



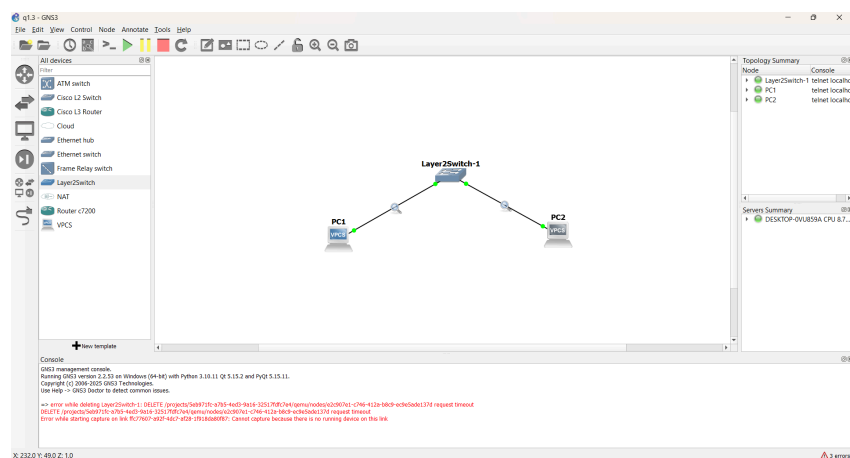
## Networking Assessment 3 - Module 5

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 communicate 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.



```

Welcome to Virtual PC Simulator, version 0.6.2
Dedicated to Daling.
Build time: Apr 10 2019 02:42:20
Copyright (c) 2007-2014, Paul Meng (mirnshi@gmail.com)
All rights reserved.

VPCS is free software, distributed under the terms of the "BSD" licence.
Source code and license can be found at vpcs.sf.net.
For more information, please visit wiki.freecode.com.cn.

Press '?' to get help.

Executing the startup file

PC1> ip 192.168.1.10 255.255.255.0
Checking for duplicate address...
PC1 : 192.168.1.10 255.255.255.0

PC1> ping 192.168.1.11
84 bytes from 192.168.1.11 icmp_seq=1 ttl=64 time=1.880 ms
84 bytes from 192.168.1.11 icmp_seq=2 ttl=64 time=2.653 ms
84 bytes from 192.168.1.11 icmp_seq=3 ttl=64 time=1.637 ms
84 bytes from 192.168.1.11 icmp_seq=4 ttl=64 time=4.106 ms
84 bytes from 192.168.1.11 icmp_seq=5 ttl=64 time=2.277 ms

PC1> 
```

Captured an ARP request and reply packets in wireshark.

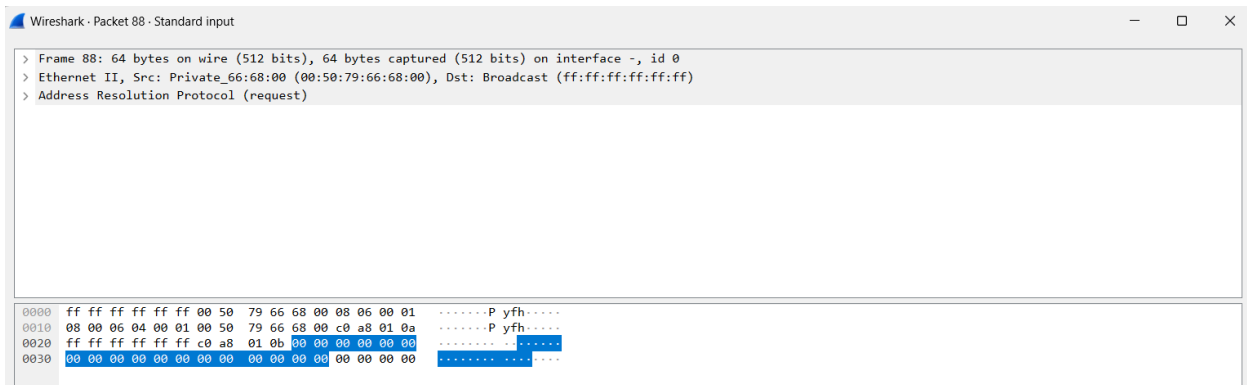
The image shows a Wireshark packet capture window titled "Capturing from Standard input [PC1 Ethernet0 to Layer2Switch-1 Ethernet0]". The packet list on the left shows several packets, with packets 88 and 89 highlighted in red. Packet 88 is an ARP request from 192.168.1.10 to 192.168.1.11. Packet 89 is an ARP reply from 192.168.1.11 to 192.168.1.10. The packet details pane on the right shows the structure of the ARP request packet, including the Ethernet II header, Internet Protocol Version 4 header, and ARP request structure. The packet bytes pane on the right shows the raw data of the packet.

No.	Time	Source	Destination	Protocol	Length	Info
81	119.575397	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
82	122.171897	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
83	124.532100	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
84	127.134206	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
85	129.862055	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
86	130.551586	0c:af:3d:90:00:00	0c:af:3d:90:00:00	LOOP	60	Reply
87	132.175390	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
88	132.987275	Private_66:68:00	Broadcast	ARP	64	Who has 192.168.1.11? Tell 192.168.1.10
89	132.989198	Private_66:68:01	Private_66:68:00	ARP	64	192.168.1.11 is at 00:50:79:66:68:01
90	132.921520	192.168.1.10	192.168.1.11	ICMP	98	Echo (ping) request id=0x6893, seq=1/256, ttl=64 (reply in 91)
91	132.923188	192.168.1.11	192.168.1.10	ICMP	98	Echo (ping) reply id=0x6893, seq=1/256, ttl=64 (request in 90)
92	133.951660	192.168.1.10	192.168.1.11	ICMP	98	Echo (ping) request id=0x6993, seq=2/512, ttl=64 (reply in 93)
93	133.954095	192.168.1.11	192.168.1.10	ICMP	98	Echo (ping) reply id=0x6993, seq=2/512, ttl=64 (request in 92)
94	134.977074	0c:af:3d:90:00:00	Spanning-tree-(for-bridges)_00	STP	60	Conf. Root = 32768/1/0c:af:3d:90:00:00 Cost = 0 Port = 0x8001
95	134.980306	192.168.1.10	192.168.1.11	ICMP	98	Echo (ping) request id=0x6a93, seq=3/768, ttl=64 (reply in 96)
96	134.981720	192.168.1.11	192.168.1.10	ICMP	98	Echo (ping) reply id=0x6a93, seq=3/768, ttl=64 (request in 95)
97	136.001063	192.168.1.10	192.168.1.11	ICMP	98	Echo (ping) request id=0x6b93, seq=4/1024, ttl=64 (reply in 98)
98	136.005551	192.168.1.11	192.168.1.10	ICMP	98	Echo (ping) reply id=0x6b93, seq=4/1024, ttl=64 (request in 97)

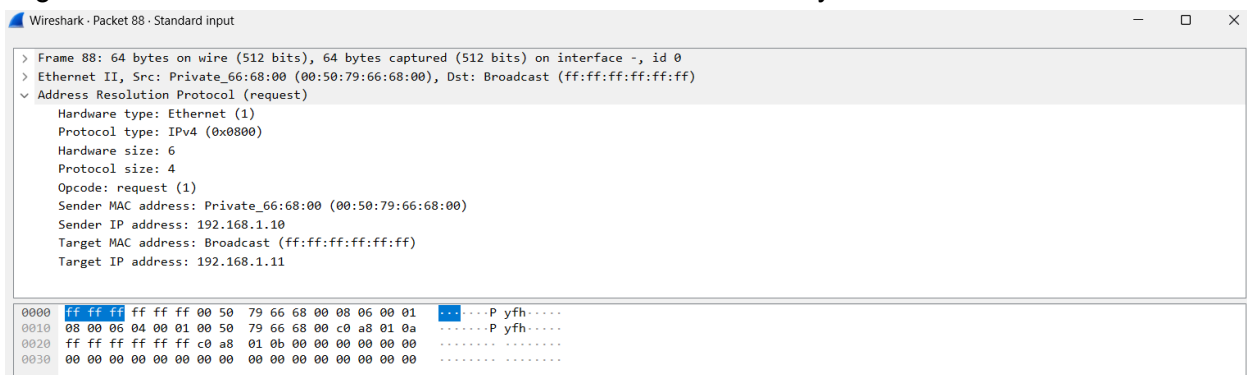
> Frame 88: 64 bytes on wire (512 bits), 64 bytes captured (512 bits) on interface -, id 0  
> Ethernet II, Src: Private\_66:68:00 (00:50:79:66:68:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
▼ Address Resolution Protocol (request)  
Hardware type: Ethernet (1)  
Protocol type: IPv4 (0x0800)  
Hardware size: 6  
Protocol size: 4  
Opcode: request (1)  
Sender MAC address: Private\_66:68:00 (00:50:79:66:68:00)  
Sender IP address: 192.168.1.10  
Target MAC address: Broadcast (ff:ff:ff:ff:ff:ff)  
Target IP address: 192.168.1.11

Standard input: <live capture in progress>      Packets: 114      Profile: Default

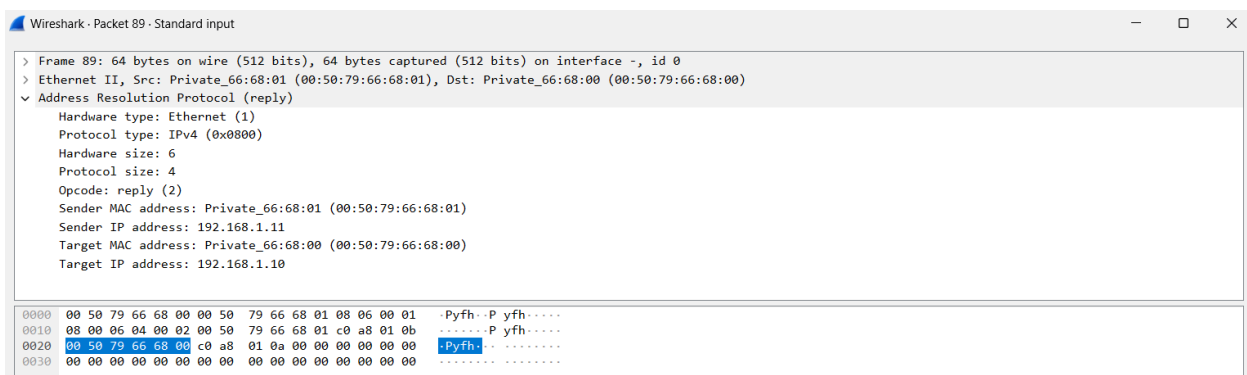
Let's examine the Request ARP packet, The frame is of 43 bytes and ethernet frame consist of the source and destination IP address.



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.

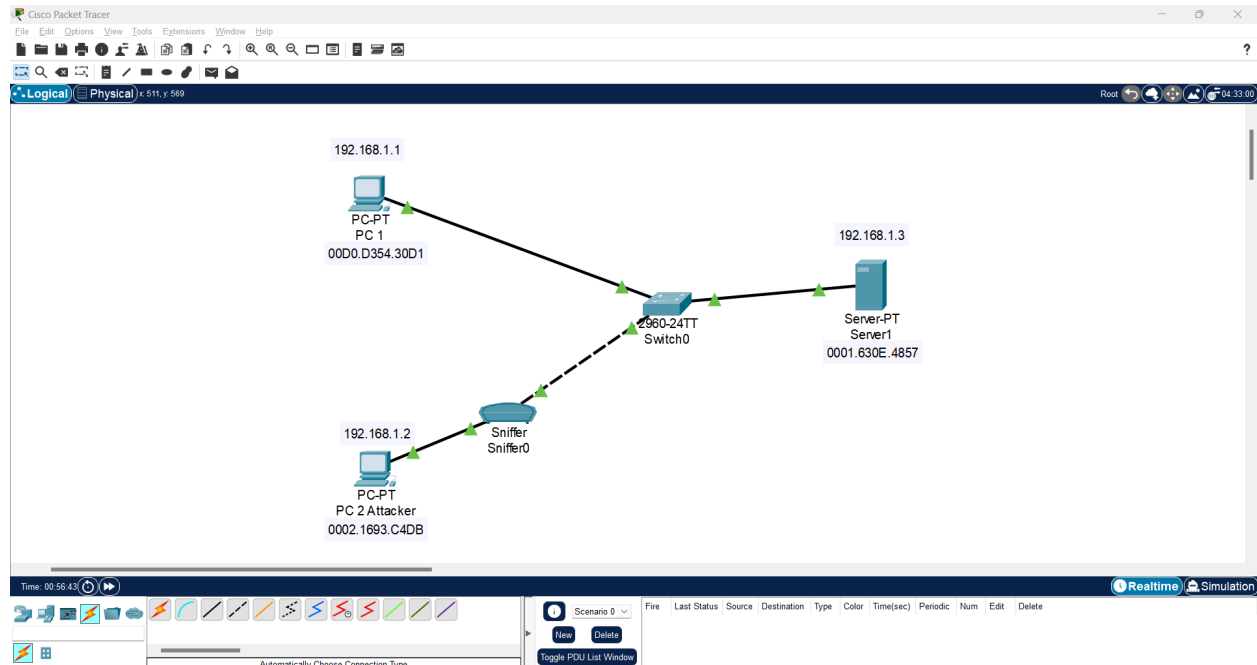


Only the intended device with the target IP address will reply, with its own IP and MAC as the sender address, and the the previously sender IP and MAC address as the current target IP and MAC Address.



- 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 :

The screenshot shows the 'PC 1' window with the 'Desktop' tab selected. A 'Command Prompt' window is open, displaying the following commands and output:

```
Cisco Packet Tracer PC Command Line 1.0
C:\>arp -a
No ARP Entries Found
C:\>
C:\>ping 192.168.1.3

Pinging 192.168.1.3 with 32 bytes of data:

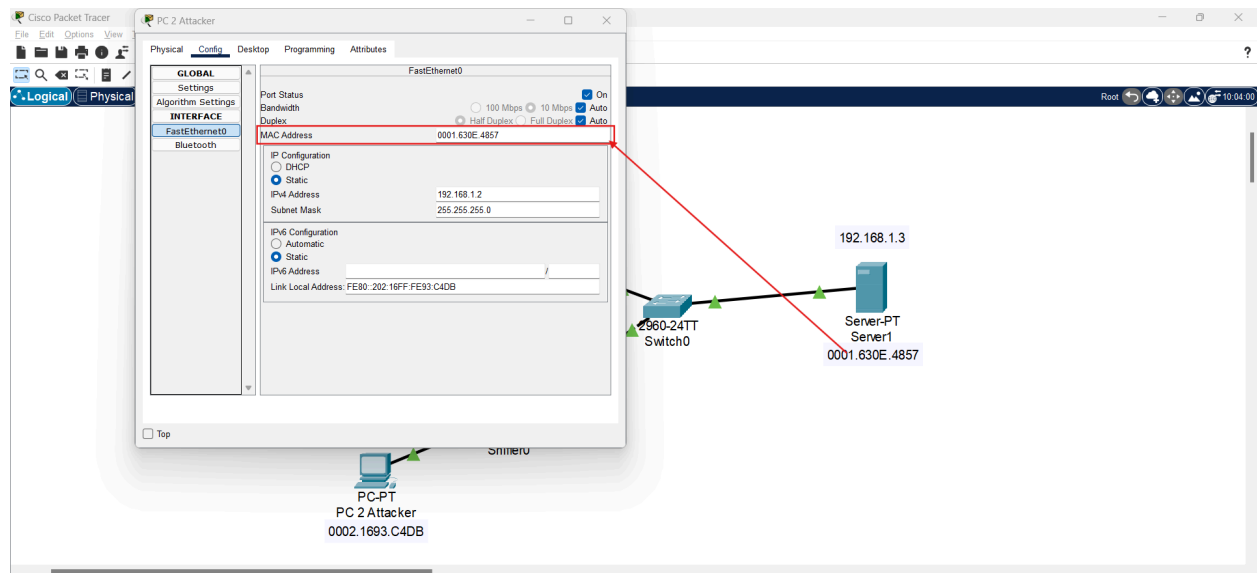
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time<1ms TTL=128
Reply from 192.168.1.3: bytes=32 time=6ms TTL=128

Ping statistics for 192.168.1.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 6ms, Average = 1ms

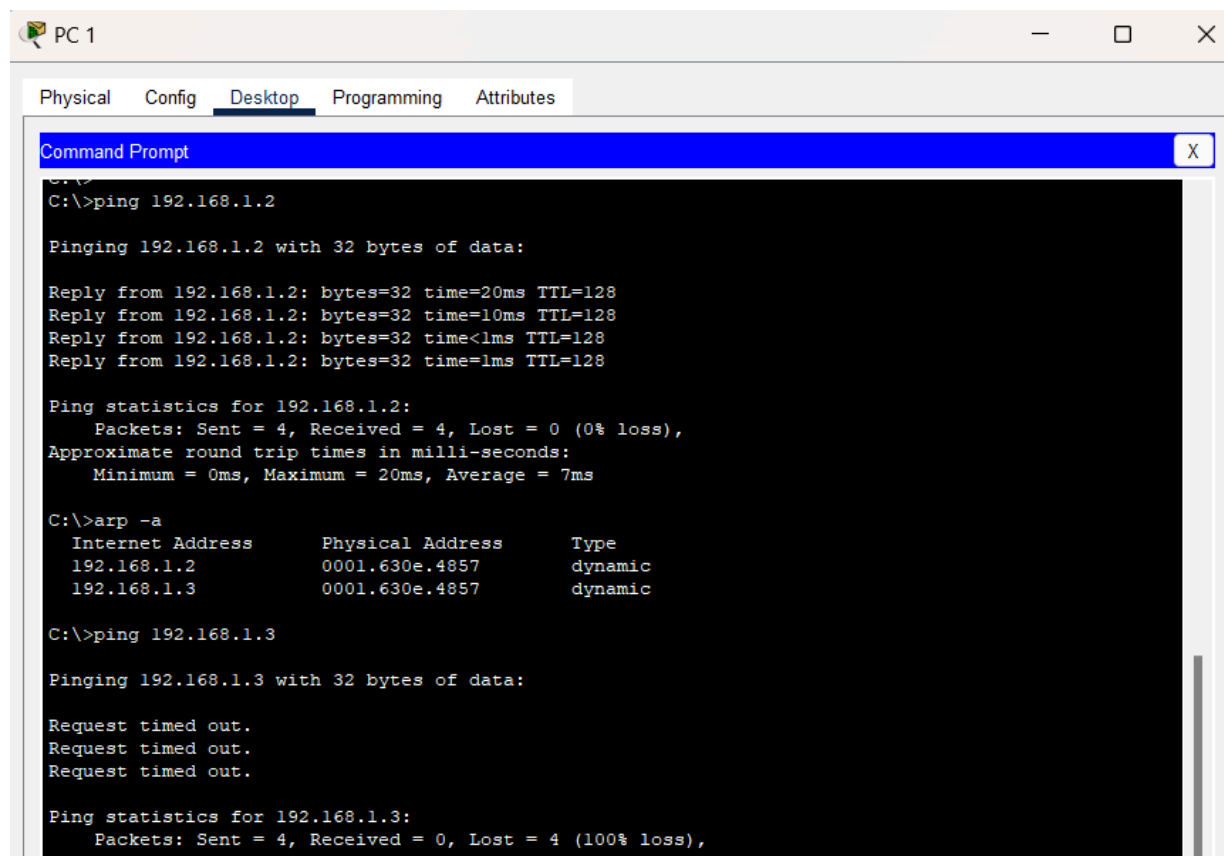
C:\>
C:\>
C:\>arp -a
    Internet Address      Physical Address      Type
    192.168.1.3           0001.630e.4857       dynamic

C:\>|
```

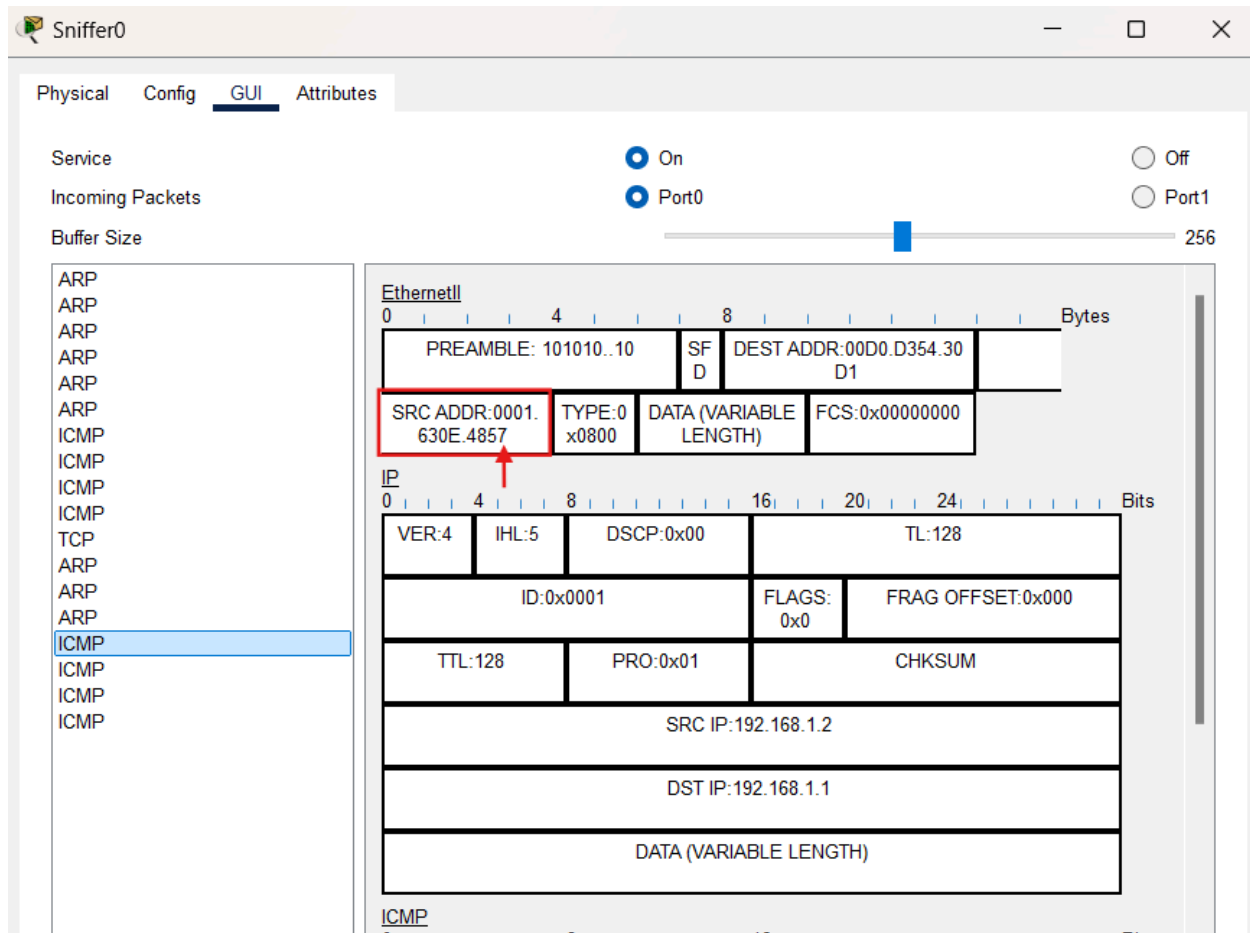
Now, let's change the MAC address of the PC 2 to the MAC address of the server to spoof the MAC address.



Now again pinging from PC1 to the Server it is being timed out, so let's 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 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 ethernet 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
  ethernet:
    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$
```

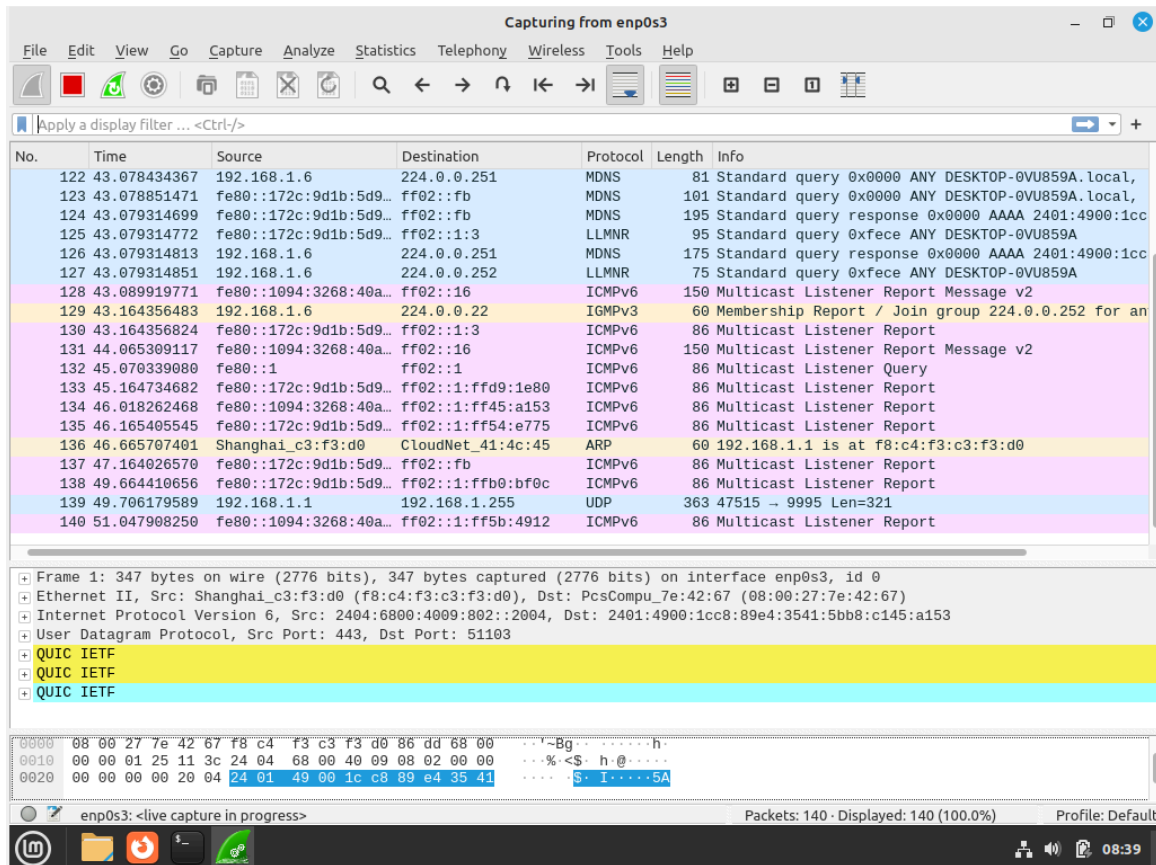
Now the changes have been applied you can check it using ip a, and we see that the static IP has been updated with 192.168.1.100/24 :

```
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.

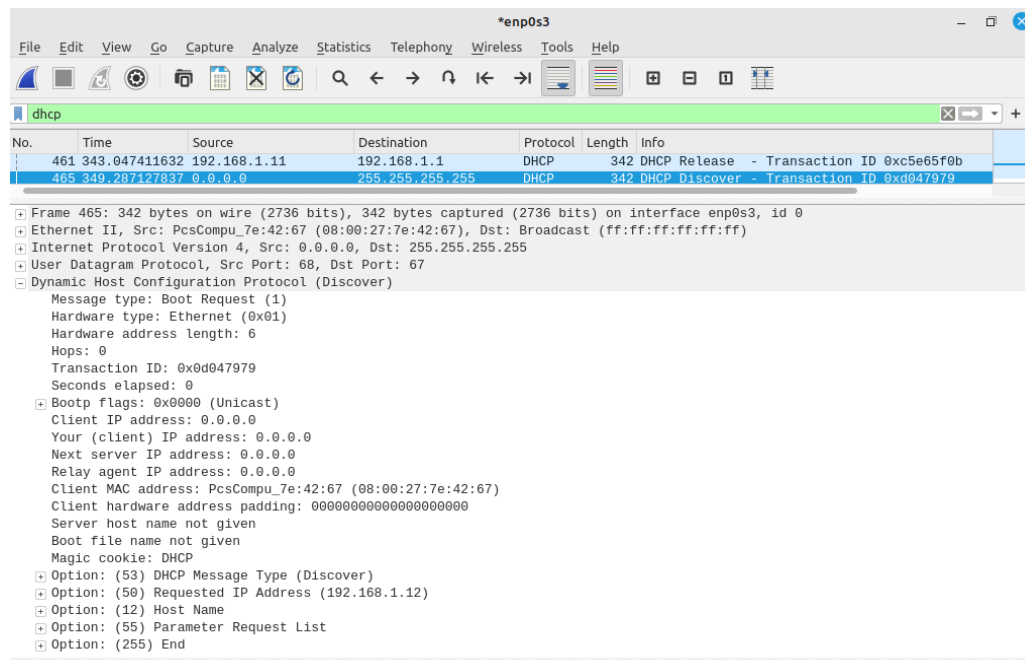
```
mint@mint:~$ sudo dhclient enp0s3
mint@mint:~$ sudo dhclient -r enp0s3
Killed old client process
mint@mint:~$ sudo dhclient enp0s3
mint@mint:~$
```

Wireshark interface showing a packet capture on the 'enp0s3' interface. The packet list displays a series of DHCP messages. The selected packet is a DHCP ACK (packet 469) with transaction ID 0xd047979.

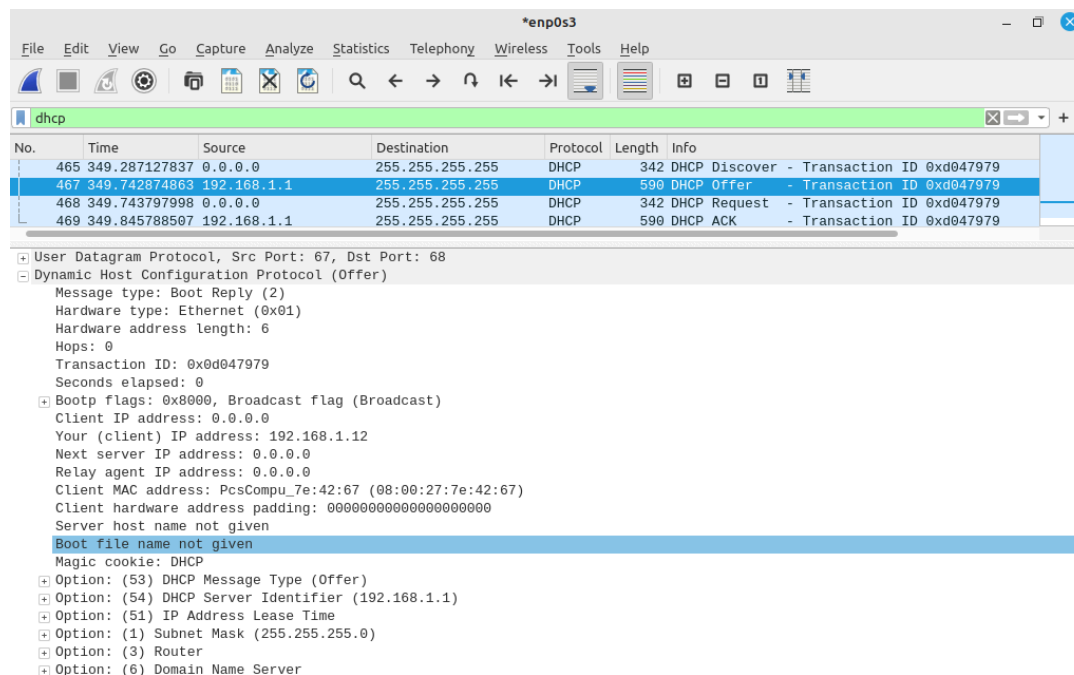
No.	Time	Source	Destination	Protocol	Length	Info
14	10.931403865	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0x66d600b2
249	199.349220082	0.0.0.0	255.255.255.255	DHCP	364	DHCP Request - Transaction ID 0x557edd0d
417	321.586783116	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xd89b080d
421	322.503891362	192.168.1.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0xd89b080d
422	322.504272341	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xd89b080d
423	322.609732574	192.168.1.1	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0xd89b080d
461	343.047411632	192.168.1.11	192.168.1.1	DHCP	342	DHCP Release - Transaction ID 0xc5e65f0b
465	349.287127837	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xd047979
467	349.742874863	192.168.1.1	255.255.255.255	DHCP	590	DHCP Offer - Transaction ID 0xd047979
468	349.743797998	0.0.0.0	255.255.255.255	DHCP	342	DHCP Request - Transaction ID 0xd047979
469	349.845788507	192.168.1.1	255.255.255.255	DHCP	590	DHCP ACK - Transaction ID 0xd047979

[illegible]

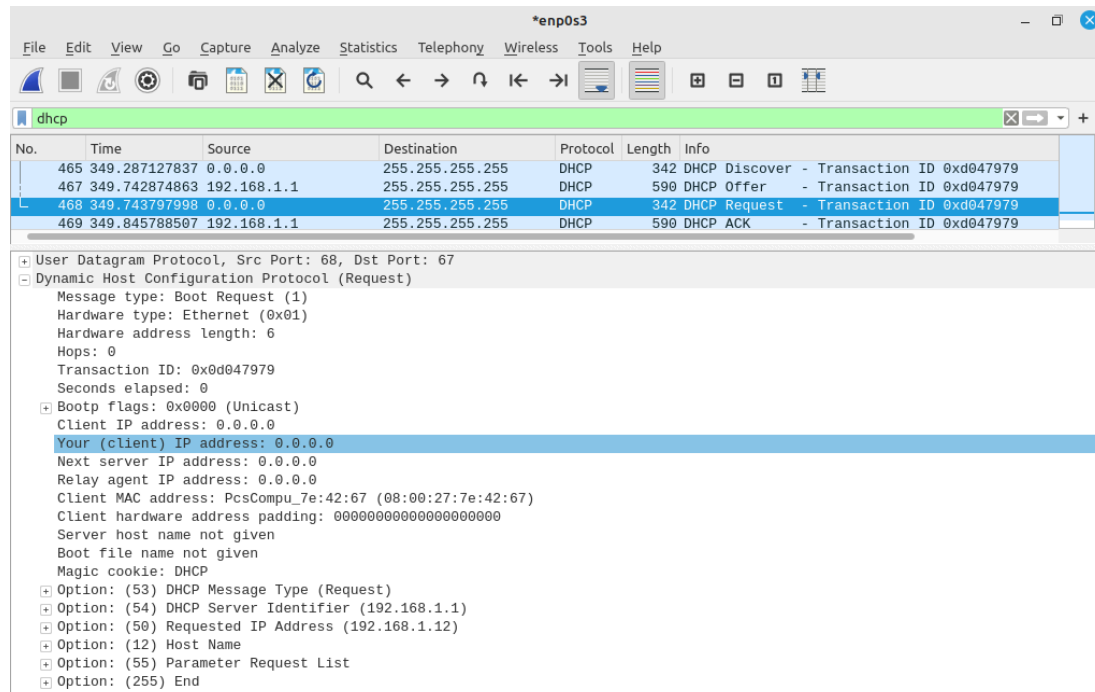
The DHCP discover is sent by a client to all the devices in the network to the DHCP servers, as a broadcast, since the device doesn't know the DHCP server.



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.



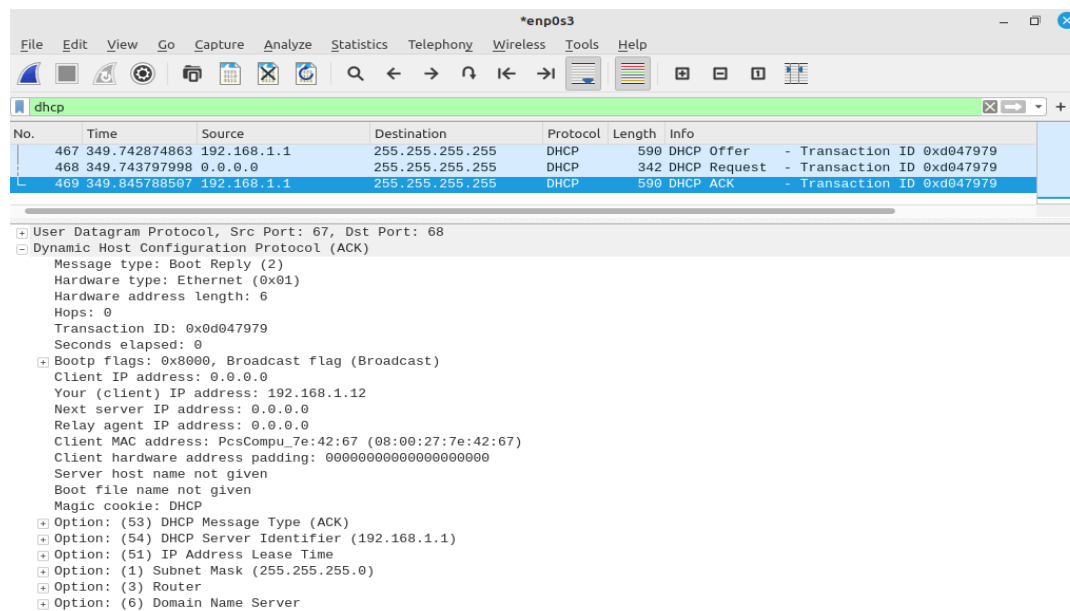
The DHCP request is when the offers made by the servers are received by the client and it gets to choose one of them. It then sends the request message as a broadcast to indicate to all the devices in the network that it is going to use the IP address given. The broadcast also informs the DHCP servers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



The screenshot shows a Wireshark packet capture on interface \*enp0s3. The packet list shows four DHCP messages: Discover (342 bytes), Offer (590 bytes), Request (342 bytes), and ACK (590 bytes). The packet details pane for the selected DHCP Request (No. 468) shows the following information:

Field	Value
Message type	Boot Request (1)
Hardware type	Ethernet (0x01)
Hardware address length	6
Hops	0
Transaction ID	0x0d047979
Seconds elapsed	0
Bootp flags	0x0000 (Unicast)
Client IP address	0.0.0.0
Your (client) IP address	0.0.0.0
Next server IP address	0.0.0.0
Relay agent IP address	0.0.0.0
Client MAC address	PcsCompu_7e:42:67 (08:00:27:7e:42:67)
Client hardware address padding	00000000000000000000
Server host name	not given
Boot file name	not given
Magic cookie	DHCP
Option (53)	DHCP Message Type (Request)
Option (54)	DHCP Server Identifier (192.168.1.1)
Option (50)	Requested IP Address (192.168.1.12)
Option (12)	Host Name
Option (55)	Parameter Request List
Option (255)	End

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.



The screenshot shows a Wireshark packet capture on interface \*enp0s3. The packet list shows three DHCP messages: Offer (590 bytes), Request (342 bytes), and ACK (590 bytes). The packet details pane for the selected DHCP ACK (No. 469) shows the following information:

Field	Value
Message type	Boot Reply (2)
Hardware type	Ethernet (0x01)
Hardware address length	6
Hops	0
Transaction ID	0x0d047979
Seconds elapsed	0
Bootp flags	0x8000, Broadcast flag (Broadcast)
Client IP address	0.0.0.0
Your (client) IP address	192.168.1.12
Next server IP address	0.0.0.0
Relay agent IP address	0.0.0.0
Client MAC address	PcsCompu_7e:42:67 (08:00:27:7e:42:67)
Client hardware address padding	00000000000000000000
Server host name	not given
Boot file name	not given
Magic cookie	DHCP
Option (53)	DHCP Message Type (ACK)
Option (54)	DHCP Server Identifier (192.168.1.1)
Option (51)	IP Address Lease Time
Option (1)	Subnet Mask (255.255.255.0)
Option (3)	Router
Option (6)	Domain Name Server

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 divide 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  $\rightarrow$  00111111 = 0  $\rightarrow$  63  
01000000  $\rightarrow$  01111111 = 64  $\rightarrow$  127  
10000000  $\rightarrow$  10111111 = 128  $\rightarrow$  191  
11000000  $\rightarrow$  11111111 = 192  $\rightarrow$  255

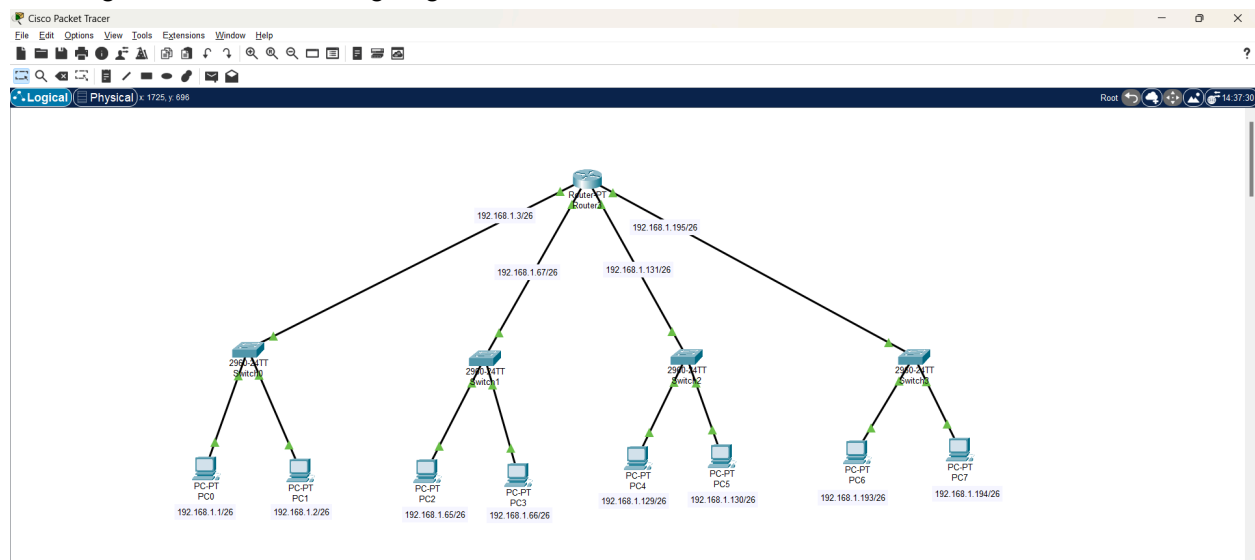
So, the 4 subnets will be :

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.

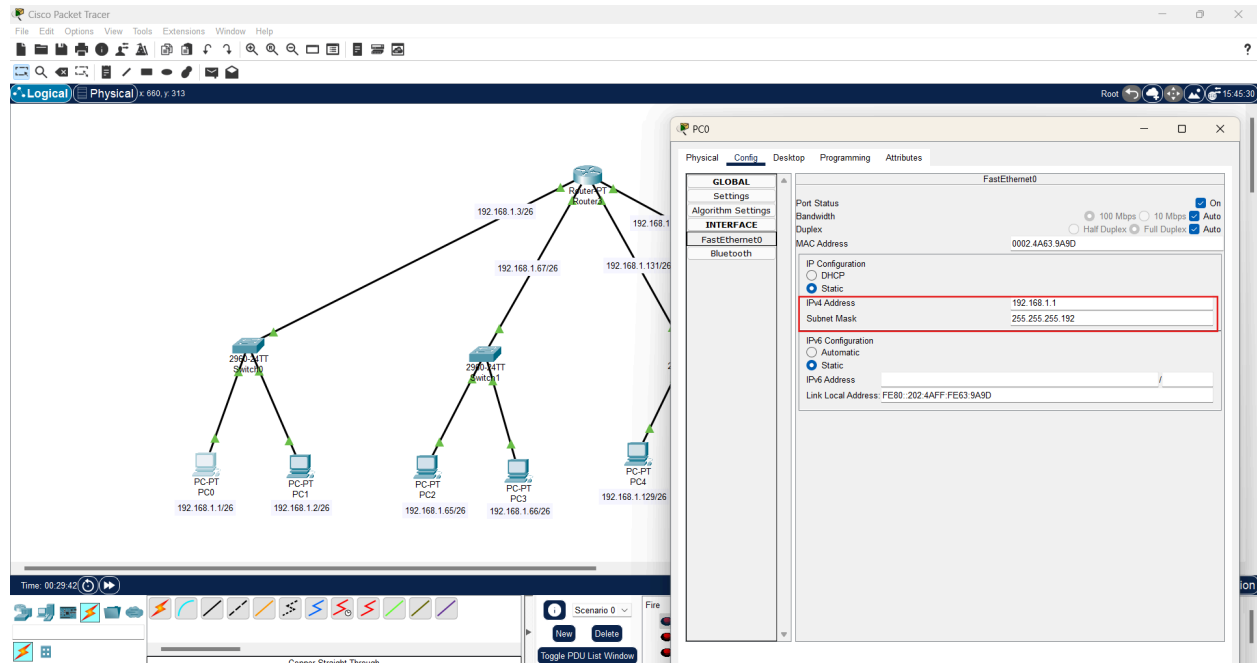
11111111.11111111.11111111.11000000 = 255.255.255.192 which is the subnet mask

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



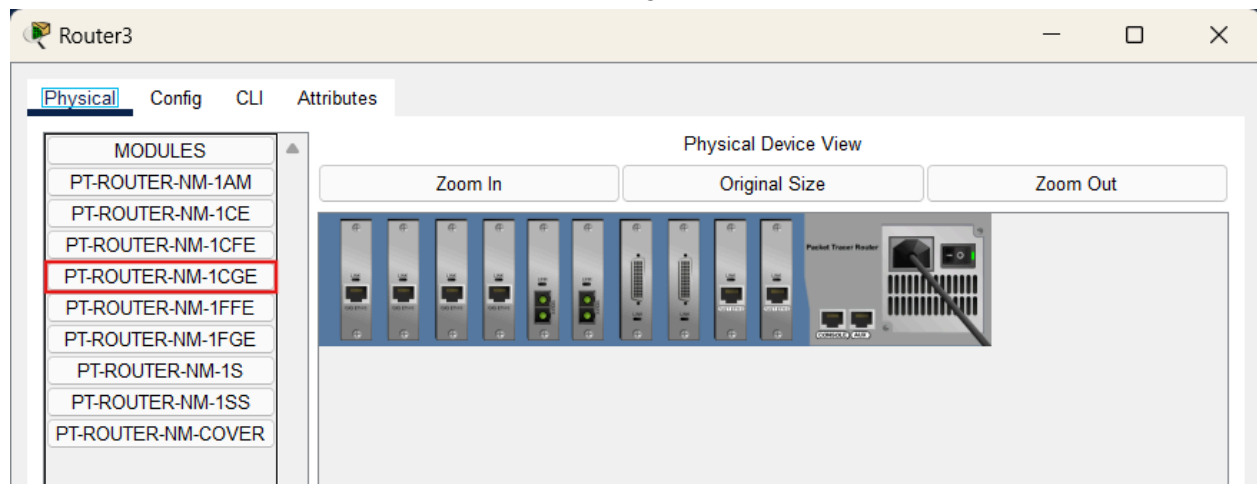
To Note : the first and last address of every subnet is reserved and hence cannot be used.

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.

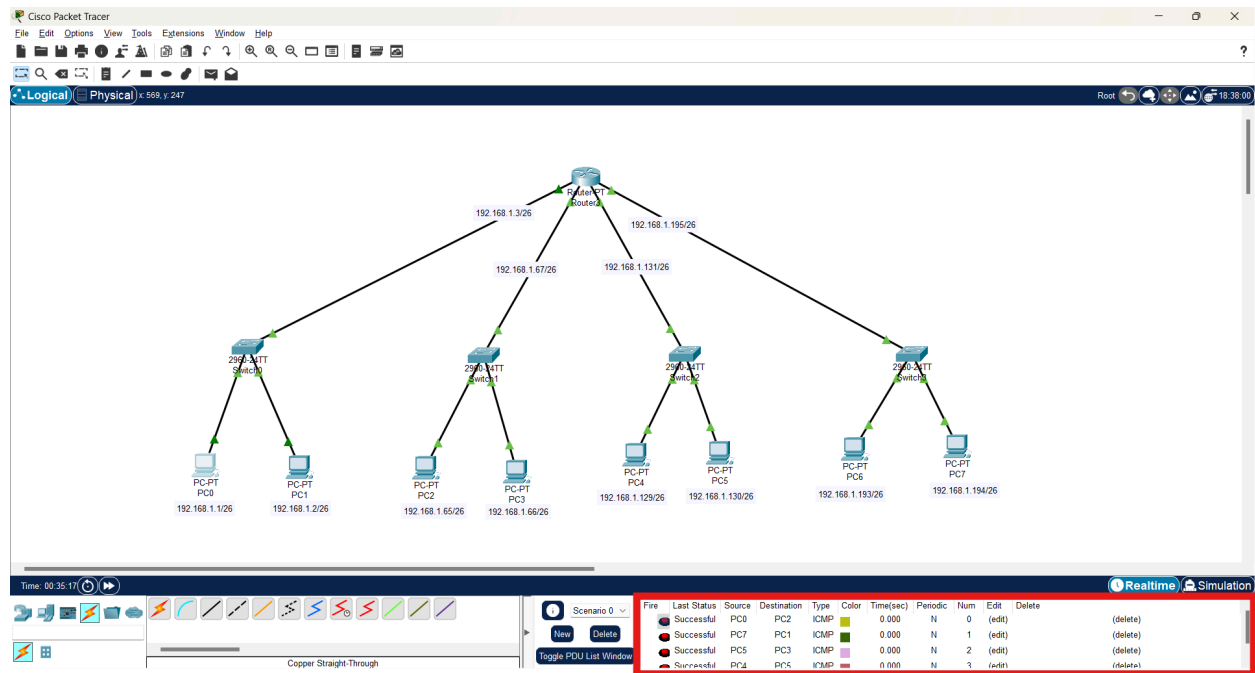
The screenshot shows the Cisco Packet Tracer interface with a network diagram and the configuration window for Router3. The network diagram shows a central router (Router3) connected to two switches (Switch0 and Switch1). Switch0 is connected to PC0 and PC1, and Switch1 is connected to PC2 and PC3. The IP addresses for the switches are 192.168.1.3/26, 192.168.1.67/26, 192.168.1.1/26, and 192.168.1.2/26. The configuration window for Router3 is open, showing the 'Config' tab. The 'INTERFACE' section is expanded, and the 'GigabitEthernet0/0' interface is selected. The 'IP Configuration' section shows the IP address 192.168.1.3 and the Subnet Mask 255.255.255.192. The 'Equivalent IOS Commands' section shows the following commands:

```
Router(config-if)#  
Router(config-if)#exit  
Router(config)#  
Router(config)#  
Router(config)#  
Router(config)#  
Router(config)#interface GigabitEthernet6/0  
Router(config-if)#  
Router(config-if)#exit  
Router(config)#interface GigabitEthernet9/0  
Router(config-if)#
```

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

The screenshot shows the Cisco Packet Tracer interface with the same network diagram as the previous image. The configuration window for PC0 is open, showing the 'Config' tab. The 'Global Settings' section is expanded, and the 'Default Gateway' field is highlighted with a red box, showing the value 192.168.1.3. The 'DNS Server' field is also visible.

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

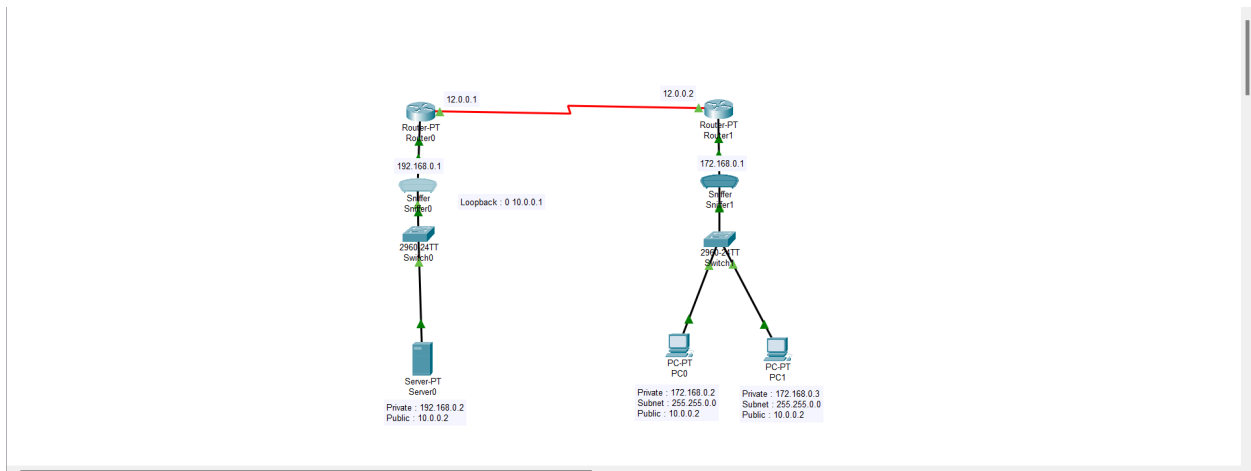


7. In Cisco Packet Tracer, create a small network with multiple devices (e.g., 2 PCs and a router). Use private IP addresses (e.g., 192.168.1.x) on the PCs and configure the router to perform NAT to allow the PCs to access the internet.

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

PC0

Physical Config Desktop Programming Attributes

IP Configuration

Interface: FastEthernet0

IP Configuration

☐ DHCP ☒ Static

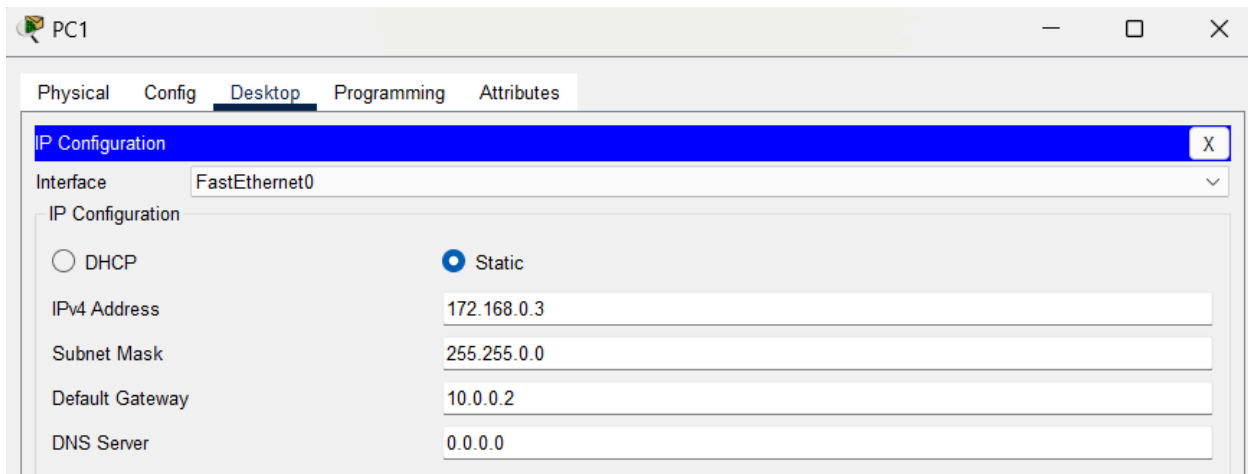
IPv4 Address: 172.168.0.2

Subnet Mask: 255.255.0.0

Default Gateway: 10.0.0.2

DNS Server: 0.0.0.0

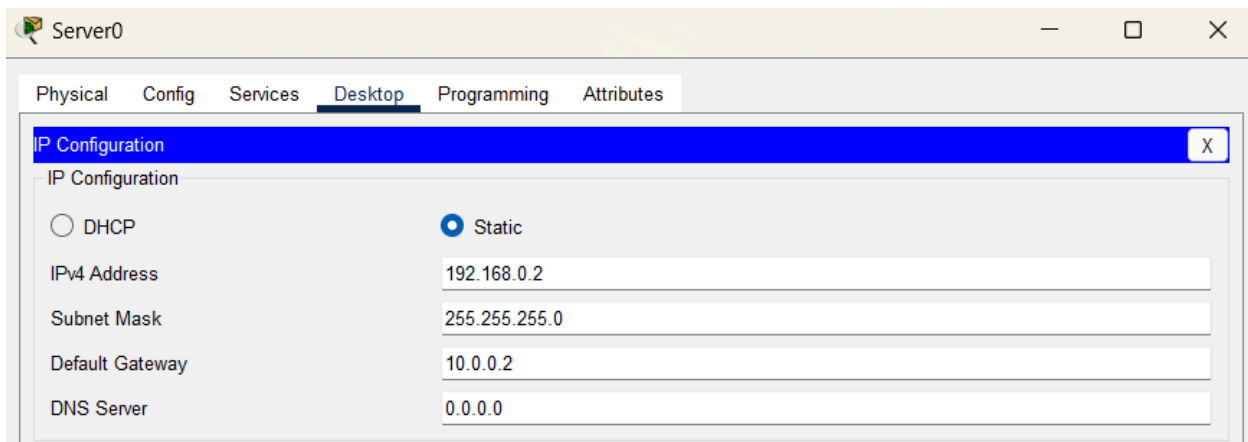
Assigning PC1 with IP, Subnet and default gateway :



The screenshot shows the 'PC1' configuration window with the 'Desktop' tab selected. The 'IP Configuration' section is highlighted. The 'Interface' is set to 'FastEthernet0'. The 'IP Configuration' section has two radio buttons: 'DHCP' (unselected) and 'Static' (selected). The 'Static' configuration fields are as follows:

Field	Value
IPv4 Address	172.168.0.3
Subnet Mask	255.255.0.0
Default Gateway	10.0.0.2
DNS Server	0.0.0.0

Assigning Server0 with IP, Subnet and default gateway



The screenshot shows the 'Server0' configuration window with the 'Desktop' tab selected. The 'IP Configuration' section is highlighted. The 'IP Configuration' section has two radio buttons: 'DHCP' (unselected) and 'Static' (selected). The 'Static' configuration fields are as follows:

Field	Value
IPv4 Address	192.168.0.2
Subnet Mask	255.255.255.0
Default Gateway	10.0.0.2
DNS Server	0.0.0.0

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 :

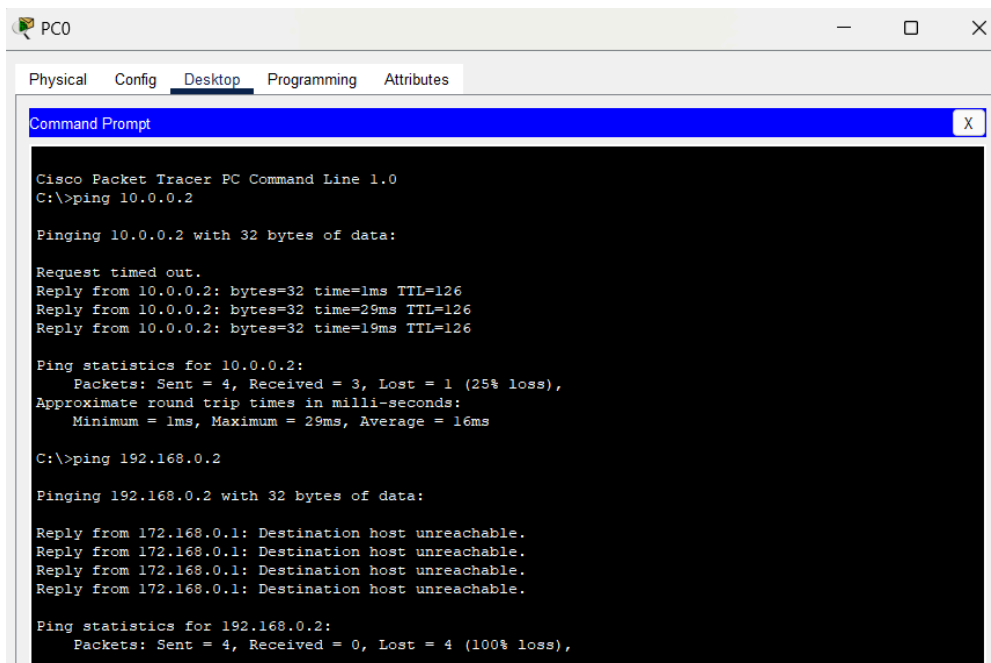
```
r2(config)#int s2/0
r2(config-if)#ip add 12.0.0.2 255.0.0.0
r2(config-if)#no shut
```

Initialising 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
```

Pinging From PC0 / PC 1 to Server :

PC 0 :



```
PC0
Physical Config Desktop Programming Attributes
Command Prompt
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

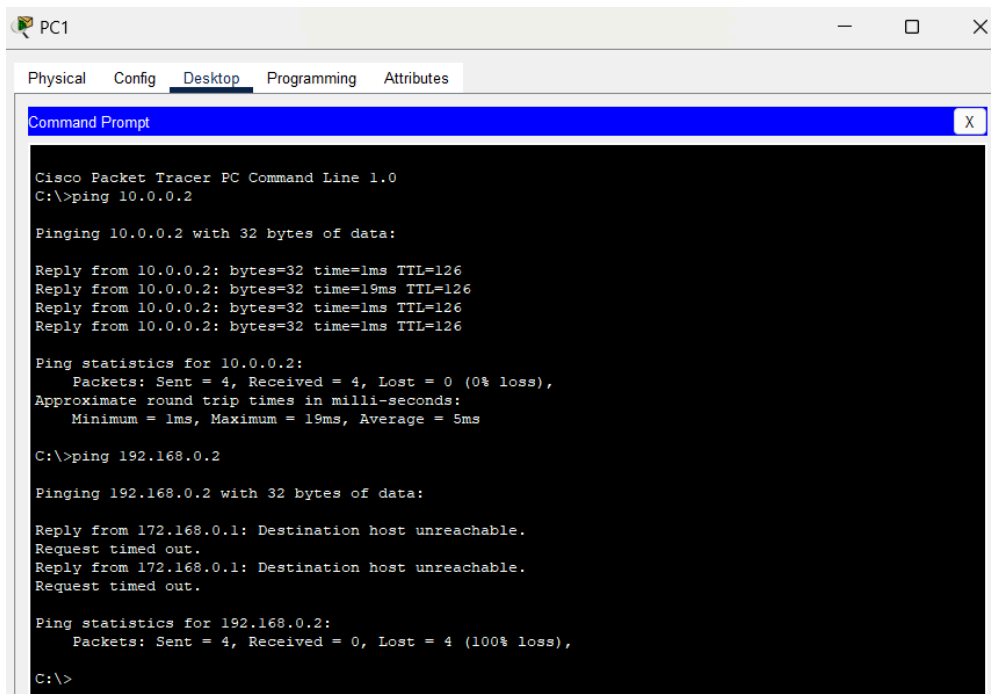
C:\>ping 192.168.0.2

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.

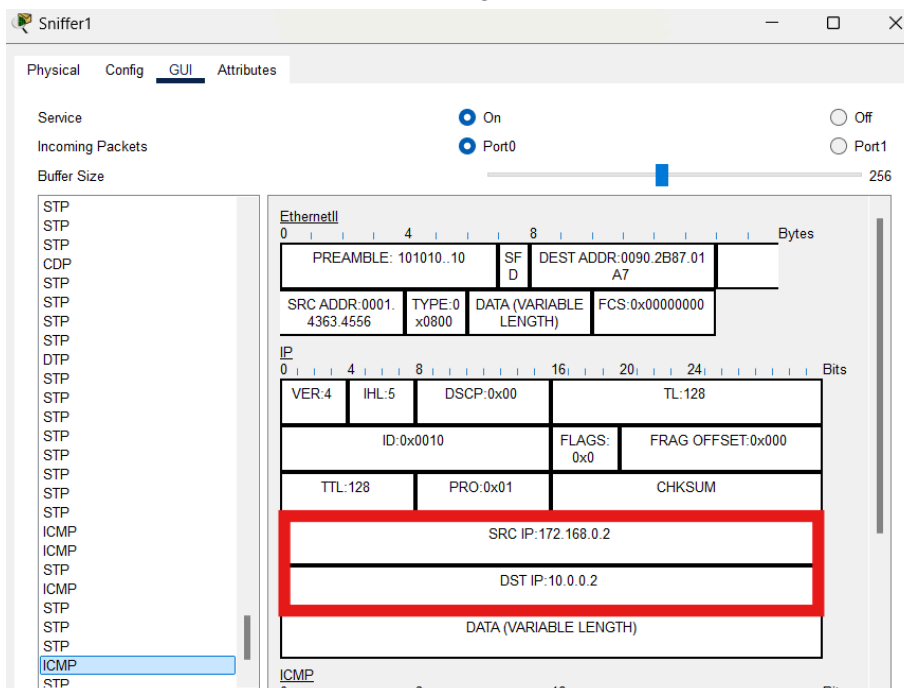
Ping statistics for 192.168.0.2:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

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

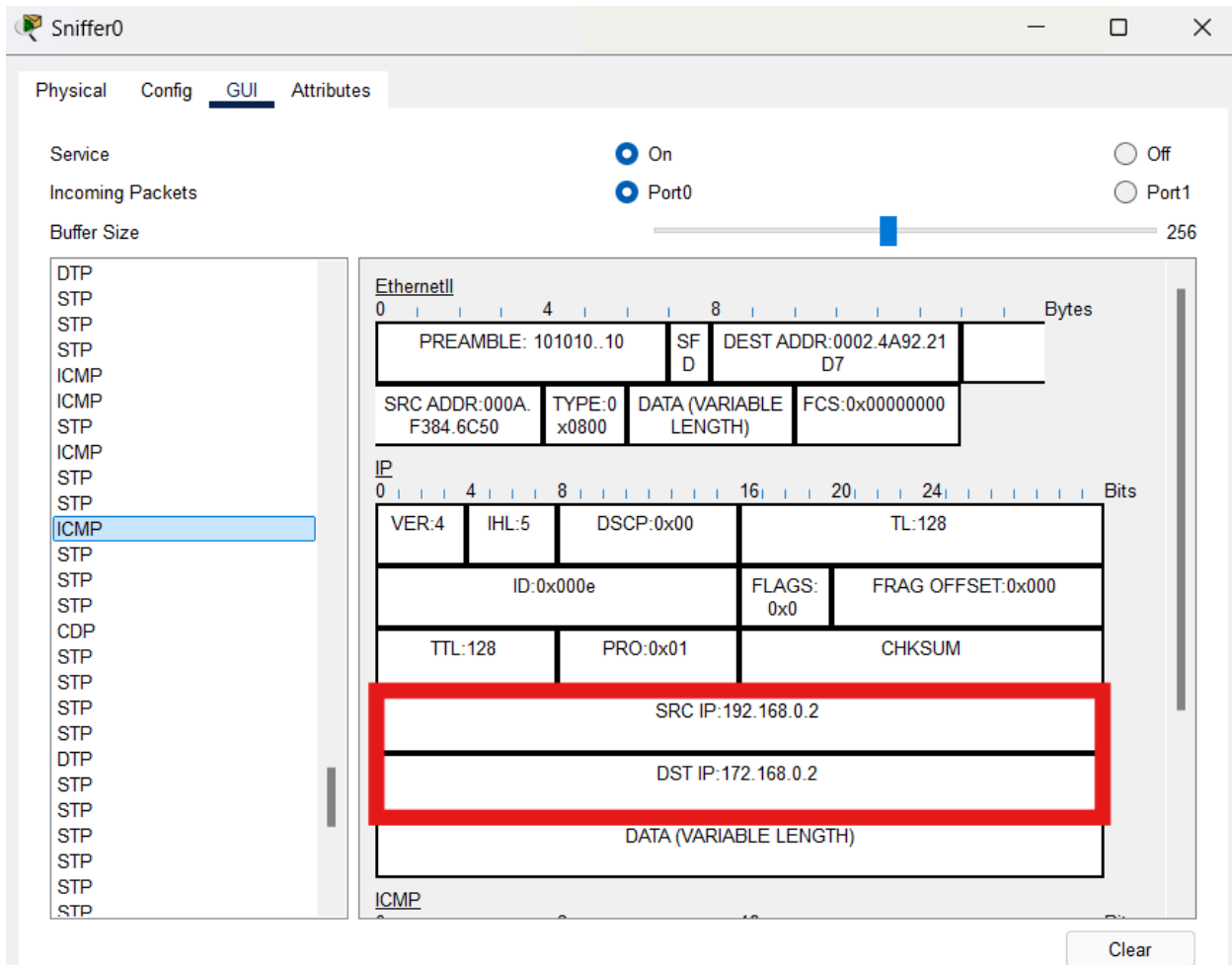
### PC Side Sniffer :

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.