



FACULTY OF ELECTRONICS

COMPUTER ENGINEERING AND TELECOMMUNICATIONS DEPARTMENT

LABORATORY WORK #6

Internet and NAT Address Translation

Student: Mark Mikula

Teacher: Artūras Medeišis

Vilnius, 2025

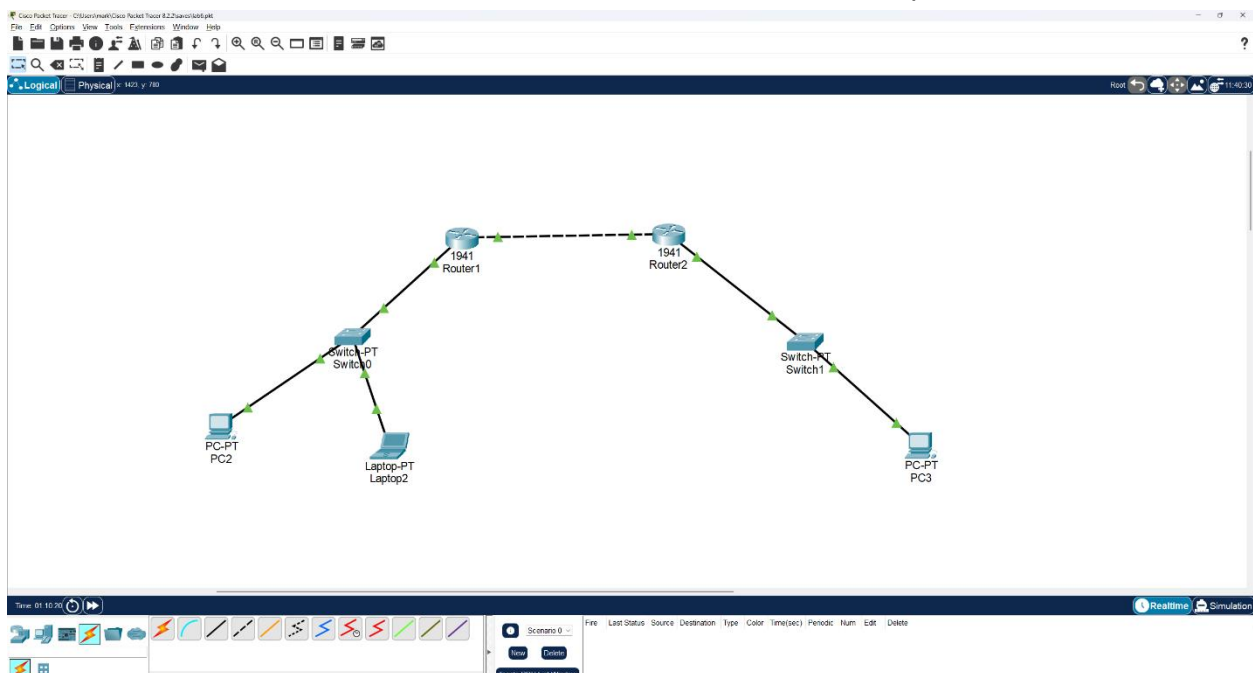
Objective

The primary objective of this lab work was to model the connection of two LANs over a public Internet network

This involved understanding the difference between private and public IP addresses and investigating the conversion of private-public IP addresses through the Network Address Translation (NAT) mechanism. The scenario simulated connecting two remote LANs over a public Internet connection

Topology Overview

The modelled scenario featured two LANs linked over a simulated public Internet network

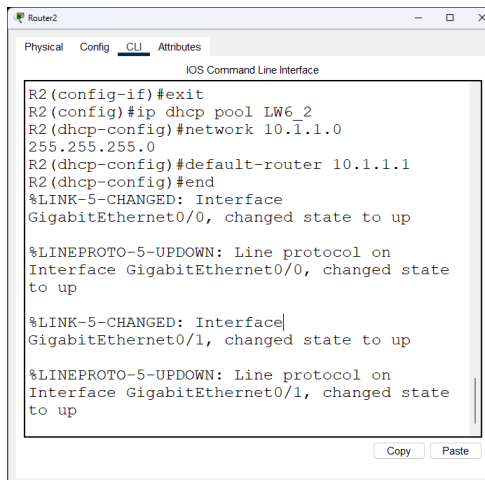


The connection diagram (Figure 1) involved two Cisco 1941 routers and two switches. The routers were connected directly to each other using a crossover cable, modelling the public Internet link. Each router was connected to a switch, which in turn connected to end devices (lab computers like PC1, Laptop on one side and PC2 on the other)

The simulated LANs used IP addresses from reserved private IP address ranges

One LAN was configured with the network 172.16.1.0/24, and the other with 10.1.1.0/24. The interconnection between the two networks was realized through the public Internet, which was modelled by having the two gateway routers use public IP addresses on their externally facing ports. Specifically, the external interface of R1 (Gi0/0) was assigned 8.8.8.11/8, and the external interface of R2 (Gi0/0) was assigned 8.8.8.22/8. R1's internal

interface (Gi0/1) used 172.16.1.1/24, serving the 172.16.1.0/24 network, and R2's internal interface (Gi0/1) used 10.1.1.1/24, serving the 10.1.1.0/24 network. DHCP pools were configured on each router to service their respective LAN subnets



```
R2(config-if)#exit
R2(config)#ip dhcp pool LW6_2
R2(dhcp-config)#network 10.1.1.0
255.255.255.0
R2(dhcp-config)#default-router 10.1.1.1
R2(dhcp-config)#end
%LINK-5-CHANGED: Interface
GigabitEthernet0/0, changed state to up

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

%LINK-5-CHANGED: Interface
GigabitEthernet0/1, changed state to up

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

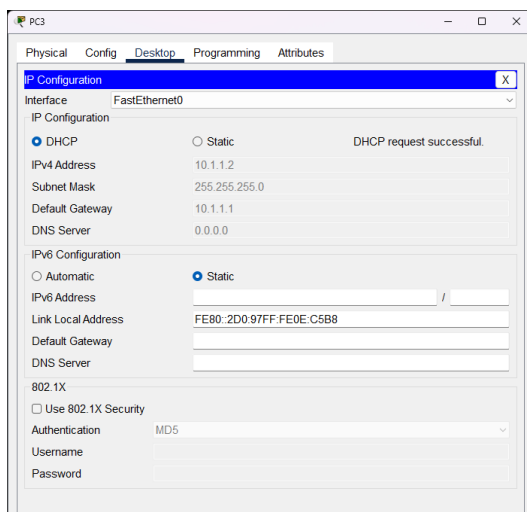
Lab Work Steps

The lab work followed a series of steps

1.

Initial Configuration: Cisco 1941 routers (R1 and R2) and switches were connected

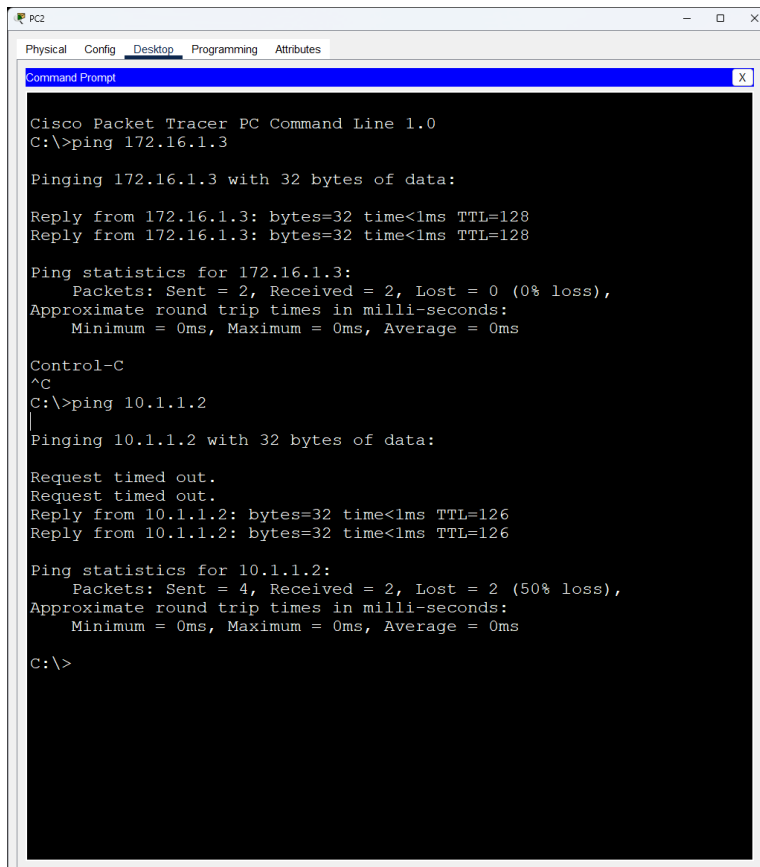
Console cables were used to configure each router according to the diagram and IP addressing scheme. Abbreviated CLI commands were used where possible. End devices (PC1, Laptop, PC2) were connected to their corresponding switches. Ethernet interfaces on lab computers were configured to belong to a 'Private network' to potentially disable firewall protection. Initial IP configurations were obtained, for example: PC1: 172.16.1.3, Laptop: 172.16.1.4, PC2: 10.1.1.3



2.

Initial Connectivity Testing (Before NAT): Connectivity between all terminals and router ports was tested using ping commands

The results were documented in a connectivity matrix. Firewall protection on destination computers was checked if ping responses were not received



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 172.16.1.3

Pinging 172.16.1.3 with 32 bytes of data:

Reply from 172.16.1.3: bytes=32 time<1ms TTL=128
Reply from 172.16.1.3: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.1.3:
    Packets: Sent = 2, Received = 2, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

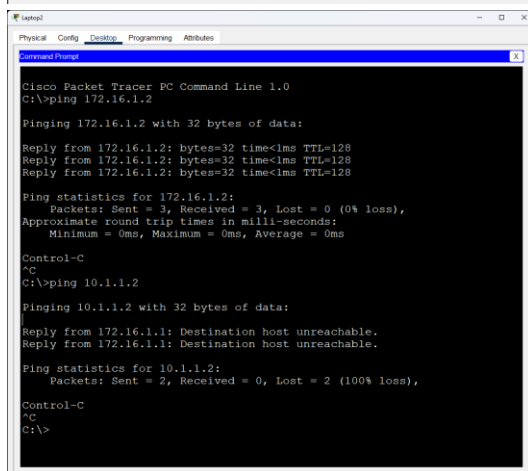
Control-C
^C
C:\>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 10.1.1.2: bytes=32 time<1ms TTL=126
Reply from 10.1.1.2: bytes=32 time<1ms TTL=126

Ping statistics for 10.1.1.2:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```



```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 172.16.1.2

Pinging 172.16.1.2 with 32 bytes of data:

Reply from 172.16.1.2: bytes=32 time<1ms TTL=128
Reply from 172.16.1.2: bytes=32 time<1ms TTL=128
Reply from 172.16.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 172.16.1.2:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

Control-C
^C
C:\>ping 10.1.1.2

Pinging 10.1.1.2 with 32 bytes of data:

Reply from 172.16.1.1: Destination host unreachable.
Reply from 172.16.1.1: Destination host unreachable.

Ping statistics for 10.1.1.2:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss),

Control-C
^C
C:\>
```

3.

Initial Connectivity Matrix (Before R1 Routing and NAT):

Terminal	IP address	PC1	Laptop	R1 Gi0/1	R1 Gi0/0	R2 Gi0/0	R2 Gi0/1	PC2
PC1	172.16.1.3		YES	YES	YES	NO	NO	NO
Laptop	172.16.1.4	YES		YES	YES	NO	NO	NO
R1 Gi0/1	172.16.1.1	YES	YES		YES	YES	NO	NO
R1 Gi0/0	8.8.8.11	YES	YES	YES		YES	NO	NO
R2 Gi0/0	8.8.8.22	NO	NO	NO	YES		YES	YES
R2 Gi0/1	10.1.1.1	NO	NO	NO	YES	YES		YES
PC2	10.1.1.3	NO	NO	NO	NO	YES	YES	

4.

Based on the results

Successful pings occurred between devices on the same subnet and switch (e.g., PC1 to Laptop, PC1/Laptop to R1 Gi0/1)

Pings between R1's external interface (8.8.8.11) and R2's external interface (8.8.8.22) were also successful as they are on the same "public internet" segment

Failed pings occurred between devices on different LANs (e.g., PC1/Laptop to R2 interfaces or PC2) due to different subnets and the lack of routing to bridge these networks

PC2 could ping R2's interfaces (Gi0/0 and Gi0/1) and vice-versa as they are on the same network segment or directly connected

5.

Additional R1 Configuration (Static Route and NAT): As full connectivity between both LANs was not achieved initially

additional configuration was performed on R1

A static default route was added to R1, instructing it to send all packets with an unknown destination network (0.0.0.0/0) to R2's public IP address (8.8.8.22)

This ensures traffic destined for networks beyond R1's directly connected segments (like R2's internal network) is forwarded towards R2

An NAT address pool named LW6 was created, consisting of a single public IP address (8.8.8.11)

NAT was configured to translate packets from internal IP addresses defined in source list #1 to the public IP address in the LW6 pool, using the NAT overload method

An access-list 1 permit any was configured, setting any internally connected terminals to be part of internal access list #1

Interface Gi0/1 was configured as the internal side of the NAT conversion (ip nat inside)

Interface Gi0/0 was configured as the external side of the NAT conversion (ip nat outside)

6.

Retesting Connectivity (After NAT): Connectivity was re-tested from PC1 (and Laptop) to destinations that were previously unreachable

A second version of the connectivity matrix was prepared

7.

Second Connectivity Matrix (After R1 Routing and NAT):

Terminal	IP address	PC1	Laptop	R1 Gi0/1	R1 Gi0/0	R2 Gi0/0	R2 Gi0/1	PC2
PC1	172.16.1.3		YES	YES	YES	YES	YES	YES
Laptop	172.16.1.4	YES		YES	YES	YES	YES	YES
R1 Gi0/1	172.16.1.1	YES	YES		YES	YES	YES	YES
R1 Gi0/0	8.8.8.11	YES	YES	YES		YES	YES	YES
R2 Gi0/0	8.8.8.22	NO	NO	NO	YES		YES	YES
R2 Gi0/1	10.1.1.1	NO	NO	NO	YES	YES		YES
PC2	10.1.1.3	YES	YES	YES	YES	YES	YES	

8.

After configuring static routing and NAT on R1, full connectivity was achieved between most devices

The default route allowed R1 to forward packets to R2, and NAT enabled communication between different subnets by translating internal IPs.

9.

Packet Analysis (Wireshark): Wireshark was launched on PC1 and PC2 to log network traffic

A ping was sent from PC1 to PC2. The logs were analyzed to observe Layer 2 (MAC) and Layer 3 (IP) addressing changes due to NAT

10.

Outbound ping packet sent from PC1 to PC2 (before NAT by R1):

PDU Information at Device: PC3

OSI Model Inbound PDU Details Outbound PDU Details

At Device: PC3
Source: PC2
Destination: 10.1.1.2

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 8.8.8.11, Dest. IP: 10.1.1.2 ICMP Message Type: 8	Layer 3: IP Header Src. IP: 10.1.1.2, Dest. IP: 8.8.8.11 ICMP Message Type: 0
Layer 2: Ethernet II Header 0001.42EB.4202 >> 00D0.970E.C5B8	Layer 2: Ethernet II Header 00D0.970E.C5B8 >> 0001.42EB.4202
Layer 1: Port FastEthernet0	Layer 1: Port(s): FastEthernet0

1. The ICMP process replies to the Echo Request by setting ICMP type to Echo Reply.
2. The ICMP process sends an Echo Reply.
3. The destination IP address 8.8.8.11 is not in the same subnet and is not the broadcast address.
4. The default gateway is set. The device sets the next-hop to default gateway.

Challenge Me << Previous Layer Next Layer >>

11.

Inbound ping packet from PC1, as received at PC2 (after NAT by R1):

PDU Information at Device: PC2

OSI Model Inbound PDU Details

At Device: PC2
Source: PC2
Destination: 10.1.1.2

In Layers	Out Layers
Layer7	Layer7
Layer6	Layer6
Layer5	Layer5
Layer4	Layer4
Layer 3: IP Header Src. IP: 10.1.1.2, Dest. IP: 172.16.1.2 ICMP Message Type: 0	Layer3
Layer 2: Ethernet II Header 0090.2B58.7B02 >> 0001.6357.E9E2	Layer2
Layer 1: Port FastEthernet0	Layer1

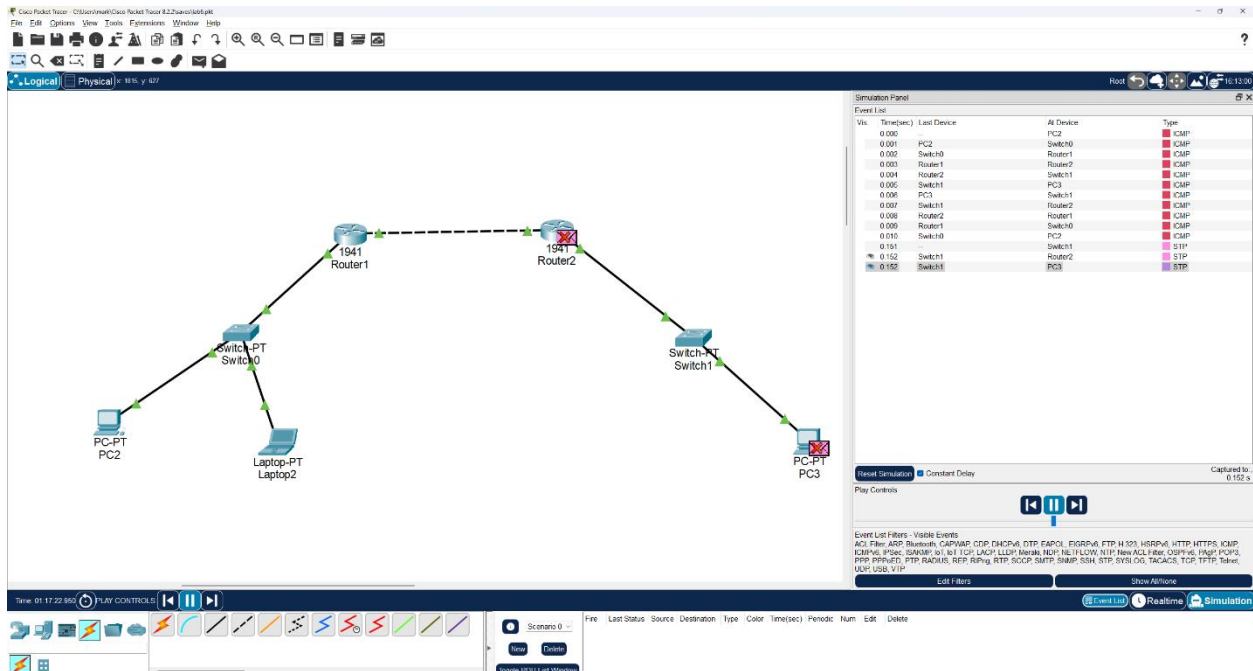
1. FastEthernet0 receives the frame.

Challenge Me << Previous Layer Next Layer >>

12.

When PC1 pings PC2, R1 performs NAT, replacing PC1's private source IP (172.16.1.3) with its public IP (8.8.8.11)

PC2 receives the ping with the source IP of 8.8.8.11. The MAC addresses change at each hop; the source MAC arriving at PC2 would be R2's internal interface MAC, and the destination MAC would be PC2's MAC



13.

Answering Questions:

What is NAT and why did enabling it on the R1 router help with specific connectivity issues? Network Address Translation (NAT) allows multiple devices on a local (private) network to share a single public IP address for accessing external networks like the Internet

It translates private IP addresses of local devices to a public IP address when traffic leaves the private network and translates the public IP back to the correct private IP for incoming responses. Enabling NAT on R1 resolved connectivity issues by allowing devices on the private network behind R1 (PC1, Laptop) to communicate with devices on the private network behind R2 (PC2) by translating their private source IPs to R1's public IP. This makes the traffic appear to originate from R1's public address, which is routable across the "public internet" segment between R1 and R2

Why was the 'NAT overload' method chosen when programming the NAT functionality? NAT overload, also known as Port Address Translation (PAT), was chosen because it is an efficient method that allows numerous devices on a local network to share a single public IP address

It achieves this by using unique source port numbers to keep track of the different connections from each internal device. This method is highly effective for conserving limited public IP addresses, making it a cost-effective and scalable solution for providing Internet access to multiple devices behind a single router

Conclusion

This lab successfully demonstrated how to connect two LANs using reserved private IP addresses over a simulated public Internet network

The core concept of NAT was explored, highlighting the difference between private and public IP addresses. By implementing NAT on the gateway router (R1), devices on the private network were able to communicate with devices on the other private network across the public link. The lab provided practical experience with configuring static routing and NAT, including the use of the NAT overload method, and troubleshooting connectivity issues. Analyzing packet headers with Wireshark confirmed the IP address translation performed by NAT. The use of connectivity matrices proved valuable in understanding the state of inter-subnet communication before and after the implementation of routing and NAT. This exercise enhanced practical skills in network configuration, particularly concerning IP addressing, routing, and address translation