Dinesh Prabhu S
Thiagarajar College of Engineering

Module 6

Q1. Capture and analyze ARP packets using Wireshark. Inspect the ARP request and reply frames when your device attempts to find the router's MAC address.
Discuss the importance of ARP in packet forwarding.

- 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.

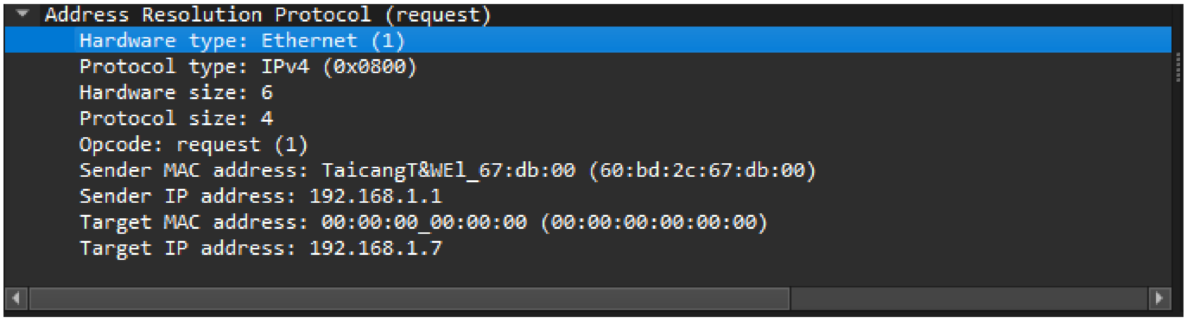
- If found, communication proceeds using that MAC.
- If not, the sender broadcasts an ARP request with its own MAC/IP, leaving the receiver’s MAC blank.

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:

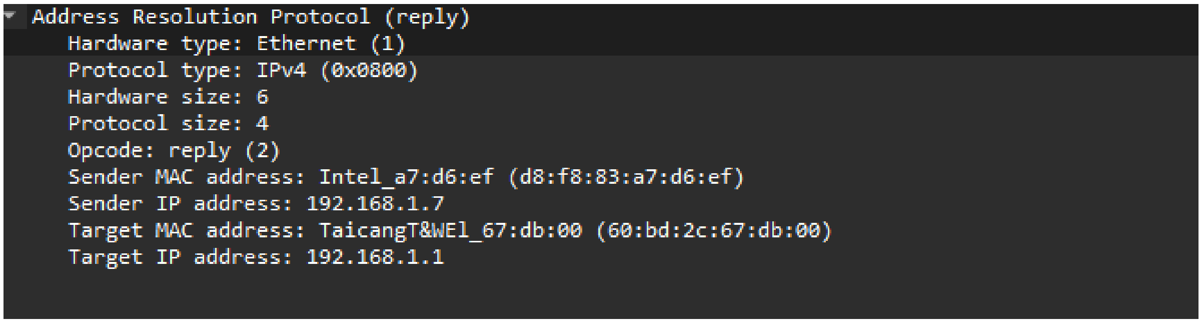
Opcode: request (1)



```
▾ Address Resolution Protocol (request)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (1)
    Sender MAC address: TaicangT&WEl_67:db:00 (60:bd:2c:67:db:00)
    Sender IP address: 192.168.1.1
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.1.7
```

ARP reply:

Opcode: reply (2)

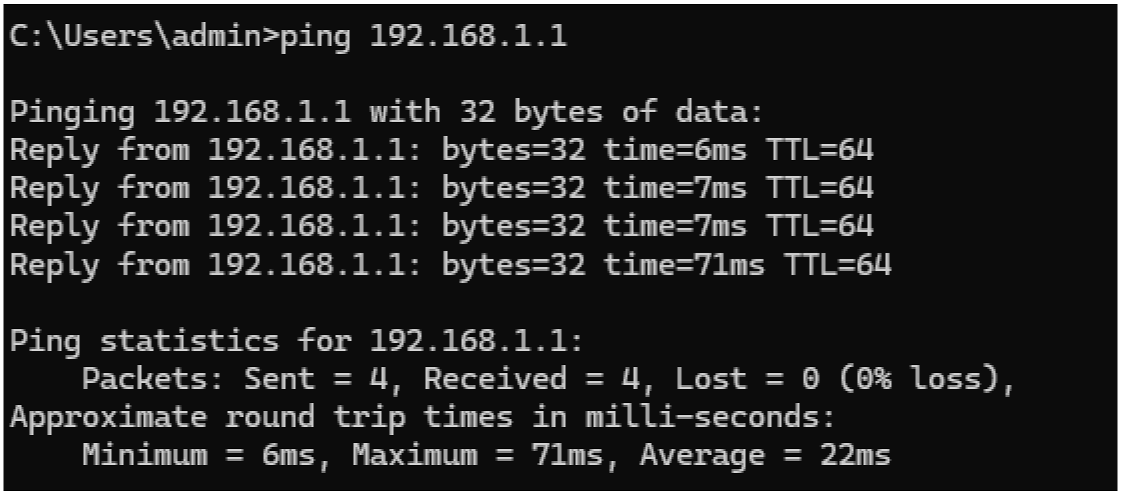


```
▾ Address Resolution Protocol (reply)
    Hardware type: Ethernet (1)
    Protocol type: IPv4 (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: reply (2)
    Sender MAC address: Intel_a7:d6:ef (d8:f8:83:a7:d6:ef)
    Sender IP address: 192.168.1.7
    Target MAC address: TaicangT&WEl_67:db:00 (60:bd:2c:67:db:00)
    Target IP address: 192.168.1.1
```

Importance of ARP in data forwarding:

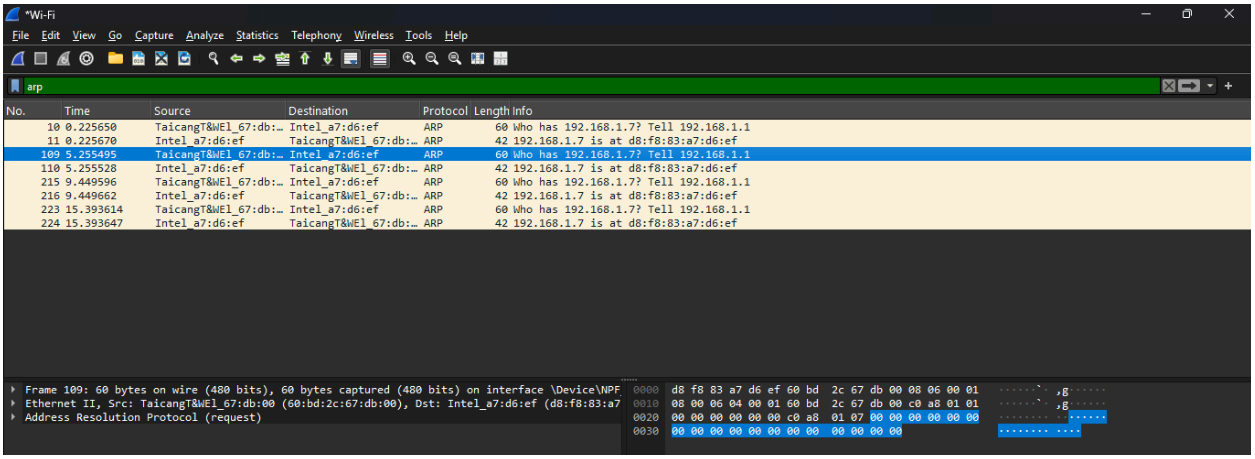
- When a device wants to send data to another device on the same network, it needs the MAC address of the recipient, not just the IP address. ARP facilitates this mapping by broadcasting a request to find the MAC address associated with a specific IP address.
- ARP resolves IP to MAC address.
- When a device wants to send packets within a LAN, ARP finds the correct MAC address for the destination,thus enabling LAN Communication.
- It stores the MAC address in arp cache to reduce the network traffic.
- Without ARP, devices couldn’t communicate in Ethernet-based LANs.
- ARP dynamically updates MAC addresses using ARP requests and replies.

Using ping command with 192.168.1.1:



```
C:\Users\admin>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=6ms TTL=64
Reply from 192.168.1.1: bytes=32 time=7ms TTL=64
Reply from 192.168.1.1: bytes=32 time=7ms TTL=64
Reply from 192.168.1.1: bytes=32 time=71ms TTL=64

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 6ms, Maximum = 71ms, Average = 22ms
```

Filtering the arp packets:



*Wi-Fi
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

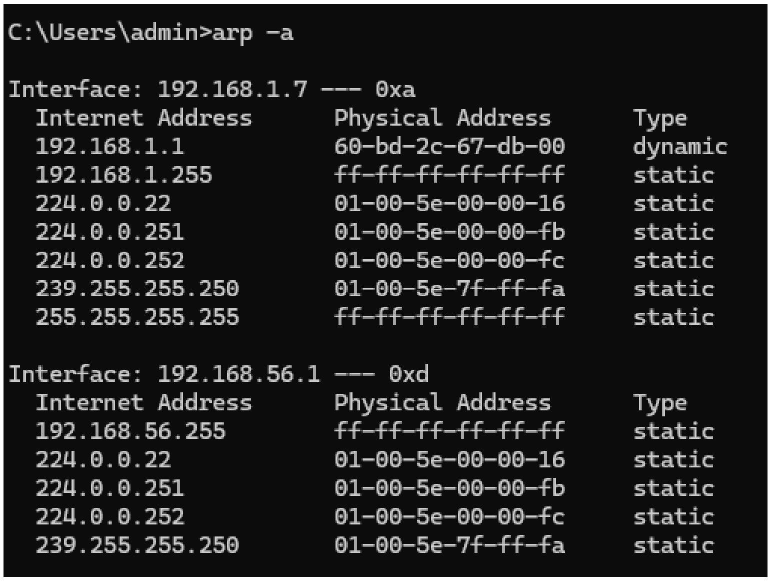
arp

```
No.   Time       Source                Destination           Protocol Length Info
   10 0.225650   TaicangT&WEl_67:db:…  Intel_a7:d6:ef        ARP        60 Who has 192.168.1.7? Tell 192.168.1.1
   11 0.225670   Intel_a7:d6:ef        TaicangT&WEl_67:db:…  ARP        42 192.168.1.7 is at d8:f8:83:a7:d6:ef
  109 5.255495   TaicangT&WEl_67:db:…  Intel_a7:d6:ef        ARP        60 Who has 192.168.1.7? Tell 192.168.1.1
  110 5.255528   Intel_a7:d6:ef        TaicangT&WEl_67:db:…  ARP        42 192.168.1.7 is at d8:f8:83:a7:d6:ef
  215 9.449596   TaicangT&WEl_67:db:…  Intel_a7:d6:ef        ARP        60 Who has 192.168.1.7? Tell 192.168.1.1
  216 9.449662   Intel_a7:d6:ef        TaicangT&WEl_67:db:…  ARP        42 192.168.1.7 is at d8:f8:83:a7:d6:ef
  223 15.393614  TaicangT&WEl_67:db:…  Intel_a7:d6:ef        ARP        60 Who has 192.168.1.7? Tell 192.168.1.1
  224 15.393647  Intel_a7:d6:ef        TaicangT&WEl_67:db:…  ARP        42 192.168.1.7 is at d8:f8:83:a7:d6:ef
```

```
> Frame 109: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interface \Device\NPF
> Ethernet II, Src: TaicangT&WEl_67:db:00 (60:bd:2c:67:db:00), Dst: Intel_a7:d6:ef (d8:f8:83:a7
> Address Resolution Protocol (request)
```

```
0000  d8 f8 83 a7 d6 ef 60 bd  2c 67 db 00 08 06 00 01   ······`· ,g······
0010  08 00 06 04 00 01 60 bd  2c 67 db 00 c0 a8 01 01   ······`· ,g······
0020  00 00 00 00 00 00 c0 a8  01 07 00 00 00 00 00 00   ········ ········
0030  00 00 00 00 00 00 00 00  00 00 00 00               ········ ····
```

ARP Table:



```
C:\Users\admin>arp -a

Interface: 192.168.1.7 --- 0xa
  Internet Address      Physical Address      Type
  192.168.1.1           60-bd-2c-67-db-00     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

Interface: 192.168.56.1 --- 0xd
  Internet Address      Physical Address      Type
  192.168.56.255        ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.251           01-00-5e-00-00-fb     static
  224.0.0.252           01-00-5e-00-00-fc     static
  239.255.255.250       01-00-5e-7f-ff-fa     static
```

Q2. Manually configure static routes on a router to direct packets to different subnets.
Use the ip route command and verify connectivity using ping and traceroute.

Routing is a process that shows how data packets are forwarded between different networks and devices. Routers are devices that perform routing.

Here I used Static routing since it is easy to perform in simple procedures .In static routing, I explicitly defined the path that data packets should take from one network to another.

I configured the router 5 and router 6 using the commands:
Router 0:

```
enable
configure terminal
interface GigabitEthernet0/0
ip address 192.168.1.1 255.255.255.0
no shutdown
exit
```

Router 1:

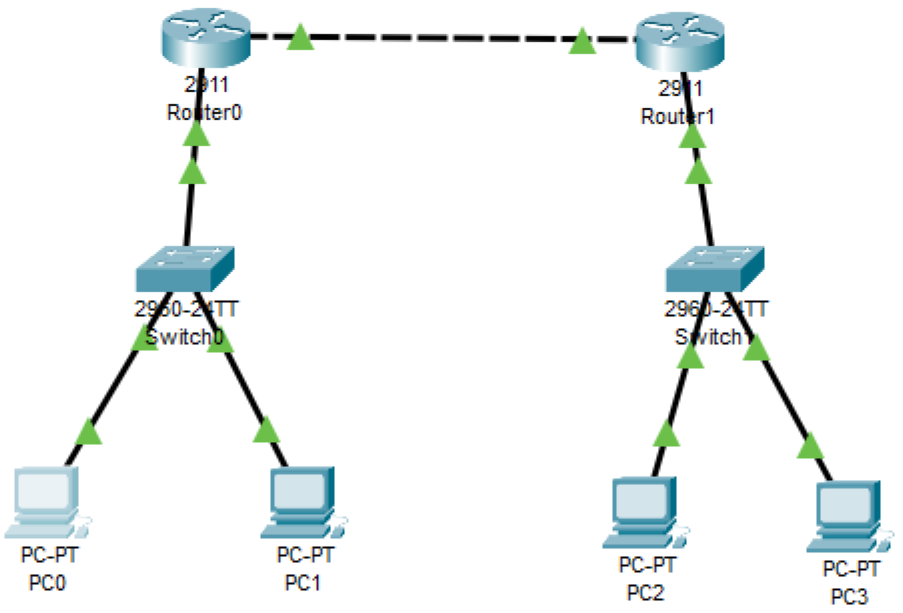
```
enable
configure terminal
interface GigabitEthernet0/0
ip address 192.168.2.1 255.255.255.0
no shutdown
```

```
exit
```

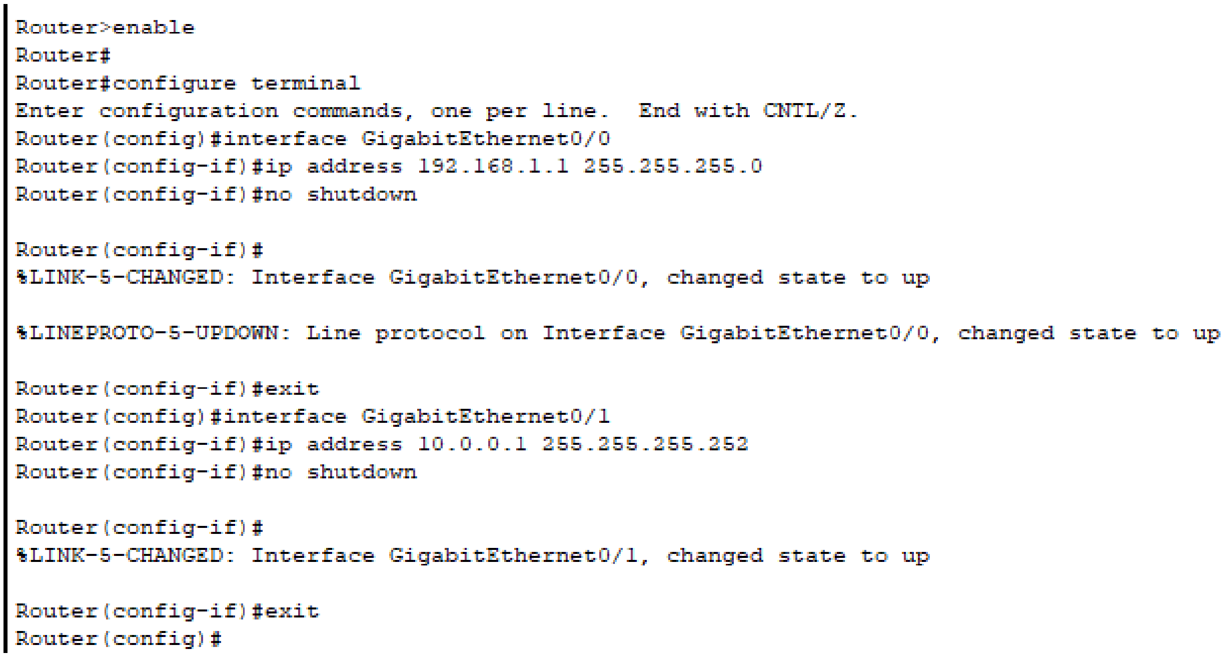
- I used ip route command in Router0 to configure Static Route to 192.168.2.0/24 via Router1:
  ```
  ip route 192.168.2.0 255.255.255.0 10.0.0.2.
  ```
- I used ip route command in Router1 to configure Static Route to 192.168.1.0/24 via Router0:
  ```
  ip route 192.168.1.0 255.255.255.0 10.0.0.1
  ```

PC0
IP Address: 192.168.1.10
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.1
PC1
IP Address: 192.168.1.11
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.1
PC2
IP Address: 192.168.2.10
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.2.1
PC3
IP Address: 192.168.2.11
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.2.1

Network setup:

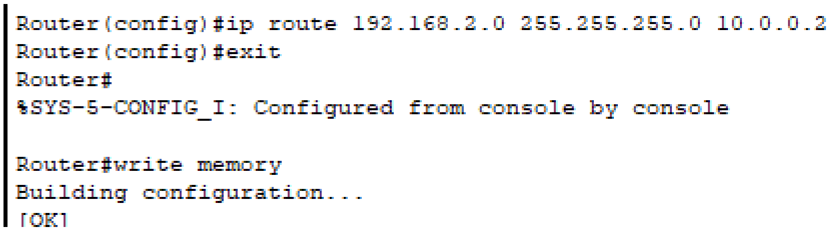


Configuring Router0:



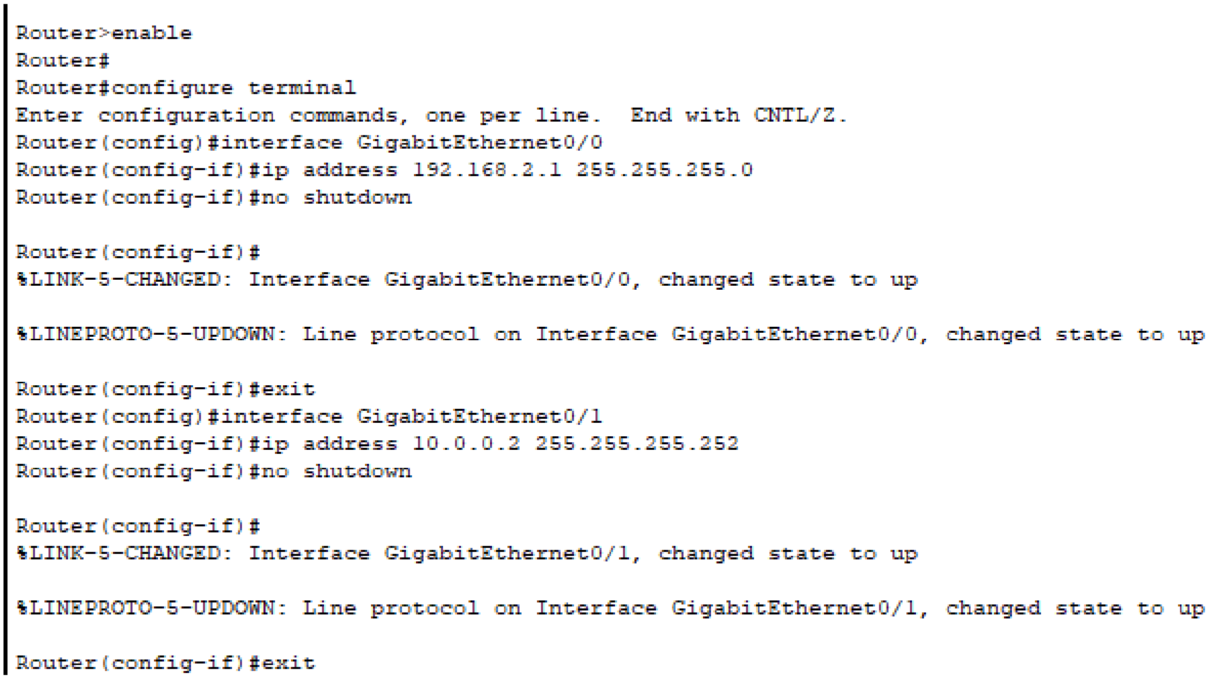
```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip address 10.0.0.1 255.255.255.252
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
Router(config)#
```

Configure Static Route to 192.168.2.0/24 via Router0:



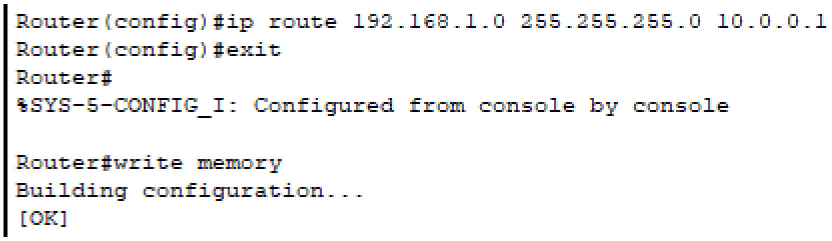
```
Router(config)#ip route 192.168.2.0 255.255.255.0 10.0.0.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#write memory
Building configuration...
[OK]
```

Configuring Router1:



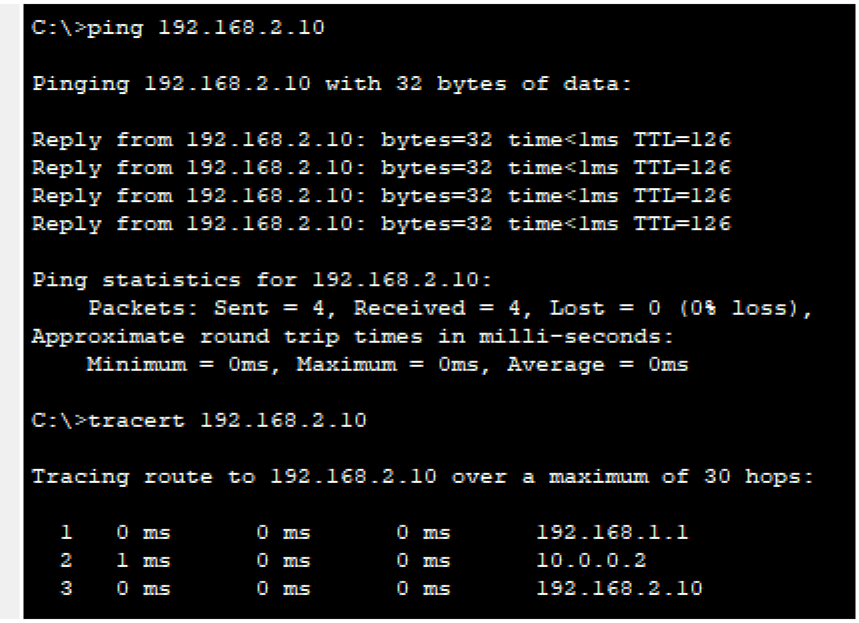
```
Router>enable
Router#
Router#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface GigabitEthernet0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/0, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/0, changed state to up

Router(config-if)#exit
Router(config)#interface GigabitEthernet0/1
Router(config-if)#ip address 10.0.0.2 255.255.255.252
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface GigabitEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface GigabitEthernet0/1, changed state to up

Router(config-if)#exit
```

Configure Static Route to 192.168.1.0/24 via Router0:



```
Router(config)#ip route 192.168.1.0 255.255.255.0 10.0.0.1
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#write memory
Building configuration...
[OK]
```

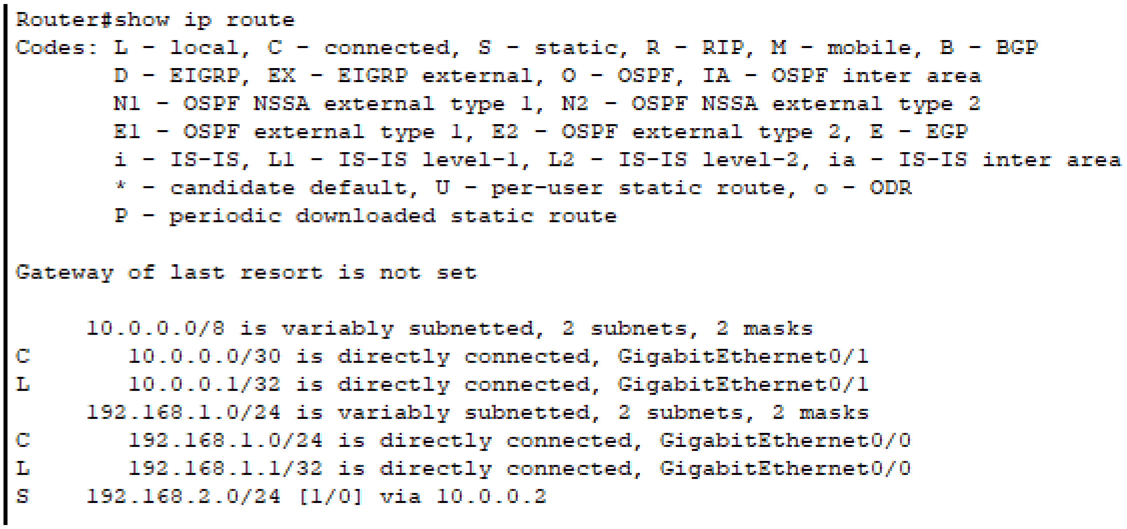
From PC0, pinging to PC2 and checking the path using tracert:



```
C:\>ping 192.168.2.10

Pinging 192.168.2.10 with 32 bytes of data:

Reply from 192.168.2.10: bytes=32 time<1ms TTL=126
Reply from 192.168.2.10: bytes=32 time<1ms TTL=126
Reply from 192.168.2.10: bytes=32 time<1ms TTL=126
Reply from 192.168.2.10: bytes=32 time<1ms TTL=126

Ping statistics for 192.168.2.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>tracert 192.168.2.10

Tracing route to 192.168.2.10 over a maximum of 30 hops:

  1   0 ms      0 ms      0 ms      192.168.1.1
  2   1 ms      0 ms      0 ms      10.0.0.2
  3   0 ms      0 ms      0 ms      192.168.2.10
```

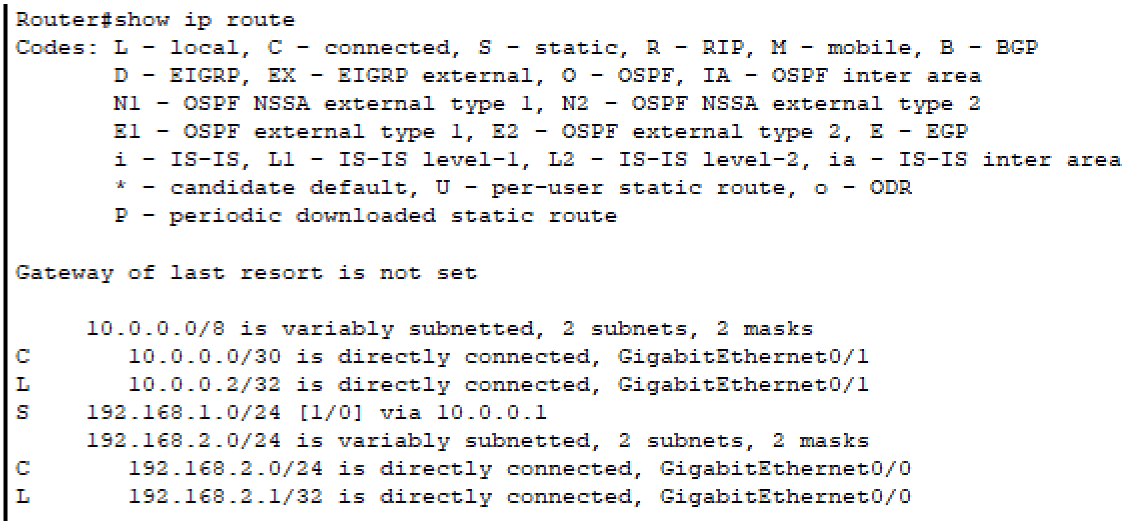
- 192.168.1.1 (Router0) → The default gateway for PC0
- 10.0.0.2 (Router1) → Router0 forwards the packet to Router1 via the link between routers.
- 192.168.2.10 (PC2) → Router1 sees that the destination is on its directly connected network (192.168.2.0/24) and forwards it to PC2.

On Router0, checking the routing table:



```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/30 is directly connected, GigabitEthernet0/1
L       10.0.0.1/32 is directly connected, GigabitEthernet0/1
     192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.1.0/24 is directly connected, GigabitEthernet0/0
L       192.168.1.1/32 is directly connected, GigabitEthernet0/0
S    192.168.2.0/24 [1/0] via 10.0.0.2
```

On Router1, checking the routing table:



```
Router#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.0.0.0/30 is directly connected, GigabitEthernet0/1
L       10.0.0.2/32 is directly connected, GigabitEthernet0/1
S    192.168.1.0/24 [1/0] via 10.0.0.1
     192.168.2.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.2.0/24 is directly connected, GigabitEthernet0/0
L       192.168.2.1/32 is directly connected, GigabitEthernet0/0
```

Q3. Given a network address of 10.0.0.0/24, divide it into 4 equal subnets.

Calculate the new subnet mask.
Determine the valid host range for each subnet.
Assign IP addresses to devices in Packet Tracer and verify connectivity.

Given network: 10.0.0.0/24
Subnet mask: 255.255.255.0
Binary representation : 11111111.11111111.11111111.00000000
Here, the first 24 bits are for the network and the last 4 bits are assigned for the hosts.

In order to divide 4 equal subnets ,We need 2 additional bits which can be taken from the host position.thus it is now converted into 10.0.0.0/26
we require 2^2 = 4 subnets.
Thus ,the new subnet is :
Binary: 11111111.11111111.11111111.11000000
Decimal: 255.255.255.192
For each subnet:
Subnet size: 2^(6) = 2^6 = 64 IPs
Usable hosts per subnet: 64 - 2 = 62 hosts.

Subnet 1
N/w address:10.0.0.0/26
First host address:10.0.0.1
Last host address:10.0.0.62
Broadcast address:10.0.0.63
Subnet 2
N/w address:10.0.0.64/26

First host address:10.0.0.65
Last host address:10.0.0.126
Broadcast address:10.0.0.127
Subnet 3
N/w address:10.0.0.128/26
First host address:10.0.0.129
Last host address:10.0.0.190
Broadcast address:10.0.0.191
Subnet 4
N/w address:10.0.0.192/26
First host address:10.0.0.193
Last host address:10.0.0.254
Broadcast address:10.0.0.255

Q4. You are given three IP addresses: 192.168.10.5, 172.20.15.1, and 8.8.8.8. Identify the class of each IP address. Determine if it is private or public. Explain how NAT would handle a private IP when accessing the internet.

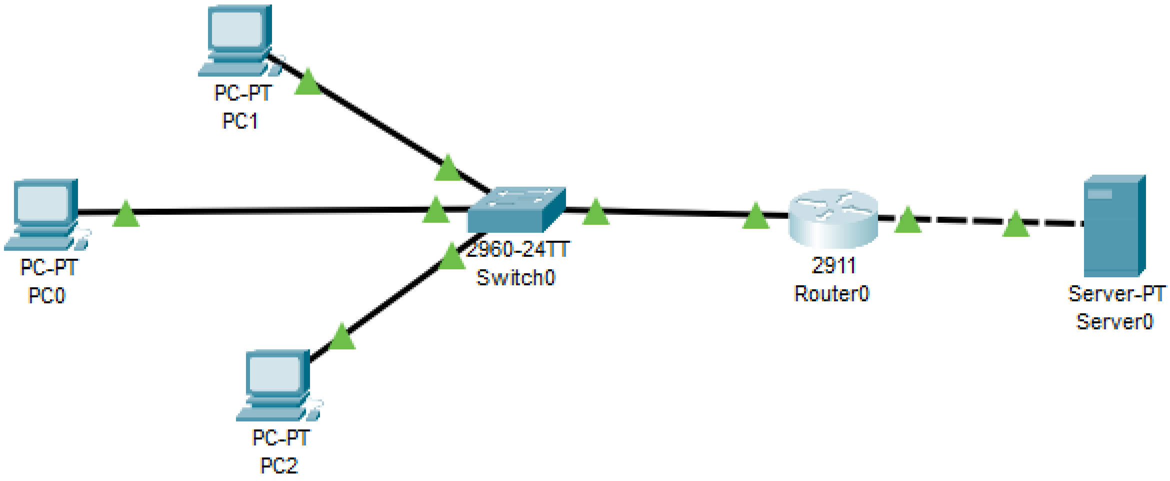
192.168.10.5
Class: Class C
Public/Private: Private
Description: Falls within 192.168.0.0 - 192.168.255.255
172.20.15.1
Class:Class B
Public/Private: Private
Description: Falls within 172.16.0.0 - 172.31.255.255
8.8.8.8
Class: Class A
Public/Private: Public
Description : Does NOT fall in any private range

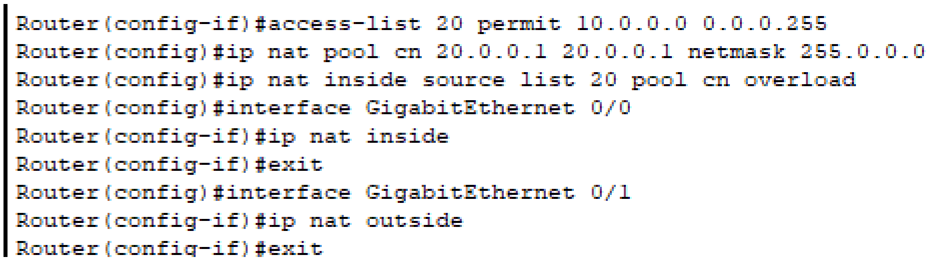
- Private IP addresses are designated for local networks and cannot directly access the internet.
- Network Address Translation (NAT) enables devices with private IPs to connect to the internet by converting them into a public IP address.

Steps:

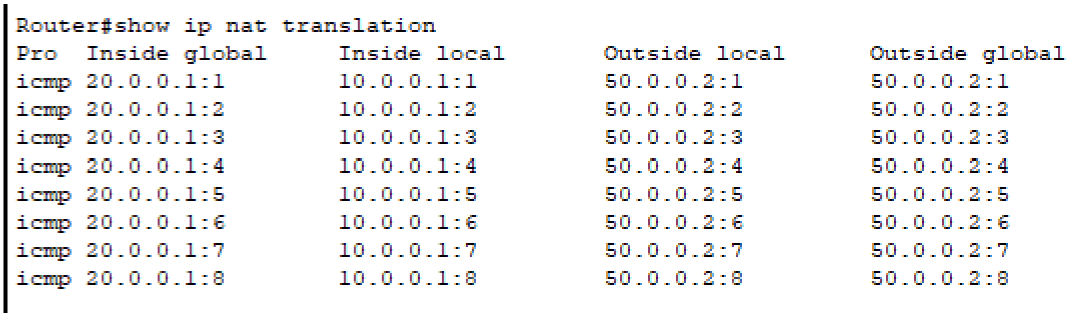
- A device with a private IP attempts to access an external website, such as 8.8.8.8
- The router, configured with NAT, swaps the private IP with its own public IP.
- The request is then sent over the internet to the destination IP.
- When the response comes back, it is directed to the router's public IP.
- NAT translates the public IP back to the original private IP and forwards the response to the correct device.

Q5. In Cisco Packet Tracer, configure NAT on a router to allow internal devices (192.168.1.x) to access the internet.
Test connectivity by pinging an external public IP.
Capture the traffic in Wireshark and analyze the source IP before and after NAT translation.





```
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
```