Dinesh Prabhu S
Thiagarajar College of Engineering

Module 5

Q1) 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 stands for “Address Resolution Protocol”. It is used to determine the MAC address from an IP address.
- This protocol is used when a device wants to communicate with another device over a local area network.

When a sender wants to communicate, it first checks its ARP cache for the receiver's MAC address.

- If found, communication proceeds using that MAC.
- If not, the sender broadcasts an ARP request with its own MAC/IP, leaving the receiver’s MAC blank.

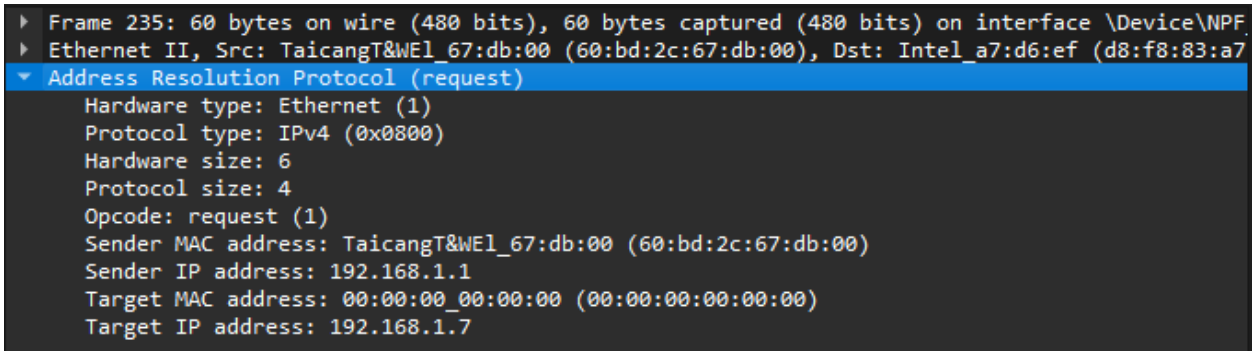
All devices on the LAN receive this request, but only the device with the matching IP responds with an ARP reply, providing its MAC address.
The sender updates its ARP cache and can now communicate directly.

ARP Request Frame contains:

- Sender MAC: The MAC address of the device sending the request.
- Sender IP: The IP address of the sender asking for the MAC address.
- Target MAC: Set to 00:00:00:00:00:00 (unknown).
- Target IP: The IP address of the destination device whose MAC is being requested.
- Broadcast MAC: Sent to ff:ff:ff:ff:ff:ff (broadcast) to reach all devices on the network.

ARP Reply Frame contains:

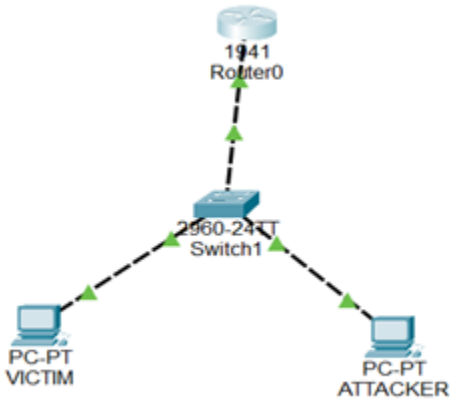
- Sender MAC: The MAC address of the responding device.
- Sender IP: The IP address of the responding device.
- Target MAC: The MAC address of the original sender.
- Target IP: The IP address of the sender who made the request.
- Unicast MAC: Sent directly to the requester’s MAC address.



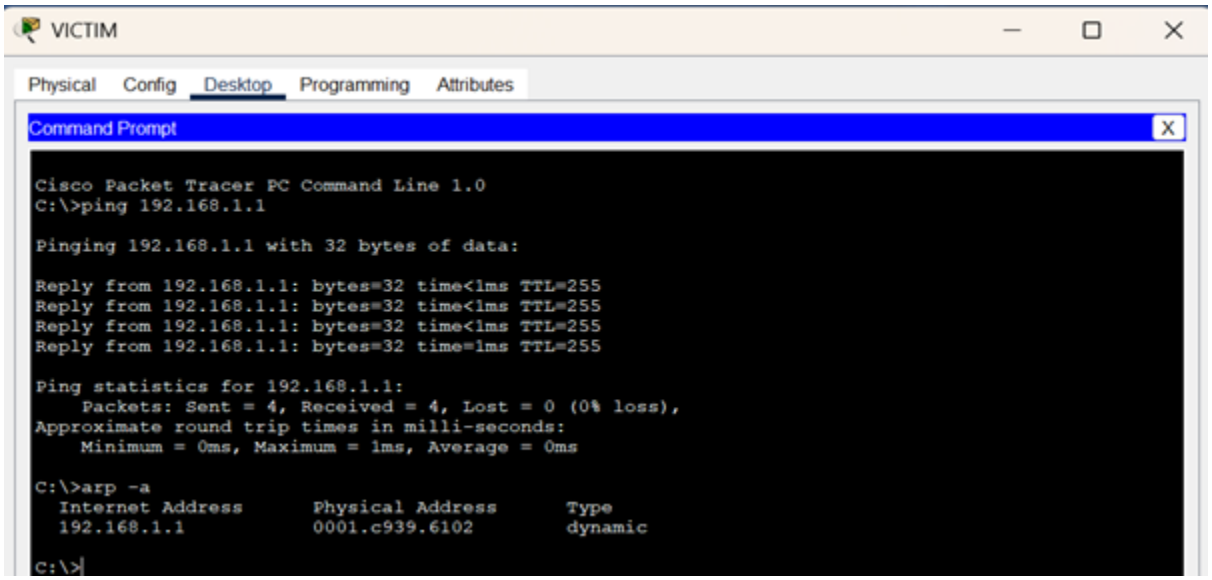
Q2) Using Packet Tracer, simulate an ARP spoofing attack. Analyze the behavior of devices on the network when they receive a malicious ARP response.

ARP spoofing is a cyber attack that allows hackers to intercept communications between network devices on a network. Hackers can also use ARP spoofing to alter or block all traffic between devices on the network.

- I have created a simple system consisting of 2 PC's connected to a switch which in turn is connected to a router.
- I have assigned a static IP to all the individual components.

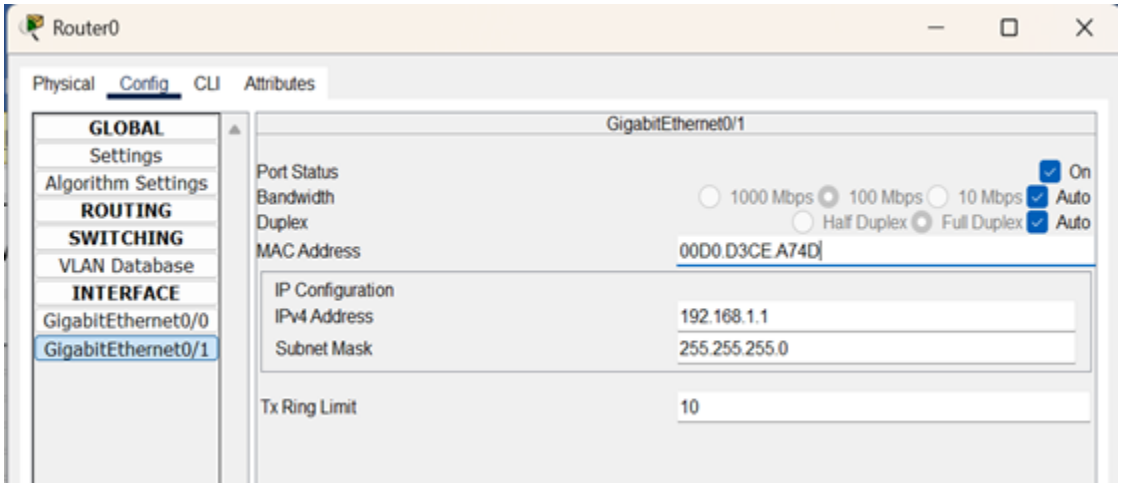


Using ping command in victim's PC to the router:

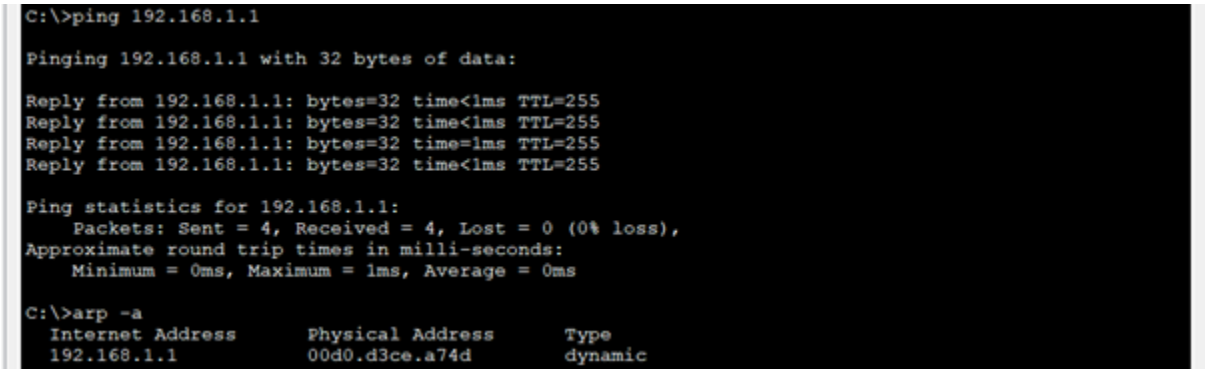


- Now change the MAC address of the router to the attacker PC.
- ping PC1 to the router.

Changing the MAC address of the router to the MAC address of the attacker :



Updation of the arp table:

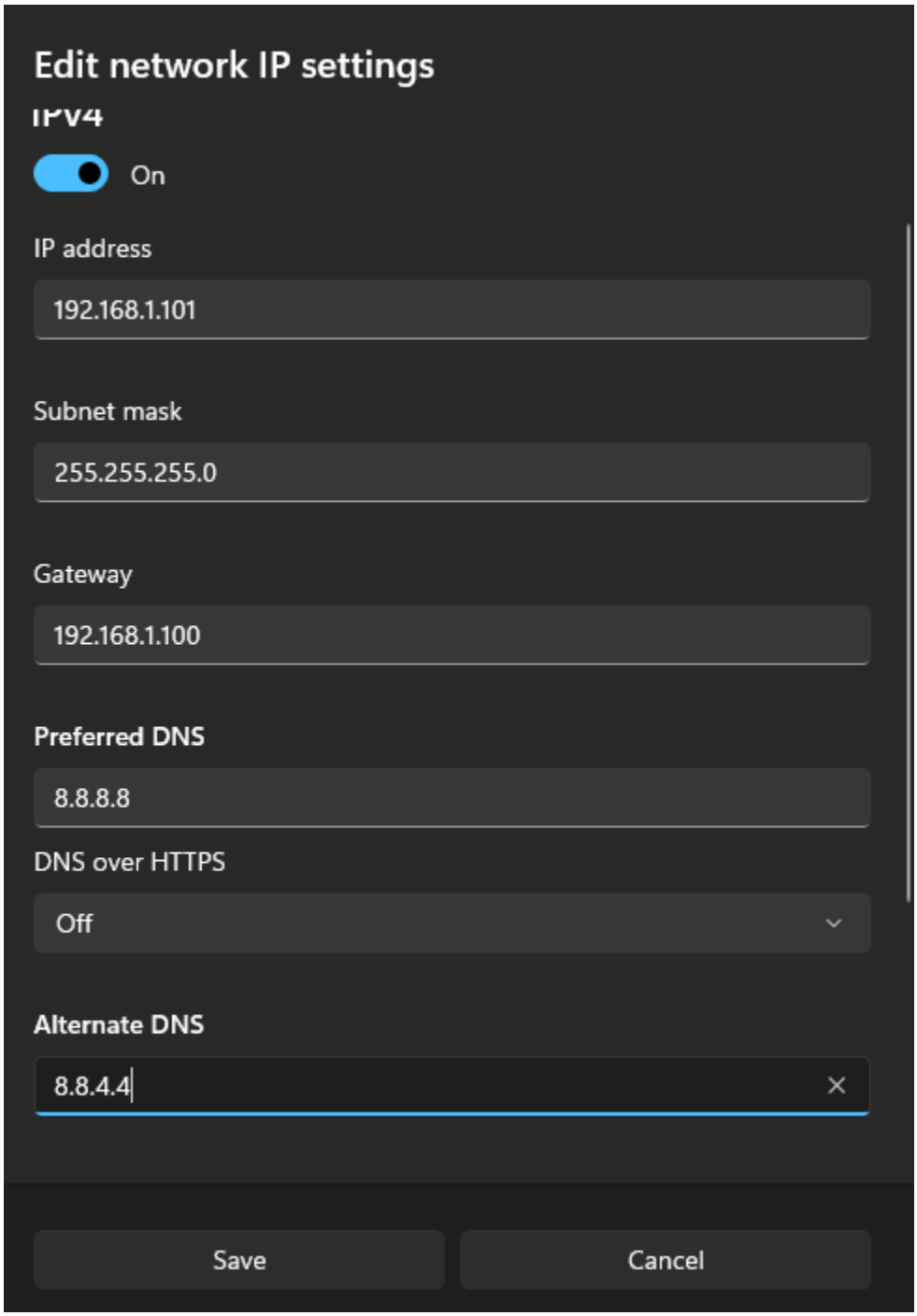


- On normal pinging, the packet transfer happens between the victim and the router.
- On modifying the MAC of the router with the attacker's MAC address, The transfer happens between the attacker PC and the victim PC.
- Victim PC assumes that the data transfer is happening between it and router but the data is transferred to a different destination causing a malware attack.

Q3) Manually configure static IPs on the client devices (like PC or your mobile phone) and verify connectivity using ping.

A static IP address is a fixed, manually assigned IP address that does not change over time. Unlike a dynamic IP address, which is assigned automatically by a DHCP server and may change periodically, a static IP remains constant.

Settings -> Network and Internet -> Wifi -> Edit



Checking with ipconfig in cmd

```
Windows IP Configuration


Ethernet adapter Ethernet:

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

Ethernet adapter Ethernet 2:

   Connection-specific DNS Suffix  . :
   Link-local IPv6 Address . . . . . : fe80::5444:5cb4:108f:4193%13
   IPv4 Address. . . . . . . . . . . : 192.168.56.1
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

Wireless LAN adapter Local Area Connection* 1:

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

Wireless LAN adapter Local Area Connection* 10:

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

Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2409:408d:3e9a:78d0:9c26:c487:43f2:5676
   Temporary IPv6 Address. . . . . . : 2409:408d:3e9a:78d0:34c3:f248:adb6:1a84
   Link-local IPv6 Address . . . . . : fe80::635e:67cd:5bcf:ea56%10
   IPv4 Address. . . . . . . . . . . : 192.168.1.101
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::6c0d:37ff:fe02:c01f%10
                                       192.168.1.100
```

Pinging the IPv4 address:

I want to check whether the network is working or not so I used the ping command to check

ping 192.168.1.101

I have got replies ensuring proper transmission without losses

```
C:\Users\admin>ping 192.168.1.101

Pinging 192.168.1.101 with 32 bytes of data:
Reply from 192.168.1.101: bytes=32 time<1ms TTL=128
Reply from 192.168.1.101: bytes=32 time<1ms TTL=128
Reply from 192.168.1.101: bytes=32 time<1ms TTL=128
Reply from 192.168.1.101: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Q4) Use Wireshark to capture DHCP Discover, Offer, Request, and Acknowledge messages and explain the process.

ipconfig /release and ifconfig /renew is used to release the existing IP and add the new IP address dynamically using DHCP.

Discover (D) – The client sends a broadcast request (DHCP Discover) asking for an available IP address.
Offer (O) – The DHCP server responds with an available IP address and other network details (DHCP Offer).
Request (R) – The client sends a DHCP Request, confirming it wants the offered IP.
Acknowledge (A) – The DHCP server sends a DHCP Acknowledge, confirming the lease and assigning the IP.

Using ipconfig /release:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2409:408d:4dc4:f395:3c12:c37d:7c96:a87c
   Temporary IPv6 Address. . . . . . : 2409:408d:4dc4:f395:4171:d251:8829:ef5b
   Link-local IPv6 Address . . . . . : fe80::635e:67cd:5bcf:ea56%10
   Default Gateway . . . . . . . . . : fe80::6c0d:37ff:fe02:c01f%10
```

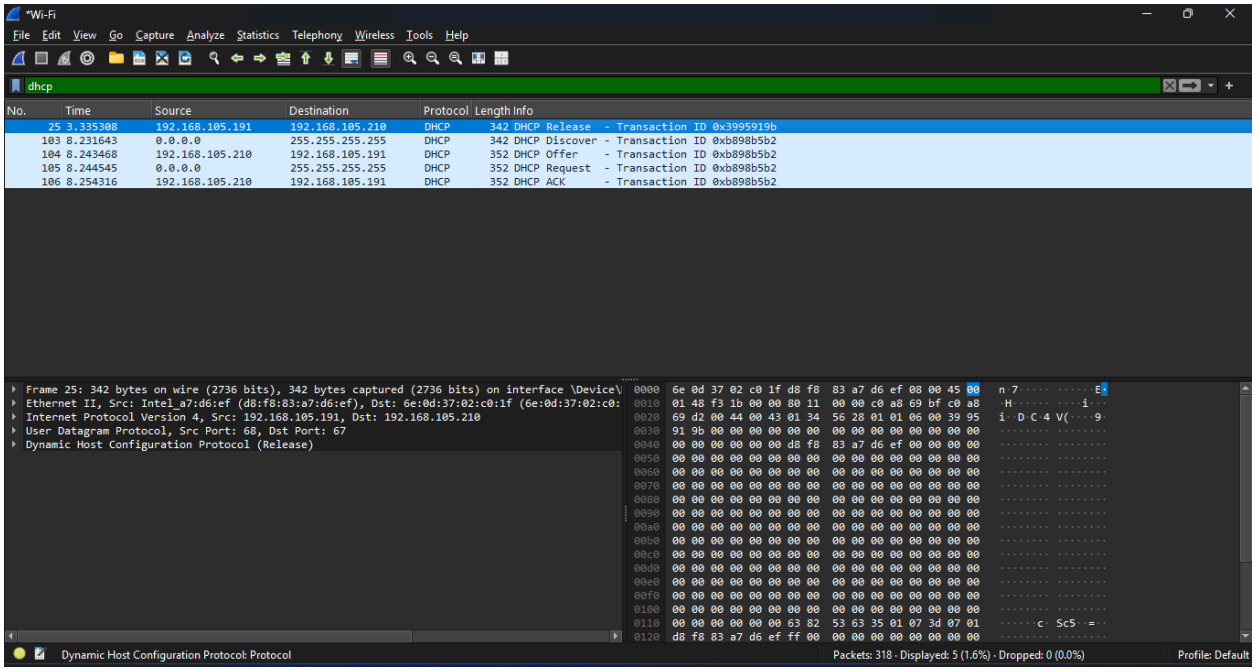
Using ipconfig /renew:

```
Wireless LAN adapter Wi-Fi:

   Connection-specific DNS Suffix  . :
   IPv6 Address. . . . . . . . . . . : 2409:408d:4dc4:f395:3c12:c37d:7c96:a87c
   Temporary IPv6 Address. . . . . . : 2409:408d:4dc4:f395:4171:d251:8829:ef5b
   Link-local IPv6 Address . . . . . : fe80::635e:67cd:5bcf:ea56%10
   IPv4 Address. . . . . . . . . . . : 192.168.105.191
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . : fe80::6c0d:37ff:fe02:c01f%10
                                       192.168.105.210
```

The Wireless LAN adapter Wi-Fi shows a change:
Before: No IPv4 address
After: Changed to 192.168.105.191
The Default Gateway is also set to 192.168.105.210

Capturing DHCP packets using wireshark:



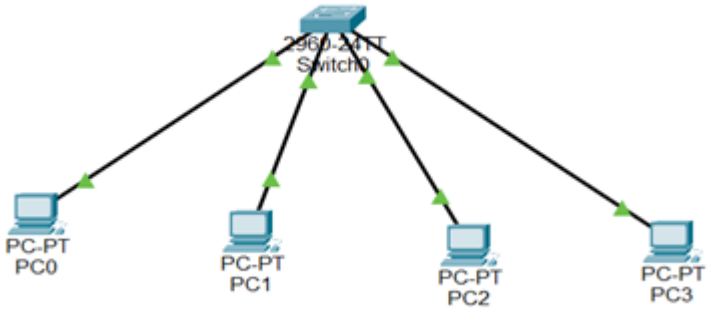
Q5) Given an IP address range of 192.168.1.0/24, divide the network into 4 subnets.
Task: Manually calculate the new subnet mask and the range of valid IP addresses for each subnet.
Assign IP addresses from these subnets to devices in Cisco Packet Tracer and verify connectivity using ping between them.

- Given Network: 192.168.1.0/24 (Subnet Mask: 255.255.255.0)
- Dividing into 4 subnets → Need 2 additional bits for subnetting (/24 → /26).
- New Subnet Mask: 255.255.255.192
- Each Subnet Contains: 2^(6) = 64 total addresses (62 usable IPs per subnet).

Ranges:

Subnet	Network Address	Usable IP Range	Broadcast Address
Subnet 1	192.168.1.0/26	192.168.1.1 – 192.168.1.62	192.168.1.63
Subnet 2	192.168.1.64/26	192.168.1.65 – 192.168.1.126	192.168.1.127
Subnet 3	192.168.1.128/26	192.168.1.129 – 192.168.1.190	192.168.1.191
Subnet 4	192.168.1.192/26	192.168.1.193 – 192.168.1.254	192.168.1.255

- Divided 192.168.1.0/24 into 4 subnets using /26 mask.
- Assigned IP addresses to PCs and configured the router.
- Used ping to test connectivity.



Pinging PC0 to all other PC's

```
C:\>ping 192.168.1.80

Pinging 192.168.1.80 with 32 bytes of data:

Reply from 192.168.1.80: bytes=32 time<1ms TTL=128
Reply from 192.168.1.80: bytes=32 time<1ms TTL=128
Reply from 192.168.1.80: bytes=32 time<1ms TTL=128
Reply from 192.168.1.80: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.80:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.130

Pinging 192.168.1.130 with 32 bytes of data:

Reply from 192.168.1.130: bytes=32 time<1ms TTL=128
Reply from 192.168.1.130: bytes=32 time<1ms TTL=128
Reply from 192.168.1.130: bytes=32 time<1ms TTL=128
Reply from 192.168.1.130: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.130:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>ping 192.168.1.200

Pinging 192.168.1.200 with 32 bytes of data:

Reply from 192.168.1.200: bytes=32 time<1ms TTL=128
Reply from 192.168.1.200: bytes=32 time<1ms TTL=128
Reply from 192.168.1.200: bytes=32 time=11ms TTL=128
Reply from 192.168.1.200: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.200:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 11ms, Average = 2ms
```

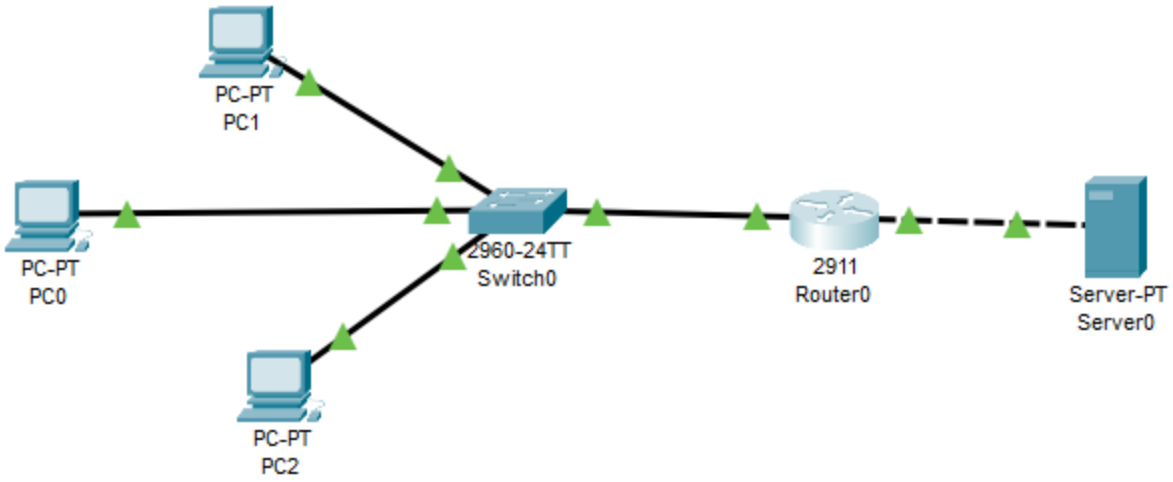
Q6) You are given three IP addresses: 10.1.1.1, 172.16.5.10, and 192.168.1.5.
Task: Identify the class of each IP address (Class A, B, or C). What is the default subnet mask for each class?
Provide the range of IP addresses for each class.

10.1.1.1
First octet: 10
Falls within the Class A range (1 - 126).
Default subnet mask: 255.0.0.0
IP range: 1.0.0.0 - 126.255.255.255

172.16.5.10
First octet: 172
Falls within the Class B range (128 - 191).
Default subnet mask: 255.255.0.0
IP range: 128.0.0.0 - 191.255.255.255

192.168.1.5
First octet: 192
Falls within the Class C range (192 - 223).
Default subnet mask: 255.255.255.0
IP range: 192.0.0.0 - 223.255.255.255

Q7) 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?




```
Router(config-if)#access-list 20 permit 10.0.0.0 0.0.0.255
Router(config)#ip nat pool cn 20.0.0.1 20.0.0.1 netmask 255.0.0.0
Router(config)#ip nat inside source list 20 pool cn overload
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#interface GigabitEthernet 0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
```

```
Router#show ip nat translation
Pro  Inside global     Inside local       Outside local      Outside global
icmp 20.0.0.1:1        10.0.0.1:1         50.0.0.2:1         50.0.0.2:1
icmp 20.0.0.1:2        10.0.0.1:2         50.0.0.2:2         50.0.0.2:2
icmp 20.0.0.1:3        10.0.0.1:3         50.0.0.2:3         50.0.0.2:3
icmp 20.0.0.1:4        10.0.0.1:4         50.0.0.2:4         50.0.0.2:4
icmp 20.0.0.1:5        10.0.0.1:5         50.0.0.2:5         50.0.0.2:5
icmp 20.0.0.1:6        10.0.0.1:6         50.0.0.2:6         50.0.0.2:6
icmp 20.0.0.1:7        10.0.0.1:7         50.0.0.2:7         50.0.0.2:7
icmp 20.0.0.1:8        10.0.0.1:8         50.0.0.2:8         50.0.0.2:8
```