

NAT 概述

NAT 是一个 IETF 标准，允许一个机构以一个地址出现在 Internet 上。NAT 技术使得一个私有网络可以通过 Internet 注册 IP 连接到外部世界，位于 Inside 网络和 Outside 网络中的 NAT 路由器在发送数据包之前，负责把内部 IP 地址翻译成外部合法 IP 地址。

NAT 将每个局域网节点的 IP 地址转换成一个合法 IP 地址，反之亦然。它也可以应用到防火墙技术里，把个别 IP 地址隐藏起来不被外界发现，对内部网络设备起到保护的作用，同时，它还帮助网络可以超越地址的限制，合理地安排网络中的公有 Internet 地址和私有 IP 地址的使用。

NAT 有三种类型：静态 NAT、动态 NAT 和端口地址转换（PAT）。

1. 静态 NAT

静态 NAT 中，内部网络中的每个主机都被永久映射成外部网络中的某个合法的地址。静态地址转换将内部本地地址与内部合法地址进行一对一的转换，且需要指定和哪个合法地址进行转换。如果内部网络有 E-mail 服务器或 FTP 服务器等可以为外部用户提供的服务，这些服务器的 IP 地址必须采用静态地址转换，以便外部用户可以使用这些服务。

2. 动态 NAT

动态NAT 首先要定义合法地址池，然后采用动态分配的方法映射到内部网络。动态NAT是动态一对一的映射。

3. PAT

PAT 则是把内部地址映射到外部网络的IP 地址的不同端口上，从而可以实现多对一的映射。PAT 对于节省IP 地址是最为有效的。

实验1：静态NAT 配置

1. 实验目的

通过本实验可以掌握

- (1) 静态NAT 的特征
- (2) 静态NAT 基本配置和调试

2. 拓扑结构

实验拓扑如图1所示。

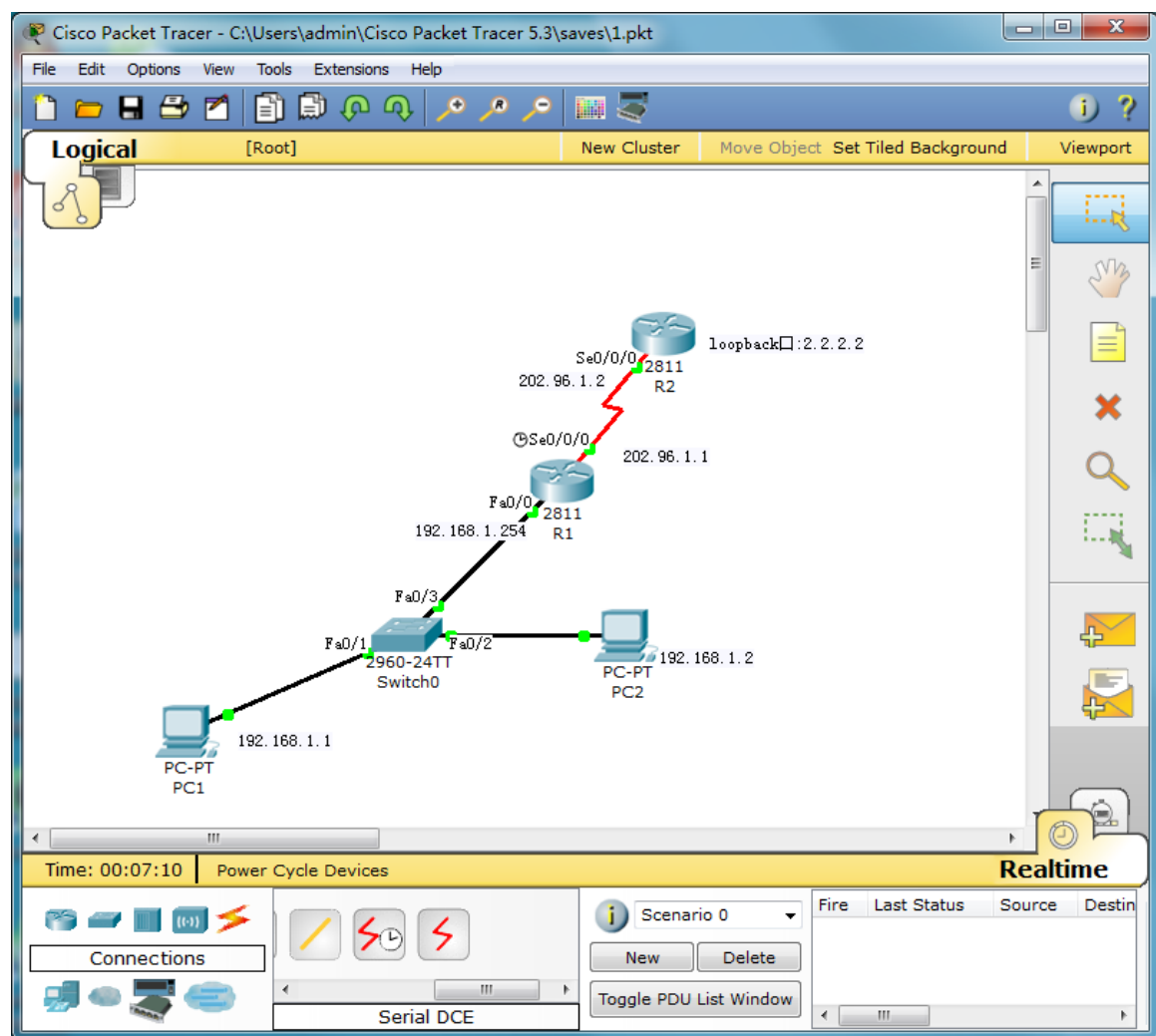


图1 实验拓扑

首先完成基本配置，关闭域名查询、IP地址等

注意：在R2上有个loopback口，loopback口是个逻辑接口，可以用来模拟一个网段。只要有一个物理接口UP，loopback口就会UP，因为其稳定性和方便性，很受欢迎。

配置方法和普通物理接口一样，`interface loopback 接口 ID`就可进入 loopback 口

3. 实验步骤

(1) 根据拓扑结构配置好PC，路由器和交换机端口的IP地址

(2) 配置R2的loopback口

```
R2(config)#interface loopback 1
```

```
%LINK-5-CHANGED: Interface Loopback1, changed state to up
```

```
%LINEPROTO-5-UPDOWN: Line protocol on Interface Loopback1,  
changed state to up
```

```
R2(config-if)#ip add 2.2.2.2 255.255.255.0  
R2(config-if)#exit  
R2(config)#
```

(3) 配置NAT 内部接口

```
R1(config)#int s0/0/0  
R1(config-if)#ip nat outside
```

//配置NAT 外部接口

```
R1(config)#router rip  
R1(config-router)#version 2  
R1(config-router)#no auto-summary  
R1(config-router)#network 202.96.1.0  
R1(config-router)#network 192.168.1.0
```

(4) 配置路由器R2

```
R2(config)#router rip
R2(config-router)#version 2
R2(config-router)#no auto-summary
R2(config-router)#network 202.96.1.0
R2(config-router)#network 192.168.1.0
R2(config-router)#network 2.2.2.0
```

(5) 配置路由器R1 提供NAT 服务

```
R1(config)#ip nat inside source static 192.168.1.1 202.96.1.3
//配置静态NAT 映射
R1(config)#ip nat inside source static 192.168.1.2 202.96.1.4

R1(config)#int f0/0
R1(config-if)#ip nat inside
```

4. 实验调试

(1) 在R1上执行 debug ip nat

该命令可以查看地址翻译的过程。

在PC1 和PC2 上Ping 2.2.2.2（路由器R2 的环回接口），此时应该是通的，路由器R1

的输出信息如下：

```
R1#debug ip nat
*Mar 4 02:02:12.779: NAT*: s=192.168.1.1->202.96.1.3,
d=2.2.2.2 [20240]
*Mar 4 02:02:12.791: NAT*: s=2.2.2.2,
d=202.96.1.3->192.168.1.1 [14435]
.....
*Mar 4 02:02:25.563: NAT*: s=192.168.1.2->202.96.1.4,
d=2.2.2.2 [25]
*Mar 4 02:02:25.579: NAT*: s=2.2.2.2,
d=202.96.1.4->192.168.1.2 [25]
```

.....

以上输出表明了NAT 的转换过程。首先把私有地址“192.168.1.1”和“192.168.1.2”分别转换成公网地址“202.96.1.3”和“202.96.1.4”访问地址“2.2.2.2”，然后回来的时候把公网地址“202.96.1.3”和 “202.96.1.4”分别转换成私有地址“192.168.1.1”和“192.168.1.2”。

(2) show ip nat translations

该命令用来查看NAT 表。静态映射时，NAT 表一直存在。

```
R1#show ip nat translations
Pro Inside global Inside local Outside local Outside global
--- 202.96.1.3 192.168.1.1 --- ---
--- 202.96.1.4 192.168.1.2 --- ---
```

以上输出表明了内部全局地址和内部局部地址的对应关系。

【术语】

- ① 内部局部 (inside local) 地址：在内部网络使用的地址，往往是RFC1918 地址；
- ② 内部全局 (inside global) 地址：用来代替一个或多个本地IP 地址的、对外的、向NIC 注册过的地址；
- ③ 外部局部 (outside local) 地址：一个外部主机相对于内部网络所用的IP 地址。不一定是合法的地址；
- ④ 外部全局 (outside global) 地址：外部网络主机的合法IP 地址。

实验2：动态NAT

1. 实验目的

通过本实验可以掌握：

- (1) 动态NAT 的特征
- (2) 动态NAT 配置和调试

2. 拓扑结构

实验拓扑如图1所示。

3. 实验步骤

- (1) 步骤1：配置路由器R1 提供NAT 服务

```
R1(config)#ip nat pool NAT 202.96.1.3 202.96.1.100 netmask 255.255.255.0
```

//配置动态NAT 转换的地址池

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

//配置动态NAT 映射

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

//允许动态NAT 转换的内部地址范围

```
R1(config)#int f0/0
```

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

```
R1(config-if)#int s0/0/0
```

```
R1(config-if)#ip nat outside
```

4. 实验调试

在PC1, PC2 上ping 2.2.2.2（路由器R2 的环回接口），调试结果如下：

- (1) debug ip nat

```
R1#debug ip nat
```

```
IP NAT debugging is on
```

```
R1#clear ip nat translation * //清除动态NAT 表
```

```

*Mar 4 01:34:23.075: NAT*: s=192.168.1.1->202.96.1.4,
d=2.2.2.2 [19833]
*Mar 4 01:34:23.087: NAT*: s=2.2.2.2,
d=202.96.1.4->192.168.1.1 [62333]
.....
*Mar 4 01:28:49.867: NAT*: s=192.168.1.2->202.96.1.3,
d=2.2.2.2 [62864]
*Mar 4 01:28:49.875: NAT*: s=2.2.2.2,
d=202.96.1.3->192.168.1.2 [54062]
.....

```

【提示】

如果动态NAT 地址池中没有足够的地址作动态映射，则会出现类似下面的信息，提示NAT 转换失败，并丢弃数据包。

```

*Feb 22 09:02:59.075: NAT: translation failed (A), dropping
packet s=192.168.1.2 d=2.2.2.2

```

(2) show ip nat translations

```

R1#show ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 202.96.1.4:1721 192.168.1.1:1721 2.2.2.2:80 2.2.2.2:80
--- 202.96.1.4 192.168.1.1 --- ---
icmp 202.96.1.3:3 192.168.1.2:3 2.2.2.2:3 2.2.2.2:3
tcp 202.96.1.3:14347 192.168.1.2:14347 2.2.2.2:23
2.2.2.2:23
--- 202.96.1.3 192.168.1.2 --- ---

```

以上信息表明当PC1 和PC2 第一次访问“2.2.2.2”地址的时候，NAT 路由器R1 为主机PC1 和PC2 动态分配两个全局地址

“202.96.1.4”和“202.96.1.3”，在NAT 表表中生成两

条动态映射的记录，同时会在NAT 表中生成和应用向对应的协议和

端口号的记录（过期时间为60 秒）。在动态映射没有过期（过期时

间为86400 秒）之前，再有应用从相同主机发起时，NAT 路由器直

接查NAT 表，然后为应用分配相应的端口号。

(3) show ip nat statistics

该命令用来查看NAT 转换的统计信息。

```
R1#show ip nat statistics
```

```
Total active translations: 5 (0 static, 5 dynamic; 3 extended)
```

```
//有5 个转换是动态转化,
```

```
Outside interfaces:
```

```
Serial0/0/0
```

```
//NAT 外部接口
```

```
Inside interfaces:
```

```
GigabitEthernet0/0
```

```
//NAT 内部接口
```

```
Hits: 54 Misses: 6
```

```
CEF Translated packets: 60, CEF Punted packets: 5
```

```
Expired translations: 12 //NAT 表中过期的转换
```

```
Dynamic mappings: //动态映射
```

```
-- Inside Source
```

```
[Id: 1] access-list 1 pool NAT refcount 2
```

```
pool NAT: netmask 255.255.255.0 //地址池名字和掩码
```

```
start 202.96.1.3 end 202.96.1.100 //地址池范围
```

```
type generic, total addresses 98, allocated 2 (2%), misses  
0
```

```
//共98 个地址, 分出去2 个
```

```
Queued Packets: 0
```

实验3: PAT 配置

1.实验目的

通过本实验可以掌握：

- (1) PAT 的特征
- (2) overload 的使用
- (3) PAT 配置和调试

2.拓扑结构

实验拓扑如图1所示。

3.实验步骤

- (1) 步骤1：配置路由器R1 提供NAT 服务

```
R1(config)#ip nat pool NAT 202.96.1.3 202.96.1.100 netmask 255.255.255.0
```

```
R1(config)#ip nat inside source list 1 pool NAT overload //
```

配置PAT

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

```
R1(config)#int f0/0
```

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

```
R1(config-if)#int s0/0
```

```
R1(config-if)#ip nat outside
```

4.实验调试

在PC1 上访问2.2.2.2（路由器R2 的环回接口）的www 服务，在PC2 上分别telnet 和

ping 2.2.2.2（路由器R2 的环回接口），调试结果如下：

- (1) debug ip nat

```

*Mar 4 01:53:47.983: NAT*: s=192.168.1.1->202.96.1.3,
d=2.2.2.2 [20056]
*Mar 4 01:53:47.995: NAT*: s=2.2.2.2,
d=202.96.1.3->192.168.1.1 [46201]
.....
*Mar 4 01:54:03.015: NAT*: s=192.168.1.2->202.96.1.3,
d=2.2.2.2 [20787]
*Mar 4 01:54:03.219: NAT*: s=2.2.2.2,
d=202.96.1.3->192.168.1.2 [12049]
.....

```

(2) show ip nat translations

```

R1#show ip nat translations
Pro Inside global Inside local Outside local Outside global
tcp 202.96.1.3:1732 192.168.1.1:1732 2.2.2.2:80 2.2.2.2:80
icmp 202.96.1.3:4 192.168.1.2:4 2.2.2.2:4 2.2.2.2:4
tcp 202.96.1.3:12320 192.168.1.2:12320 2.2.2.2:23
2.2.2.2:23

```

以上输出表明进行PAT 转换使用的是同一个IP 地址的不同端口号。

(3) show ip nat statistics

```

Total active translations: 3 (0 static, 3 dynamic; 3 extended)
Outside interfaces:
Serial0/0/0
Inside interfaces:
GigabitEthernet0/0
Hits: 762 Misses: 22
CEF Translated packets: 760, CEF Punted packets: 47
Expired translations: 19
Dynamic mappings:
-- Inside Source
[Id: 2] access-list 1 pool NAT refcount 3
pool NAT: netmask 255.255.255.0
start 202.96.1.3 end 202.96.1.100
type generic, total addresses 98, allocated 1 (1%), misses
0
Queued Packets: 0

```

【提示】

动态NAT 的过期时间是86400 秒，PAT 的过期时间是60 秒，通过命令“**show ip nat translations verbose**”可以查看。也可以通过下面的命令来修改超时时间：

```
R1(config)#ip nat translation timeout timeout
```

参数timeout 的范围是0-2147483。

如果主机的数量不是很多，可以直接使用outside 接口地址配置PAT，不必定义地址池，

命令如下：

```
R1(config)#ip nat inside source list 1 interface s0/0/0  
overload
```

NAT 命令汇总

命令	作用
clear ip nat translation *	清除动态NAT 表
show ip nat translation	查看NAT 表
show ip nat statistics	查看NAT 转换的统计信息
debug ip nat	动态查看NAT 转换过程
ip nat inside source static	配置静态NAT
ip nat inside	配置NAT 内部接口
ip nat outside	配置NAT 外部接口
ip nat pool	配置动态NAT 地址池
ip nat inside source list access-list-number pool name	配置动态NAT
ip nat inside source list access-list-number pool name overload	配置 PAT

练习：

根据下面的拓扑图，自行完成拓扑结构的连接，IP 地址分配，最后，实现全网能相互 PING 通。

