EXPERIMENT 7

AIM

Examine network address translation and configure a NAT simulation using cisco packet tracer.

THEORY

Definition
Network Address Translation (NAT) is a method that modifies IP address information in packet headers while they traverse a router or firewall. It translates between private (non-routable) IP addresses used inside a local network and public IP addresses used on the global Internet. NAT operates at the IP layer and can also change transport ports when required.

How NAT works (basic mechanics)
1. A host on the private network sends a packet with source = inside-local (private IP) and destination = outside-global (public server).
2. The NAT device rewrites the source address to an inside-global (public) address and may rewrite the source port.
3. The NAT device stores an entry in its translation table that maps inside-local:port to inside-global:port.
4. The packet travels on the public network to the destination.
5. The server replies to the inside-global address. The NAT device receives the reply and consults the translation table.
6. The NAT device rewrites the destination address (and port if needed) back to the inside-local address and forwards the packet to the host.
7. Translation entries are created on demand and may expire after an inactivity timeout.

Key terms: inside local (private IP), inside global (public IP used by private host on Internet), outside local (private IP representation of remote host, rarely used), outside global (actual remote public IP).
NAT maintains a translation table. For PAT entries the table includes ports.

Types of NAT (detailed)
- **Static NAT (one-to-one)**
  Maps one inside-local address to one inside-global address permanently.
  Use case: publish an internal server with a fixed public IP.
  Behavior: predictable. Outside hosts can initiate connections to mapped public IP.
  Configuration concept: ip nat inside source static <local-ip> <global-ip>.
- **Dynamic NAT (many-to-many using pool)**
  Maps inside-local addresses to addresses from a pool of inside-global addresses.
  Mappings are temporary and allocated from the pool when needed.
  Use case: limited public address pool for outbound connections.
  Limitation: if pool is exhausted new translations fail.
- **Port Address Translation (PAT) / NAT Overload (many-to-one)**
  Maps multiple inside-local addresses to a single inside-global address by using unique source ports.

Default method for conserving IPv4 addresses.
Translation table stores source port mappings.
Use case: home routers, ISPs, enterprise edge when few public IPs available.
Configuration concept: ip nat inside source list <acl> interface <if> overload or pool with overload.

- **Static Port Forwarding (Destination NAT / Port forwarding)**
  Forwards traffic arriving at a specific public IP:port to a specific internal host:port.
  Use case: host internal services (HTTP, SSH) behind NAT.
  Example: map public 203.0.113.10:80 to 192.168.1.100:80.
- **Twice NAT / Policy NAT / Identity NAT**
  Allows independent translation of source and destination addresses and ports in a single rule.
  Use case: complex policies where both source and destination must change.
  Supported on advanced platforms.
- **NAT64 / NAT46 (protocol translation) — brief note**
  Translate between IPv6 and IPv4 in specialized deployments.
  Not relevant for basic IPv4 PAT but important where IPv4/IPv6 coexist.

Objectives of NAT

- Conserve public IPv4 addresses.
- Enable hosts with private addresses to access the public Internet.
- Allow selective publishing of internal services to the Internet.
- Provide a simple layer of topology hiding between internal hosts and the public network.
- Simplify address renumbering of internal networks without changing global allocations.

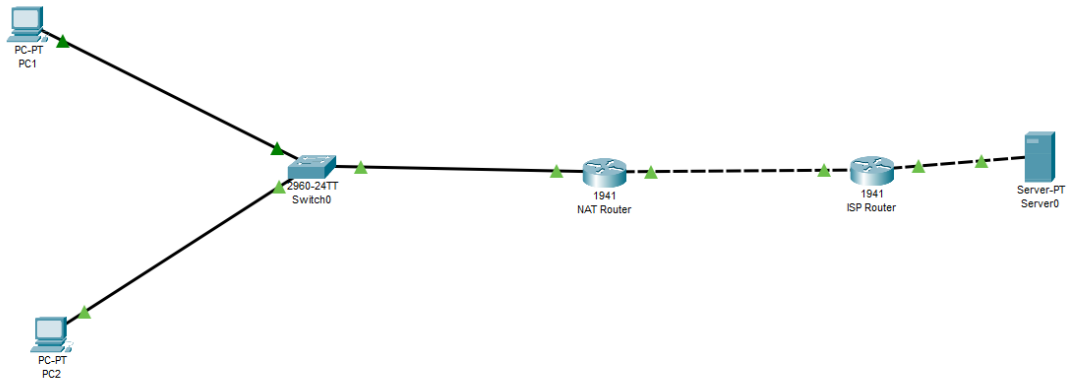
Advantages of NAT

- IPv4 address economy. Many internal hosts share few public IPs.
- Security by obscurity. Internal addresses are not directly visible to the Internet.
- Flexibility in internal addressing. Internal networks can use RFC1918 ranges without coordination.
- Enables legacy networks to connect to the Internet without readdressing.
- Simple port forwarding supports hosting services behind NAT.

Disadvantages and limitations of NAT

- Breaks end-to-end IP semantics. Source address changes interfere with applications that embed IP addresses.
- Complicates protocols that use IPs or ports inside payloads (FTP active mode, SIP, H.323, some VPNs). Application-level gateways or ALG may be required.
- Limits inbound connections unless explicit static mappings or port forwards exist.
- Adds translation state. Devices must maintain translation table that can be exhausted under high load.
- Introduces NAT timeouts that can disrupt long-lived or idle connections.
- Makes troubleshooting and forensics harder because source IPs on the public side differ from internal hosts.
- Interferes with end-to-end authentication that relies on original IP addresses.
- Can complicate multicast and some routing scenarios.
- Performance cost. NAT requires CPU and memory on the device doing translation.

PROCEDURE:



1. Build topology

1. Place devices.
    - 2 × PC-PT (PC1, PC2).
    - 1 × Switch-2960.
    - 2 × Router-1941 (R1 = NAT router, R2 = ISP router).
    - 1 × Server-PT.
2. Cable devices.
    - PC1 → Switch port Fa0/1 (copper straight-through).
    - PC2 → Switch port Fa0/2.
    - Switch Fa0/24 → R1 GigabitEthernet0/0.
    - R1 GigabitEthernet0/1 → R2 GigabitEthernet0/0 (copper).
    - R2 GigabitEthernet0/1 → Server NIC.
3. Power on devices. Wait for interface LEDs to turn green.

2. Configure end hosts (PCs and Server)

Use Desktop > IP Configuration on each device in Packet Tracer or configure via CLI where available.

PC1:

```
IP: 192.168.1.10
Mask: 255.255.255.0
Gateway: 192.168.1.1
DNS: leave blank or 8.8.8.8
```

PC2:

```
IP: 192.168.1.11
Mask: 255.255.255.0
Gateway: 192.168.1.1
```

Server:

```
IP: 198.51.100.10
Mask: 255.255.255.0
Gateway: 198.51.100.1
Optional: enable HTTP service in Server > Services > HTTP for web testing.
```

3. Basic interface configuration on routers

Enter Router CLI. Use enable then configure terminal.

R1 (NAT router):

```
interface GigabitEthernet0/0
 description LAN to switch
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 no shutdown

interface GigabitEthernet0/1
 description WAN to ISP
 ip address 203.0.113.2 255.255.255.252
 ip nat outside
 no shutdown

! Optional: enable CEF for performance
ip cef

! Save later
end
write memory
```

R2 (ISP router):

```
interface GigabitEthernet0/0
 description Link to R1
 ip address 203.0.113.1 255.255.255.252
 no shutdown

interface GigabitEthernet0/1
 description Link to Server LAN
 ip address 198.51.100.1 255.255.255.0
 no shutdown

ip route 0.0.0.0 0.0.0.0 Null0   ! optional placeholder
end
write memory
```

4. Configure routing (return routes)

On R1 add default route to ISP:

```
configure terminal
ip route 0.0.0.0 0.0.0.0 203.0.113.1
end
write memory
```

On R2 add route back to 192.168.1.0/24 via R1:

```
configure terminal
ip route 192.168.1.0 255.255.255.0 203.0.113.2
end
write memory
```

5. Configure NAT — PAT (NAT overload) — primary lab step

1. Create ACL that matches the inside network.

```
access-list 1 permit 192.168.1.0 0.0.0.255
```

2. Configure PAT using R1 outside interface as global address.

```
ip nat inside source list 1 interface GigabitEthernet0/1 overload
```

3. Save config.

```
end
write memory
```

OBSERVATION:

Perform tests in this order. Run commands on the device indicated.

1. Local link tests
    - From PC1 > Command Prompt: ping 192.168.1.1 (R1 LAN IP). Expect success.

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.1.1

Pinging 192.168.1.1 with 32 bytes of data:

Reply from 192.168.1.1: bytes=32 time<1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=1ms TTL=255
Reply from 192.168.1.1: bytes=32 time=2ms TTL=255
Reply from 192.168.1.1: bytes=32 time=9ms TTL=255

Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 9ms, Average = 3ms

C:\>
```

    - From R1 CLI: ping 203.0.113.1 (R2). Expect success.

```
Router#ping 203.0.113.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 203.0.113.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/0 ms
```

- From R2 CLI: ping 198.51.100.10 (Server). Expect success.

```
Router#ping 198.51.100.10

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 198.51.100.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/0/1 ms
```

2. NAT test (outbound)
   - From PC1: ping 198.51.100.10 (Server). Expect replies if NAT and routes correct.

```
C:\>ping 198.51.100.10

Pinging 198.51.100.10 with 32 bytes of data:

Reply from 198.51.100.10: bytes=32 time<1ms TTL=126
Reply from 198.51.100.10: bytes=32 time<1ms TTL=126
Reply from 198.51.100.10: bytes=32 time<1ms TTL=126
Reply from 198.51.100.10: bytes=32 time<1ms TTL=126

Ping statistics for 198.51.100.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

   - From PC1: open Server HTTP in Packet Tracer browser http://198.51.100.10 if HTTP service enabled.

3. Check translation table on R1

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
tcp 203.0.113.2:1025   192.168.1.10:1025  198.51.100.10:80   198.51.100.10:80
```

4. Check NAT counters

```
Router#show ip nat statistics
Total translations: 1 (0 static, 1 dynamic, 1 extended)
Outside Interfaces: GigabitEthernet0/1
Inside Interfaces: GigabitEthernet0/0
Hits: 39  Misses: 28
Expired translations: 8
Dynamic mappings:
```

5. Confirm routing tables

```
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 203.0.113.1 to network 0.0.0.0

     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
     203.0.113.0/24 is variably subnetted, 2 subnets, 2 masks
C       203.0.113.0/30 is directly connected, GigabitEthernet0/1
L       203.0.113.2/32 is directly connected, GigabitEthernet0/1
S*   0.0.0.0/0 [1/0] via 203.0.113.1
```

6. Trace path if ping fails

- From PC1 CLI: `tracert 198.51.100.10` (Windows style in Packet Tracer PC).
- On routers use `debug ip packet` and `debug ip nat` briefly if needed.

CONCLUSION:

In this lab experiment, we studied and configured various aspects of network IP addressing. We explored the structure and classification of IPv4 addresses, including Classes A, B, C, and D, understanding their ranges, default subnet masks, and typical use cases. We also examined IPv6 addressing, learning about its expanded 128-bit format, address types such as unicast, multicast, and anycast, and the notation rules including zero compression and prefix length.