Chapitre 11 : Configuration du NAT/PAT sur un routeur Cisco

Sommaire :

1. Configuration initiale du routeur ISP.
2. Configuration initiale du routeur R1.
3. Configuration commune à tout type de NAT.
4. Configuration du NAT statique pour PC3.
5. Configuration du NAT dynamique avec pool d'adresses (sans et avec surcharge).
6. Configuration du NAT dynamique sans pool d'adresses (avec surcharge : fonction PAT).

1. Configuration initiale du routeur ISP.

```
interface Loopback0
 ip address 8.8.8.8 255.255.255.255
!
interface GigabitEthernet0/0
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface GigabitEthernet0/1
 no ip address
 duplex auto
 speed auto
 shutdown
!
interface Serial0/0/0
 ip address 80.79.100.1 255.255.255.252
!
```

1ère étape : Nous avons ici configurer l'interface **loopback** sur le routeur **ISP** ainsi que l'interface **s0/0/0**.

```
ISP(config)#ip route 201.49.10.16 255.255.255.240 80.79.100.2
```

2ème étape : Par la suite nous configurons la **route** vers le **pool d'adresses publique**.

2. Configuration initiale du routeur R1.

```
interface GigabitEthernet0/0
 ip address 192.168.0.1 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 192.168.1.1 255.255.255.0
 duplex auto
 speed auto
!
interface Serial0/0/0
 ip address 80.79.100.2 255.255.255.252
 clock rate 64000
!
```

1ère étape : Configuration des trois interfaces **s0/0/0**, **G0/0** et **G0/1** sur le routeur **R1**.

```
R1(config)#ip route 0.0.0.0 0.0.0.0 80.79.100.1
```

2ème étape : Nous configurons cette fois-ci la **route par défaut**.

PC1 vers PC2

```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time<1ms TTL=127
Reply from 192.168.1.10: bytes=32 time<1ms TTL=127
Reply from 192.168.1.10: bytes=32 time<1ms TTL=127
Reply from 192.168.1.10: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC1 vers PC3

```
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time<1ms TTL=127
Reply from 192.168.1.100: bytes=32 time<1ms TTL=127
Reply from 192.168.1.100: bytes=32 time<1ms TTL=127
Reply from 192.168.1.100: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC1 vers R1

PC1

Physical | Config | Desktop | Programming | Attributes

Command Prompt

```
Cisco Packet Tracer PC Command Line 1.0
C:\>ping 192.168.0.1

Pinging 192.168.0.1 with 32 bytes of data:

Reply from 192.168.0.1: bytes=32 time<1ms TTL=255
Reply from 192.168.0.1: bytes=32 time<1ms TTL=255
Reply from 192.168.0.1: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.0.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

3ème étape : Test de **ping** depuis **PC1** afin de vérifier si il est actuellement possible d'obtenir une réponse.

PC1 vers PC2 (Screen 1)
PC1 vers PC3 (Screen 2)
PC1 vers R1 (Screen 3)

PC2 vers PC1

```
Pinging 192.168.0.10 with 32 bytes of data:

Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC2 vers PC3

```
C:\>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:

Reply from 192.168.1.100: bytes=32 time=1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC2 vers R1

```
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<1ms TTL=255

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

4ème étape : Depuis **PC2** nous effectuons un test de ping.

PC2 vers PC1 (Screen 1)
PC2 vers PC3 (Screen 2)
PC2 vers R1 (Screen 3)

PC3 vers PC1

```
C:\>ping 192.168.0.10

Pinging 192.168.0.10 with 32 bytes of data:

Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127
Reply from 192.168.0.10: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.0.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

PC3 vers PC2

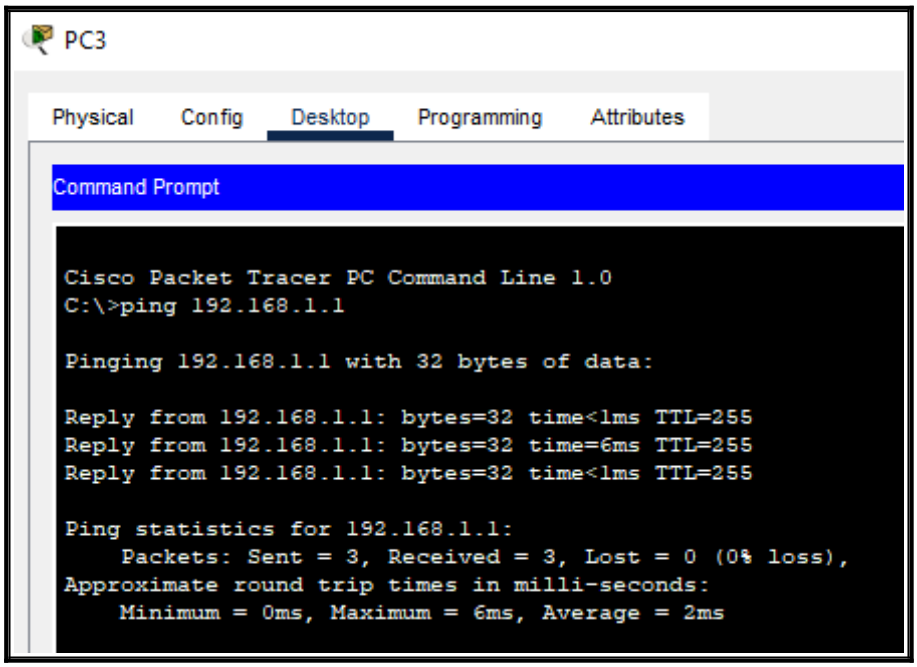
```
C:\>ping 192.168.1.10

Pinging 192.168.1.10 with 32 bytes of data:

Reply from 192.168.1.10: bytes=32 time<1ms TTL=128
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128
Reply from 192.168.1.10: bytes=32 time<1ms TTL=128
Reply from 192.168.1.10: bytes=32 time=1ms TTL=128

Ping statistics for 192.168.1.10:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

PC3 vers R1



5ème étape : Test de ping depuis **PC3**.

PC3 vers PC1 (Screen 1)
PC3 vers PC2 (Screen 2)
PC3 vers R1 (Screen 3)

R1 vers 4x8

```
R1#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/10/15 ms
```

R1 vers ISP

```
C:\>ping 80.79.100.1

Pinging 80.79.100.1 with 32 bytes of data:

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

Ping statistics for 80.79.100.1:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

6ème étape : Test de ping depuis **R1**.

3. Configuration commune à tout type de NAT.

```
R1(config)#int g0/0
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#int g0/1
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#int s0/0/0
R1(config-if)#ip nat outside
R1(config-if)#exit
```

1ère étape : Nous configurons le **NAT** en commençant par les **interfaces** et en indiquant de si elles sont du côté **privé « inside »** ou bien **public « outside »**.

4. Configuration du NAT statique pour PC3.

```
R1(config)#ip nat inside source static 192.168.1.100 201.49.10.30
```

1ère étape : Nous configurons une translation statique afin d'indiquer au routeur **R1** que les **paquets** qui arrivent sur son interface publique **S0/0/0** avec l'adresse de destination **201.49.10.30** doivent être redirigés vers **192.168.1.100**

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  201.49.10.30      192.168.1.100      ---                ---
```

2ème étape : Voici à quoi ressemble la **table de translation NAT** après la saisit de la translation.

```
C:\>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=16ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 16ms, Average = 4ms
```

3ème étape : On observe donc que **PC3** est maintenant capable de **communiquer** avec le **réseau public** avec un **ping à 4x8**.

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  201.49.10.30      192.168.1.100      ---                ---

R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 201.49.10.30:10   192.168.1.100:10   8.8.8.8:10         8.8.8.8:10
icmp 201.49.10.30:11   192.168.1.100:11   8.8.8.8:11         8.8.8.8:11
icmp 201.49.10.30:12   192.168.1.100:12   8.8.8.8:12         8.8.8.8:12
icmp 201.49.10.30:9    192.168.1.100:9    8.8.8.8:9          8.8.8.8:9
---  201.49.10.30      192.168.1.100      ---                ---
```

4ème étape : Après le ping sur **PC3** nous consultons de nouveau la **table de translation NAT**, nous constatons donc l'évolution de celle-ci.

```
ISP#ping 201.49.10.30

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 201.49.10.30, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/6/14 ms
```

5ème étape : Depuis **ISP** nous effectuons un **ping** vers le **pool d'adresses publiques** achetées.

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 201.49.10.30:1    192.168.1.100:1    80.79.100.1:1      80.79.100.1:1
icmp 201.49.10.30:2    192.168.1.100:2    80.79.100.1:2      80.79.100.1:2
icmp 201.49.10.30:3    192.168.1.100:3    80.79.100.1:3      80.79.100.1:3
icmp 201.49.10.30:4    192.168.1.100:4    80.79.100.1:4      80.79.100.1:4
icmp 201.49.10.30:5    192.168.1.100:5    80.79.100.1:5      80.79.100.1:5
---  201.49.10.30      192.168.1.100      ---                ---
```

6ème étape : Nous affichons de nouveau la **translation NAT** du routeur **R1**.

5. Configuration du NAT dynamique avec pool d'adresses (sans et avec surcharge).

```
R1(config)#ip nat pool POOL-NAT-LAN2 201.49.10.17 201.49.10.29 netmask 255.255.255.240
```

1ère étape : Nous créons le **pool** d'adresses **LAN2** sur **R1**.

```
R1(config)#access-list 1 deny 192.168.1.100
R1(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

2ème étape : Nous définissons par la suite les **adresses IP sources** qui peuvent potentiellement être translatées, nous créons donc une **ACL**.

```
R1(config)#ip nat inside source list 1 pool POOL-NAT-LAN2
```

3ème étape : Nous configurons donc le NAT.

```
R1(config)#ip nat inside source list 1 pool POOL-NAT-LAN2 overload
```

4ème étape : Même commande saisit juste avant mais cette fois-ci nous ajoutons « **overload** », puisqu'il y a **plus de machines dans le réseau privé** que d'**adresses publiques disponibles**.

```
C:\>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=15ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 15ms, Average = 4ms
```

5ème étape : Maintenant **PC2** peut également communiquer avec l'extérieur, nous effectuons donc un ping vers **4x8**.

6. Configuration du NAT dynamique sans pool d'adresses (avec surcharge : fonction PAT).

```
R1(config)#access-list 2 permit 192.168.0.0 0.0.0.255
```

1ère étape : Nous créons une nouvelle ACL

```
R1(config)#ip nat inside source list 2 interface s0/0/0 overload
```

2ème étape : Nous configurons donc le troisième type **NAT** en y ajoutant aussi « **overload** ».

```
ISP#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 0/5/10 ms
```

PC1

```
C:\>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=10ms TTL=254
Reply from 8.8.8.8: bytes=32 time=2ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 10ms, Average = 3ms
```

PC2

```
C:\>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=9ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=10ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 10ms, Average = 5ms
```

PC3

```
C:\>ping 8.8.8.8

Pinging 8.8.8.8 with 32 bytes of data:

Reply from 8.8.8.8: bytes=32 time=11ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254
Reply from 8.8.8.8: bytes=32 time=1ms TTL=254

Ping statistics for 8.8.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 11ms, Average = 3ms
```

```
R1#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/4/12 ms
```

3ème étape : Désormais **toutes les machines** présentes dans la topologie peuvent **communiquer** avec **l'extérieur**, nous effectuons donc un ping vers **4x8** afin de vérifier cela.