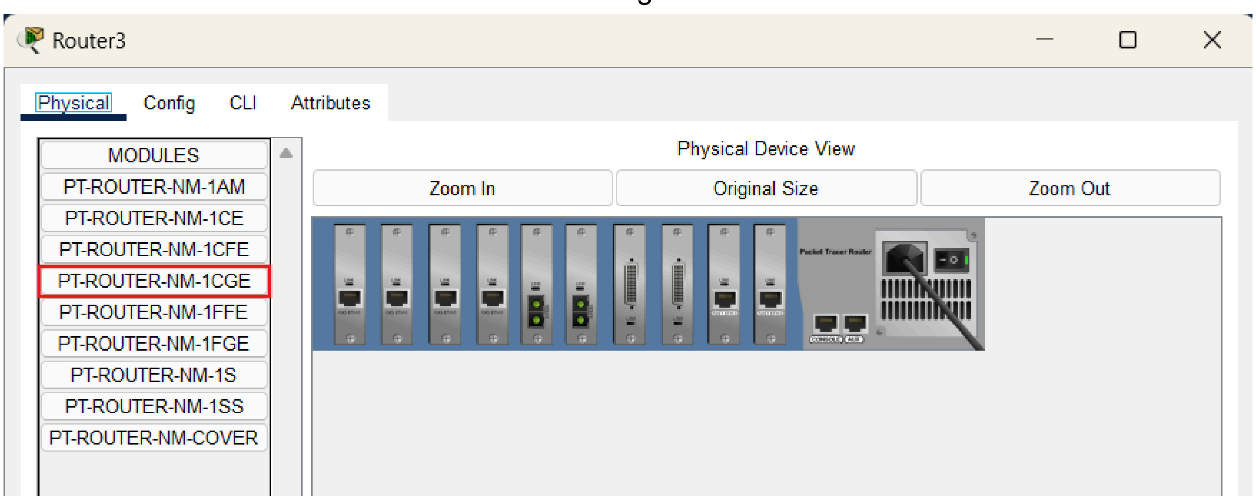
To Note : the first and last address of every subnet is reserved and hence cannot be used.

We added the PCs, Switches and the Router accordingly. The IP address of each PC is written below it and their subnet mask is 255.255.255.192.

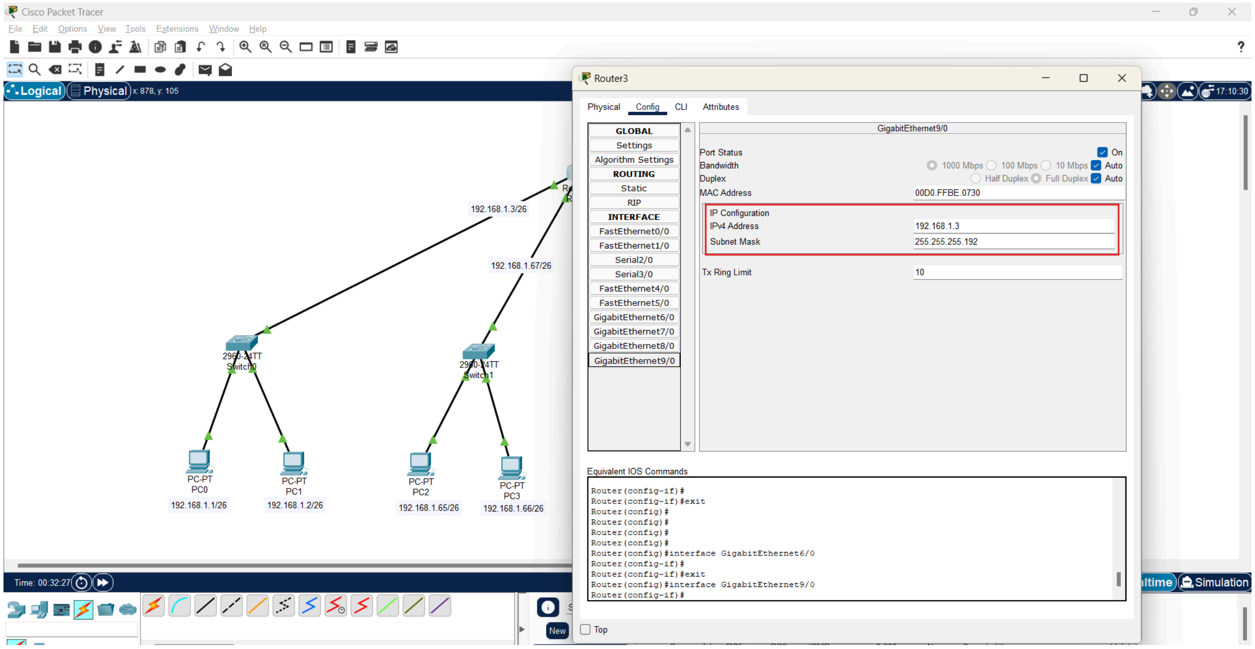


Connect them to their switches using Copper Straight through cable on Fast ethernet port.

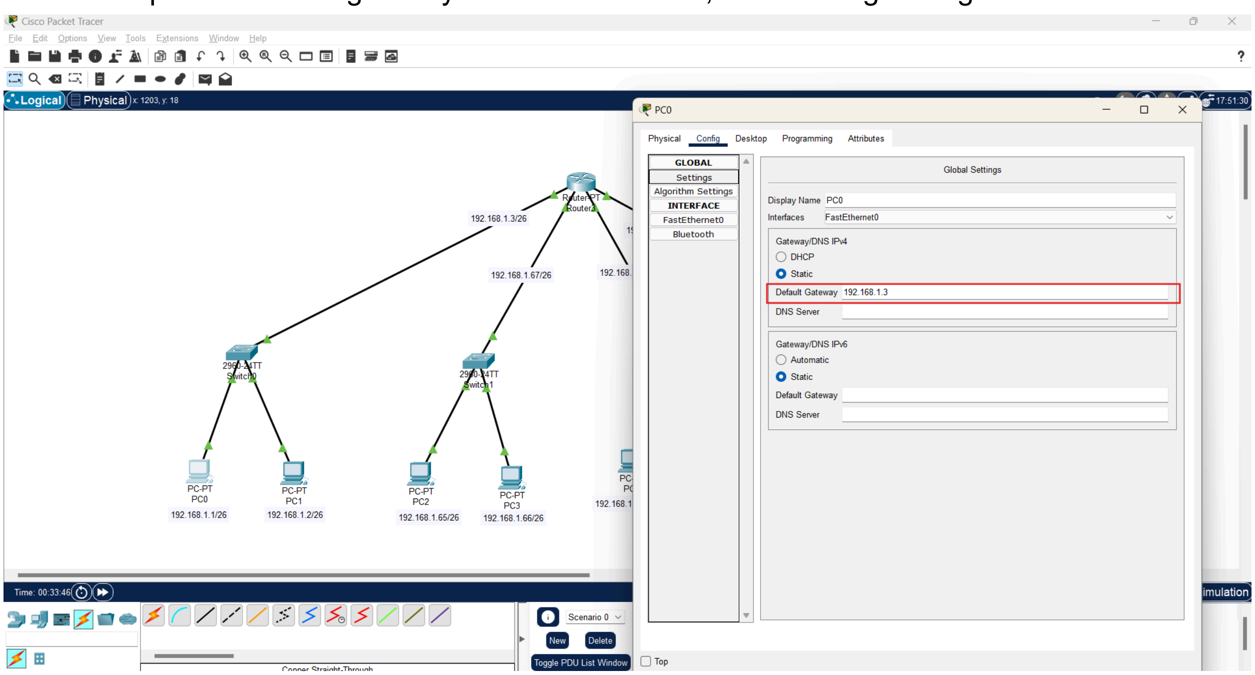
On the router, add the CGE module to the Router-PT which is used for the Gigabit-ethernet connection and add 4 of them since we are having four networks.



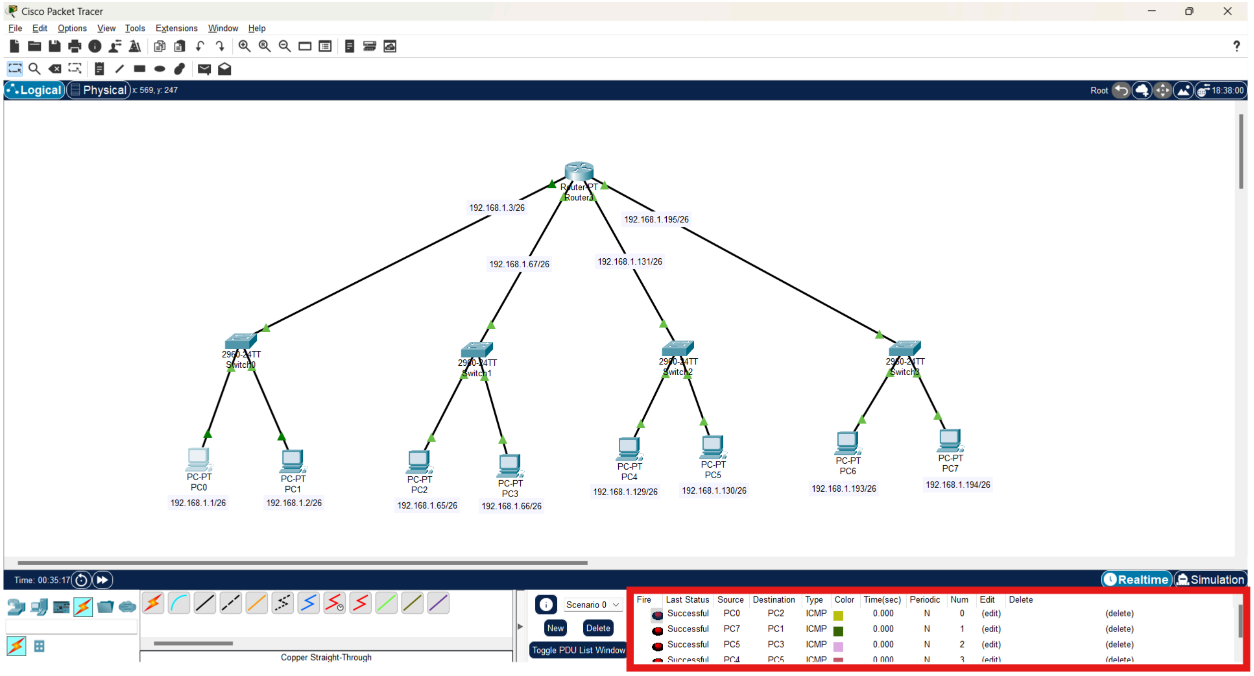
Add the IP address of each port of the router accordingly, which will be used as the default gateway.



Add the respective default gateway to each of the PCs, in the config settings section.



Now add the PDU from any PC to any PC with any Subnet and check if it is successful.



6. You are given three IP addresses: 10.1.1.1, 172.16.5.10, and 192.168.1.5. Task: Identify the class of each IP address (Class A, B, or C). What is the default subnet mask for each class? Provide the range of IP addresses for each class.

Given IP address	Class of IP address	Default Subnet Mask	IP Range of Class
10.1.1.1	Class A	255.0.0.0/8	0.0.0.0 → 127.255.255.255
172.16.5.10	Class B	255.255.0.0/16	128.0.0.0 → 191.255.255.255
192.168.1.5	Class C	255.255.255.0/24	192.0.0.0 → 223.255.255.255

We can identify the Class of the IP address in classful addressing by checking where the first octet lies in which range of class.

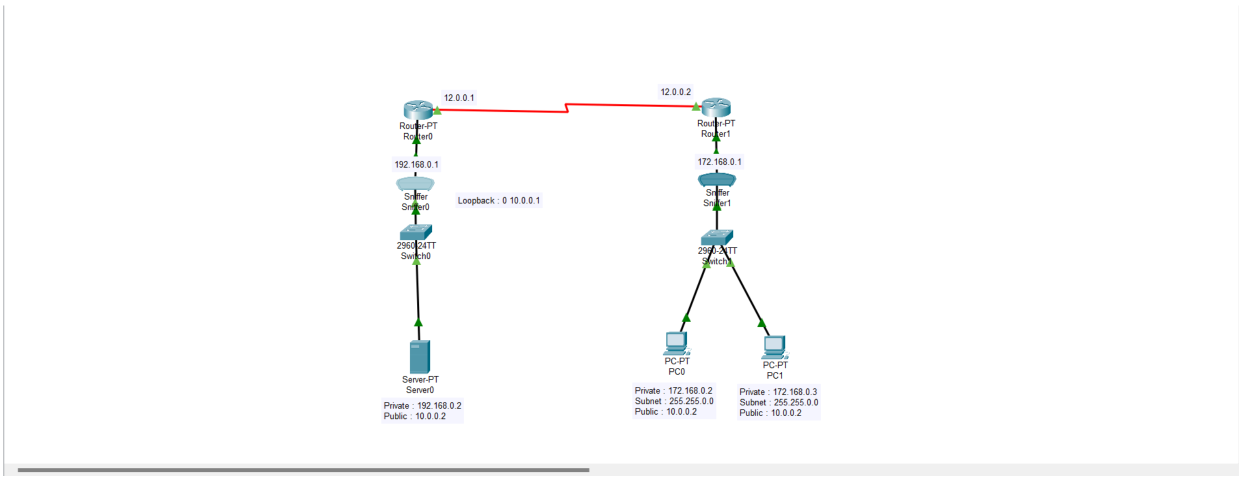
In first octet :
Class A → 0 to 127
Class B → 128 to 191
Class C → 192 to 223

7. In Cisco Packet Tracer, create a small network with multiple devices (e.g., 2 PCs and a router). Use private IP addresses (e.g., 192.168.1.x) on the PCs and configure the router to perform NAT to allow the PCs to access the internet.

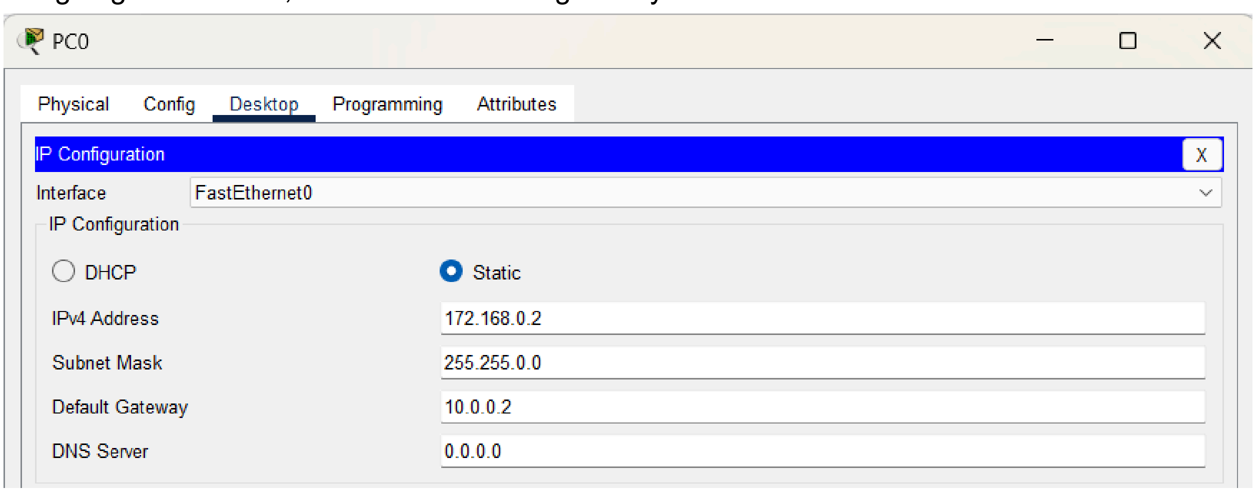
   Task: Test the NAT configuration by pinging an external IP address from the PCs and capture the traffic using Wireshark.

   What is the source IP address before and after NAT?

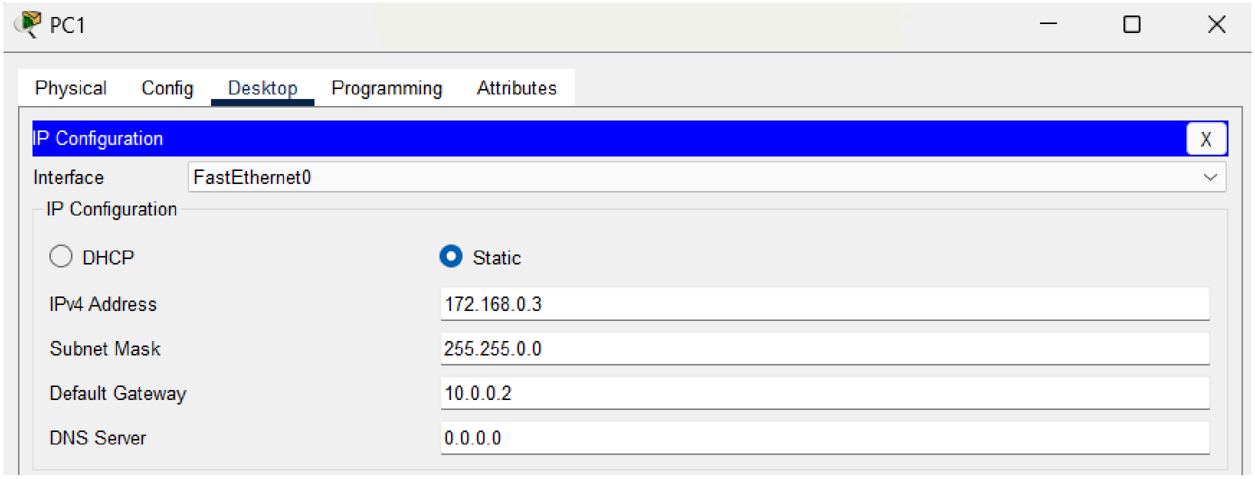
Creating this topology in the Cisco Packet Tracer, and connecting the devices appropriately.



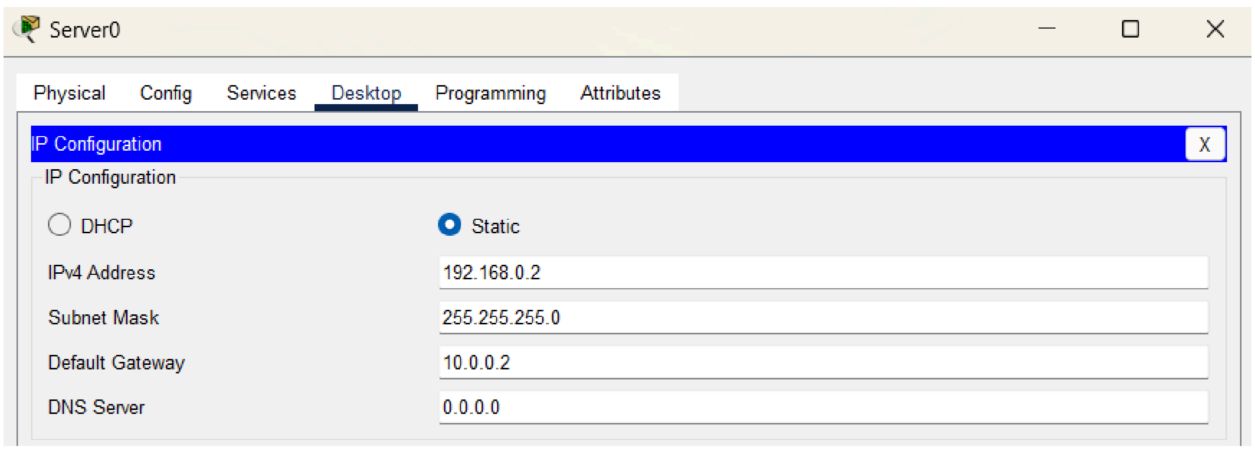
Assigning PC0 with IP, Subnet and default gateway



Assigning PC1 with IP, Subnet and default gateway :



Assigning Server0 with IP, Subnet and default gateway



Configuring Router 0 with IP address and subnet on fastEthernet0:

```
Router>
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#host r1
r1(config)#int f0/0
r1(config-if)#ip add 192.168.0.1 255.255.255.0
r1(config-if)#no shut
```

Changed hostname from Router to r1 for simplicity.

Adding Loopback interface and IP address to it, it is used in NAT to represent static IP address :

```
r1(config-if)#int loopback 0

r1(config-if)#
%LINK-5-CHANGED: Interface Loopback0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback0, changed state to up

r1(config-if)#int loopback 0
r1(config-if)#ip add 10.0.0.1 255.0.0.0
```

Adding IP address to Serial Connection from router to router, the clock rate states that it transfers at speed of 64 kbps :

```
r1(config)#int s2/0
r1(config-if)#ip add 12.0.0.1 255.0.0.0
r1(config-if)#clock rate 64000
r1(config-if)#no shut
```

Initializing RIP and adding the IP addresses to the network, by doing this any router running RIP will identify these two IP address :

```
r1(config-if)#router rip
r1(config-router)#net 10.0.0.0
r1(config-router)#net 12.0.0.0
r1(config-router)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state to up
```

Configuring the IP address and stating the NAT inside :

```
r1(config-router)#ip nat inside source static 192.168.0.2 10.0.0.2
```

Configuring the NAT outside part as the serial port side :

```
r1(config)#int s2/0
r1(config-if)#ip nat outside
r1(config-if)#int f0/0
r1(config-if)#ip nat inside
```

Checking NAT address :

```
r1(config-if)#^Z
r1#
%SYS-5-CONFIG_I: Configured from console by console

r1#
r1#sh ip nat tran
r1#sh ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  10.0.0.2          192.168.0.2        ---                ---

r1#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to down

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
```

Configuring on Router 1 with IP address:

```
Router>en

Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#
Router(config)#host r2
r2(config)#int f0/0
r2(config-if)#ip add 172.168.0.1 255.255.0.0
r2(config-if)#no shut
```

Adding IP address to Serial port on router 1 :

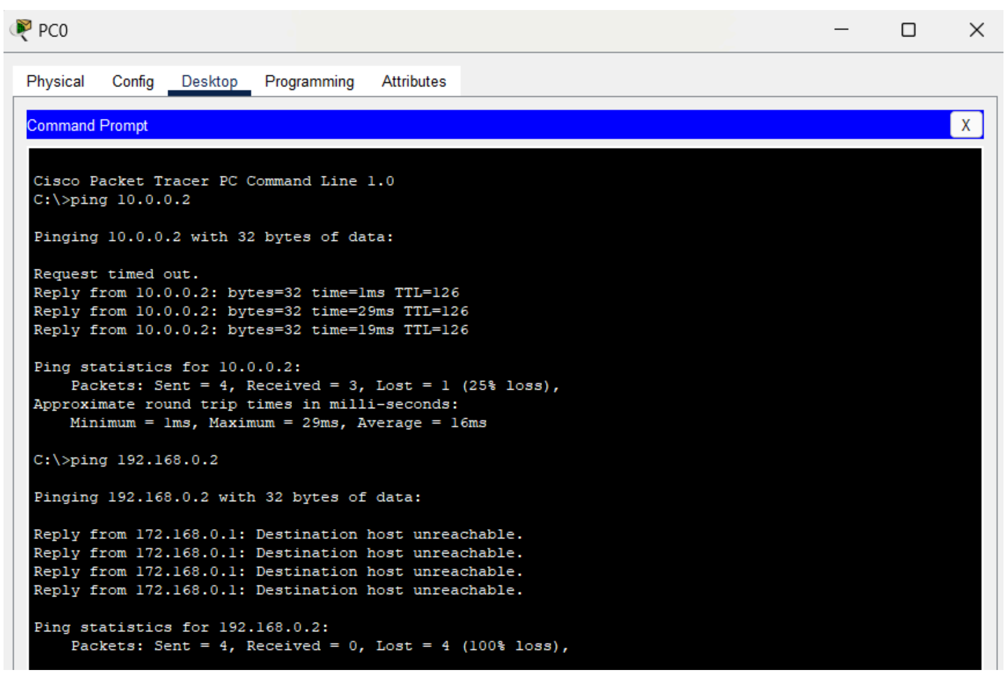
```
r2(config)#int s2/0
r2(config-if)#ip add 12.0.0.2 255.0.0.0
r2(config-if)#no shut
```

Intialising RIP and adding IP address to the network :

```
r2(config-if)#router rip
r2(config-router)#router rip
r2(config-router)#net 12.0.0.0
r2(config-router)#net 172.168.0.0

r2(config-router)#end
```

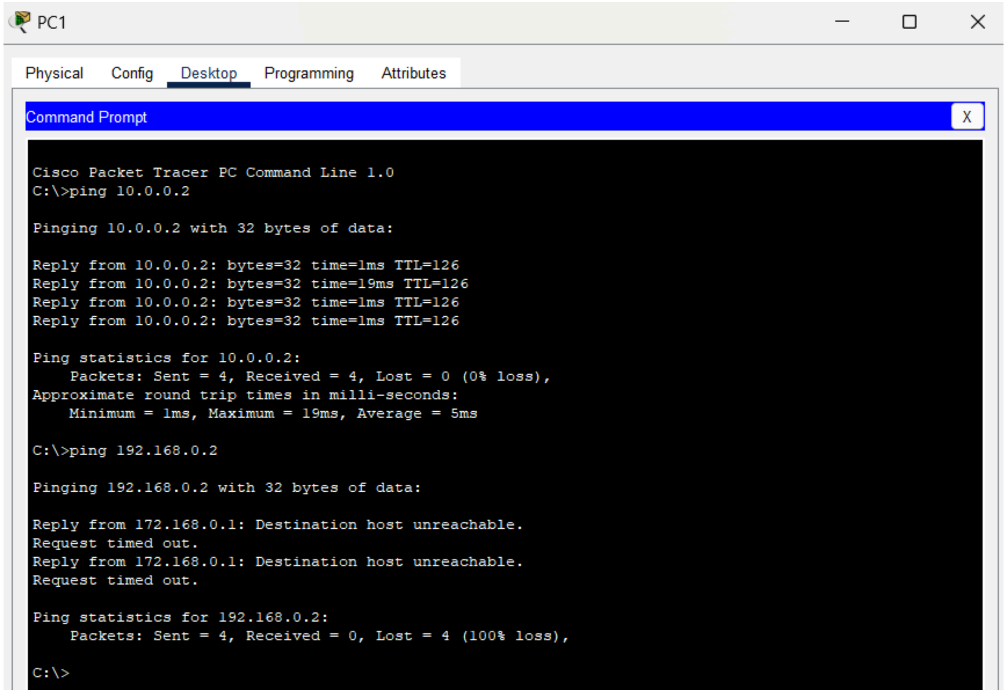
<u>Pinging From PC0 / PC 1 to Server :</u>

PC 0 :




```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Request timed out.
Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time=29ms TTL=126
Reply from 10.0.0.2: bytes=32 time=19ms TTL=126

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 29ms, Average = 16ms

C:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Reply from 172.168.0.1: Destination host unreachable.
Reply from 172.168.0.1: Destination host unreachable.
Reply from 172.168.0.1: Destination host unreachable.
Reply from 172.168.0.1: Destination host unreachable.

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

PC1 :

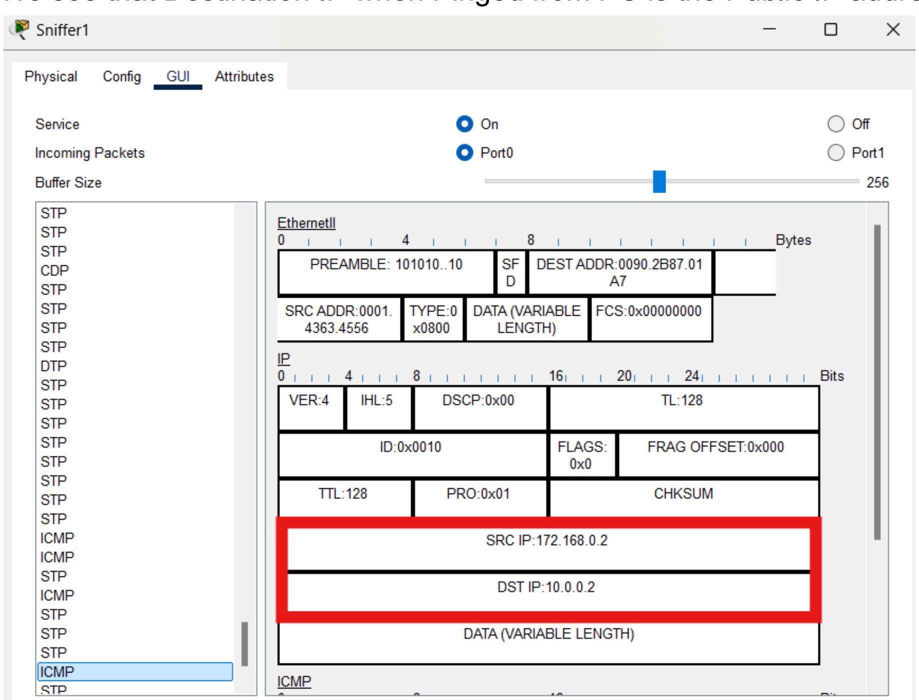



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 10.0.0.2

Pinging 10.0.0.2 with 32 bytes of data:

Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time=19ms TTL=126
Reply from 10.0.0.2: bytes=32 time=1ms TTL=126
Reply from 10.0.0.2: bytes=32 time=1ms TTL=126

Ping statistics for 10.0.0.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 19ms, Average = 5ms

C:\>ping 192.168.0.2

Pinging 192.168.0.2 with 32 bytes of data:

Reply from 172.168.0.1: Destination host unreachable.
Request timed out.
Reply from 172.168.0.1: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

We see that Server is not reachable from its own Private IP address but only reachable through its allowed Public IP address.

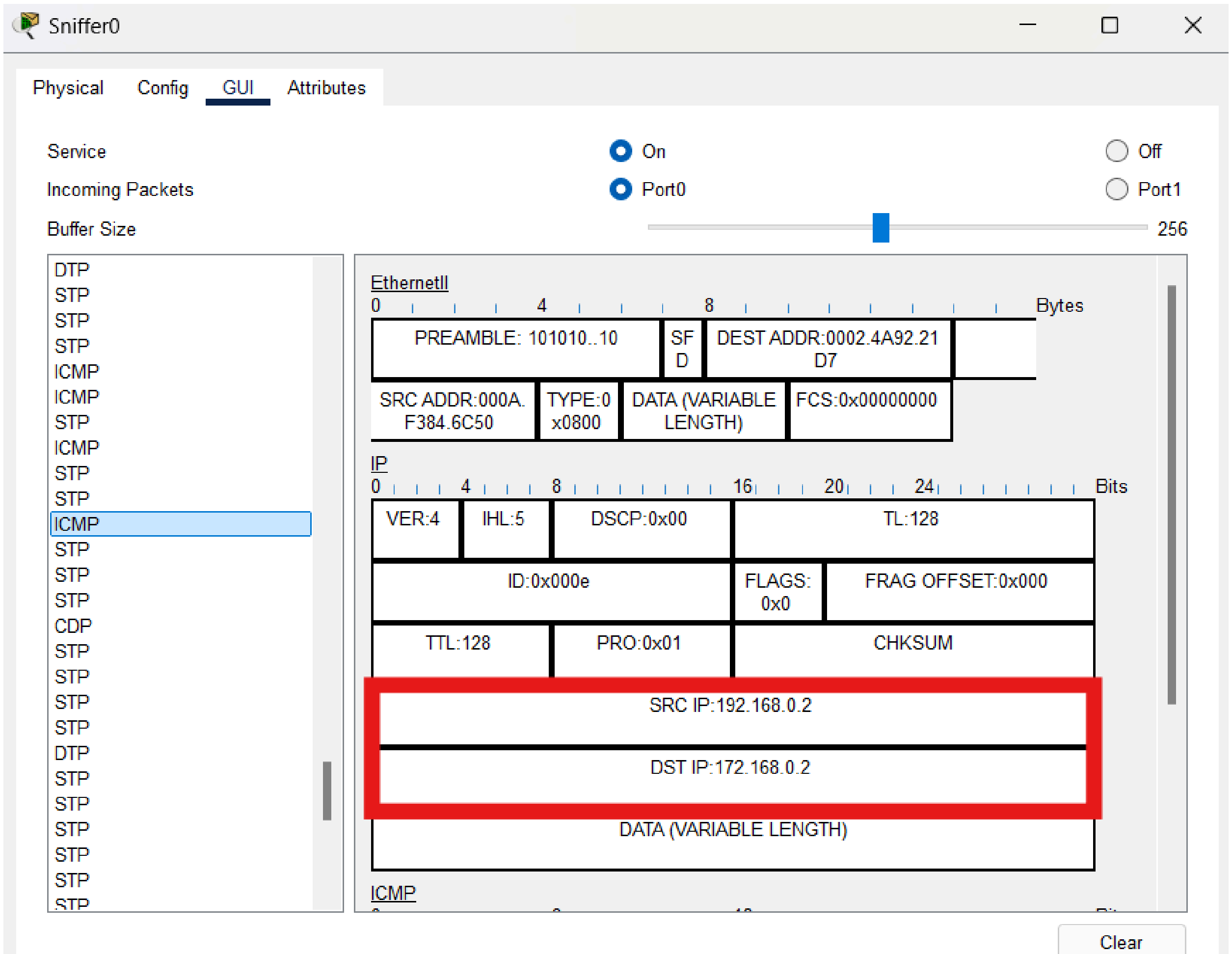
<u>PC Side Sniffer :</u>

We see that Destination IP when Pinged from PC is the Public IP address.



Server Side Sniffer :

On the server side, examining the packet, we see that the IP address has changed from Public IP to Private IP address by NAT.



Using a sniffer device to capture packets as Wireshark is not supported by the Cisco Packet Tracer.