UCCN2243 Internetworking Principles and Practices
FEB 2025

ASSIGNMENT REPORT

No.	Student ID	Name	Programme	Practical Group
1	2301973	Richard Ting Li Zeng	CS	P13
2	2205631	HENG XING LING	CS	P13
3	2207932	Khor Zhe Khai	CS	P13
4	1901991	Ting Tieng Wei	CS	P13

INDEX

Q1 Assignment

PART A: Show the running configurations of Router0 and Router1. Highlight the relevant parts and provide a brief explanation.

ROUTER 0 Configuration

```
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.254 255.255.255.0
Router(config-if)#ip nat inside
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

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

Router(config-if)#interface FastEthernet0/1
Router(config-if)#ip address 201.1.1.1 255.255.255.0
Router(config-if)#ip nat outside
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

Router(config-if)#access-list 1 permit 192.168.1.0 0.0.0.255
Router(config)#ip nat inside source list 1 interface FastEthernet0/1
overload
Router(config)#interface FastEthernet0/0
Router(config-if)#ip helper-address 201.1.1.2
```

Figure 1.1 *Interface configuration on router 0*

```
Router(config-if)#ip route 0.0.0.0 0.0.0.0 201.1.1.2
Router(config)#write memory
                ^
% Invalid input detected at '^' marker.

Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
write memory
Building configuration...
[OK]
Router#
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
```

Figure 1.2 *set the ip route*

Explanation:

In Figure 1.1 shows interface Fa0/0 was configured inside the NAT interface with **192.168.1.0/24** as the final valid IP address (254). Interface Fa0/1 was configured as an external NAT interface using the public IP address **201.1.1.1**. NAT overload (PAT) is configured for the internal network **192.168.1.0/24**. In

Figure 1.2 the default route is set to **201.1.1.2** for the following hop. The IP helper address is added to relay DHCP queries to the remote server. The commands "**end**" and "**write memory**" are to save the configuration.

ROUTER 1 Configuration

```
Router>en
Router#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.2.254 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up

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

Router(config-if)#interface FastEthernet0/1
Router(config-if)#ip address 201.1.1.2 255.255.255.0
Router(config-if)#no shutdown

Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to up

%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
```

Figure 2.1 *interface configuration on router 1*

```
Router(config-if)#ip dhcp excluded-address 192.168.1.254
Router(config)#ip dhcp pool COMPANY_AA_POOL
Router(dhcp-config)# network 192.168.1.0 255.255.255.0
Router(dhcp-config)# default-router 192.168.1.254
Router(dhcp-config)#dns-server 8.8.8.8
Router(dhcp-config)#ex
Router(config)#ip route 0.0.0.0 0.0.0.0 201.1.1.1
Router(config)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
write memory
Building configuration...
[OK]
```

Figure 2.2 *DHCP Configuration on router 1*

Explanation:

In figure 2.1 shows interface Fa0/0 is set as the DHCP server interface, using **192.168.2.0/24** as the last valid IP (254). Interface Fa0/1 was configured with public IP **201.1.1.2**. Set the default route back to Router0. In figure 2.2 shows configured a DHCP pool for the Company AA network with the correct default gateway. The commands "**end**" and "**write memory**" are to save the configuration.

Verification Commands:

On Router0:

1) **Check NAT/PAT and helper-address**

```
Router#show running-config
Building configuration...

Current configuration : 792 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
!
!
!
!
ip cef
no ipv6 cef
!
!
 --More--
```

Figure 3 *Router 0 running configuration*

2) Check the PAT is working

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside
global
udp 201.1.1.1:68       192.168.1.254:68   201.1.1.2:67
201.1.1.2:67
```

Figure 4 *Router 0 Nat Translation*

3) Test connectivity to Router 1

```
Router#ping 201.1.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 201.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/3 ms
```

Figure 5 *Pinging 201.1.1.2 from router 0*

On Router1

1) Verify DHCP pool

```
Router#show running-config
Building configuration...

Current configuration : 779 bytes
!
version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
no service password-encryption
!
hostname Router
!
!
!
!
ip dhcp excluded-address 192.168.1.254
!
ip dhcp pool COMPANY_AA_POOL
 network 192.168.1.0 255.255.255.0
 default-router 192.168.1.254
 dns-server 8.8.8.8
!
!
 --More--
```

Figure 6 *Router 1 running configuration*

2) Verify DHCP Lease on Router 1:

Check if Router1 has assigned Ips to the PCs:

```
Router#show ip dhcp binding
IP address       Client-ID/              Lease expiration        Type
                 Hardware address
192.168.1.1      00E0.A3BA.504B           --
Automatic
192.168.1.2      0009.7C0C.7507           --
Automatic
192.168.1.3      0001.9794.0E3C           --
Automatic
```

Figure 7 DHCP Binding Table on Router1

<u>Part B: Show that PC0, PC1, and PC2 have successfully obtained their IP addresses from the Remote DHCP Server 0. Include descriptions.</u>

On PCs:

PC0:

Check the Assigned IP for PC0

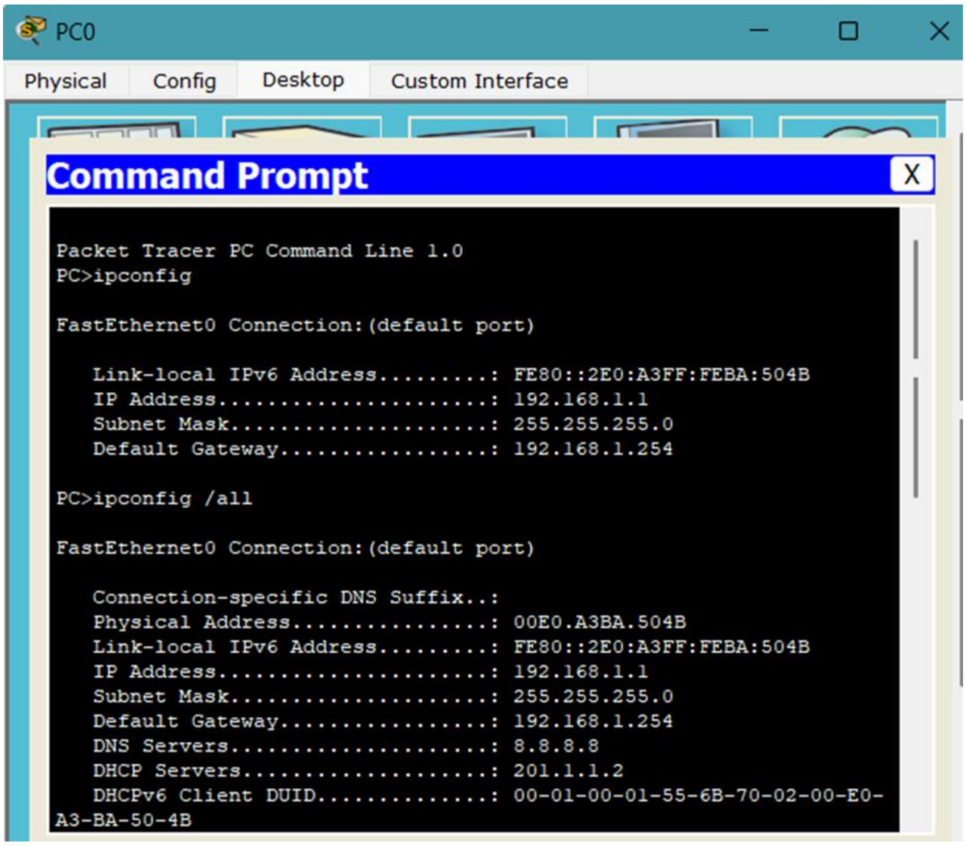


Figure 8 IP Configuration on PC0

Ping the Default Gateway (Router 0)

```
PC>ping 192.168.1.254

Pinging 192.168.1.254 with 32 bytes of data:

Reply from 192.168.1.254: bytes=32 time=1ms TTL=255
Reply from 192.168.1.254: bytes=32 time=0ms TTL=255
Reply from 192.168.1.254: bytes=32 time=1ms TTL=255
Reply from 192.168.1.254: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.1.254:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figure 9 *Pinging the router 0 default gateway from PC0*

Check the PC can reach the internet

```
PC>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Figure 10.1 *Ping Test to 8.8.8.8 from PC0*

```
PC>ping 201.1.1.2

Pinging 201.1.1.2 with 32 bytes of data:

Reply from 201.1.1.2: bytes=32 time=0ms TTL=254
Reply from 201.1.1.2: bytes=32 time=0ms TTL=254
Reply from 201.1.1.2: bytes=32 time=1ms TTL=254
Reply from 201.1.1.2: bytes=32 time=0ms TTL=254

Ping statistics for 201.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figure 10.2 *Ping test to 201.1.1.2 from PC0*

Explanation for the failure in 8.8.8.8 but successful in 201.1.1.2:

In figure 10.1, the failing ping 8.8.8.8 is due to Packet Tracer's simulated environment lacking real internet access, although PAT is correctly configured, as evidenced by the successful ping **201.1.1.2** (Router1's public IP). While PAT transforms private traffic from **192.168.1.0/24** to Router0's public IP (**201.1.1.1**) for outward communication, **8.8.8.8** is inaccessible because Packet Tracer does not imitate external internet routes, and Router1's default route loops back to Router0 in the absence of an upstream ISP connection. Pinging successfully to

201.1.1.2 in figure 10.2 verifies that NAT/PAT is operational, whereas the unsuccessful attempt to reach **8.8.8.8** is solely due to a simulation limitation and not a configuration mistake.

On PC1:

Check the Assigned IP for PC1

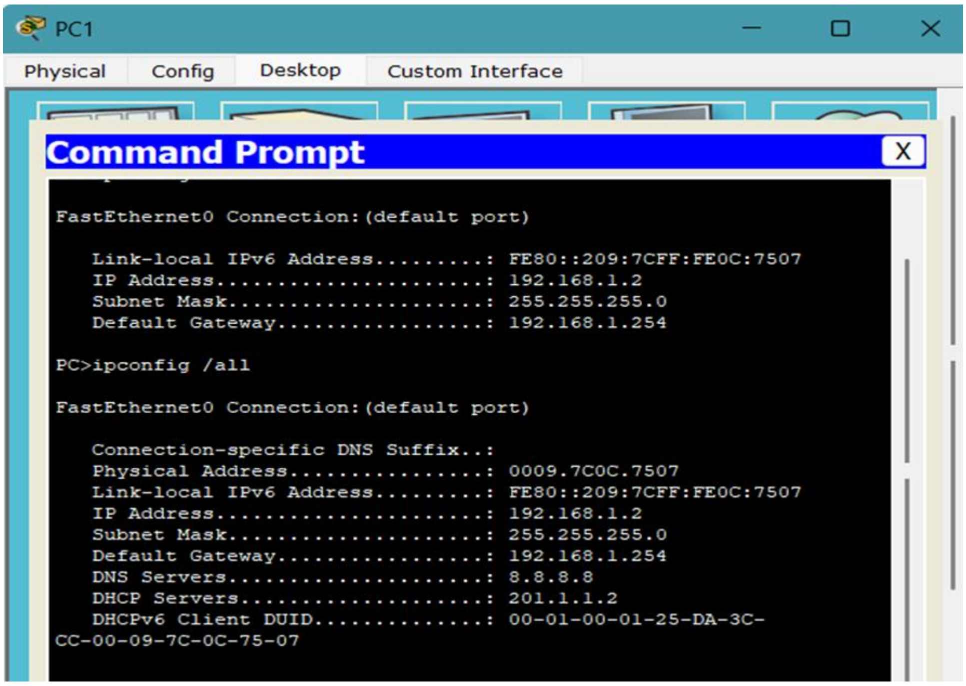


***Figure 11** IP Configuration on PC1*

Ping the Default Gateway (Router 0)

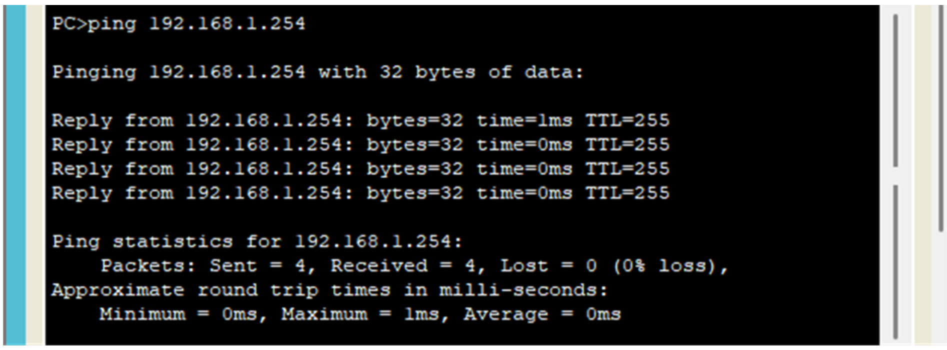


Figure 12 .1 *Pinging the router 0 default gateway from PC1*

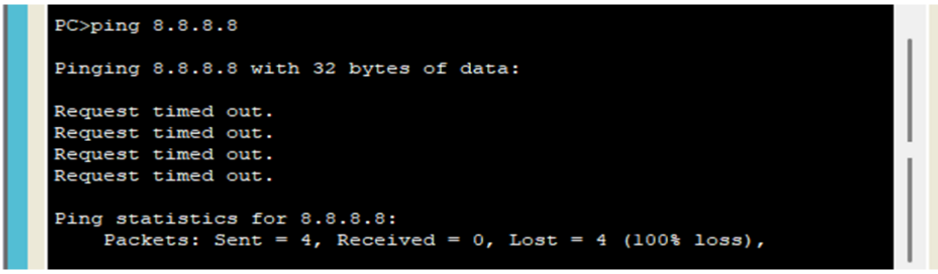


Figure 12 .2 *Ping Test to 8.8.8.8 from PC1*

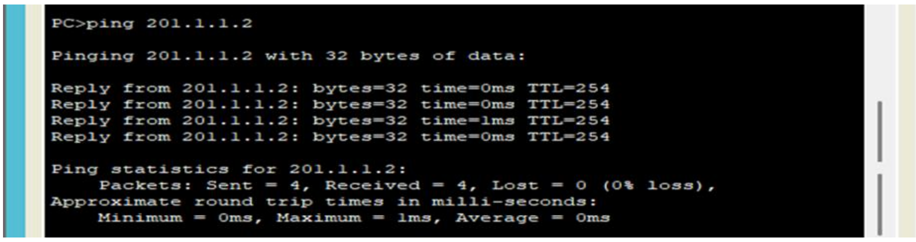


Figure 12 .3 *Ping test to 201.1.1.2 from PC1*

Explanation:

In figure 12 .1 shows PC1 is assigned the IP address **192.168.1.2** with a subnet mask of **255.255.255.0** and a default gateway of **192.168.1.254**, confirming it is properly connected to the local network. The successful ping test to the default gateway shows all packets were sent and received with 0% loss and minimal delay, indicating that PC1 can communicate effectively with the local router. However, in figure 12.2, when attempting to ping the external IP address **8.8.8.8** (Google DNS), all requests time out, resulting in 100% packet loss. This suggests that PC1 is unable to reach that external address, possibly due to a firewall rule, missing NAT configuration on the router, or ICMP being blocked by the destination. On the other hand, in the figure 12.3, pinging another external IP **201.1.1.2** is successful, with 0% packet loss and quick response times. This proves that PC1 has partial external connectivity, and the issue is likely specific to certain external IPs, not the entire internet.

Check the Assigned IP for PC2

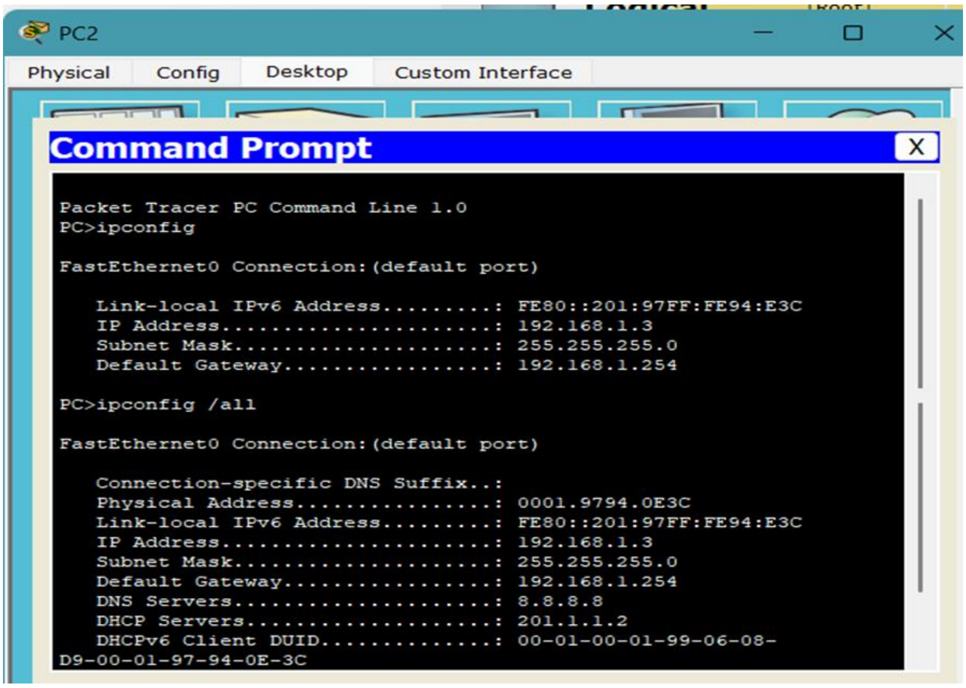


Figure 13 *IP Configuration on PC2*

Ping the Default Gateway (Router 0)

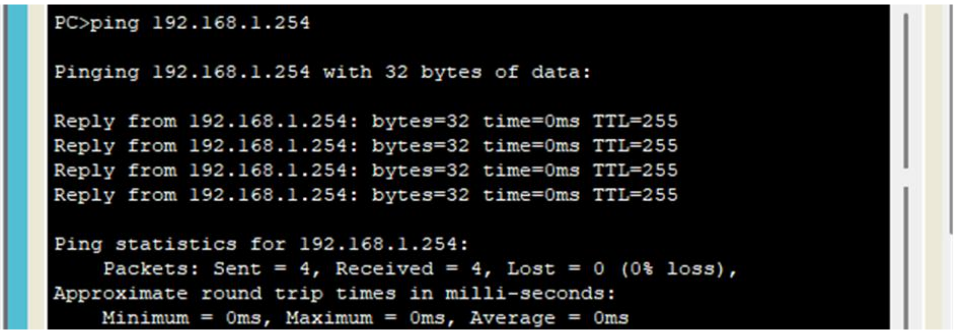


Figure 14 .1 *Pinging the router 0 default gateway from PC2*

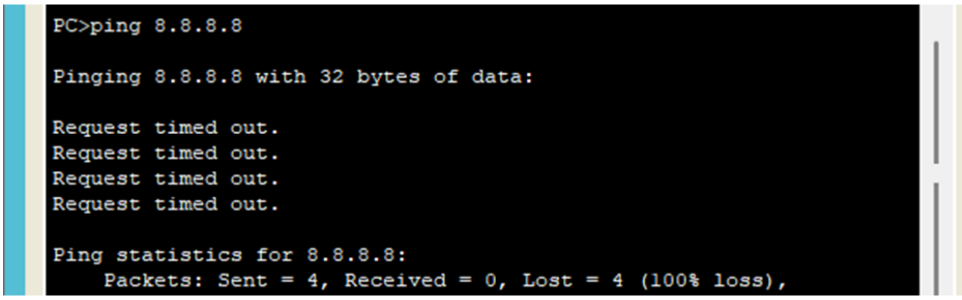
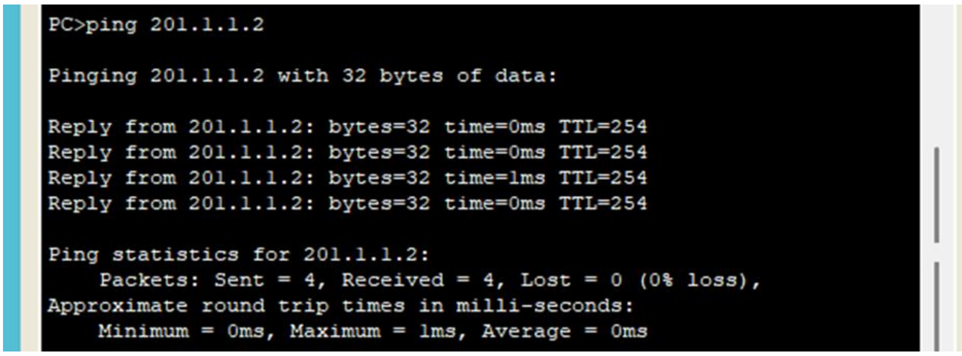


Figure 14 .2 *Ping Test to 8.8.8.8 from PC2*



```
PC>ping 201.1.1.2

Pinging 201.1.1.2 with 32 bytes of data:

Reply from 201.1.1.2: bytes=32 time=0ms TTL=254
Reply from 201.1.1.2: bytes=32 time=0ms TTL=254
Reply from 201.1.1.2: bytes=32 time=1ms TTL=254
Reply from 201.1.1.2: bytes=32 time=0ms TTL=254

Ping statistics for 201.1.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

Figure 14 .3 *Ping test to 201.1.1.2 from PC2*

Explanation:

PC2 is assigned the IP address **192.168.1.3** with a subnet mask of **255.255.255.0** shown in figure 13 and a default gateway of **192.168.1.254** shown in figure 14.1, indicating it is properly connected to the local network. A successful ping to the default gateway confirms that the local connection between PC2 and the router is working correctly. However, in figure 14.2, when attempting to ping the external IP **8.8.8.8** (Google DNS), all requests time out, showing that PC2 cannot reach this external server—possibly due to missing NAT configuration, a firewall rule, or ICMP being blocked or filtered. Interestingly, in figure 14.3, a ping to another external IP, **201.1.1.2**, is successful with no packet loss, which proves that PC2 does have some access to external networks. This suggests that internet connectivity is partially functional, and the issue is specific to certain destinations. The problem could be caused by selective routing, firewall restrictions, or limitations on external server responses, rather than a complete lack of internet access.

PART C: Could PC0, PC1, and PC2 DIRECTLY ping the Remote DHCP Server0? Explain WHY / WHY NOT. You may use "print screen" to present the results.:

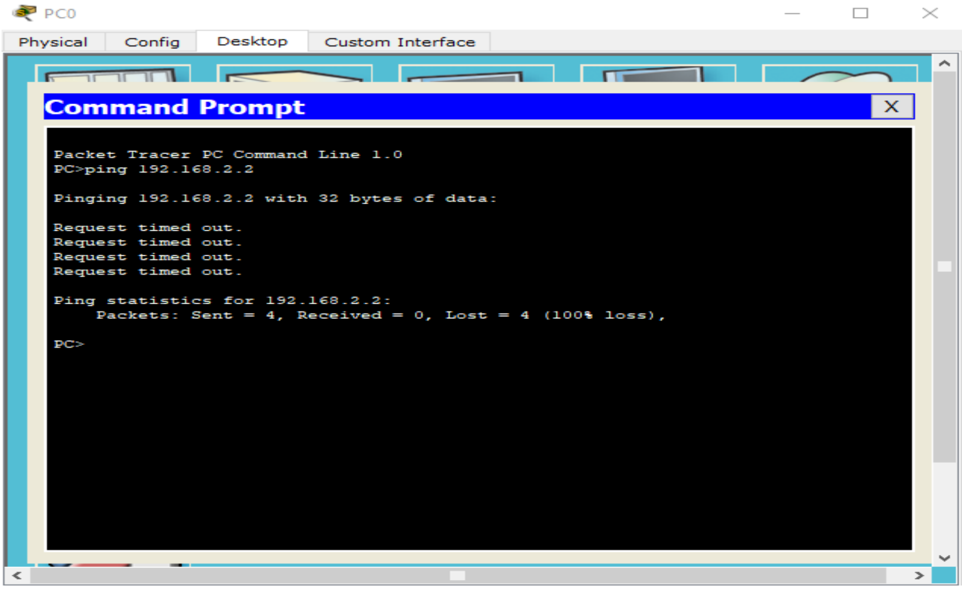


Figure 15.1 *Ping test 192.168.2.2 from PC0*

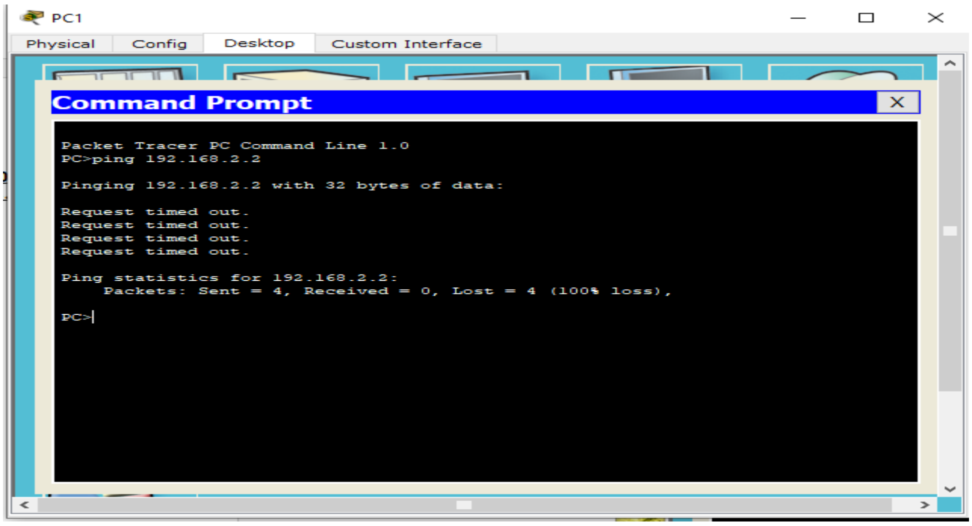


Figure 15.2 *Ping test 192.168.2.2 from PC1*

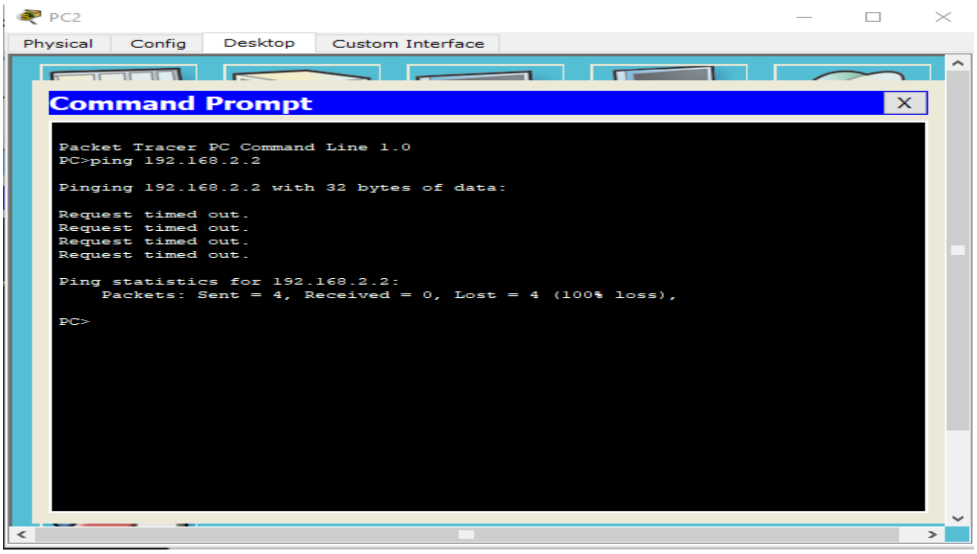


Figure 15.3 *Ping test 192.168.2.2 from PC2*

<u>Explanation:</u>

No, PC0, PC1, and PC2 cannot directly ping the Remote DHCP Server by default because they are in the **192.168.1.0/24** network while the DHCP server is in the **192.168.2.0/24** network, making them part of separate subnets that require proper routing configurations on both Router0 and Router1 to enable communication. Without these routes, and unless no firewalls or ACLs are blocking the traffic and NAT (if used) is correctly set up, the ICMP ping packets from the PCs cannot reach the DHCP server, even if DHCP is successful in assigning IP addresses. This is because DHCP and ICMP are different protocols, and ICMP requires routing for ICMP to work across networks.

Assignment Q2

The network is built as shown in figure 2.

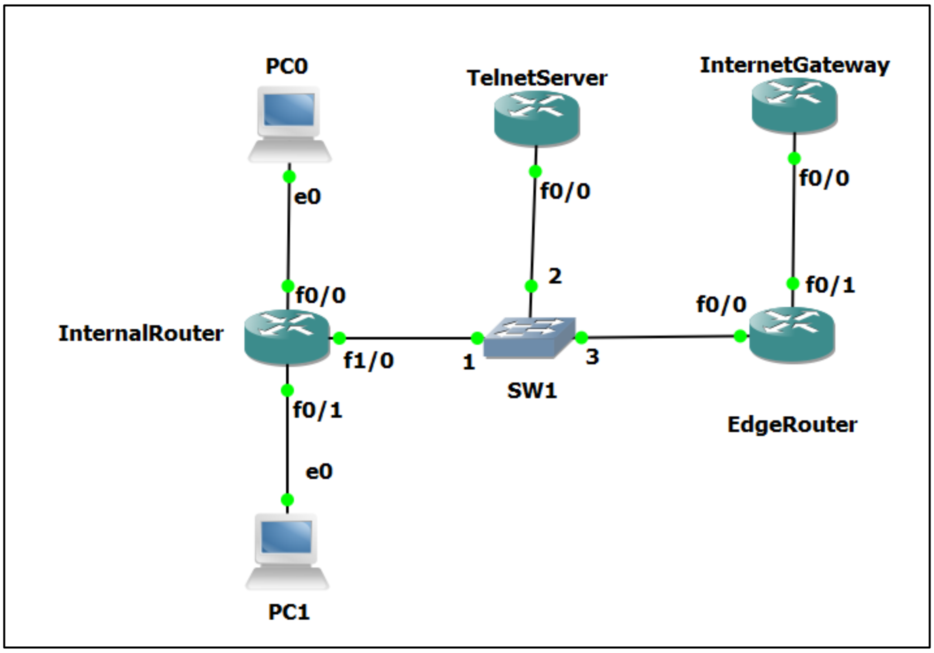


Figure 2 The structure of the network

Part for Configurations of the Network

Setup for PC0:

```
QEMU (QEMU1)
tc@box:~$ sudo su
root@box:~# hostname PC0
root@PC0:~# ip addr add 10.1.1.1/24 brd + dev eth0
root@PC0:~# ip route add default via 10.1.1.254
root@PC0:~# _
```

Figure 2.0.1 Configuration on PC0

- IP for PC0 is assigned as 10.1.1.1/24. Its default gateway is 10.1.1.254.

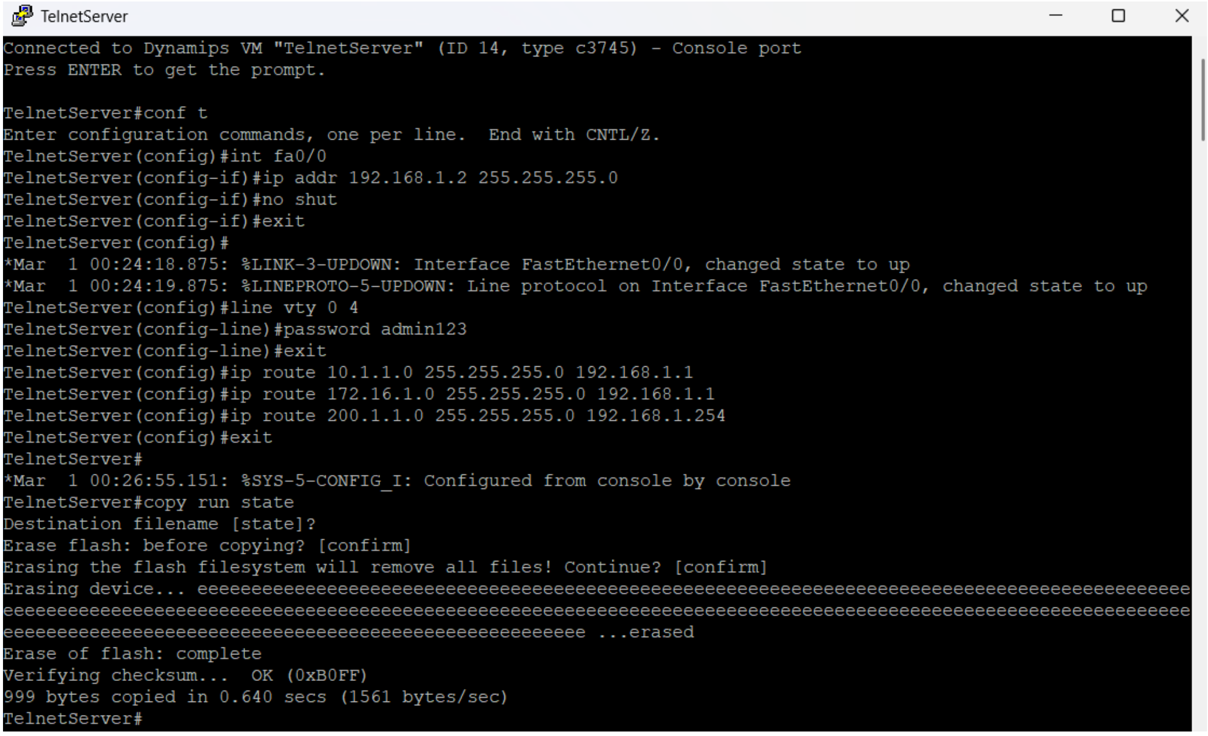
Setup for PC1:

```
QEMU (QEMU2)
tc@box:~$ sudo su
root@box:~# hostname PC1
root@PC1:~# ip addr add 172.16.1.1/24 brd + dev eth0
root@PC1:~# ip route add default via 172.16.1.254
root@PC1:~# _
```

Figure 2.0.2 Configuration on PC1

- PC1’s IP address is assigned as 172.16.1.1/24 and its default gateway is 172.16.1.254.

<u>Configuration on Telnet Server:</u>



```
TelnetServer
Connected to Dynamips VM "TelnetServer" (ID 14, type c3745) - Console port
Press ENTER to get the prompt.

TelnetServer#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
TelnetServer(config)#int fa0/0
TelnetServer(config-if)#ip addr 192.168.1.2 255.255.255.0
TelnetServer(config-if)#no shut
TelnetServer(config-if)#exit
TelnetServer(config)#
*Mar  1 00:24:18.875: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:24:19.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
TelnetServer(config)#line vty 0 4
TelnetServer(config-line)#password admin123
TelnetServer(config-line)#exit
TelnetServer(config)#ip route 10.1.1.0 255.255.255.0 192.168.1.1
TelnetServer(config)#ip route 172.16.1.0 255.255.255.0 192.168.1.1
TelnetServer(config)#ip route 200.1.1.0 255.255.255.0 192.168.1.254
TelnetServer(config)#exit
TelnetServer#
*Mar  1 00:26:55.151: %SYS-5-CONFIG_I: Configured from console by console
TelnetServer#copy run state
Destination filename [state]?
Erase flash: before copying? [confirm]
Erasing the flash filesystem will remove all files! Continue? [confirm]
Erasing device... eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee
eeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeeee ...erased
Erase of flash: complete
Verifying checksum...  OK (0xB0FF)
999 bytes copied in 0.640 secs (1561 bytes/sec)
TelnetServer#
```

Figure 2.0.3 Configuration on Telnet Server

The command used in Telnet Server:

```
conf t
int fa0/0
ip addr 192.168.1.2 255.255.255.0
no shut
exit
line vty 0 4
password admin123
exit
ip route 10.1.1.0 255.255.255.0 192.168.1.1
ip route 172.16.1.0 255.255.255.0 192.168.1.1
ip route 200.1.1.0 255.255.255.0 192.168.1.254
copy run state
```

- The Telnet Server is configured with interface fa0/0 set to IP address "192.168.1.2" and enabled.
- VTY lines (0 to 4) are configured with a login password set to admin123 to allow Telnet Service.
- Static routes are added to reach network 10.1.1.0/24 and 172.16.1.0/24 via 192.168.1.1 and network 200.1.1.0/24 via 192.168.254.
- The configuration is saved using "copy run state".

Configuration on Internet Gateway:

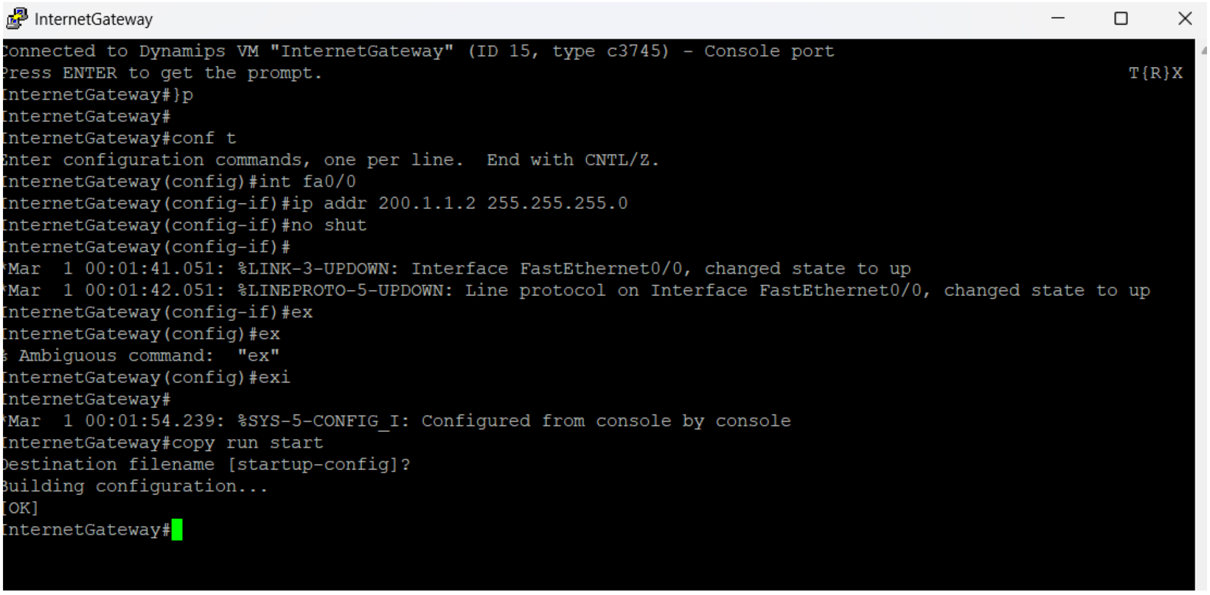


Figure 2.0.4 Configuration on Internet Gateway

The command used in Internet Gateway is as below:

```
conf t
int fa0/0
ip addr 200.1.1.2 255.255.255.0
no shut
ex
exit
copy run start
```

- In internet Gateway, the interface fa0/0 is assigned the IP address 200.1.1.2/24.
- The configuration is saved using “copy run start”

Configuration on Edge Router:

```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/1
Router(config-if)#ip address 200.1.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
*Mar  1 00:01:53.035: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:01:54.035: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to upint
fa0/0
Router(config-if)#int fa0/0
Router(config-if)#ip address 192.168.1.254 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
*Mar  1 00:02:31.831: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:02:32.831: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
Router(config)#ip route 10.1.1.0 255.255.255.0 192.168.1.1
Router(config)#ip route 172.16.1.0 255.255.255.0 192.168.1.1
Router(config)#int fa0/0
Router(config-if)#ip nat inside
Router(config-if)#
*Mar  1 00:06:24.875: %LINEPROTO-5-UPDOWN: Line protocol on Interface NVI0, changed state to up
Router(config-if)#exit
Router(config)#int fa0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
Router(config)#ip nat inside source list 10 int fa0/1 overload
Router(config)#ip nat inside source static tcp 192.168.1.2 23 200.1.1.1 23
Router(config)#no ip nat inside source list 10 int fa0/1 overload
Router(config)#ip nat inside source list 1 int fa0/1 overload
Router(config)#exit
Router#cop
*Mar  1 00:11:59.755: %SYS-5-CONFIG_I: Configured from console by consol
% Ambiguous command:  "co"
Router#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

Figure 2.0.5 Configuration on Edge Router

The command used on Edge Router:

```
config t
int fa0/1
ip addr 200.1.1.1 255.255.255.0
no shut
exit
int fa0/0
```

```
ip addr 192.168.1.254 255.255.255.0
no shut
exit
ip route 10.1.1.0 255.255.255.0 192.168.1.1
ip route 172.16.1.0 255.255.255.0 192.168.1.1
int fa0/0
ip nat inside
exit
int fa0/1
ip nat outside
exit
access-list 1 permit 192.168.1.0 0.0.0.255
ip nat inside source list 1 int fa0/1 overload
ip nat inside source static tcp 192.168.1.2 23 200.1.1.1 23
exit
copy run start
```

<u>Configuration on Internal Router:</u>

<u>IP Configuration</u>

```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip address 10.1.1.254 255.255.255.0
Router(config-if)#no shut
Router(config-if)#
*Mar  1 00:02:44.291: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar  1 00:02:45.291: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to upexit
Router(config)#int fa0/1
Router(config-if)#ip address 172.16.1.254 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
*Mar  1 00:03:14.587: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
*Mar  1 00:03:15.587: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to up
Router(config)#int fa1/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shut
Router(config-if)#exit
Router(config)#
*Mar  1 00:03:58.695: %LINK-3-UPDOWN: Interface FastEthernet1/0, changed state to up
*Mar  1 00:03:59.695: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet1/0, changed state to up
Router(config)#
```

Figure 2.0.6.1 IP Configuration on Internal Router

The command used for IP configuration on Internal Router:

```
conf t
int fa0/0
ip address 10.1.1.254 255.255.255.0
no shut
int fa0/1
ip address 172.16.1.254 255.255.255.0
no shut
exit
int fa1/0
ip address 192.168.1.1 255.255.255.0
no shut
exit
```

<u>Static route configuration on the Internal Router</u>

Figure 2.0.6.2 Static Route on Internal Router

The command used for static route configuration on Internal Router:

```
ip route 200.1.1.0 255.255.255.0 192.168.1.254
```

Others Configuration on Internal Router

```
Router>en
Router#config t
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#
*Mar  1 00:23:06.947: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#int fa0/1
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#
*Mar  1 00:23:41.195: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config)#int fa1/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#
*Mar  1 00:24:06.959: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config)#ip nat inside source static 192.168.1.2 10.1.1.2
*Mar  1 00:24:41.199: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config)#ip nat inside source static 192.168.1.3 10.1.1.3
Router(config)#
*Mar  1 00:25:06.959: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config)#ip nat inside source 172.16.1.1 192.168.1.3
                                    ^
% Invalid input detected at '^' marker.

Router(config)#ip nat inside source static 172.16.1.1 192.168.1.3
Router(config)#
*Mar  1 00:25:41.163: %CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet1/0 (not half duplex),
 with Router FastEthernet0/0 (half duplex).
Router(config)#exit
Router#
*Mar  1 00:25:47.087: %SYS-5-CONFIG_I: Configured from console by console
Router#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

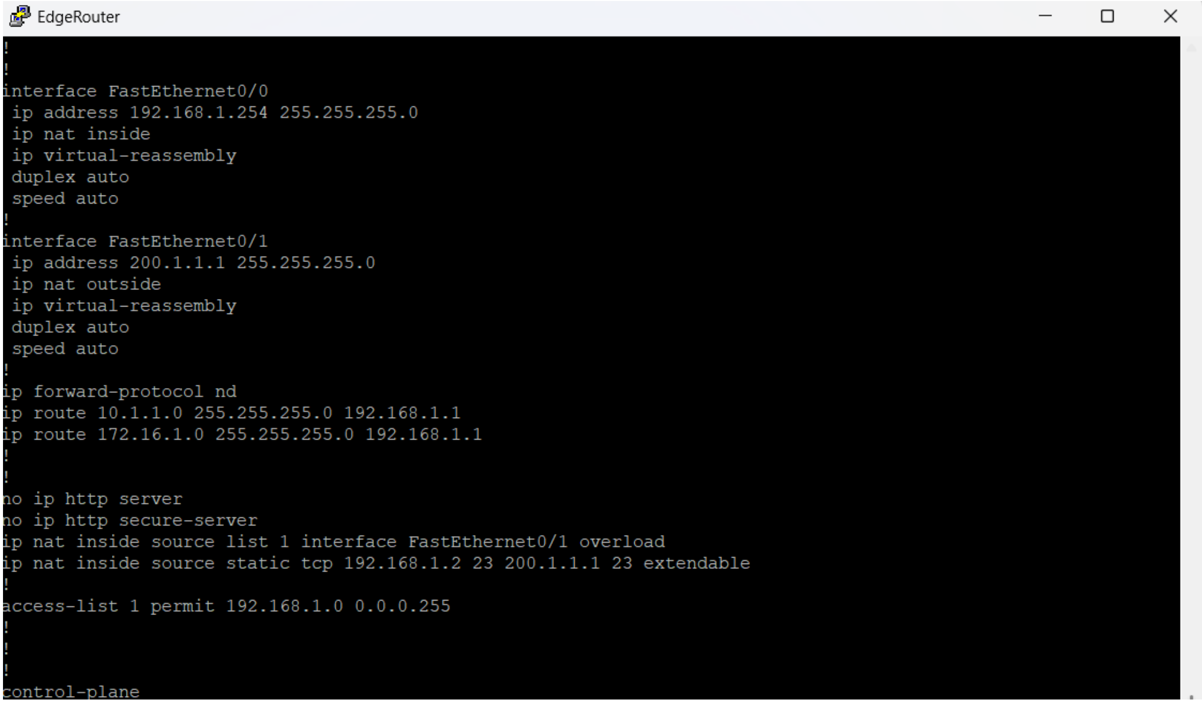
Figure 2.0.6.3 Others Configuration on Internal Router

The command used for others configuration on Internal Router:

conf t int fa0/0 ip nat inside exit int fa0/1 ip nat inside exit int fa1/0 ip nat outside exit ip nat inside source static 192.168.1.2 10.1.1.2

```
ip nat inside source static 192.168.1.3 10.1.1.3
ip nat inside source static 172.16.1.1 192.168.1.3
exit
copy run start
```

PART A: The relevant running configuration for the Edge Router



```
!
!
interface FastEthernet0/0
 ip address 192.168.1.254 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 200.1.1.1 255.255.255.0
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
ip forward-protocol nd
ip route 10.1.1.0 255.255.255.0 192.168.1.1
ip route 172.16.1.0 255.255.255.0 192.168.1.1
!
!
no ip http server
no ip http secure-server
ip nat inside source list 1 interface FastEthernet0/1 overload
ip nat inside source static tcp 192.168.1.2 23 200.1.1.1 23 extendable
!
access-list 1 permit 192.168.1.0 0.0.0.255
!
!
!
control-plane
```

Figure 2.1 Running Configuration of Edge Router

- The Edge Router is set up to connect the internal network (192.168.1.0/24) to the outside world (200.1.1.0/24).
- It gives the IP "192.168.1.254" to interface fa0/0 (for inside traffic) and 200.1.1.1 to interface fa0/1(for outside traffic).
- Both interfaces are brought up using "no shut".
- To make sure it knows how to reach the other internal networks (10.1.1.0/24 and 172.16.1.0/24), two static routes that pointing to 192.168.1.1 are added to the router.
- NAT is then configured:
    - fa0/0 is designated as the NAT inside interface (for internal devices),
    - fa0/1 is marked as the NAT outside interface (for external access).
- An access list is created to permit traffic from the internal network, and NAT overload is enabled using the command "ip nat inside source list 1 interface fa0/1 overload".

- This allows multiple internal devices to share the single public IP “200.1.1.1” by using different ports.
- Additionally, a static NAT mapping is configured to forward any Telnet traffic (TCP port 23) received on 200.1.1.1 to an internal host at 192.168.1.2.

<u>PART B: The relevant running configuration for the Internal Router to enable PC1 to access the Internet. PC0 however could only access the internal devices (PC1, Telnet Server).</u>

<u>Running Configuration of Internal Router</u>

InternalRouter

```
!
!
interface FastEthernet0/0
 ip address 10.1.1.254 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet0/1
 ip address 172.16.1.254 255.255.255.0
 ip nat inside
 ip virtual-reassembly
 duplex auto
 speed auto
!
interface FastEthernet1/0
 ip address 192.168.1.1 255.255.255.0
 ip nat outside
 ip virtual-reassembly
 duplex auto
 speed auto
!
ip forward-protocol nd
ip route 200.1.1.0 255.255.255.0 192.168.1.254
!
!
no ip http server
no ip http secure-server
ip nat inside source static 192.168.1.2 10.1.1.2
ip nat inside source static 192.168.1.3 10.1.1.3
ip nat inside source static 172.16.1.1 192.168.1.3
!
!
!
```

Figure 2.2 Running Configuration of Internal Router

- The Internal Router is configured with three interfaces to connect different internal networks.
- Interface fa0/0 is assigned the IP address 10.1.1.254 and connects to PC0, while fa0/1 has the IP 172.16.1.254 and connects to PC1.
- The third interface, fa1/0, is configured with the IP 192.168.1.1, linking to the Edge Router.

- A static route pointing to 192.168.1.254 (Edge Router) is configured so that the router able to reach to outside network (200.1.1.0/24)
- The interface fa0/0 and fa0/1 are marked as NAT inside interfaces, meaning they belong to internal/private networks, while fa1/0 serves as the NAT outside interface.
- To allow PC1 (172.16.1.1) to reach the internet through the Edge Router, a static NAT rule is used to translate its IP to 192.168.1.3, making it part of the 192.168.1.0/24 network.
- Additional NAT rules translate the IP address of the Telnet Server (192.168.1.2) to 10.1.1.2, and PC1’s translated IP (192.168.1.3) to 10.1.1.3.

PART C: Usage of Wireshark with following situations

i) PC1 is able to Ping the Internet Gateway and all other internal devices.

PC1 is able to ping PC0

```
user@debian-i386:~$ ping 10.1.1.1
PING 10.1.1.1 (10.1.1.1) 56(84) bytes of data.
64 bytes from 10.1.1.1: icmp_seq=1 ttl=63 time=34.6 ms
64 bytes from 10.1.1.1: icmp_seq=2 ttl=63 time=31.6 ms
64 bytes from 10.1.1.1: icmp_seq=3 ttl=63 time=34.2 ms
64 bytes from 10.1.1.1: icmp_seq=4 ttl=63 time=22.0 ms
64 bytes from 10.1.1.1: icmp_seq=5 ttl=63 time=39.7 ms
64 bytes from 10.1.1.1: icmp_seq=6 ttl=63 time=24.6 ms
64 bytes from 10.1.1.1: icmp_seq=7 ttl=63 time=29.8 ms
64 bytes from 10.1.1.1: icmp_seq=8 ttl=63 time=25.6 ms
64 bytes from 10.1.1.1: icmp_seq=9 ttl=63 time=27.8 ms
64 bytes from 10.1.1.1: icmp_seq=10 ttl=63 time=30.2 ms
64 bytes from 10.1.1.1: icmp_seq=11 ttl=63 time=21.5 ms
64 bytes from 10.1.1.1: icmp_seq=12 ttl=63 time=23.3 ms
^Z
[3]+  Stopped                 ping 10.1.1.1
user@debian-i386:~$ _
```

Figure 2.3.1 PC1 is pinging to PC0

857	4459.88900	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=1/256, ttl=64
858	4459.93500	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=1/256, ttl=63
859	4460.12100	c4:08:3d:c0:00:01	c4:08:3d:c0:00:01	LOOP	60	Reply	
860	4460.85000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=2/512, ttl=64
861	4460.88000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=2/512, ttl=63
862	4461.86200	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=3/768, ttl=64
863	4461.88900	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=3/768, ttl=63
864	4462.86800	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=4/1024, ttl=64
865	4462.89800	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=4/1024, ttl=63
866	4463.87100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=5/1280, ttl=64
867	4463.89000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=5/1280, ttl=63
868	4464.87700	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=6/1536, ttl=64
869	4464.89700	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=6/1536, ttl=63
870	4465.88100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=7/1792, ttl=64
871	4465.90900	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=7/1792, ttl=63
872	4466.88400	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=8/2048, ttl=64
873	4466.90700	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=8/2048, ttl=63
874	4467.88900	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=9/2304, ttl=64
875	4467.91400	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=9/2304, ttl=63
876	4468.89700	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) request	id=0x4f09, seq=10/2560, ttl=64
877	4468.92600	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply	id=0x4f09, seq=10/2560, ttl=63

Figure 2.3.2 WireShark Result for PC1 pinging to PC0

Based on the figures that were provided for this report, we are able to find that PC1 is able to ping PC0. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC1 is able to ping PC0.

When the PC1 ping to PC0, Wireshark shows the request packets from PC1 and send it to PC0, and there would be a reply packet from PC0 and send to PC1. With this situation, we are able to say that the PC1 is able to ping PC0.

PC 1 is able to ping Telnet Server

```
debian-i386:/home/user# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=254 time=90.2 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=254 time=53.0 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=254 time=44.7 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=254 time=42.1 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=254 time=63.1 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=254 time=139 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=254 time=54.7 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=254 time=38.8 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=254 time=62.9 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=254 time=55.1 ms
64 bytes from 192.168.1.2: icmp_seq=11 ttl=254 time=40.4 ms
64 bytes from 192.168.1.2: icmp_seq=12 ttl=254 time=56.8 ms
64 bytes from 192.168.1.2: icmp_seq=13 ttl=254 time=67.7 ms
64 bytes from 192.168.1.2: icmp_seq=14 ttl=254 time=52.4 ms
64 bytes from 192.168.1.2: icmp_seq=15 ttl=254 time=45.5 ms
^Z
[5]+  Stopped                 ping 192.168.1.2
debian-i386:/home/user# _
```

Figure 2.3.3 PC1 is pinging to Telnet Router

1179	2250.59900	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=1/256, ttl=64
1180	2250.65900	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=1/256, ttl=254
1181	2251.57500	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=2/512, ttl=64
1182	2251.62800	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=2/512, ttl=254
1183	2252.29300	c4:08:3d:c0:00:01	c4:08:3d:c0:00:01	LOOP	60	Reply
1184	2252.57900	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=3/768, ttl=64
1185	2252.62000	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=3/768, ttl=254
1186	2253.58200	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=4/1024, ttl=64
1187	2253.62400	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=4/1024, ttl=254
1188	2254.03500	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0x3fec40
1189	2254.59000	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=5/1280, ttl=64
1190	2254.65200	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=5/1280, ttl=254
1191	2255.59800	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=6/1536, ttl=64
1192	2255.64600	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=6/1536, ttl=254
1193	2256.60700	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=7/1792, ttl=64
1194	2256.66100	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=7/1792, ttl=254
1195	2257.62800	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=8/2048, ttl=64
1196	2257.66600	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=8/2048, ttl=254
1197	2258.62700	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=9/2304, ttl=64
1198	2258.68900	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=9/2304, ttl=254
1199	2259.63400	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=10/2560, ttl=64
1200	2259.68600	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=10/2560, ttl=254
1201	2260.64600	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=11/2816, ttl=64
1202	2260.68500	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=11/2816, ttl=254
1203	2261.65300	172.16.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x4b09, seq=12/3072, ttl=64
1204	2261.70800	192.168.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4b09, seq=12/3072, ttl=254

Figure 2.3.4 WireShark Result for PC1 pinging to Telnet Router

Based on the figures that were provided for this report, we are able to find that PC1 is able to ping the Telnet Server. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC1 is able to ping the Telnet Server.

When the PC1 pings to Telnet Server, Wireshark shows the request packets from PC1 and send it to Telnet Server, and there would be a reply packet from Telnet Server and send to PC1. With this situation, we are able to say that the PC1 is able to ping Telnet Server.

<u>PC 1 is able to ping Edged Server</u>

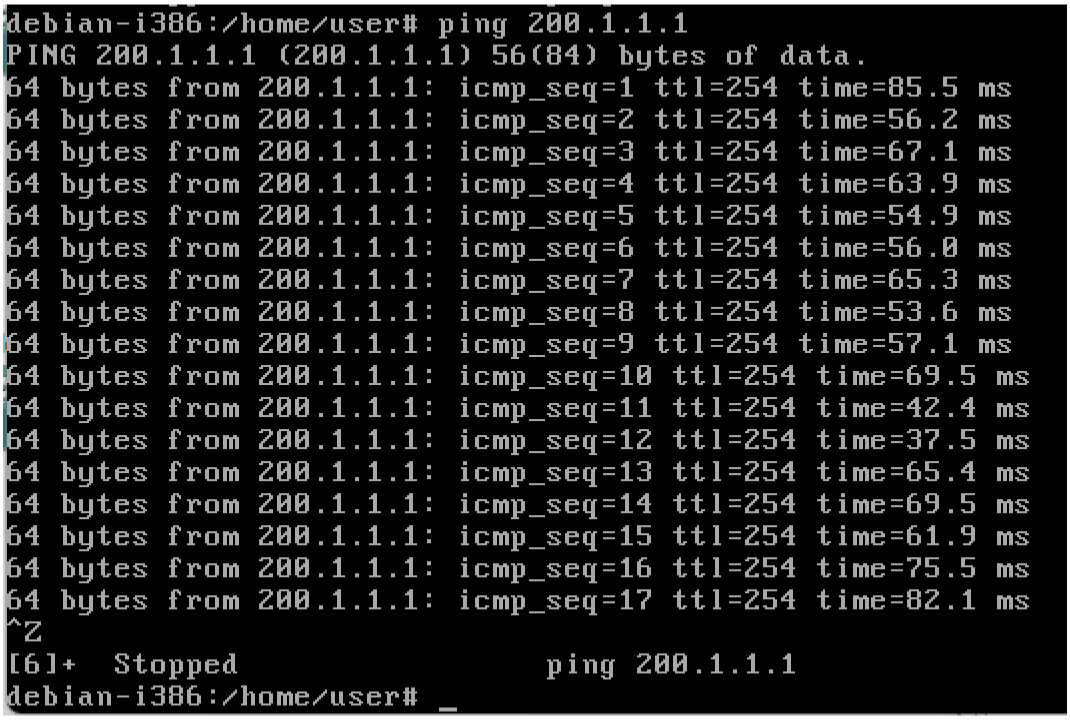


Figure 2.3.5 PC1 is pinging to Edged Server

1220	2297.15500	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=1/256, ttl=64
1221	2297.23800	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=1/256, ttl=254
1222	2298.15700	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=2/512, ttl=64
1223	2298.21300	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=2/512, ttl=254
1224	2299.16100	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=3/768, ttl=64
1225	2299.22700	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=3/768, ttl=254
1226	2300.17000	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=4/1024, ttl=64
1227	2300.23200	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=4/1024, ttl=254
1228	2301.17500	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=5/1280, ttl=64
1229	2301.22700	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=5/1280, ttl=254
1230	2302.18200	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=6/1536, ttl=64
1231	2302.23700	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=6/1536, ttl=254
1232	2302.31400	c4:08:3d:c0:00:01	c4:08:3d:c0:00:01	LOOP	60	Reply
1233	2303.18500	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=7/1792, ttl=64
1234	2303.24800	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=7/1792, ttl=254
1235	2304.18900	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=8/2048, ttl=64
1236	2304.24200	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=8/2048, ttl=254
1237	2305.19400	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=9/2304, ttl=64
1238	2305.25000	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=9/2304, ttl=254
1239	2306.19900	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=10/2560, ttl=64
1240	2306.26700	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=10/2560, ttl=254
1241	2307.20100	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=11/2816, ttl=64
1242	2307.24300	200.1.1.1	172.16.1.1	ICMP	98	Echo (ping) reply id=0x4c09, seq=11/2816, ttl=254
1243	2308.20500	172.16.1.1	200.1.1.1	ICMP	98	Echo (ping) request id=0x4c09, seq=12/3072, ttl=64

Figure 2.3.6 WireShark Result for PC1 pinging to Edged Server

Based on the figures that were provided for this report, we are able to find that PC1 is able to ping Edged Server. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC1 is able to ping Edged Server.

When the PC1 ping to Edged Server, Wireshark shows the request packets from PC1 and send it to Edged Server, and there would be a reply packet from Edged Server and send to PC1. With this situation, we are able to say that the PC1 is able to ping Edged Server.

PC 1 is able to ping Internet Gateway

```
PING 200.1.1.2 (200.1.1.2) 56(84) bytes of data.
64 bytes from 200.1.1.2: icmp_seq=1 ttl=253 time=99.3 ms
64 bytes from 200.1.1.2: icmp_seq=2 ttl=253 time=106 ms
64 bytes from 200.1.1.2: icmp_seq=3 ttl=253 time=107 ms
64 bytes from 200.1.1.2: icmp_seq=4 ttl=253 time=70.6 ms
64 bytes from 200.1.1.2: icmp_seq=5 ttl=253 time=72.2 ms
64 bytes from 200.1.1.2: icmp_seq=6 ttl=253 time=72.1 ms
64 bytes from 200.1.1.2: icmp_seq=7 ttl=253 time=77.2 ms
64 bytes from 200.1.1.2: icmp_seq=8 ttl=253 time=64.2 ms
64 bytes from 200.1.1.2: icmp_seq=9 ttl=253 time=69.4 ms
64 bytes from 200.1.1.2: icmp_seq=10 ttl=253 time=63.5 ms
64 bytes from 200.1.1.2: icmp_seq=11 ttl=253 time=67.1 ms
64 bytes from 200.1.1.2: icmp_seq=12 ttl=253 time=82.0 ms
^Z
[7]+  Stopped                 ping 200.1.1.2
debian-i386:/home/user# _
```

Figure 2.3.7 PC1 is pinging to Internet Gateway

1261	2356.54100	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=1/256, ttl=64
1262	2356.63700	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=1/256, ttl=253
1263	2357.54100	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=2/512, ttl=64
1264	2357.64600	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=2/512, ttl=253
1265	2358.54400	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=3/768, ttl=64
1266	2358.64400	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=3/768, ttl=253
1267	2359.54900	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=4/1024, ttl=64
1268	2359.61900	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=4/1024, ttl=253
1269	2360.55700	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=5/1280, ttl=64
1270	2360.62800	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=5/1280, ttl=253
1271	2361.56000	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=6/1536, ttl=64
1272	2361.63100	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=6/1536, ttl=253
1273	2362.31500	c4:08:3d:c0:00:01	c4:08:3d:c0:00:01	LOOP	60	Reply
1274	2362.56700	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=7/1792, ttl=64
1275	2362.64300	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=7/1792, ttl=253
1276	2363.57300	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=8/2048, ttl=64
1277	2363.63500	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=8/2048, ttl=253
1278	2364.57700	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=9/2304, ttl=64
1279	2364.64600	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=9/2304, ttl=253
1280	2365.58100	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=10/2560, ttl=64
1281	2365.64400	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=10/2560, ttl=253
1282	2366.58500	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=11/2816, ttl=64
1283	2366.65200	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=11/2816, ttl=253
1284	2367.59300	172.16.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x5009, seq=12/3072, ttl=64
1285	2367.67400	200.1.1.2	172.16.1.1	ICMP	98	Echo (ping) reply id=0x5009, seq=12/3072, ttl=253

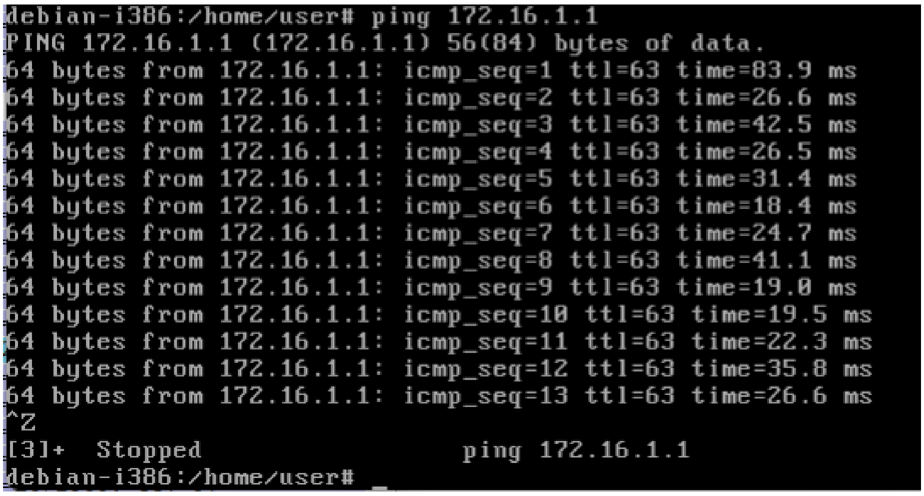
Figure 2.3.8 WireShark Result for PC1 pinging to Internet Gateway

Based on the figures that were provided for this report, we are able to find that PC1 is able to ping Internet Gateway. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC1 is able to ping Internet Gateway.

When the PC1 ping to Internet Gateway, Wireshark shows the request packets from PC1 and send it to Internet Gateway, and there would be a reply packet from Internet Gateway and send to PC1. With this situation, we are able to say that the PC1 is able to ping Internet Gateway.

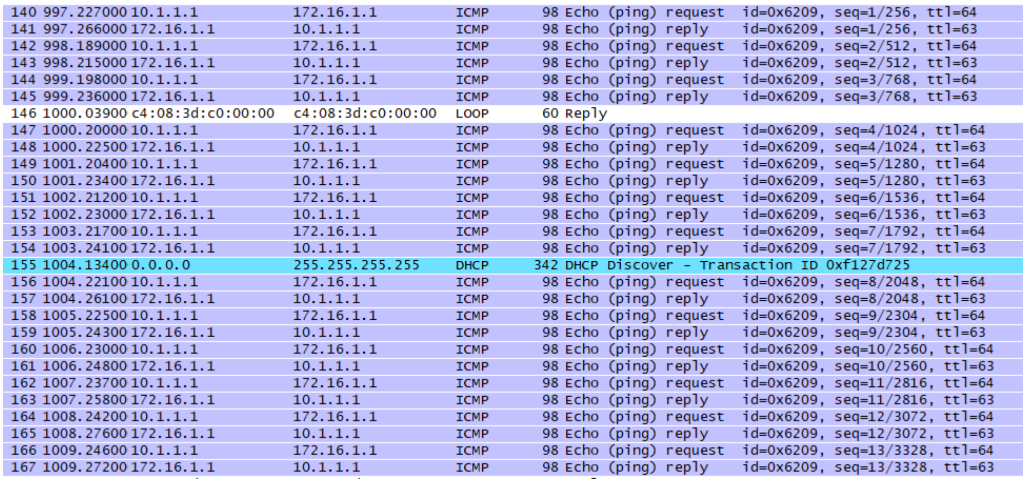
ii) PC0 is unable to Ping the Internet Gateway but able to Ping all other internal devices.

PC0 is able to ping PC1



```
debian-i386:/home/user# ping 172.16.1.1
PING 172.16.1.1 (172.16.1.1) 56(84) bytes of data.
64 bytes from 172.16.1.1: icmp_seq=1 ttl=63 time=83.9 ms
64 bytes from 172.16.1.1: icmp_seq=2 ttl=63 time=26.6 ms
64 bytes from 172.16.1.1: icmp_seq=3 ttl=63 time=42.5 ms
64 bytes from 172.16.1.1: icmp_seq=4 ttl=63 time=26.5 ms
64 bytes from 172.16.1.1: icmp_seq=5 ttl=63 time=31.4 ms
64 bytes from 172.16.1.1: icmp_seq=6 ttl=63 time=18.4 ms
64 bytes from 172.16.1.1: icmp_seq=7 ttl=63 time=24.7 ms
64 bytes from 172.16.1.1: icmp_seq=8 ttl=63 time=41.1 ms
64 bytes from 172.16.1.1: icmp_seq=9 ttl=63 time=19.0 ms
64 bytes from 172.16.1.1: icmp_seq=10 ttl=63 time=19.5 ms
64 bytes from 172.16.1.1: icmp_seq=11 ttl=63 time=22.3 ms
64 bytes from 172.16.1.1: icmp_seq=12 ttl=63 time=35.8 ms
64 bytes from 172.16.1.1: icmp_seq=13 ttl=63 time=26.6 ms
^Z
[3]+  Stopped                 ping 172.16.1.1
debian-i386:/home/user# _
```

Figure 2.3.9 PC0 is pinging to PC1



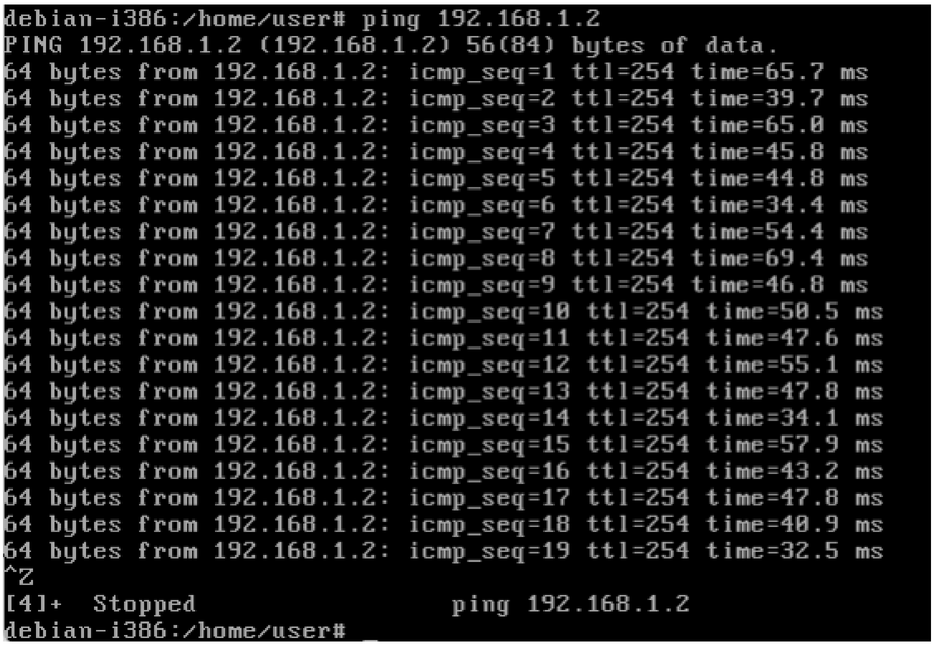
140	997.227000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=1/256, ttl=64
141	997.266000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=1/256, ttl=63
142	998.189000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=2/512, ttl=64
143	998.215000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=2/512, ttl=63
144	999.198000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=3/768, ttl=64
145	999.236000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=3/768, ttl=63
146	1000.03900	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
147	1000.20000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=4/1024, ttl=64
148	1000.22500	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=4/1024, ttl=63
149	1001.20400	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=5/1280, ttl=64
150	1001.23400	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=5/1280, ttl=63
151	1002.21200	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=6/1536, ttl=64
152	1002.23000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=6/1536, ttl=63
153	1003.21700	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=7/1792, ttl=64
154	1003.24100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=7/1792, ttl=63
155	1004.13400	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf127d725
156	1004.22100	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=8/2048, ttl=64
157	1004.26100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=8/2048, ttl=63
158	1005.22500	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=9/2304, ttl=64
159	1005.24300	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=9/2304, ttl=63
160	1006.23000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=10/2560, ttl=64
161	1006.24800	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=10/2560, ttl=63
162	1007.23700	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=11/2816, ttl=64
163	1007.25800	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=11/2816, ttl=63
164	1008.24200	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=12/3072, ttl=64
165	1008.27600	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=12/3072, ttl=63
166	1009.24600	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=13/3328, ttl=64
167	1009.27200	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=13/3328, ttl=63

Figure 2.3.10 WireShark result PC0 is pinging to PC1

Based on the figures that were provided for this report, we are able to find that PC0 is able to ping PC1. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC0 is able to ping PC1.

When the PC0 ping to PC1, Wireshark shows the request packets from PC0 and send it to PC1, and there would be a reply packet from PC1 and send to PC0. With this situation, we are able to say that the PC0 is able to ping PC1.

PC0 is able to ping Telnet Server



```
debian-i386:/home/user# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=254 time=65.7 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=254 time=39.7 ms
64 bytes from 192.168.1.2: icmp_seq=3 ttl=254 time=65.0 ms
64 bytes from 192.168.1.2: icmp_seq=4 ttl=254 time=45.8 ms
64 bytes from 192.168.1.2: icmp_seq=5 ttl=254 time=44.8 ms
64 bytes from 192.168.1.2: icmp_seq=6 ttl=254 time=34.4 ms
64 bytes from 192.168.1.2: icmp_seq=7 ttl=254 time=54.4 ms
64 bytes from 192.168.1.2: icmp_seq=8 ttl=254 time=69.4 ms
64 bytes from 192.168.1.2: icmp_seq=9 ttl=254 time=46.8 ms
64 bytes from 192.168.1.2: icmp_seq=10 ttl=254 time=50.5 ms
64 bytes from 192.168.1.2: icmp_seq=11 ttl=254 time=47.6 ms
64 bytes from 192.168.1.2: icmp_seq=12 ttl=254 time=55.1 ms
64 bytes from 192.168.1.2: icmp_seq=13 ttl=254 time=47.8 ms
64 bytes from 192.168.1.2: icmp_seq=14 ttl=254 time=34.1 ms
64 bytes from 192.168.1.2: icmp_seq=15 ttl=254 time=57.9 ms
64 bytes from 192.168.1.2: icmp_seq=16 ttl=254 time=43.2 ms
64 bytes from 192.168.1.2: icmp_seq=17 ttl=254 time=47.8 ms
64 bytes from 192.168.1.2: icmp_seq=18 ttl=254 time=40.9 ms
64 bytes from 192.168.1.2: icmp_seq=19 ttl=254 time=32.5 ms
^Z
[4]+  Stopped                 ping 192.168.1.2
debian-i386:/home/user# _
```

Figure 2.3.11 PC0 is pinging to Telnet Server

215	1328.83700	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=1/256, ttl=64
216	1328.90100	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=1/256, ttl=254
217	1329.84100	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=2/512, ttl=64
218	1329.88000	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=2/512, ttl=254
219	1330.05200	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
220	1330.84600	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=3/768, ttl=64
221	1330.90900	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=3/768, ttl=254
222	1331.84900	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=4/1024, ttl=64
223	1331.89200	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=4/1024, ttl=254
224	1332.85400	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=5/1280, ttl=64
225	1332.89800	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=5/1280, ttl=254
226	1333.85800	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=6/1536, ttl=64
227	1333.89100	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=6/1536, ttl=254
228	1334.86600	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=7/1792, ttl=64
229	1334.91900	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=7/1792, ttl=254
230	1335.87400	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=8/2048, ttl=64
231	1335.94300	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=8/2048, ttl=254
232	1336.87700	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=9/2304, ttl=64
233	1336.92200	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=9/2304, ttl=254
234	1337.88600	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=10/2560, ttl=64
235	1337.93400	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=10/2560, ttl=254
236	1338.89300	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=11/2816, ttl=64
237	1338.94000	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=11/2816, ttl=254
238	1339.89700	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=12/3072, ttl=64
239	1339.95100	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=12/3072, ttl=254
240	1340.05900	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
241	1340.90000	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=13/3328, ttl=64
242	1340.94800	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=13/3328, ttl=254
243	1341.92500	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=14/3584, ttl=64
244	1341.95900	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=14/3584, ttl=254
245	1342.92800	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=15/3840, ttl=64
246	1342.98400	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=15/3840, ttl=254
247	1343.93400	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=16/4096, ttl=64
248	1343.97500	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=16/4096, ttl=254
249	1344.94300	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=17/4352, ttl=64
250	1344.98900	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=17/4352, ttl=254
251	1345.94600	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=18/4608, ttl=64
252	1345.98500	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=18/4608, ttl=254
253	1346.94900	10.1.1.1	192.168.1.2	ICMP	98	Echo (ping) request id=0x6809, seq=19/4864, ttl=64
254	1346.98000	192.168.1.2	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6809, seq=19/4864, ttl=254

Figure 2.3.12 WireShark result PC0 is pinging to Telnet Server

Based on the figures that were provided for this report, we are able to find that PC0 is able to ping Telnet Server. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC0 is able to ping Telnet Server.

When the PC0 ping to Telnet Server, Wireshark shows the request packets from PC0 and send it to Telnet Server, and there would be a reply packet from Telnet Server and send to PC0. With this situation, we are able to say that the PC0 is able to ping Telnet Server.

<u>PC0 is able to ping Edged Server</u>

```
debian-i386:/home/user# ping 200.1.1.1
PING 200.1.1.1 (200.1.1.1) 56(84) bytes of data.
64 bytes from 200.1.1.1: icmp_seq=1 ttl=254 time=44.9 ms
64 bytes from 200.1.1.1: icmp_seq=2 ttl=254 time=61.2 ms
64 bytes from 200.1.1.1: icmp_seq=3 ttl=254 time=46.5 ms
64 bytes from 200.1.1.1: icmp_seq=4 ttl=254 time=34.1 ms
64 bytes from 200.1.1.1: icmp_seq=5 ttl=254 time=42.7 ms
64 bytes from 200.1.1.1: icmp_seq=6 ttl=254 time=39.7 ms
64 bytes from 200.1.1.1: icmp_seq=7 ttl=254 time=46.6 ms
64 bytes from 200.1.1.1: icmp_seq=8 ttl=254 time=35.8 ms
64 bytes from 200.1.1.1: icmp_seq=9 ttl=254 time=45.8 ms
64 bytes from 200.1.1.1: icmp_seq=10 ttl=254 time=42.5 ms
64 bytes from 200.1.1.1: icmp_seq=11 ttl=254 time=42.3 ms
^Z
[2]+  Stopped                 ping 200.1.1.1
debian-i386:/home/user# _
```

Figure 2.3.13 PC0 is pinging to Edged Server

3	15.887000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=1/256, ttl=64
4	15.932000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=1/256, ttl=254
5	16.896000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=2/512, ttl=64
6	16.941000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=2/512, ttl=254
7	17.901000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=3/768, ttl=64
8	17.947000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=3/768, ttl=254
9	18.908000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=4/1024, ttl=64
10	18.941000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=4/1024, ttl=254
11	19.914000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=5/1280, ttl=64
12	19.956000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=5/1280, ttl=254
13	20.003000	c4:00:06:94:00:00	c4:00:06:94:00:00	LOOP	60	Reply	
14	20.920000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=6/1536, ttl=64
15	20.958000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=6/1536, ttl=254
16	21.926000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=7/1792, ttl=64
17	21.971000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=7/1792, ttl=254
18	22.932000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=8/2048, ttl=64
19	22.966000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=8/2048, ttl=254
20	23.936000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=9/2304, ttl=64
21	23.981000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=9/2304, ttl=254
22	24.940000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=10/2560, ttl=64
23	24.981000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=10/2560, ttl=254
24	25.944000	10.1.1.1	200.1.1.1	ICMP	98	Echo (ping) request	id=0x3509, seq=11/2816, ttl=64
25	25.986000	200.1.1.1	10.1.1.1	ICMP	98	Echo (ping) reply	id=0x3509, seq=11/2816, ttl=254

Figure 2.3.14 WireShark result PC0 is pinging to Edged Server

Based on the figures that were provided for this report, we are able to find that PC0 is able to ping Edged Server. In the command prompt of the Debian operating system, we are able to have a look at the message that the request and reply for the packet, and also the time used for every request and reply. Meanwhile, in Wireshark, the tracks and records of the requesting and replying, which are the records with the ICMP protocol, are able to be shown if PC0 is able to ping Edged Server.

When the PC0 ping to Edged Server, Wireshark shows the request packets from PC0 and send it to Edged Server, and there would be a reply packet from Edged Server and send to PC0. With this situation, we are able to say that the PC0 is able to ping Edged Server.

<u>PC0 is unable to ping Internet Gateway</u>

```
debian-i386:/home/user# ping 200.1.1.2
PING 200.1.1.2 (200.1.1.2) 56(84) bytes of data.
^Z
[6]+  Stopped                 ping 200.1.1.2
```

Figure 2.3.15 PC0 is pinging to Internet Gateway

372	1888.14800	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=1/256, ttl=64
373	1889.14600	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=2/512, ttl=64
374	1890.04800	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
375	1890.16100	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=3/768, ttl=64
376	1891.16100	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=4/1024, ttl=64
377	1892.16200	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=5/1280, ttl=64
378	1893.14600	00:ab:ad:22:9b:00	c4:08:3d:c0:00:00	ARP	60	Who has 10.1.1.254? Tell 10.1.1.1
379	1893.16200	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=6/1536, ttl=64
380	1893.19300	c4:08:3d:c0:00:00	00:ab:ad:22:9b:00	ARP	60	10.1.1.254 is at c4:08:3d:c0:00:00
381	1894.16400	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=7/1792, ttl=64
382	1895.17800	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=8/2048, ttl=64
383	1896.19300	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=9/2304, ttl=64
384	1897.19400	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=10/2560, ttl=64
385	1898.19400	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=11/2816, ttl=64
386	1899.19300	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=12/3072, ttl=64
387	1900.05300	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
388	1900.19400	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=13/3328, ttl=64
389	1901.21000	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=14/3584, ttl=64
390	1902.21000	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=15/3840, ttl=64
391	1903.22400	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=16/4096, ttl=64
392	1904.22500	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=17/4352, ttl=64
393	1905.22600	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=18/4608, ttl=64
394	1906.22700	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=19/4864, ttl=64
395	1907.24100	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=20/5120, ttl=64
396	1908.24100	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=21/5376, ttl=64
397	1909.24200	10.1.1.1	200.1.1.2	ICMP	98	Echo (ping) request id=0x7209, seq=22/5632, ttl=64

Figure 2.3.16 WireShark result PC0 is pinging to Internet Gateway

Based on the figures that were provided for this report, we are able to find that PC0 is unable to ping Internet Gateway. In the command prompt of the Debian operating system, we are unable to have a look at the message that the request and reply for the packet. Meanwhile, in Wireshark, the tracks and records would just show the request of the packet but not any replies from Internet Gateway. Hence, PC0 is unable to ping Internet Gateway.

When the PC0 ping to Internet Gateway, Wireshark shows the request packets from PC0 and send it to Internet Gateway, but not receives any reply packets from Internet Gateway. With this situation, we are able to say that the PC0 is unable to ping Internet Gateway.

iii) The relevant NAT occurring in the Wireshark results

961	700.367000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=425/43265, ttl=63
962	700.429000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=425/43265, ttl=254
963	701.385000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=426/43521, ttl=63
964	701.431000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=426/43521, ttl=254
965	702.399000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=427/43777, ttl=63
966	702.444000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=427/43777, ttl=254
967	703.415000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=428/44033, ttl=63
968	703.462000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=428/44033, ttl=254
969	704.435000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=429/44289, ttl=63
970	704.496000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=429/44289, ttl=254
971	705.443000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=430/44545, ttl=63
972	705.484000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=430/44545, ttl=254
973	706.454000	192.168.1.3	200.1.1.2	ICMP	98	Echo (ping) request id=0x3009, seq=431/44801, ttl=63
974	706.501000	200.1.1.2	192.168.1.3	ICMP	98	Echo (ping) reply id=0x3009, seq=431/44801, ttl=254

Figure 2.3.17 WireShark result for Packet transfer from 192.168.1.3 to 200.1.1.2

140	997.227000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=1/256, ttl=64
141	997.266000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=1/256, ttl=63
142	998.189000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=2/512, ttl=64
143	998.215000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=2/512, ttl=63
144	999.198000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=3/768, ttl=64
145	999.236000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=3/768, ttl=63
146	1000.03900	c4:08:3d:c0:00:00	c4:08:3d:c0:00:00	LOOP	60	Reply
147	1000.20000	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=4/1024, ttl=64
148	1000.22500	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=4/1024, ttl=63
149	1001.20400	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=5/1280, ttl=64
150	1001.23400	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=5/1280, ttl=63
151	1002.21200	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=6/1536, ttl=64
152	1002.23000	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=6/1536, ttl=63
153	1003.21700	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=7/1792, ttl=64
154	1003.24100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=7/1792, ttl=63
155	1004.13400	0.0.0.0	255.255.255.255	DHCP	342	DHCP Discover - Transaction ID 0xf127d725
156	1004.22100	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=8/2048, ttl=64
157	1004.26100	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=8/2048, ttl=63
158	1005.22500	10.1.1.1	172.16.1.1	ICMP	98	Echo (ping) request id=0x6209, seq=9/2304, ttl=64
159	1005.24300	172.16.1.1	10.1.1.1	ICMP	98	Echo (ping) reply id=0x6209, seq=9/2304, ttl=63

Figure 2.3.18 WireShark result for Packet transfer from 200.1.1.1 to 200.1.1.2

For these figures that we have provided, we are able to found that actually this is the pinging from 172.16.1.1, which is the PC1, to the Internet Gateway with 200.1.1.2. We use the NAT and PAT configuration for this situation and we are able to find that the IP address of PC1 would be changed into the public IP given to the PC1, which is 192.168.1.3. Then the public Ip would be used to transfer the packet to the public and the packet have to be translated by 200.1.1.1, the Edged Router before the request packet sent to the Internet Gateway. The transfer from private IP of PC1 to the public IP is based on the NAT configuration, while the PAT configuration with ACL 1 is used between the PC1 with public IP, the Edged Router and the Internet Gateway.

PART D: Configuration and result for Internet Gateway is able to connect to the Telnet Server using telnet, but unable to Ping to the Telnet Server

Internet Gateway is able to connect Telnet Server

```
Router>telnet 200.1.1.1
Trying 200.1.1.1 ... Open


User Access Verification

Password:
Router>exit

[Connection to 200.1.1.1 closed by foreign host]
Router>
```

Figure 2.3.19 Internet Gateway is connecting to Telnet Server

26	101.364000	c4:07:3d:c0:00:01	c4:07:3d:c0:00:01	LOOP	60	Reply
27	105.420000	200.1.1.2	200.1.1.1	TCP	60	38955 > telnet [SYN] Seq=0 Win=4128 Len=0 MSS=1460
28	105.481000	200.1.1.1	200.1.1.2	TCP	58	telnet > 38955 [SYN, ACK] Seq=0 Ack=1 Win=4128 Len=0 MSS=536
29	105.513000	200.1.1.2	200.1.1.1	TCP	60	38955 > telnet [ACK] Seq=1 Ack=1 Win=4128 Len=0
30	105.529000	200.1.1.2	200.1.1.1	TELNET	63	Telnet Data ...
31	105.544000	200.1.1.2	200.1.1.1	TCP	60	[TCP Dup ACK 30#1] 38955 > telnet [ACK] Seq=10 Ack=1 Win=4128 Len=0
32	105.576000	200.1.1.1	200.1.1.2	TELNET	66	Telnet Data ...
33	105.592000	200.1.1.1	200.1.1.2	TELNET	96	Telnet Data ...
34	105.592000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
35	105.607000	200.1.1.1	200.1.1.2	TELNET	57	Telnet Data ...
36	105.607000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
37	105.623000	200.1.1.1	200.1.1.2	TELNET	60	Telnet Data ...
38	105.624000	200.1.1.2	200.1.1.1	TELNET	63	Telnet Data ...
39	105.654000	200.1.1.1	200.1.1.2	TELNET	57	Telnet Data ...
40	105.795000	200.1.1.2	200.1.1.1	TCP	60	38955 > telnet [ACK] Seq=25 Ack=67 Win=4062 Len=0
41	105.887000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=25 Win=4104 Len=0
42	110.166000	c4:05:3d:c0:00:00	c4:05:3d:c0:00:00	LOOP	60	Reply
43	111.379000	c4:07:3d:c0:00:01	c4:07:3d:c0:00:01	LOOP	60	Reply
44	114.645000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
45	114.843000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
46	114.890000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=27 Win=4102 Len=0
47	115.182000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
48	115.321000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
49	115.415000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=29 Win=4100 Len=0
50	115.508000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
51	115.775000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=30 Win=4099 Len=0
52	115.807000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
53	115.979000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
54	116.072000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=32 Win=4097 Len=0
55	116.195000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
56	116.474000	200.1.1.1	200.1.1.2	TCP	54	telnet > 38955 [ACK] Seq=67 Ack=33 Win=4096 Len=0
57	116.738000	200.1.1.2	200.1.1.1	TELNET	60	Telnet Data ...
58	116.785000	200.1.1.1	200.1.1.2	TELNET	63	Telnet Data ...
59	116.986000	200.1.1.2	200.1.1.1	TCP	60	38955 > telnet [ACK] Seq=35 Ack=76 Win=4053 Len=0

Figure 2.3.20 WireShark result for Telnet Connection from Internet Gateway to Telnet Server

Based on these, we are able to find that we are using the IP address of the Telnet Server to have the connection of the Telnet sever. This is because we have used a command line, ip nat inside source tcp 192.168.1.2 23 200.1.1.1 23, for the configuration so that the Edged router will translate the traffic on reaching the

Telnet Server when telnet server have been trying to be connected by the Internet Gateway.

<u>Internet Gateway is unable to ping Telnet Server</u>

```
Router>ping 192.168.1.2

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
Router>
```

Figure 2.3.21 Internet Gateway is pinging to Telnet Server

Based on this figure, we are able to find that the Internet Gateway is unable to ping the Telnet server, this is because of the configuration of the Internet Gateway would not have any for the static route, NAT and PAT within the others internal devices or known as the private network for this situation, Hence the Internet Gateway have been configured as the public network and it is unable to ping the Telnet Server.