EXPERIMENT- 11

Implementation of Network Address Translation

By

**Madhurya Mozumder (RA1911028010036)**

Under the guidance of

**Mrs. P. Visalakshi**

*In partial fulfilment for the course*

Of

**18CSC302J- COMPUTER NETWORKS**



**FACULTY OF ENGINEERING AND TECHNOLOGY**

**SRM ISNTITUTE OF SCEINCE AND TECHNOLOGY**

Kattankulathur, Chengalpattu District

November 2021

TABLE OF CONTENTS

1. Abstract
2. Requirement Analysis
3. Procedure & Implementation
4. Architecture & Design Implementation
5. Experiment Observation
6. Result

**Abstract**

A Network Address Translation (NAT) is the process of mapping an [internet protocol (IP) address](https://whatis.techtarget.com/definition/IP-address-Internet-Protocol-Address) to another by changing the header of IP packets while in transit via a router. This helps to improve security and decrease the number of IP addresses an organization needs.

A NAT works by selecting gateways that sit between two local networks: the internal network, and the outside network. Systems on the inside network are typically assigned IP addresses that cannot be routed to external networks (e.g., networks in the 10.0.0.0/8 block).

A few externally valid IP addresses are assigned to the gateway. The gateway makes outbound traffic from an inside system appear to be coming from one of the valid external addresses. It takes incoming traffic aimed at a valid external address and sends it to the correct internal system.

This helps ensure security. Because each outgoing or incoming request must go through a translation process that offers the opportunity to qualify or authenticate incoming streams and match them to outgoing requests, for example.

NAT conserves the number of globally valid IP addresses a company needs and -- in combination with Classless Inter-Domain Routing (CIDR) -- has done a lot to extend the useful life of IPv4 as a result. NAT is described in general terms in IETF RFC 1631.

The NAT mechanism ("natting") is a router feature, and is often part of a corporate firewall. NAT gateways can map IP addresses in several ways:

* from a local IP address to one global IP address statically;
* hiding an entire IP address space comprised of private IP addresses behind a single IP address;
* to a large private network using a single public IP address using translation tables;
* from a local IP address plus a particular TCP port to a global address or a pool of public IP addresses; and
* from a global IP address to any of a pool of local IP addresses on a round-robin basis.

In some cases, network administrators define policies that allow the gateway device to assign mappings based on the intended destination ("pick this external address for communications to partner A's area network; pick that external address for communications to partner B's").

Policies can also be used on the protocols being used ("assign out of this pool for HTTP traffic, that pool for HTTPS") or on other factors.

A newer way to use NAT focuses on translating an ISP provider's IPv4 addresses to IPv6, and vice versa. This provides integration of IPv4 infrastructure and end nodes into IPv6 environments, and allows IPv6 services to interact with IPv4 systems.

Cisco, which has developed a technique that uses a NAT overload to map several private IP addresses to a single public IP address.

Conversely, a static NAT, also common in large organizations, provides a 1:1 mapping between an internal IP address and a public network IP address.

**Requirement Analysis:**

**Hardware Requirement:**

* A Server
* Two 1841 Routers
* A 2960-24TT Switch
* Two PCs
* Connecting Wires

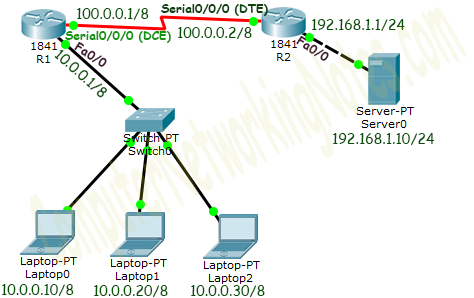
**Software Requirement:**

* Windows Server for operating system in the server
* Antivirus with Internet and network security installed in the software
* Monitoring System like Solar Winds Networking Monitor to map switches, routers and nodes and control data flow in the network

**Procedure & Implementation**

To explain Static NAT Configuration, I will use packet tracer network simulator software. You can use any network simulator software or can use real Cisco devices to follow this guide. There is no difference in output as long as your selected software contains the commands explained in this tutorial.

Create a practice lab as shown in following figure or download this pre-created practice lab and load in packet tracer

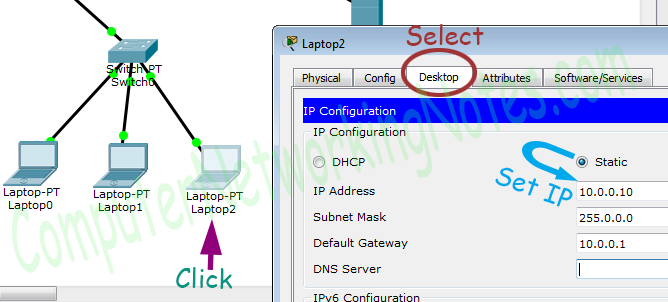


**Initial IP Configuration**

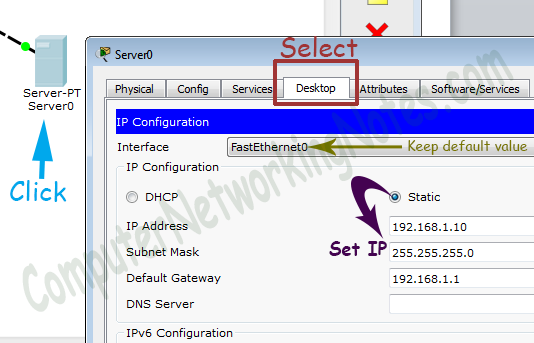
| Device / Interface | IP Address | Connected With |
| --- | --- | --- |
| Laotop0 | 10.0.0.10/8 | Fa0/0 of R0 |
| Laptop1 | 10.0.0.20/8 | Fa0/0 of R0 |
| Laptop2 | 10.0.0.30/8 | Fa0/0 of R0 |
| Server0 | 192.168.1.10/24 | Fa0/0 of R1 |
| Serial 0/0/0 of R1 | 100.0.0.1/8 | Serial 0/0/0 of R2 |
| Serial 0/0/0 of R2 | 100.0.0.2/8 | Serial 0/0/0 of R2 |

*If you are following this tutorial on my practice topology, skip this IP configuration section as that topology is already configured with this initial IP configuration*

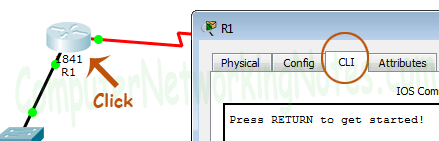
To assign IP address in Laptop click **Laptop** and click **Desktop** and **IP configuration** and Select **Static** and **set IP address** as given in above table.



Following same way configure IP address in Server.



To configure IP address in Router1 click **Router1** and select **CLI** and press **Enter key**.



Two interfaces of Router1 are used in topology; FastEthernet0/0 and Serial 0/0/0.

By default interfaces on router are remain administratively down during the start up. We need to configure IP address and other parameters on interfaces before we could actually use them for routing. Interface mode is used to assign the IP address and other parameters. Interface mode can be accessed from global configuration mode. Following commands are used to access the global configuration mode.

Router>enable

Router# configure terminal

Enter configuration commands, one per line. End with CNTL/Z.

Router(config)#

Before we configure IP address in interfaces let’s assign a unique descriptive name to router.

Router(config)#hostname R1

R1#

Now execute the following commands to set IP address in FastEthernet 0/0 interface.

R1(config)#interface FastEthernet0/0

R1(config-if)#ip address 10.0.0.1 255.0.0.0

R1(config-if)#no shutdown

R1(config-if)#exit

***interface FastEthernet 0/0*** command is used to enter in interface mode.

***ip address 10.0.0.1 255.0.0.0*** command assigns IP address to interface.

***no shutdown*** command is used to bring the interface up.

***exit*** command is used to return in global configuration mode.

Serial interface needs two additional parameters clock rate and bandwidth. Every serial cable has two ends DTE and DCE. These parameters are always configured at DCE end.

We can use show controllers interface command from privilege mode to check the cable’s end.

R1(config)#exit

R1#show controllers serial 0/0/0

Interface Serial0/0/0

Hardware is PowerQUICC MPC860

DCE V.35, clock rate 2000000

[Output omitted]

Fourth line of output confirms that DCE end of serial cable is attached. If you see DTE here instead of DCE skip these parameters.

Now we have necessary information let’s assign IP address to serial interface.

R1#configure terminal

R1(config)#interface Serial0/0/0

R1(config-if)#ip address 100.0.0.1 255.0.0.0

R1(config-if)#clock rate 64000

R1(config-if)#bandwidth 64

R1(config-if)#no shutdown

R1(config-if)#exit

R1(config)#

Router#configure terminal Command is used to enter in global configuration mode.

***Router(config)#interface serial 0/0/0*** Command is used to enter in interface mode.

***Router(config-if)#ip address 100.0.0.1 255.0.0.0*** Command assigns IP address to interface.

**Router(config-if)#clock rate 64000**

In real life environment this parameter controls the data flow between serial links and need to be set at service provider’s end. In lab environment we need not to worry about this value. We can use any valid rate here.

**Router(config-if)#bandwidth 64**

Bandwidth works as an influencer. It is used to influence the metric calculation of EIGRP or any other routing protocol which uses bandwidth parameter in route selection process.

***Router(config-if)#no shutdown*** Command brings interface up.

***Router(config-if)#exit*** Command is used to return in global configuration mode.

We will use same commands to assign IP addresses on interfaces of Router2. We need to provided clock rate and bandwidth only on DCE side of serial interface. Following command will assign IP addresses on interface of Router2.

Initial IP configuration in R2

Router>enable

Router#configure terminal

Router(config)#hostname R2

R2(config)#interface FastEthernet0/0

R2(config-if)#ip address 192.168.1.1 255.255.255.0

R2(config-if)#no shutdown

R2(config-if)#exit

R2(config)#interface Serial0/0/0

R2(config-if)#ip address 100.0.0.2 255.0.0.0

R2(config-if)#no shutdown

R2(config-if)#exit

R2(config)#

That’s all initial IP configuration we need. Now this topology is ready for the practice of static nat.

Configure Static NAT

Static NAT configuration requires three steps: -

1. Define IP address mapping
2. Define inside local interface
3. Define inside global interface

Since static NAT use manual translation, we have to map each inside local IP address (which needs a translation) with inside global IP address. Following command is used to map the inside local IP address with inside global IP address.

Router(config)#ip nat inside source static [inside local ip address] [inside global IP address]

For example in our lab Laptop1 is configured with IP address 10.0.0.10. To map it with 50.0.0.10 IP address we will use following command

Router(config)#ip nat inside source static 10.0.0.10 50.0.0.10

In second step we have to define which interface is connected with local the network. On both routers interface Fa0/0 is connected with the local network which need IP translation.

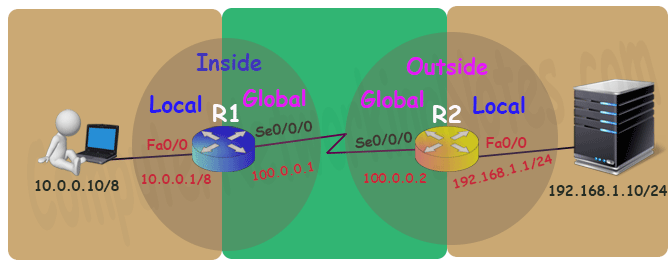
Following command will define interface Fa0/0 as inside local.

Router(config-if)#ip nat inside

In third step we have to define which interface is connected with the global network. On both routers serial 0/0/0 interface is connected with the global network. Following command will define interface Serial0/0/0 as inside global.

Router(config-if)#ip nat outside

Following figure illustrates these terms.



Let’s implement all these commands together and configure the static NAT.

R1 Static NAT Configuration

R1(config)#ip nat inside source static 10.0.0.10 50.0.0.10

R1(config)#interface FastEthernet 0/0

R1(config-if)#ip nat inside

R1(config-if)#exit

R1(config)#

R1(config)#interface Serial 0/0/0

R1(config-if)#ip nat outside

R1(config-if)#exit

For testing purpose I configured only one static translation. You may use following commands to configure the translation for remaining address.

R1(config)#ip nat inside source static 10.0.0.20 50.0.0.20

R1(config)#ip nat inside source static 10.0.0.30 50.0.0.30

R2 Static NAT Configuration

R2(config)#ip nat inside source static 192.168.1.10 200.0.0.10

R2(config)#interface FastEthernet 0/0

R2(config-if)#ip nat inside

R2(config-if)#exit

R2(config)#

R2(config)#interface Serial 0/0/0

R2(config-if)#ip nat outside

R2(config-if)#exit

Before we test this lab we need to configure the IP routing. IP routing is the process which allows router to route the packet between different networks. Following tutorial explain routing in detail with examples

Configure static routing in R1

R1(config)#ip route 200.0.0.0 255.255.255.0 100.0.0.2

Configure static routing in R2

R2(config)#ip route 50.0.0.0 255.0.0.0 100.0.0.1

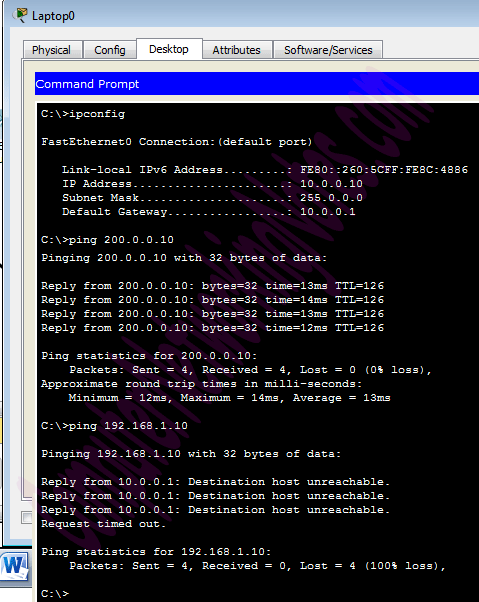
Testing Static NAT Configuration

In this lab we configured static NAT on R1 and R2. On R1 we mapped inside local IP address 10.0.0.10 with inside global address 50.0.0.10 while on R2 we mapped inside local IP address 192.168.1.10 with inside global IP address 200.0.0.10.

| Device | Inside Local IP Address | Inside Global IP Address |
| --- | --- | --- |
| Laptop0 | 10.0.0.10 | 50.0.0.10 |
| Server | 192.168.1.10 | 200.0.0.10 |

To test this setup click Laptop0 and Desktop and click Command Prompt.

* Run **ipconfig** command.
* Run **ping 200.0.0.10** command.
* Run **ping 192.168.1.10** command.

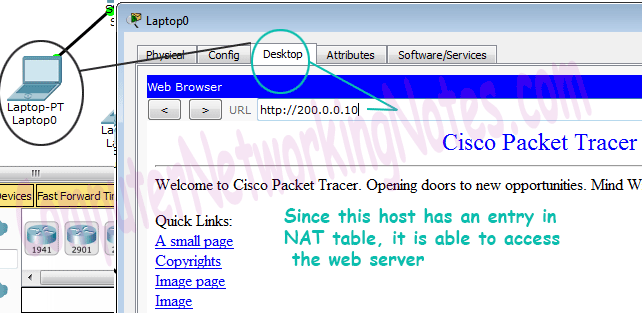


First command verifies that we are testing from correct NAT device.

Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

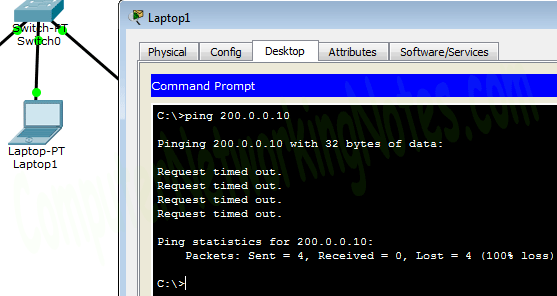
Third command checks whether we are able to access the remote device on its actual IP address or not. A ping error confirms that we are not able to connect with remote device on this IP address.

Let’s do one more testing. Click **Laptop0** and click **Desktop** and click **Web Browser** and access 200.0.0.10.



Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10.

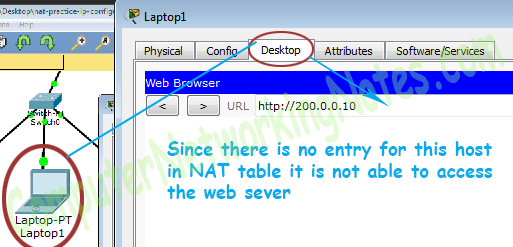
Now run **ping 200.0.0.10** command from Laptop1.



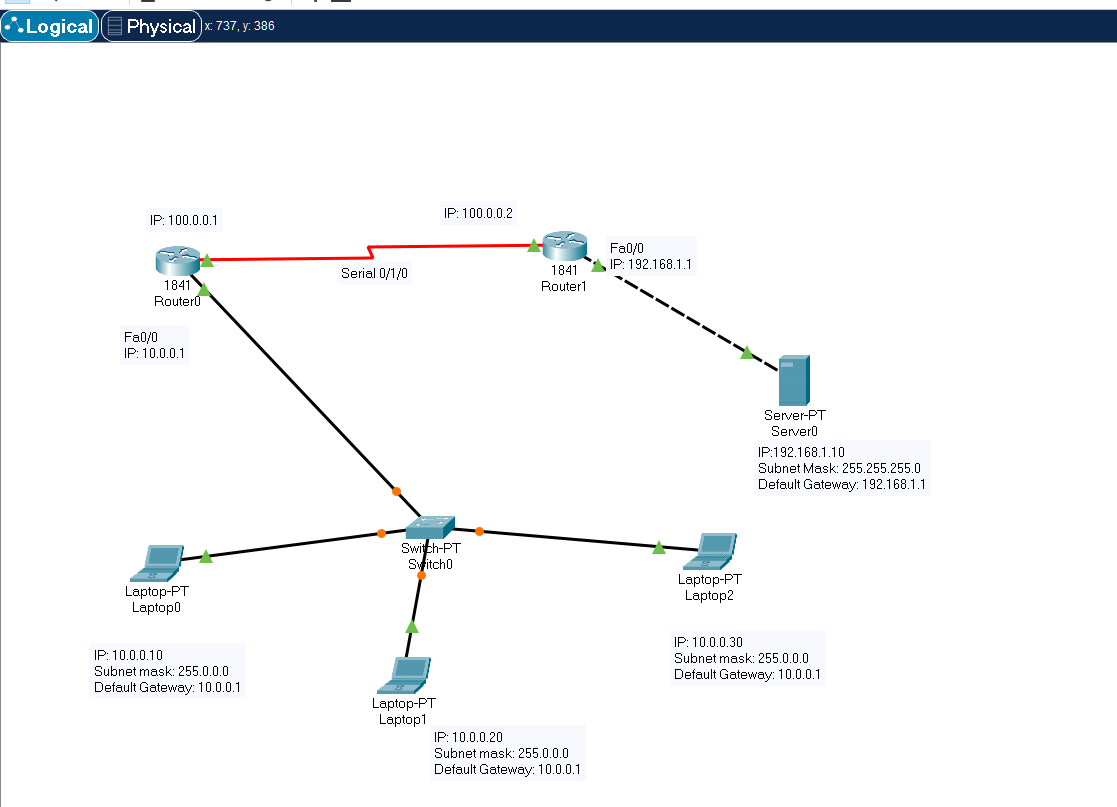
*Why we are not able to connect with the remote device from this host?*

Because we configured NAT only for one host (Laptop0) which IP address is 10.0.0.10. So only the host 10.0.0.10 will be able to access the remote device.

To confirm it again, let’s try to access web service from this host.



**Architecture & Design Implementation**

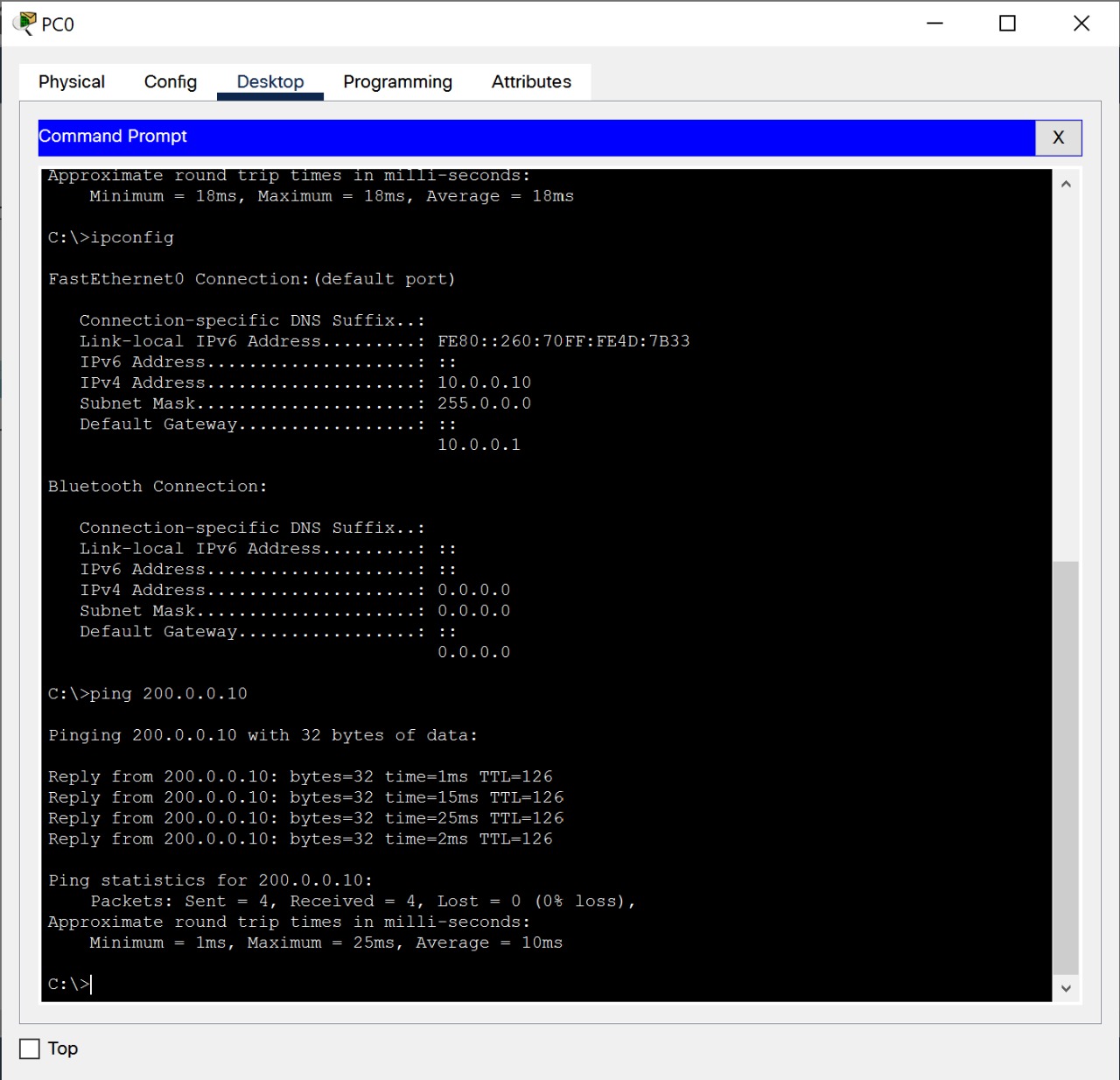
****

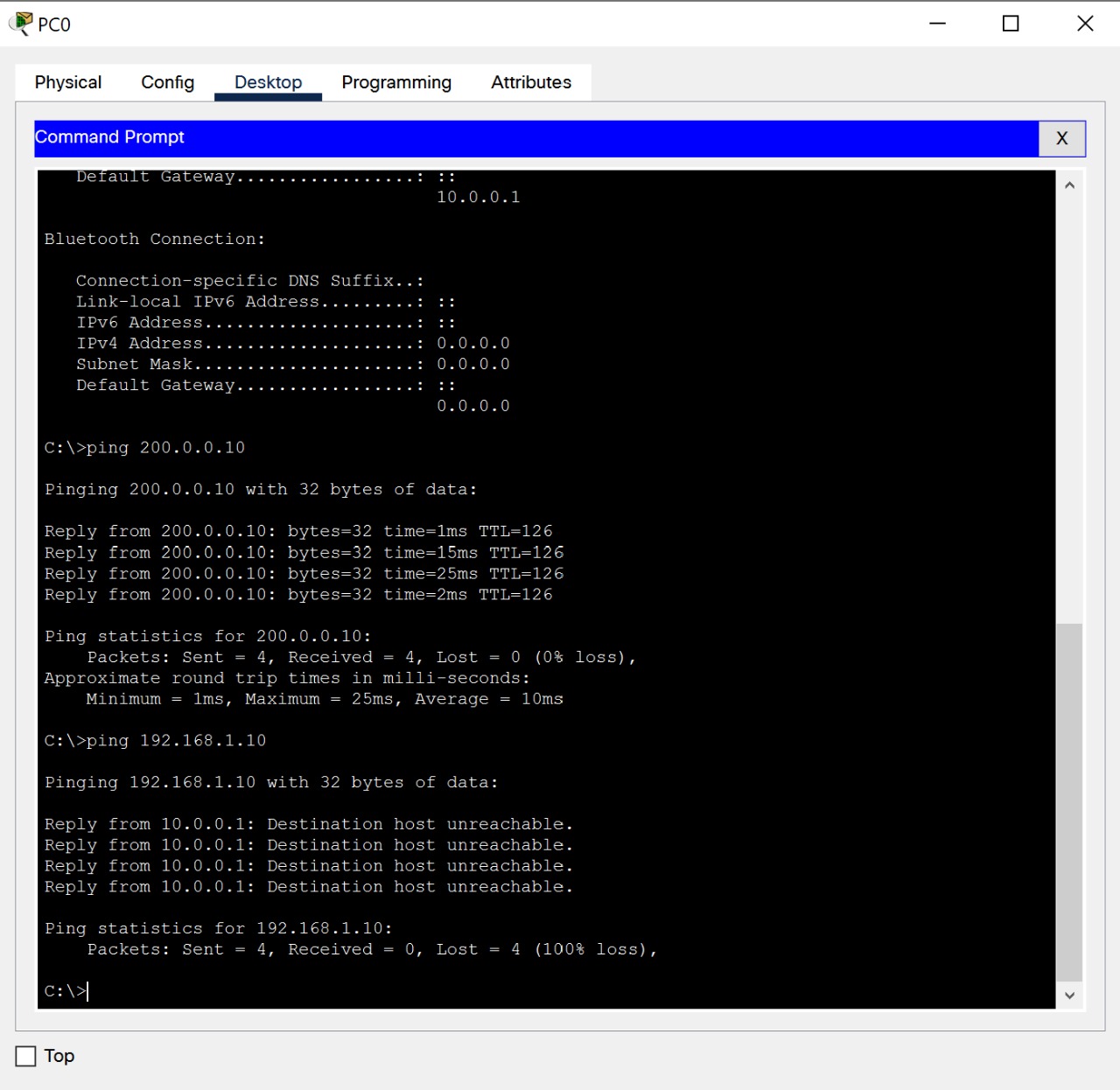
The design has been implemented in Cisco Packet Tracer, following the procedures given in the implementation section.

* First, two 1841 routers were taken and they were connected through the Serial Wire.
* Then a 2960-24TT switch was taken which was connected to the Router 0 using a copper straight wire.
* Two PCs were taken and connected to the switch with the help of copper straight wires.
* A server was taken which was connected to Router 1 with a copper cross-over cable.
* All the IP addresses are assigned as shown in the table of the Procedure and Implementation section and labelled for convenience.
* The CLI commands as given in the Procedure and Implementation section were put in the CLI of both Router 0 and Router 1 to initiate connectivity.
* Static Routing for the Inside Global IP addresses are carried out at both Router 0 and Router 1.
* Once all the connections were made, the connections were checked by pinging the PCs to the server and the Inside Global IP addresses.
* The observations are as shown below.

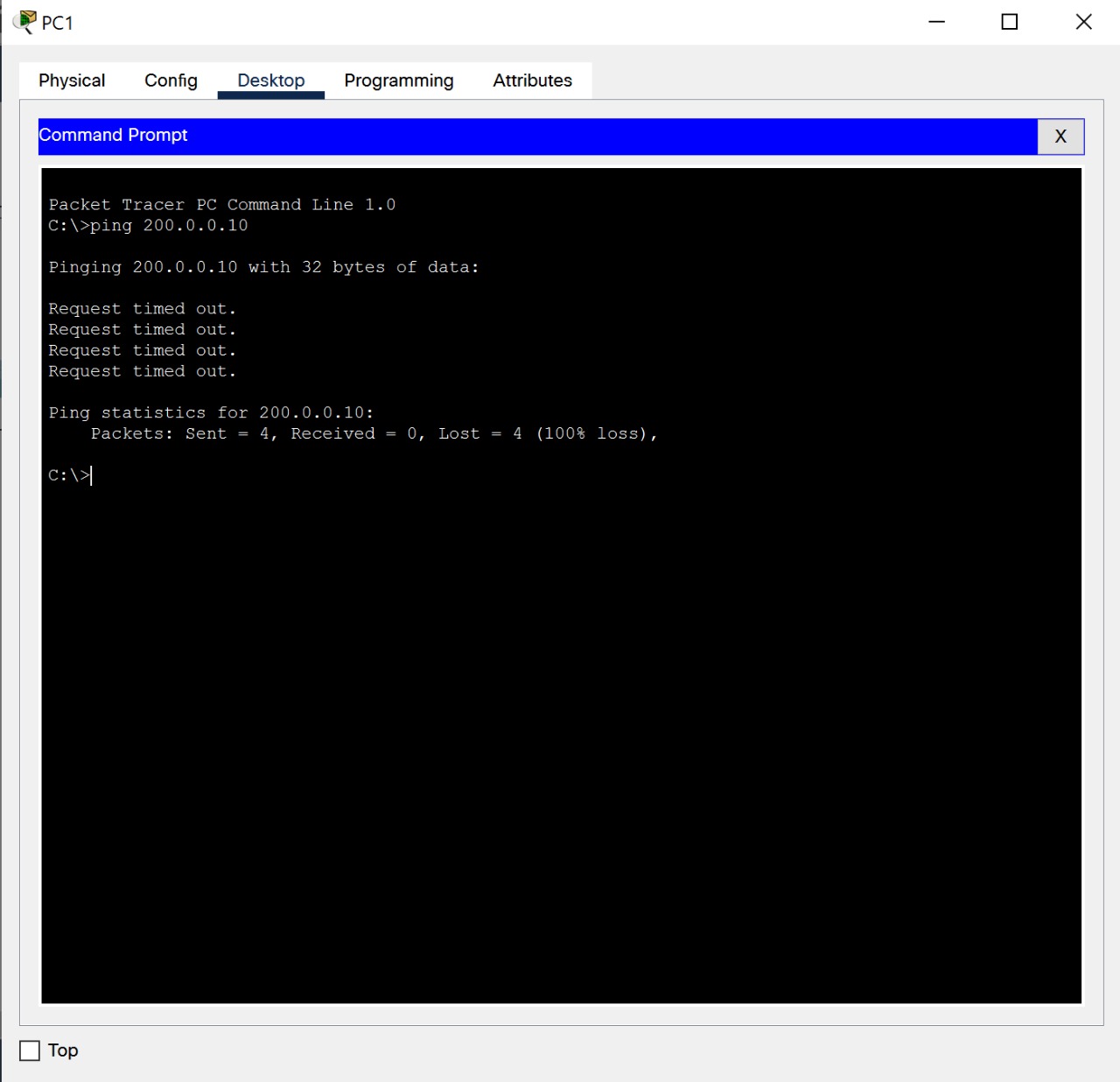
**Experiment Observation**

**PC0 Command Prompt**

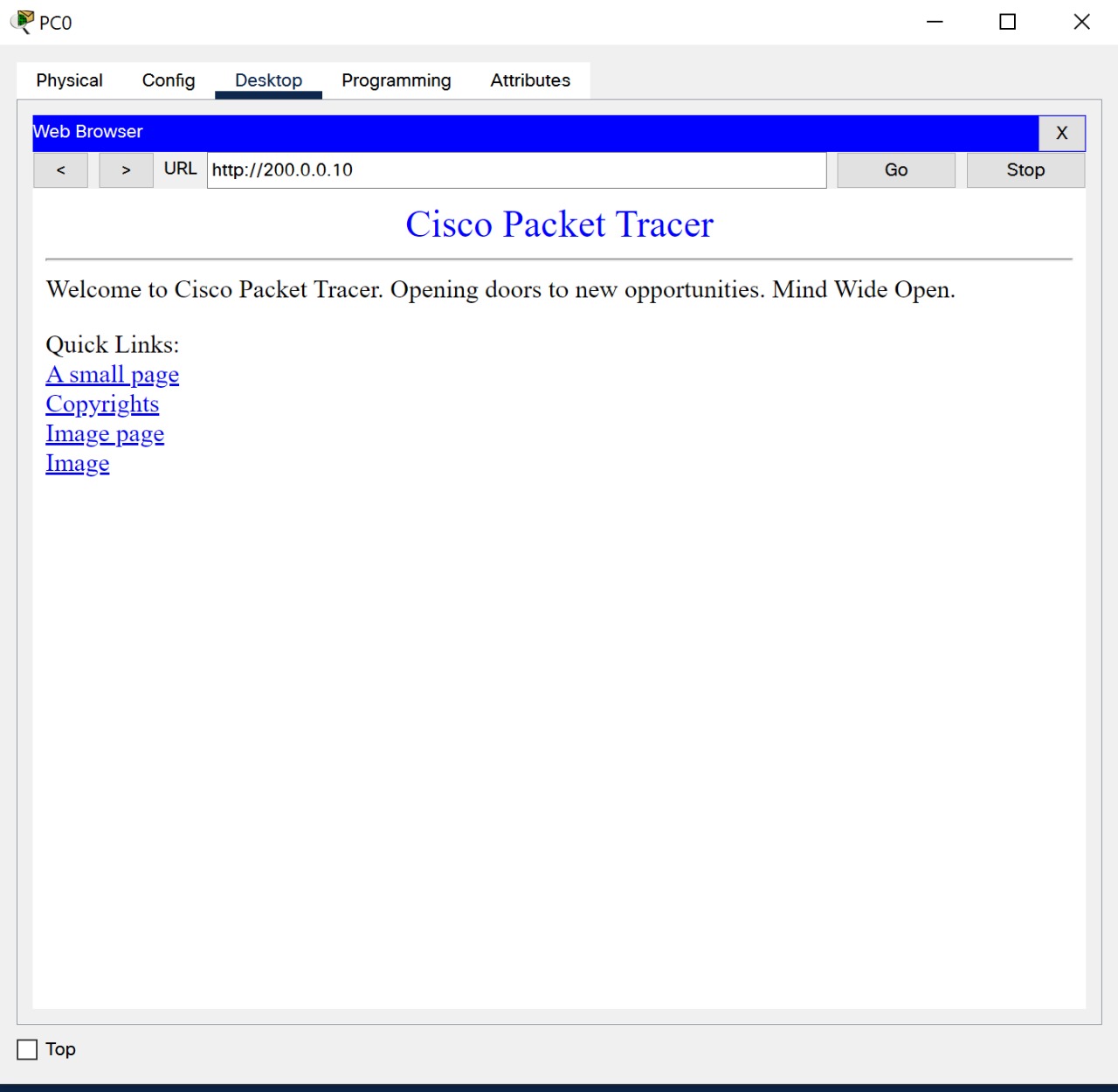
****



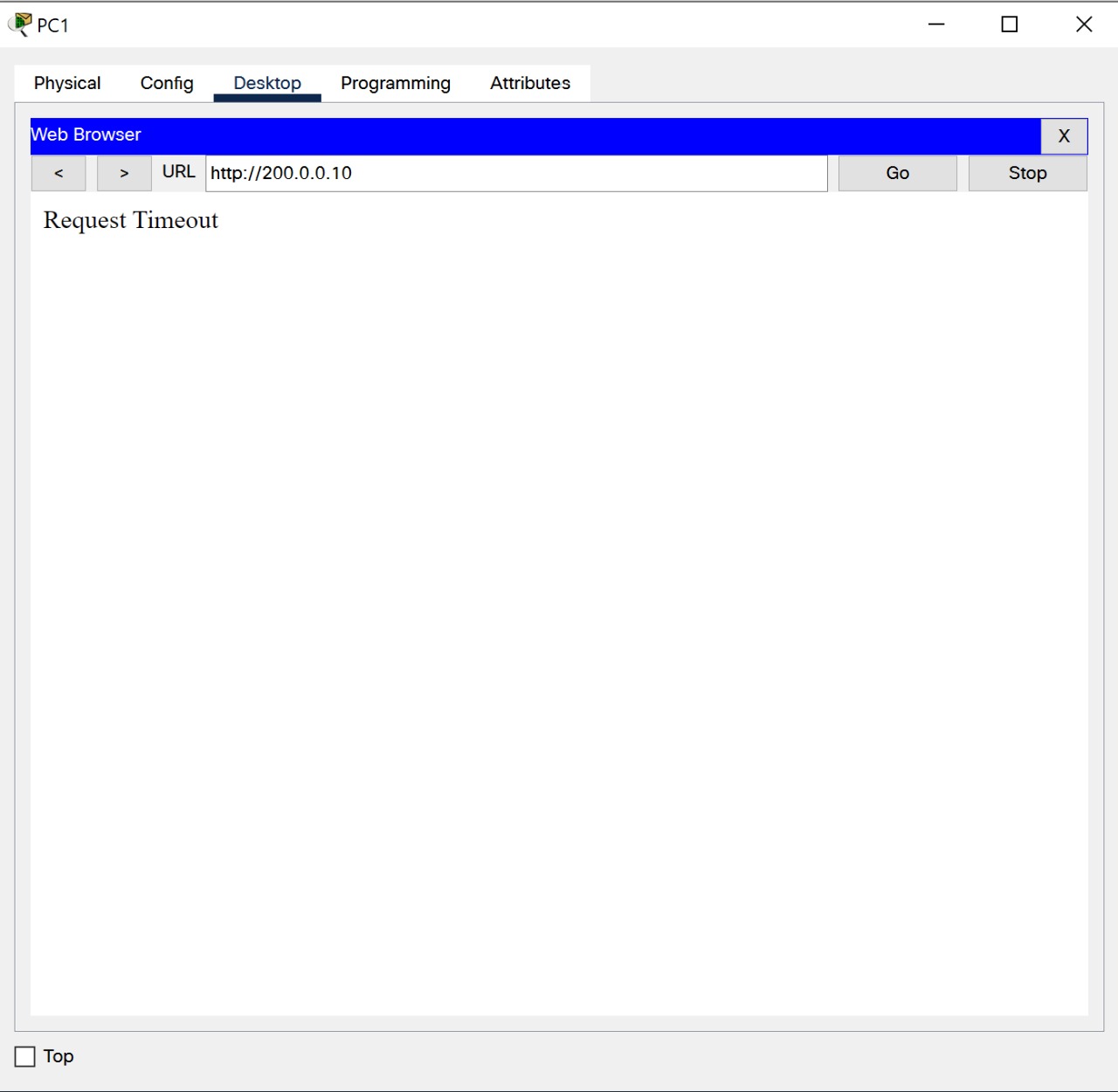
**PC1 Command Prompt**

****

**PC0 Browser**

****

**PC1 Browser**

****

**Result:**

Network Address Translation (NAT) was successfully build and implemented.