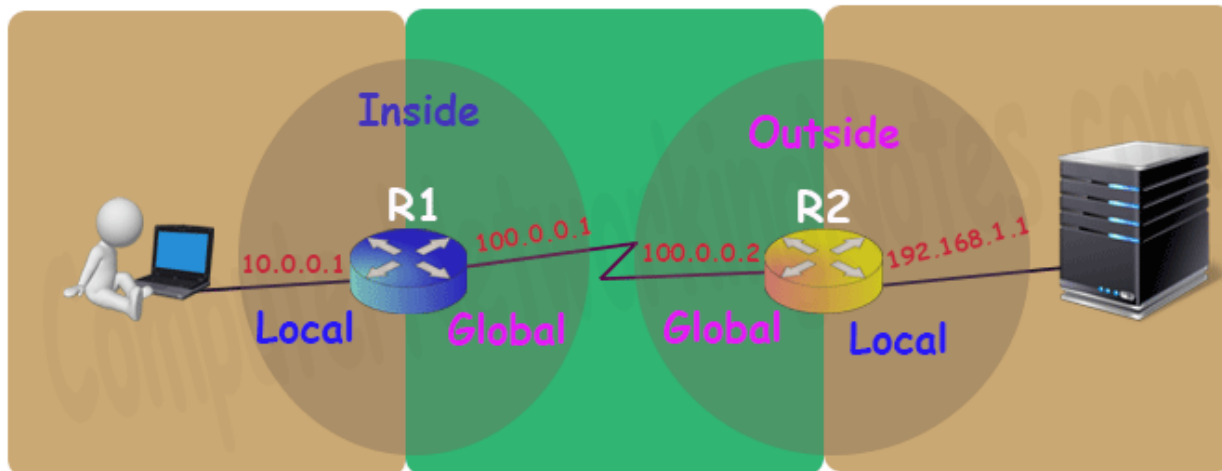


# NAT(Network Address Translation)

## NAT Terminology

Before we understand NAT in details let's get familiar with four basic terms used in NAT.

Term	Description
Inside Local IP Address	Before translation source IP address located inside the local network.
Inside Global IP Address	After translation source IP address located outside the local network.
Outside Global IP Address	Before translation destination IP address located outside the remote network.
Outside Local IP Address	After translation destination IP address located inside the remote network.



what about outside global and outside local? Well... these terms are used to explain the NAT process theoretically. Practically we never need to configure the outside local and outside global as they sound

## Types of NAT

There are three types of NAT; Static NAT, Dynamic NAT and PAT. These types define how inside local IP address will be mapped with inside global IP address.

### Static NAT

In this type we manually map each inside local IP address with inside global IP address. Since this type uses one to one mapping we need exactly same number of IP address on both sides.

### Dynamic NAT

In this type we create a pool of inside global IP addresses and let the NAT device to map inside local IP address with the available outside global IP address from the pool automatically.

## PAT

In this type a single inside global IP address is mapped with multiple inside local IP addresses using the source port address. This is also known as PAT (Port Address Translation) or NAT over load.

### *Situations where NAT is used*

There are no hard and fast rules about where we should use NAT or where we should not use the NAT. Whether we should use the NAT or not is purely depends on network requirement for example NAT is the best solution in following situations: -

- Our network is built with private IP addresses and we want to connect it with internet. As we know to connect with internet we require public IP address. In this situation we can use NAT device which will map private IP address with public IP address.
- Two networks which are using same IP address scheme want to merge. In this situation NAT device is used to avoid IP overlapping issue.
- We want to connect multiple computers with internet through the single public IP address. In this situation NAT is used to map the multiple IP addresses with single IP address through the port number.

## Advantages and disadvantages of NAT

Nat provides following advantages: -

- NAT solves IP overlapping issue.
- NAT hides internal IP structure from external world.
- NAT allows us to connect with any network without changing IP address.
- NAT allows us to connect multiple computers with internet through the single the public IP address.

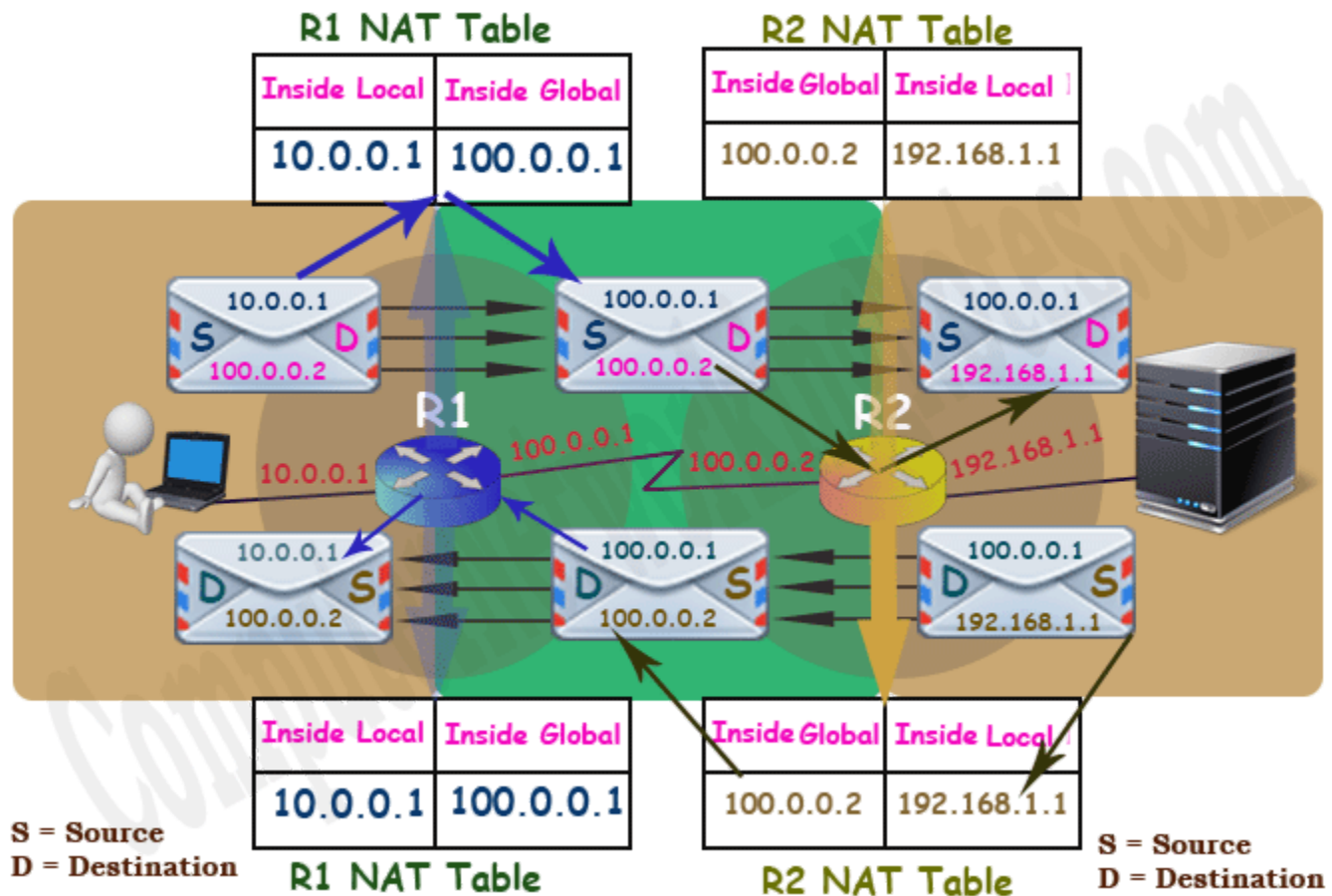
NAT has following disadvantages: -

- NAT adds additional delay in network.
- Several applications are not compatible with NAT.
- End to end IP traceability will not work with NAT.
- NAT hides actual end device.

## How NAT Works

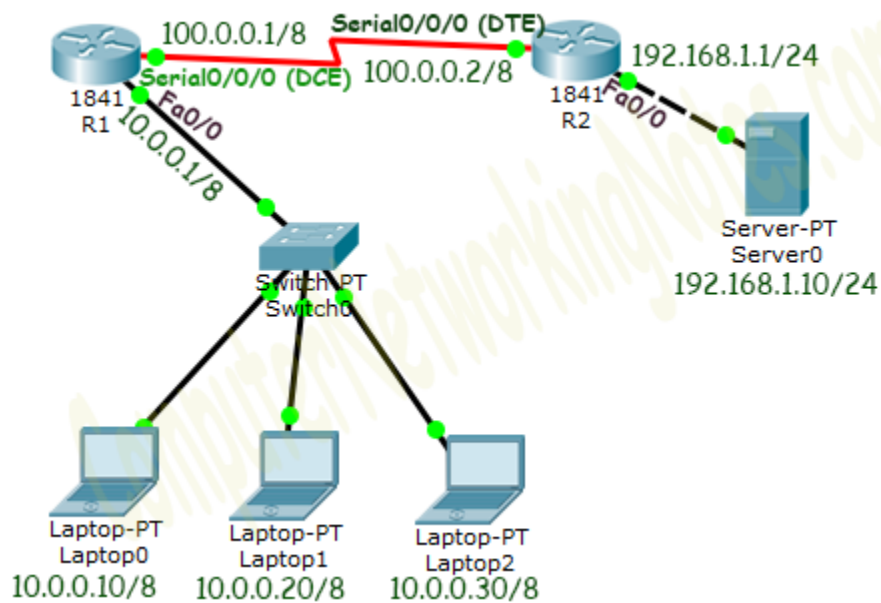
To understand how NAT works, let's take one more example. In this example a user is accessing a web server. User and Webserver both are connected through the NAT

devices. Both user and webserver are using private IP addresses which are not routable on the internet. Now let's understand how NAT makes this communication possible.



## STATIC NAT

## Static NAT Practice LAB Setup



## Initial IP Configuration

Device / Interface	IP Address	Connected With
Laptop0	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

## Configure Static NAT

Static NAT configuration requires three steps: -

- Define IP address mapping
- Define inside local interface
- 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]
```

## 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
```

## 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

We can also verify this translation on router with **show ip nat translation** command. Following figure illustrate this translation on router R1.

```
R1#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp 50.0.0.10:13      10.0.0.10:13      200.0.0.10:13      200.0.0.10:13
icmp 50.0.0.10:14      10.0.0.10:14      200.0.0.10:14      200.0.0.10:14
icmp 50.0.0.10:15      10.0.0.10:15      200.0.0.10:15      200.0.0.10:15
icmp 50.0.0.10:16      10.0.0.10:16      200.0.0.10:16      200.0.0.10:16
tcp 50.0.0.10:1030      10.0.0.10:1030      200.0.0.10:80      200.0.0.10:80
tcp 50.0.0.10:1031      10.0.0.10:1031      200.0.0.10:80      200.0.0.10:80
R1#
```

Following figure illustrate this translation on router R2

```
R2#show ip nat translations
Pro Inside global      Inside local      Outside local      Outside global
icmp 200.0.0.10:13      192.168.1.10:13      50.0.0.10:13      50.0.0.10:13
icmp 200.0.0.10:14      192.168.1.10:14      50.0.0.10:14      50.0.0.10:14
icmp 200.0.0.10:15      192.168.1.10:15      50.0.0.10:15      50.0.0.10:15
icmp 200.0.0.10:16      192.168.1.10:16      50.0.0.10:16      50.0.0.10:16
tcp 200.0.0.10:80      192.168.1.10:80      50.0.0.10:1030      50.0.0.10:1030
tcp 200.0.0.10:80      192.168.1.10:80      50.0.0.10:1031      50.0.0.10:1031
R2#
```

## DYNAMIC NAT

### Dynamic NAT Practice LAB Setup



## Initial IP Configuration

Device / Interface	IP Address	Connected With
Laptop0	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

## Configure Dynamic NAT

Dynamic NAT configuration requires four steps: -

- Create an access list of IP addresses which need translation
- Create a pool of all IP address which are available for translation
- Map access list with pool
- Define inside and outside interfaces

In first step we will create a standard access list which defines which inside local addresses are permitted to map with inside global address.

To create a standard numbered ACL following global configuration mode command is used:-

```
Router(config)# access-list ACL_Identifier_number permit/deny  
matching-parameters
```

Let's understand this command and its options in detail.

### **access-list**

Through this parameter we tell router that we are creating or accessing an access list.

### **ACL\_Identifier\_number**

With this parameter we specify the type of access list. We have two types of access list; standard and extended. Both lists have their own unique identifier numbers. Standard ACL uses numbers range 1 to 99 and 1300 to 1999. We can pick any number from this range to tell the router that we are working with standard ACL. This number is used in grouping the conditions under a single ACL. This number is also a unique identifier for this ACL in router.

### **permit/deny**

An ACL condition has two actions; permit and deny. If we use permit keyword, ACL will allow all packets from the source address specified in next parameter. If we use deny keyword, ACL will drop all packets from the source address specified in next parameter.

### **matching-parameters**

This parameter allows us to specify the contents of packet that we want to match. In a standard ACL condition it could be a single source address or a range of addresses. We have three options to specify the source address.

1. Any
2. host

### **Any**

Any keyword is used to match all sources. Every packet compared against this condition would be matched.

### **Host**

Host keyword is used to match a specific host. To match a particular host, type the keyword host and then the IP address of host.

```
R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
```



```
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
```

```
R1(config)#access-list 1 deny any
```

In second step we define a pool of inside global addresses which are available for translation.

Following command is used to define the NAT pool.

```
Router(config)#ip nat pool [Pool Name] [Start IP address] [End  
IP address] netmask [Subnet mask]
```

This command accepts four options pool name, start IP address, end IP address and Subnet mask

**Pool Name:** - This is the name of pool. We can choose any descriptive name here.

**Start IP Address:** - First IP address from the IP range which is available for translation.

**End IP Address:** - Last IP address from the IP range which is available for translation. There is no minimum or maximum criteria for IP range for example we can have a range of single IP address or we can have a range of all IP address from a subnet.

**Subnet Mask:** - Subnet mask of IP range.

Let's create a pool named **abc** with an IP range of two addresses.

```
R1(config)#ip nat pool abc 50.0.0.1 50.0.0.2 netmask 255.0.0.0
```

This pool consist two class A IP address 50.0.0.1 and 50.0.0.2.

In third step we map access list with pool. Following command will map the access list with pool and configure the dynamic NAT.

```
Router(config)#ip nat inside source list [access list name  
or number] pool [pool name]
```

This command accepts two options.

**Access list name or number:** - Name or number the access list which we created in first step.

**Pool Name:** - Name of pool which we created in second step.

In first step we created a standard access list with number **1** and in second step we created a pool named **abc**. To configure a dynamic NAT with these options we will use following command.

```
R1(config)#ip nat inside source list 1 pool abc
```

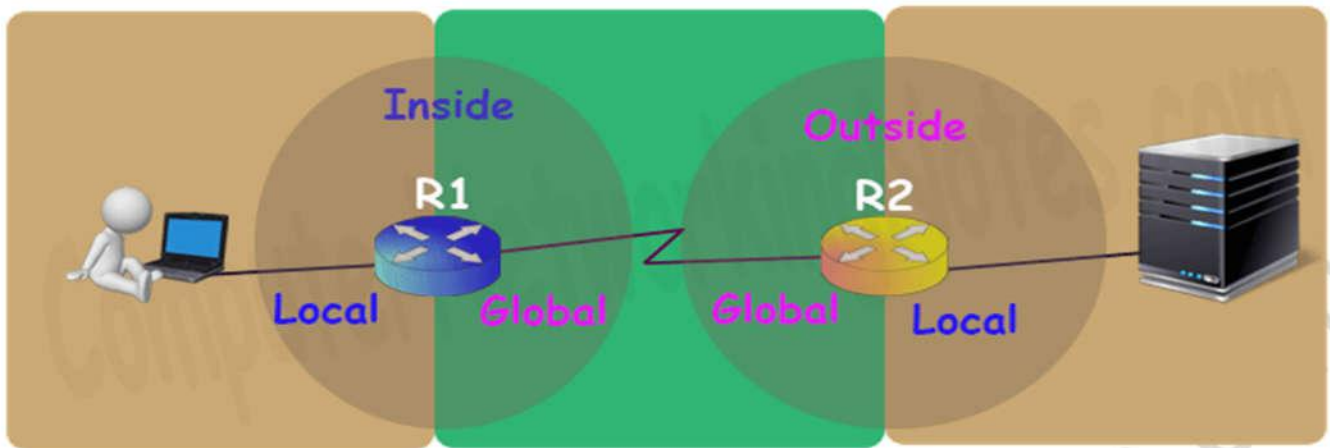
Finally we have to define which interface is connected with local network and which interface is connected with global network.

To define an inside local we use following command

```
Router(config-if)#ip nat inside
```

Following command defines inside global

```
Router(config-if)#ip nat outside
```



Let's implement all these commands together and configure the dynamic NAT.

## R1 Dynamic NAT Configuration

```
R1#configure terminal

Enter configuration commands, one per line. End with
CNTL/Z. R1(config)#access-list 1 permit 10.0.0.10 0.0.0.0
R1(config)#access-list 1 permit 10.0.0.20 0.0.0.0
R1(config)#access-list 1 deny any

R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.2 netmask
255.0.0.0 R1(config)#ip nat inside source list 1 pool ccna
R1(config)#interface FastEthernet 0/0 R1(config-if)#ip nat
inside

R1(config-if)#exit

R1(config)#interface Serial0/0/0

R1(config-if)#ip nat outside

R1(config-if)#exit

R1(config)#
```

*For testing purpose I configured dynamic translations for two addresses only.*

On R2 we can keep standard configuration or can configure dynamic NAT as we just did in R1 or can configure static NAT as we learnt in pervious part of this article.

Let's do a quick recap of what we learnt in previous part and configure static NAT on R2.

```
R2>enable
```

```
R2#configure terminal
```

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

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

```
R2(config)#interface Serial 0/0/0 R2(config-if)#ip nat outside
```

```
R2(config-if)#exit
```

```
R2(config)#interface FastEthernet 0/0
```

```
R2(config-if)#ip nat inside
```

```
R2(config-if)#exit
```

```
R2(config)#
```

### 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
```

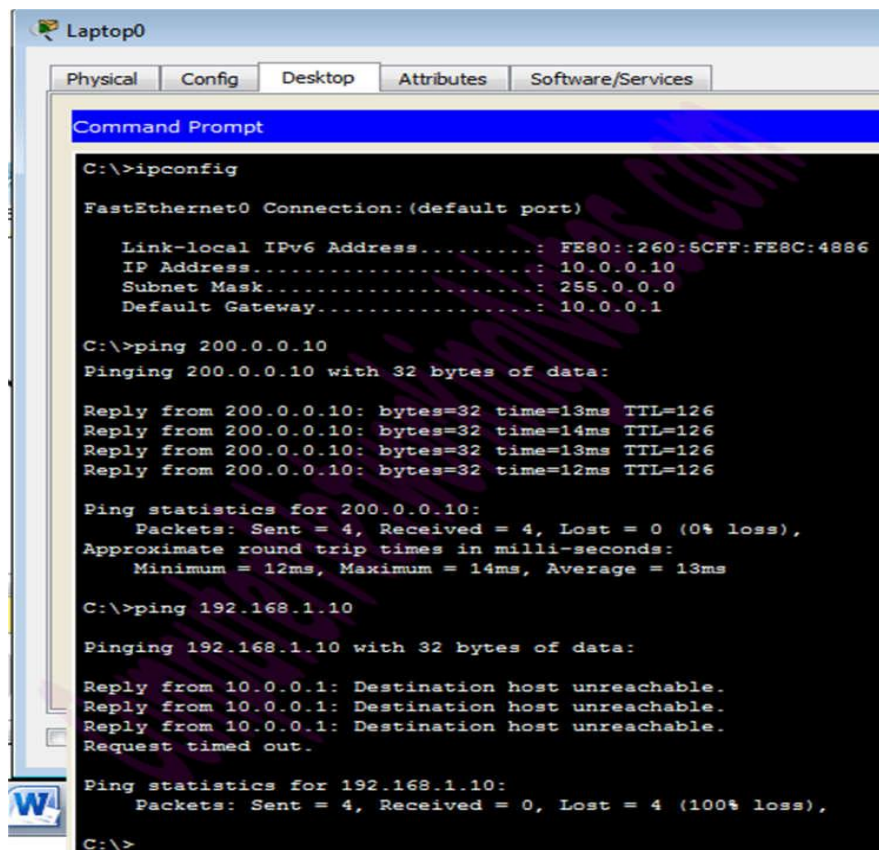
## Testing Dynamic NAT Configuration

In this lab we configured dynamic NAT on R1 for 10.0.0.10 and 10.0.0.20 and static NAT on R2 for 192.168.1.10.

Device	Inside Local IP Address	Inside Global IP Address
Laptop0	10.0.0.10	50.0.0.1
Laptop1	10.0.0.20	50.0.0.2
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.



```
Laptop0
Physical Config Desktop Attributes Software/Services
Command Prompt
C:\>ipconfig

FastEthernet0 Connection: (default port)

    Link-local IPv6 Address . . . . . : FE80::260:5CFF:FE8C:4886
    IP Address. . . . . : 10.0.0.10
    Subnet Mask . . . . . : 255.0.0.0
    Default Gateway . . . . . : 10.0.0.1

C:\>ping 200.0.0.10
Pinging 200.0.0.10 with 32 bytes of data:

Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=14ms TTL=126
Reply from 200.0.0.10: bytes=32 time=13ms TTL=126
Reply from 200.0.0.10: bytes=32 time=12ms TTL=126

Ping statistics for 200.0.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 12ms, Maximum = 14ms, Average = 13ms

C:\>ping 192.168.1.10
Pinging 192.168.1.10 with 32 bytes of data:

Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Reply from 10.0.0.1: Destination host unreachable.
Request timed out.

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

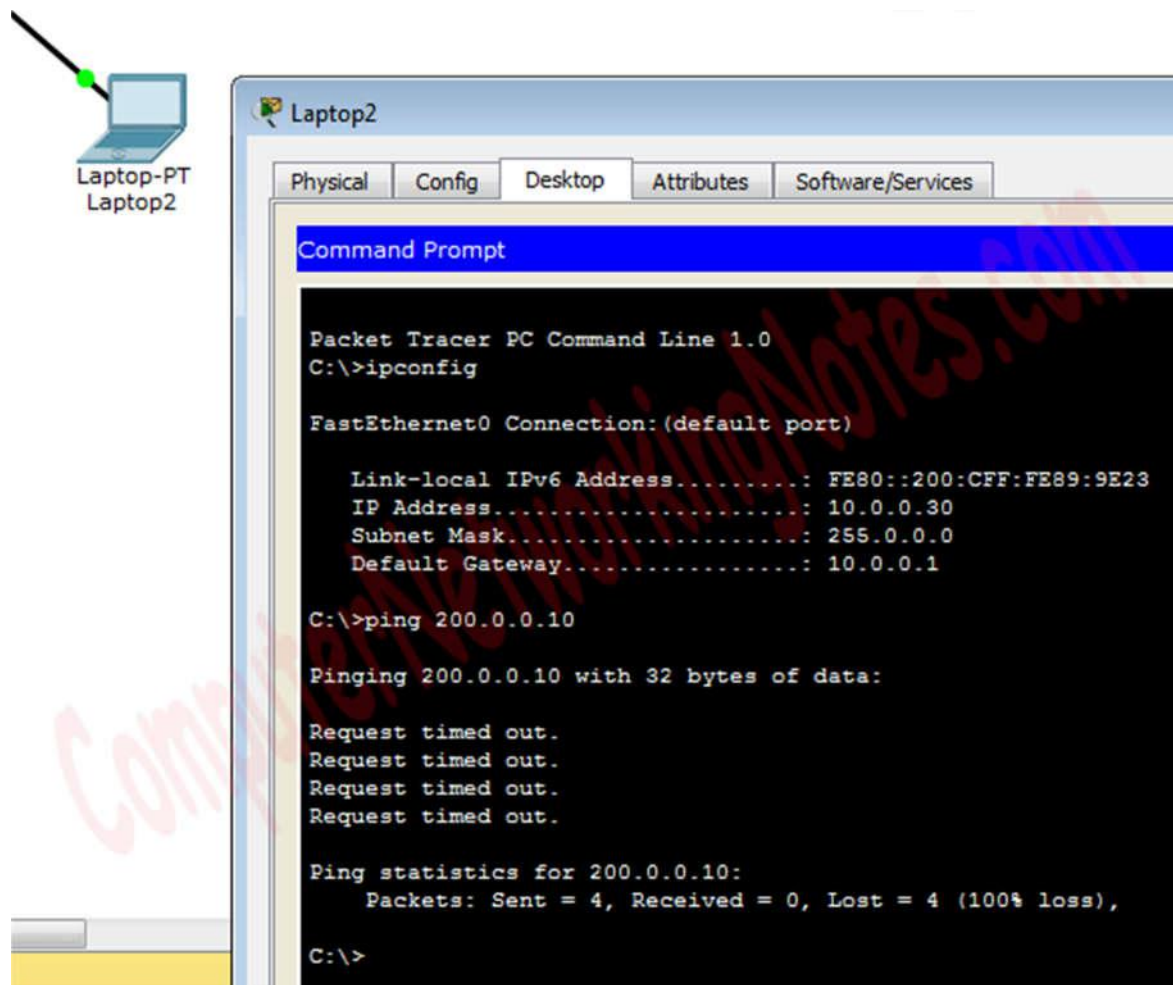
First command verifies that we are testing from correct NAT device.

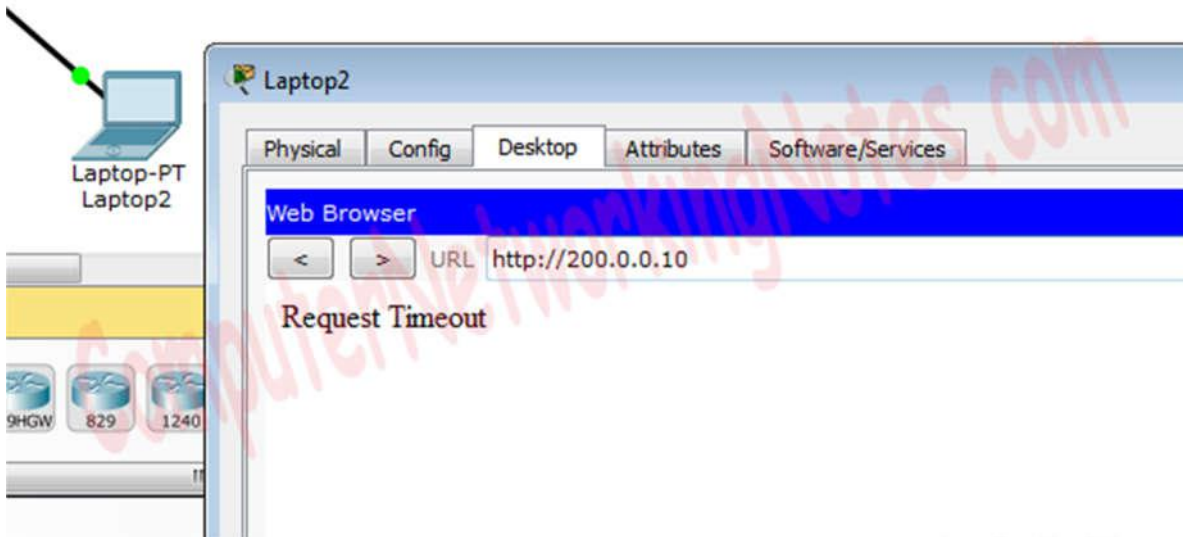
Second command checks whether we are able to access the remote device or not. A ping reply confirms that we are able to connect with remote device on this IP address.

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.

Above figure confirms that host 10.0.0.10 is able to access the 200.0.0.10. You can also do the same testing from Laptop1, result will be same.

Now run ping 200.0.0.10 command from Laptop2.





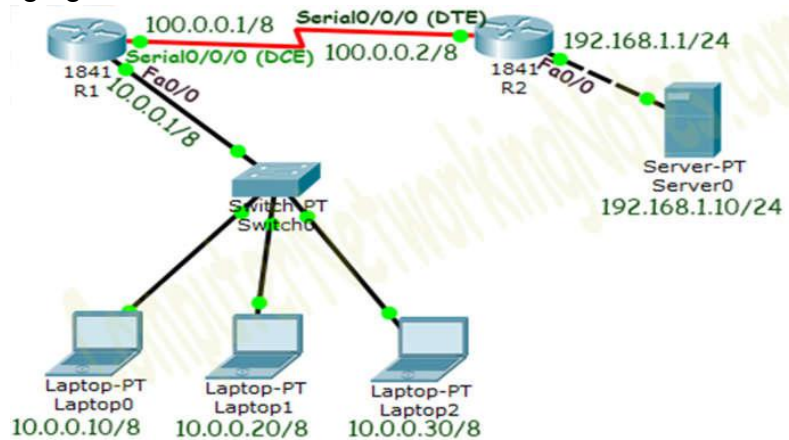
Why we are not able to connect with the remote device from this host?

Because we configured NAT only for two hosts (Laptop0 and Laptop1) which IP addresses are 10.0.0.10 and 10.0.0.20. So only the host 10.0.0.10 and 10.0.0.20 will be able to access the remote device.

# NAT OVERLOAD(PAT)

## PAT (NAT Overload) Practice LAB Setup

Create a lab as illustrates in following figure.



## 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



## Configure PAT (NAT Overload)

PAT configuration requires four steps: -

- Create an access list of IP addresses which need translation
- Create a pool of all IP address which are available for translation
- Map access list with pool
- Define inside and outside interfaces

## R1 PAT (NAT Overload) Configuration

```
R1>enable
```

```
R1#configure terminal
```

```
Enter configuration commands, one per line. End  
with CNTL/Z. R1(config)#access-list 1 permit
```

```
10.0.0.10 0.0.0.0 R1(config)#access-list 1 permit
```

```
10.0.0.20 0.0.0.0 R1(config)#access-list 1 deny any
```

```
R1(config)#ip nat pool ccna 50.0.0.1 50.0.0.1 netmask  
255.0.0.0 R1(config)#ip nat inside source list 1 pool  
ccna overload
```

```
R1(config)#interface FastEthernet 0/0
```

```
R1(config-if)#ip nat  
inside R1(config-  
if)#exit
```

```
R1(config)#interface  
Serial 0/0/0 R1(config-  
if)#ip nat outside  
R1(config-if)#exit  
R1(config)#
```

On R2 we can keep standard configuration or can configure dynamic NAT or can configure static NAT .

```
R2>enable
```

```
R2#configure terminal
```

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

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

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

```
R2(config-if)#ip nat outside
```

```
R2(config-if)#exit
```

```
R2(config)#interface FastEthernet 0/0
```

```
R2(config-if)#ip nat inside
```

```
R2(config-if)#exit
```

```
R2(config)#
```

### 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
```