第五次：思科软件入门

王崇智 2017141051019

必要的知识与概念

1. **PPP协议**
    - PPP协议是一种点——点串行通信协议。PPP具有处理错误检测、支持多个协议、允许在连接时刻协商IP地址、允许身份认证等功能，还有其他。PPP提供了3类功能：成帧；链路控制协议LCP；网络控制协议NCP。PPP是面向字符类型的协议。
    - PPP协议的验证分为两种：一种是PAP，一种是CHAP。相对来说PAP的认证方式安全性没有CHAP高。PAP在传输password是明文的，而CHAP在传输过程中不传输密码，取代密码的是hash（哈希值）。PAP认证是通过两次握手实现的，而CHAP则是通过3次握手实现的。
    - 点到点协议是为在同等单元之间传输数据包这样的简单链路设计的链路层协议。这种链路提供全双工，并按照顺序传递数据包。设计目的主要是用来通过拨号或专线方式建立点对点连接发送数据，使其成为各种主机、网桥和路由器之间简单连接的一种共通的解决方案，PPP具有以下功能：
        - 具有动态分配IP地址的能力，允许在连接时刻协商IP地址
        - 支持多种网络协议，比如TCP/IP，NetBEUI，NWLINK等
        - 具有错误检测能力，但不具备纠错能力，所以PPP是不可靠传输协议
        - 无重传的机制，网络开销小，速度快
        - 具有身份验证功能
        - 可以用于多种类型的物理介质上。
2. **交换机和路由器的区别**
    - 交换机实际上是多接口的网桥
    - 外形上
        - 交换机通常端口比较多，路由器端口少体积小，路由器一般都即成了交换机的功能，LAN口就是作为交换机的端口来使用。而WAN少用于连接外网的端口
    - 工作层次不同
        - 交换机在数据链路层，而路由器在网络层
    - 数据的转发对象不同
        - 交换机是根据mac地址转发数据帧，而路由器是根据IP地址来转发数据报。
    - 分工不同
        - 交换机主要是用于组建局域网，而路由器则负责让主机连接外网，多台主机可以通过网线连接到交换机，这时候就组建好了局域网，可以通过网线连接到交换机。
    - 冲突域和广播域
        - 交换机分割冲突域，但不分割广播域，而路由器分割广播域。
3. **默认网关**

    网关实质上是一个网络通向其他网络的IP地址。

    比如有网络A和网络B，网络A的IP地址范围为“192.168.1.1~192. 168.1.254”，子网掩码为255.255.255.0；网络B的IP地址范围为“192.168.2.1~192.168.2.254”，子网掩码为255.255.255.0。在没有路由器的情况下，两个网络之间是不能进行TCP/IP通信的，即使是两个网络连接在同一台交换机（或集线器）上，TCP/IP协议也会根据子网掩码（255.255.255.0）判定两个网络中的主机处在不同的网络里。

而要实现这两个网络之间的通信，则必须通过网关。如果网络A中的主机发现数据包的目的地主机不在本地网络中，就必须把数据包转发给它自己的网关，再由网关转发给网络B的网关，网关B的网关再转发给网络B的某个主机

解释清了网关的含义，就可以具体的介绍默认网关了：

即一台主机如果找不到可用的网关，就把数据包发给默认指定的网关，由这个网关来处理数据包。现在主机使用的网关，一般指的是默认网关。

注意到：默认网关必须是电脑自己所在网段中的IP地址，而不能填写其他网段中的IP地址。

默认网关的设定有手动设置和默认设置两种方式。

- **手动方式顾名思义**
- 自动设置就是利用DHCP（动态主机配置协议）服务器来自动给网络中的电脑分配IP地址、子网掩码和默认网关。这种方法适用于网络规模较大、TCP/IP参数有可能变动的网络。

4. **HDLC协议与PPP协议的对比**

   HDLC协议是面向比特的，而PPP协议则是面向字节的，HDLC的帧采用开头跟结尾都是01111110作为帧的边界，这样当接收方接收到一串比特的时候可以**根据它来判断该帧从哪里开始，到哪里结束，**但是，假如在两个标志字段之间的比特串中恰好出现了01111110比特串，那该怎么办呢，**HDLC采用零比特填充法，所谓零比特填充法就是每当出现5个1的时候就给它添加一个0进去**，而接收方接收到数据时凡出现5个1的时候去掉其后面一个0，这样就能很好地确定帧。

   PPP协议本来也是跟HDLC协议一样，把01111110作为边界符（一般称为标志符），但是因为PPP协议是面向字节的，所以这里不说01111110，**而是说用7E作为边界符。**PPP协议在同步传输链路中也是采用零比特填充法，而在异步传输链路中则采用特殊的字符填充法。

   HDLC在控制字段中提供了可靠的确认机制，因此它可以**实现可靠传输**，而PPP则不提供可靠传输，要靠上层实现保证其正确性，因此，曾经在误码率比较高的链路中，HDLC曾起到了极大的作用，但随着技术的发展，在数据链路层出现差错的概率不大，因此现在全世界使用得最多的数据链路层协议是PPP**协议**

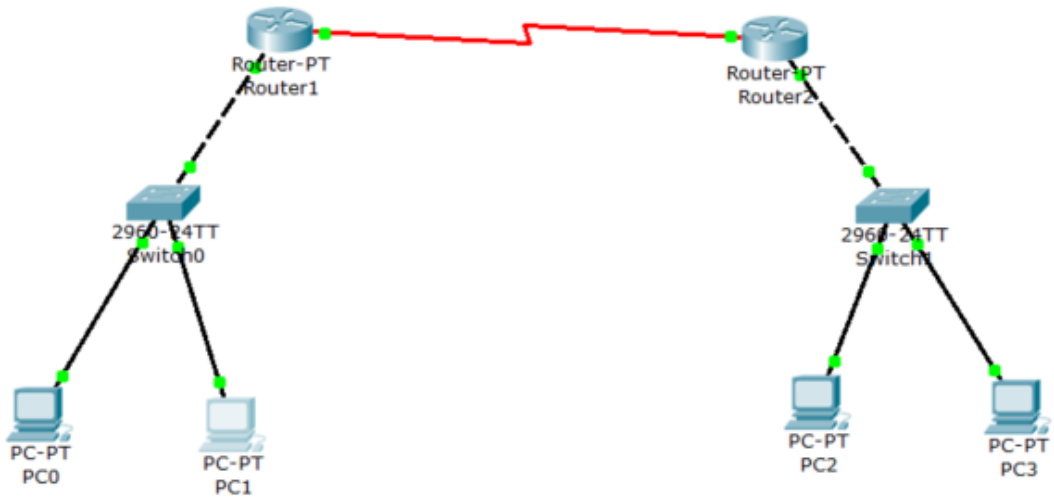
5. 静态路由等的理解

   可以理解为到达特定目的地的路由，有具体的目标网络地址。如果没有特定的目标地址，就会走缺省路由即默认路由。

   默认路由是一种特殊的静态路由，目的地址与掩码配置全为0。这样可以当路由表中所有路由都选择失败的时候，为了使报文有一个最终的目的地，会使用缺省路由。

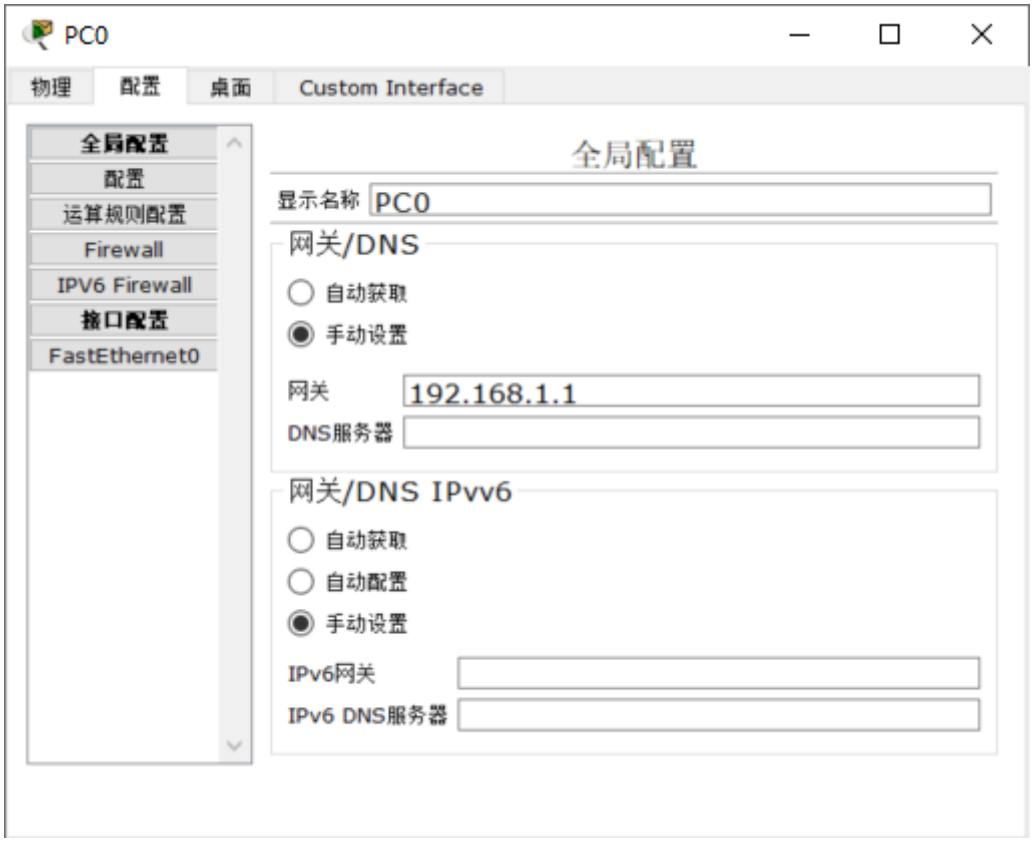
   有了默认，可以减少路由器查路由表的消耗

配置网络拓扑结构图



如上图所示，根据实验说明配置好每个节点的信息，注意到需要配置各个结点位置的接口IP，子网掩码，PC机还要额外设置自己的网关，路由器还要同时考虑路由表的设置与特殊端口的协议。

以PC0为例，展示配置流程

1. 

首先配置网关gateway，保证发数据包的时候能找到一个中转的媒介

2. 

接着，配置f0/0端口的信息，根据图示配置其ip地址与子网掩码即可

配置路由器

- 这也是本次实验中最关键的部分，很容易就跑不通并且没有报错提示。
- 在建立好拓扑图之后，针对路由器方面，可以使用命令行操作，也可以通过GUI界面直接实现，实践证明效果基本没差无非是多重复几遍即可。

1. 先配置PPP协议，以R1为例
2. 这里使用命令行操作，以增强普适性
3. 进入全局模式

```
Router1#conf te
Router1#conf terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Router1(config)#
```

4. 选择接口，注意配置时在gui中将配置的接口状态打开,即为on的状态

```
Router1(config)#inte s2/0
Router1(config-if)#enc ppp
Router1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
```

5. 调用命令观察此时该s2/0接口的配置情况

```
Router1#show interfaces s2/0
Serial2/0 is up, line protocol is down (disabled)
  Hardware is HD64570
  Internet address is 14.10.0.1/8
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  LCP Closed
```

发现PPP已经配置成功，但必要的LCP还处于closed的状态。这个是因为另一个连接与同一条线缆上的router还没有进行相同的配置操作，在对其进行同样操作之后

```
Router1#show interfaces s2/0
Serial2/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 14.10.0.1/8
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set, keepalive set (10 sec)
  LCP Open
```

LCP建立成功

6. 配置CHAP协议

```
Router1(config-if)#ppp authentication chap
Router1(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state


%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state

Router1(config-if)#username Router2 password 123456
```

7. 配置IP地址，即完善网络层的协议，这个在初始配置网络拓补结构时已经通过GUI界面完成
8. 配置路由协议，这里采用静态路由

```
Router1#show ip route
Codes: C - connected, S - static, I - IGRP, 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 inte
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    14.0.0.0/8 is directly connected, Serial2/0
     15.0.0.0/32 is subnetted, 1 subnets
C       15.10.0.1 is directly connected, Serial2/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
S    192.168.4.0/24 [1/0] via 15.10.0.1
```

这一点尤为重要，需要在每次配置完之后利用show ip route指令观察路由器的路由配置情况，有的时候gui界面中显示的信息是无效的，尤其是在重新打开网页之后。

在Router1中，处于config状态下

- 输入ip route 192.168.4.0 255.255.255.0 15.10.0.1即可；router2的配置与1对称

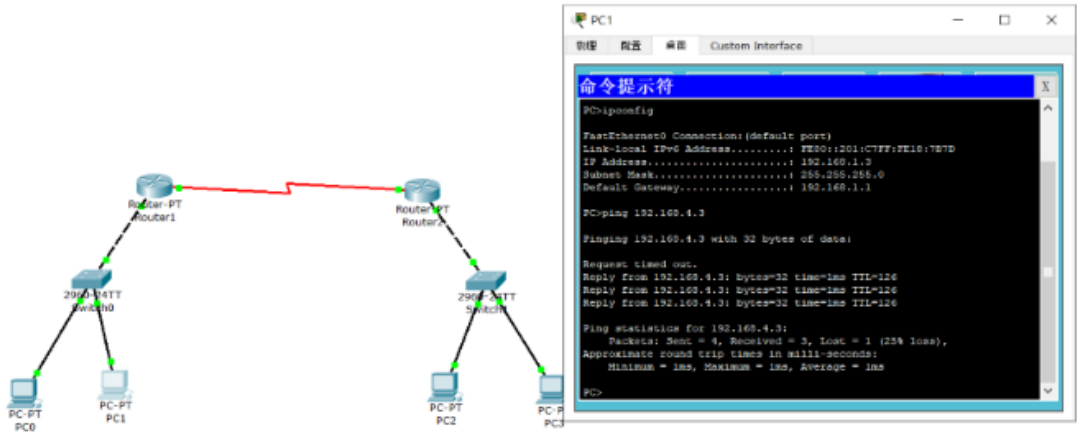
```
PC>ipconfig

FastEthernet0 Connection:(default port)
Link-local IPv6 Address.....: FE80::2E0:F7FF:FE6B:32B2
IP Address.....: 192.168.1.2
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.1.1

PC>ping 192.168.4.2

Pinging 192.168.4.2 with 32 bytes of data:

Request timed out.
Reply from 192.168.4.2: bytes=32 time=1ms TTL=126
Reply from 192.168.4.2: bytes=32 time=6ms TTL=126
Reply from 192.168.4.2: bytes=32 time=2ms TTL=126

Ping statistics for 192.168.4.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 6ms, Average = 3ms
```

```
PC>ipconfig

FastEthernet0 Connection:(default port)
Link-local IPv6 Address.....: FE80::2D0:D3FF:FE11:45D8
IP Address.....: 192.168.4.2
Subnet Mask.....: 255.255.255.0
Default Gateway.....: 192.168.4.1

PC>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time=6ms TTL=126
Reply from 192.168.1.2: bytes=32 time=1ms TTL=126
Reply from 192.168.1.2: bytes=32 time=1ms TTL=126
Reply from 192.168.1.2: bytes=32 time=8ms TTL=126

Ping statistics for 192.168.1.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 1ms, Maximum = 8ms, Average = 4ms
```

如上图所示，pc0与pc2实现了互联互通，且成功出现了首次ping是的数据报丢失问题。因为初次ping的时候两者尚未通信过，互相不知道对方的mac地址，需要付出一次代价来在网络中得到该信息。

修改协议为HDLC

在将PPP协议更换为HDLC在hi前先观测ping的情况



利用enc 指令

```
Router2(config)#enc
Router2(config)#inter s2/0
Router2(config-if)#enc hdlc
Router2(config-if)#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
```

复制　粘贴

修改原先的**ppp协议为hdlc协议**

利用指令**show interfaces s2/0** **观察到两个路由器的协议都已经成功更改，由于****hdlc没有ppp复杂**，不需要继续配置chap或者说username这样的匹配信息了。

```
Router1#show inte
Router1#show interfaces  s2/0
Serial2/0 is up, line protocol is up (connected)
  Hardware is HD64570
  Internet address is 14.10.0.1/8
  MTU 1500 bytes, BW 128 Kbit, DLY 20000 usec,
     reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation HDLC, loopback not set, keepalive set (10 sec)
  Last input never, output never, output hang never
  Last clearing of "show interface" counters never
```

此时的结果发现两个路由器无法互相ping

```
Router1#ping 15.10.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 15.10.0.1, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router1#
```

失败的原因之一发现，配置完之后静态路由发生丢失，两者无法保持像之前ppp协议状态下的正常连接

ping丢失

如实验所示，在两台PC互通的时候复现了该情况，重现该问题的步骤即更换一台新的PC，即刷新MAC以使得两台机时刻保持新鲜状态。

究其原因

- 因为第一个包还没有ARP解析，也就是没有对应的目的mac地址，在尚无mac地址的情况下，系统内核是不会发包的，路由器在一段链路中是使用mac地址的。无目的mac地址的数据包根本就不会发出去，所以不知道目的mac地址，首先先发ARP解析mac地址，同时第一个包还没有发出去就被自己kill了。后续的包因为有了对应的mac地址就顺利的发出去了。
- ping的时候，ip协议会想办法得到目的ip的MAC地址（即物理地址，这是数据链路层协议构建数据链路层的传输单元帧所必需的，以便交给数据链路层构建一个数据帧）。如果两台pc之前通信过，那么A的ARP缓存表中应该存有B的IP与其MAC的映射关系，如果没有，那么就会发一个ARP请求广播，得到B机的物理地址，一起交给数据链路层。然后组成一个数据帧，该帧的目的地址是IP层传过来的物理地址，源地址就是本机的物理地址，同时会附带一些控制信息，再根据当前网络的访问规则发出。
- 注意ping命令是针对IP的，ARP解析式为了建立MAC和IP之间的联系，所以如果没有预先建立好映射关系，ICMP式无法成功到达有效反馈的。

小结

实验内容为理论课未学习内容，过程中查阅了大量资料进行预习，很多步骤是在实践，调试中逐渐明白的。细心很重要，这些结点的端口哪些处于打开与关闭状态都要十分留意，以确保没有问题。本次实验接触了很多概念，只有不断运用才能记住。

第六次：NAT实践

> **实验目标**

本实验在上次实验的基础上，模拟配置 NAT。实验过程中，需要完成

对以下知识点的掌握：

1) 了解内网ip和外网ip的概念;

2) 了解 NAT 的工作原理;

2) 了解 NAT 和NAPT的地址转换方式。

必要的知识与概念

1. **公网与内网**

   先讲**IP**地址的分类**，以及其划分依据：

   为了便于寻址以及层次化构造网络，每个IP地址包括两个标识码（ID），即网络ID和主机ID。同一个物理网络上的所有主机都使用同一个网络ID，网络上的一个主机（包括网络上工作站，服务器和路由器等）有一个主机ID与其对应。Internet委员会定义了5种IP地址类型以适合不同容量的网络，即A类~E类。

类别	最大网络数	IP地址范围	最大主机数	私有IP地址范围
A	126（2^7-2)	0.0.0.0-127.255.255.255	16777214	10.0.0.0-10.255.255.255
B	16384(2^14)	128.0.0.0-191.255.255.255	65534	172.16.0.0-172.31.255.255
C	2097152(2^21)	192.0.0.0-223.255.255.255	254	192.168.0.0-192.168.255.255

我们一般在不同网络场景下ipconfig的结果通常是172,或者192.168。开头的都是私有IP。他们存在一些限制

1. 私有IP的路由信息不能对外散播（只能存在内部网络）
2. 使用私有IP作为来源或目的地址的封包时不能通过Internet来转送，网络会发生混乱。
3. 关于私有IP的参考记录，如DNS，只能限于内部网络使用

这样的好处时，你的私有路由信息没有散布在外，也不会被网络抓包发现。但是就不能使用私有IP连接互联网了。

平时在网络平台上查询IP的时候和自己在命令行中ipconfg得到的ip地址是不同的，这里就有一个是内网IP一个是外网IP。简单来说一个是用于连接互联网，实现上网功能的IP,一个是用于局域网内部分发一定数量网络权限的IP。

出现原因，IPv4规则下，所能表示的IP数量有限，而每个电脑必须分配不同的IP，供需不平衡的现象，所以必须区分公网与私网，且只有私网可以上网。即

- Public IP : 公共 IP ，经由 INTERNIC　所统一规划的 IP，有这种 IP 才可以连上 Internet；
- Private IP : 私有 IP 或保留 IP，不能直接连上　Internet 的 IP ，主要用于局域网络内的主机联机规划。

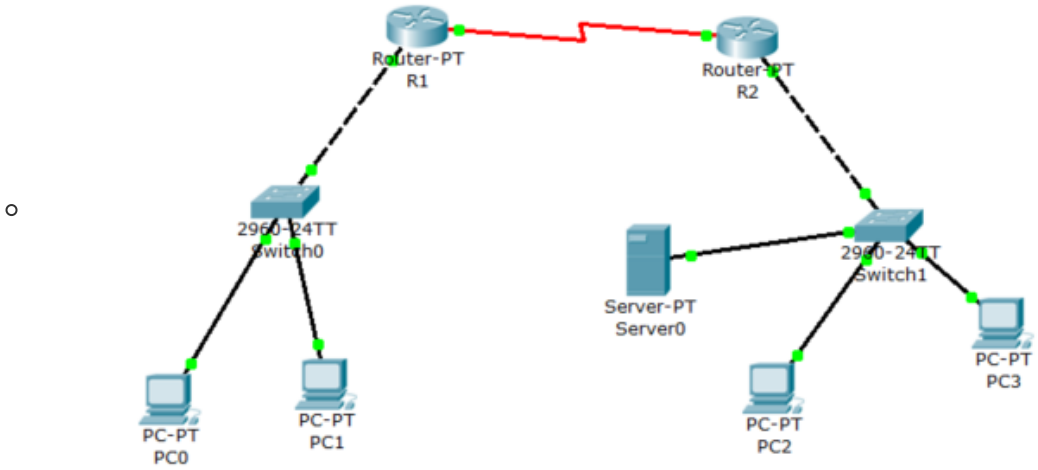
既然自己的ip是192.168开头，又是怎样转换为公网IP的呢。他们是利用到了NAT转换技术，因为我们的私网IP建立时有一定的体系，建设完毕后只需要将公网IP和私网IP做一个转换即可。

2. **NAT转换**
   - 一般的NAT设备支持UDP/TCP/ICMP，但是对于其他协议有的NAT设备支持，有的不支持，所以一些应用无法工作，遇到此类情况首先确认NAT是否支持
   - 一般的NAT设备支持UDP/TCP/ICMP，但是对于其他协议有的NAT设备支持，有的不支持，所以一些应用无法工作，遇到此类情况首先确认NAT是否支持。如果这里进行的是一对一转换，那么无法节约IPV4地址空间；所以一般NAT都是一对多的映射方式，即**一个公网**IPV4映射多个私网IPv4。**那么如何区分不同的私网IPv4的host：
   -
     - NAT使用基于session的转换规则
     -
       - 对于TCP/UDP使用：Host 's 私有IPv4 +　Port <------> NAT 公网IPv4 + Port
       - 对于ICMP使用：Host 's 私有IPv4 +　Port <------> NAT 公网IPv4 + Port
     - 由于session ID　在NAT设备上是独一无二的，所以NAT可以很容易区别局域网内部的不同host。
   - 一般路由器的1-1024端口会保留给 well-known　服务端口，所以会有65535-1024 = 64511个可用端口，换句话说可以支持64511个session，比如一个host 开了10　个session，就会耗费10个 NAT session，但是好在有NAT entry timeout，不用的session 会很快超时flush掉。NAT的性能主要由硬件影响。只要生成NAT表项，就会硬件转换。
   - NAT路由器对外界的行为反过来就如同一个具有单一IP地址的单一设备
   - 通常情况下，路由器从DHCP得到单一IP地址。路由器从ISP的DHCP服务器得到它的地址，并且路由器运行一个DHCP服务器，为位于NAT-DHCP路由器控制的家庭网络地址空间中的计算机提供地址。
   - 为了让路由器知道它应将某个分组转发给哪个内部主机，可以利用在NAT路由器上的一张NAT转换表（NAT translation table），并且在表项中包含了端口号及其IP地址。
   - NAT有三种类型:

- **静态NAT**: 一对一映射,每台主机对应一个真实的IP地址
- **动态NAT**: 映射一个未注册IP地址到注册IP地址池中的一个注册IP地址.
- **PAT**: 通过端口区分内网主机,将多个私网IP地址映射到一个公网IP

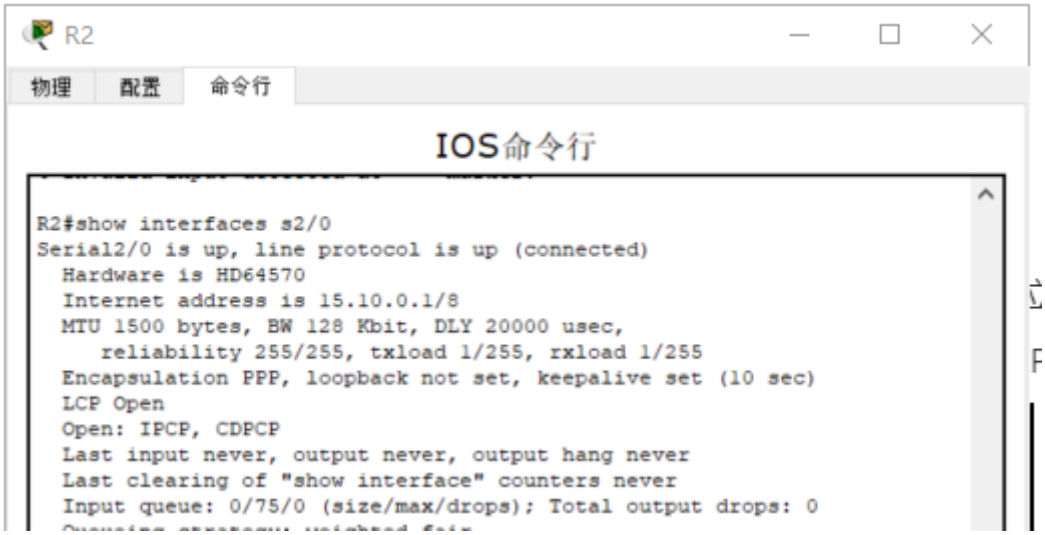
3. **NAPT**
   - **网络地址端口转换协议，把网络分成内网和外网两个部分。****NAPT网关会把任何从内网出去的数据包的源地址和源端口转换成网关自己的外网网址和一个新分配的端口，然后将数据包发送到外网，网关同时接收从目的服务器返回来的数据包并且把目的地址和目的端口转换为内网的地址和端口，然后将数据包发到内网，这样就完成了通信的过程。**
   - NAPT网络地址端口转换，我们的家用路由器全部采用NAPT的方式工作，NAPT最基本的功能就是让内部网络全部都能上网。

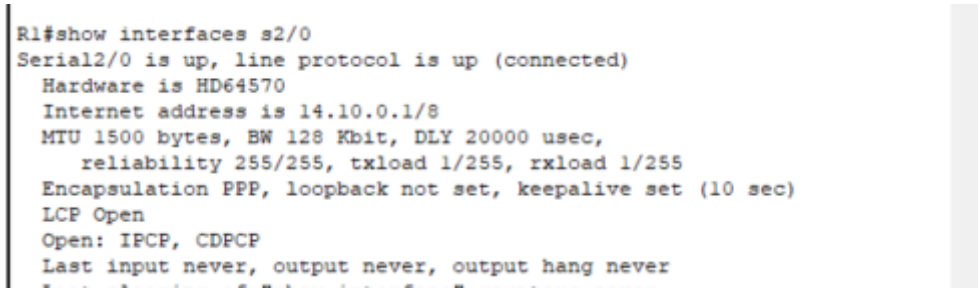
实验，以上一次的结构为基础



- 实验前需要修改部分pc的网关，路由器的ip地址，静态路由

  同样需要在两个路由器之间配置ppp协议，来保证路由器之间可以正常通信





同样对两个路由器进行静态路由的设置，但这里注意到不能像之前那样简易的直接配置成对方局域网的网端，还要同时兼顾自己的网端。要不然配置的东西都是无效的了。

```
R2#show ip route
Codes: C - connected, S - static, I - IGRP, 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 inte
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

     14.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       14.10.0.0/16 [1/0] via 14.10.0.1
C       14.10.0.1/32 is directly connected, Serial2/0
C    15.0.0.0/8 is directly connected, Serial2/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
R2#
```

```
R1#show ip route
Codes: C - connected, S - static, I - IGRP, 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 inte
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is not set

C    14.0.0.0/8 is directly connected, Serial2/0
     15.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
S       15.10.0.0/16 [1/0] via 15.10.0.1
C       15.10.0.1/32 is directly connected, Serial2/0
C    192.168.1.0/24 is directly connected, FastEthernet0/0
R1#
```

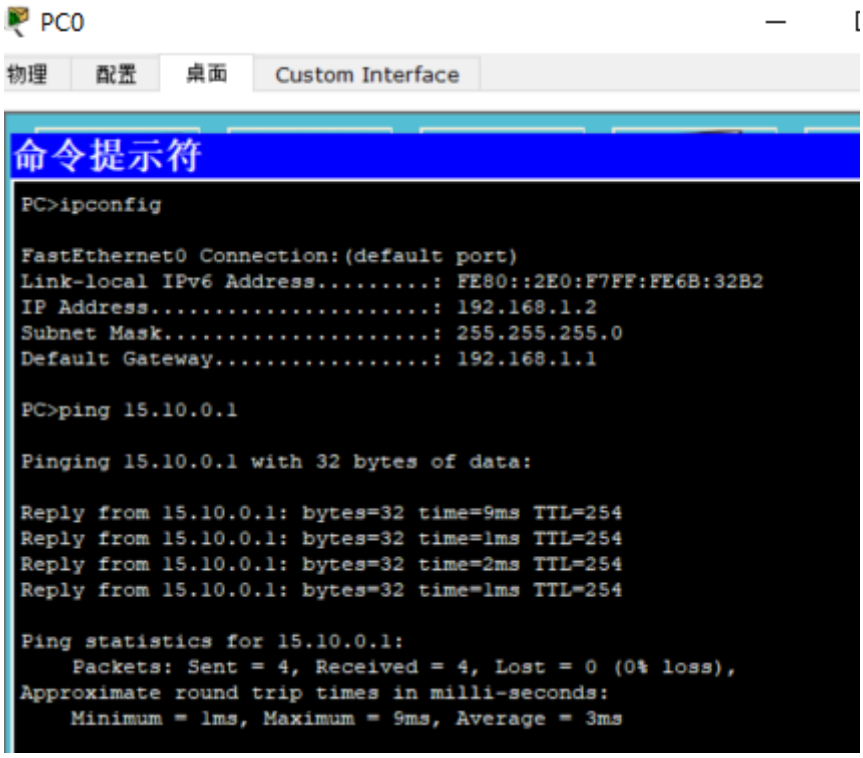
两个路由器互相执行ping操作

```
R1#ping 15.10.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 15.10.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/12 ms
```

```
R2#ping 14.10.0.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 14.10.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/11/16 ms
```

得到验证

以PC0为例进行初步检验



已经可以访问到内网路由器R1，

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 14.10.0.2:17      192.168.1.2:17     15.10.0.1:17       15.10.0.1:17
icmp 14.10.0.2:18      192.168.1.2:18     15.10.0.1:18       15.10.0.1:18
icmp 14.10.0.2:19      192.168.1.2:19     15.10.0.1:19       15.10.0.1:19
icmp 14.10.0.2:20      192.168.1.2:20     15.10.0.1:20       15.10.0.1:20
```

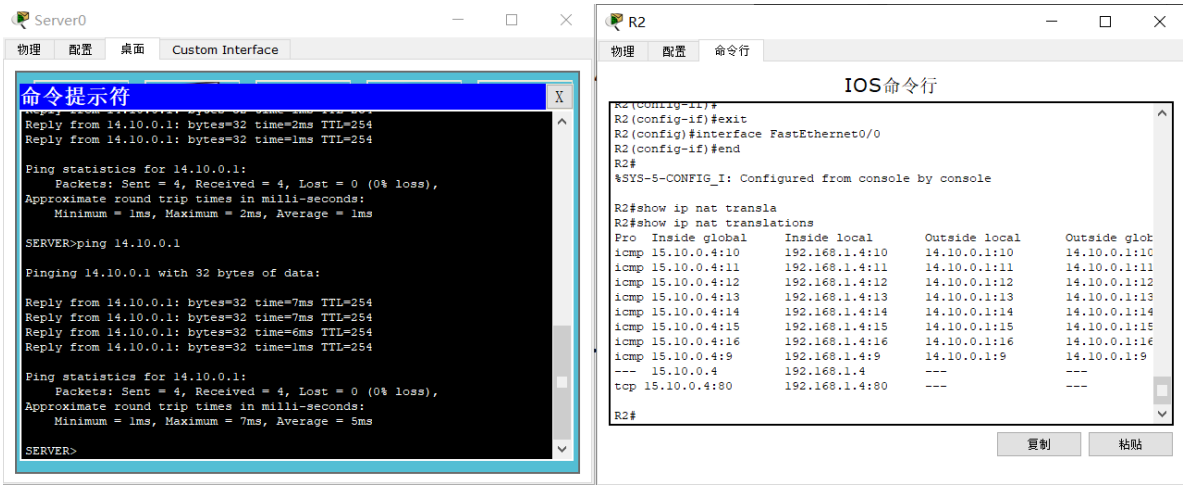
对R1进行NAT配置，分配了一个pool，其中的ip仅有14.10.0.2，该控制访问列表list的号码为1，名字为nanpool，且成功实现关联

```
R1#show ip nat st
R1#show ip nat statistics
Total translations: 2 (0 static, 2 dynamic, 2 extended)
Outside Interfaces: Serial2/0
Inside Interfaces: FastEthernet0/0
Hits: 34  Misses: 589
Expired translations: 20
Dynamic mappings:
-- Inside Source
access-list 1 pool nanpool refCount 2
 pool nanpool: netmask 255.255.255.0
       start 14.10.0.2 end 14.10.0.2
       type generic, total addresses 1 , allocated 1 (100%), misses 0
```

且此时通过检测R1作为nat设备的记录可以发现，**此时设备对外的ip已经成功修改为14.10.0.2**，并为每个数据报分配了一个独立的port，以供明确其具体信息，也进而避免了冲突。

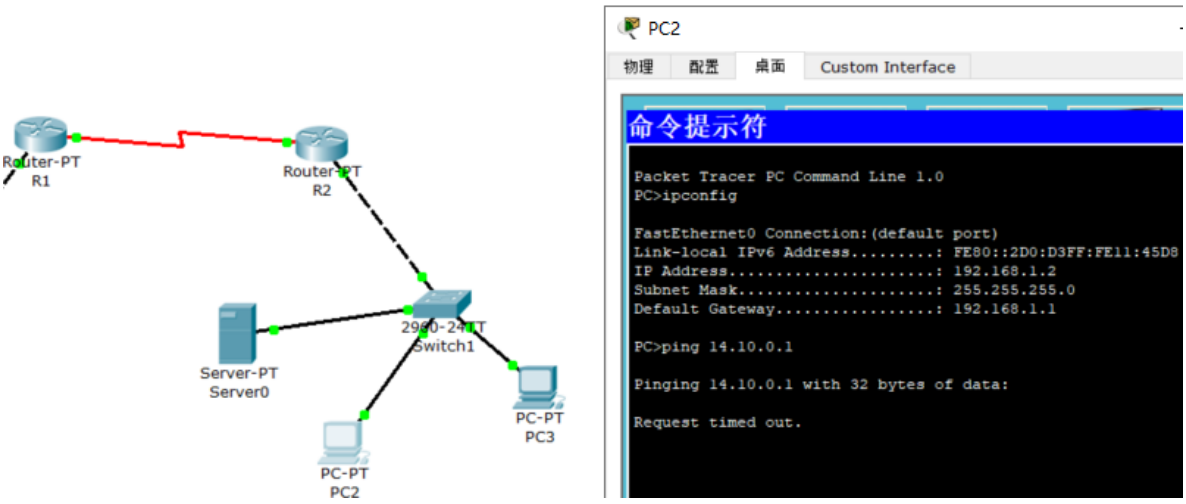
利用另一台pc： PC1进行测试

```
R1#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
icmp 14.10.0.2:1       192.168.1.3:1      15.10.0.1:1        15.10.0.1:1
icmp 14.10.0.2:2       192.168.1.3:2      15.10.0.1:2        15.10.0.1:2
icmp 14.10.0.2:3       192.168.1.3:3      15.10.0.1:3        15.10.0.1:3
icmp 14.10.0.2:4       192.168.1.3:4      15.10.0.1:4        15.10.0.1:4
```



利用已经配置公网ip的server对左侧路由器进行ping测试，发现成功。

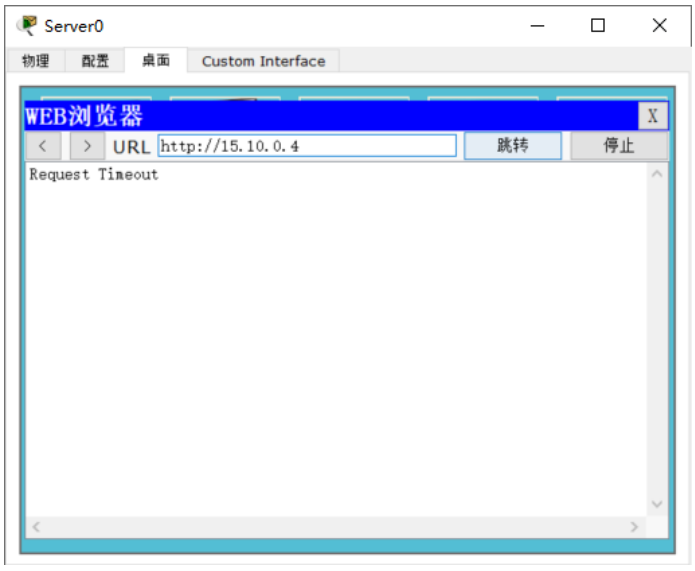
但是实际上发现无论是左侧还是右侧的pc机，均**无法实现对15.10.0.4的web内容的访问**

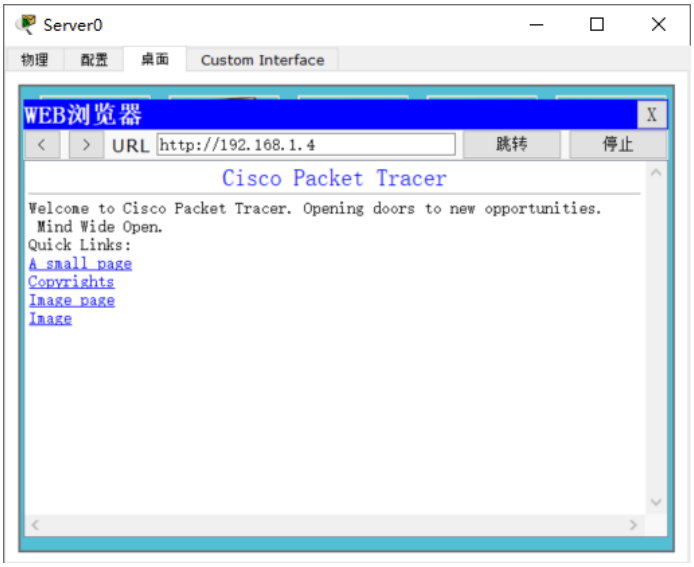


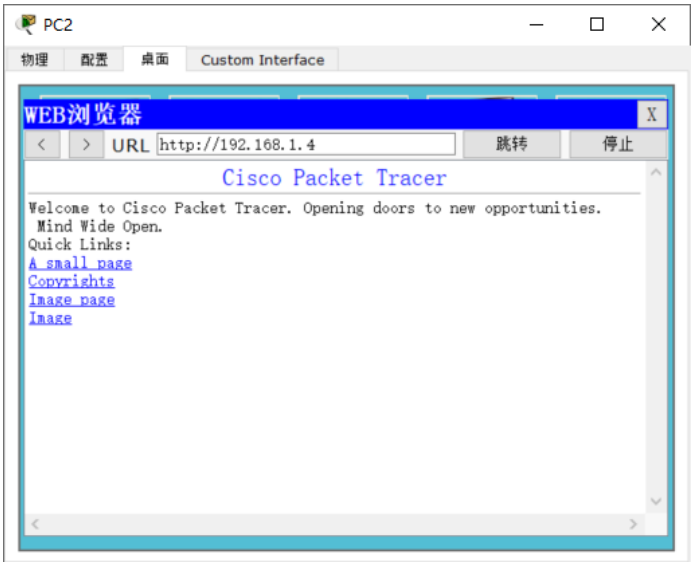
而且静态路由已经步像上一次实验那样，可以让一侧的极其ping通另一侧的路由器了，现在查找一下问题所在。

先检测web的功能是否正常运行

- 在server0上





- 在PC2上
  - 

  - 但输入15.10.0.4的时候为**超时**，在PC3上与该情况相同

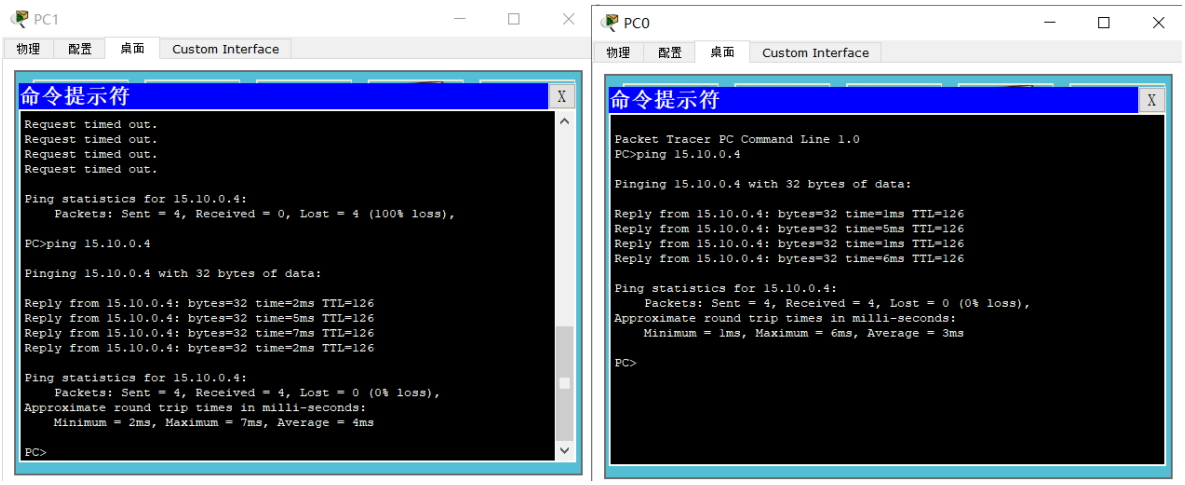
尝试修改两个关键路由器的静态路由表

但是问题仍旧出现，仅以R2为例

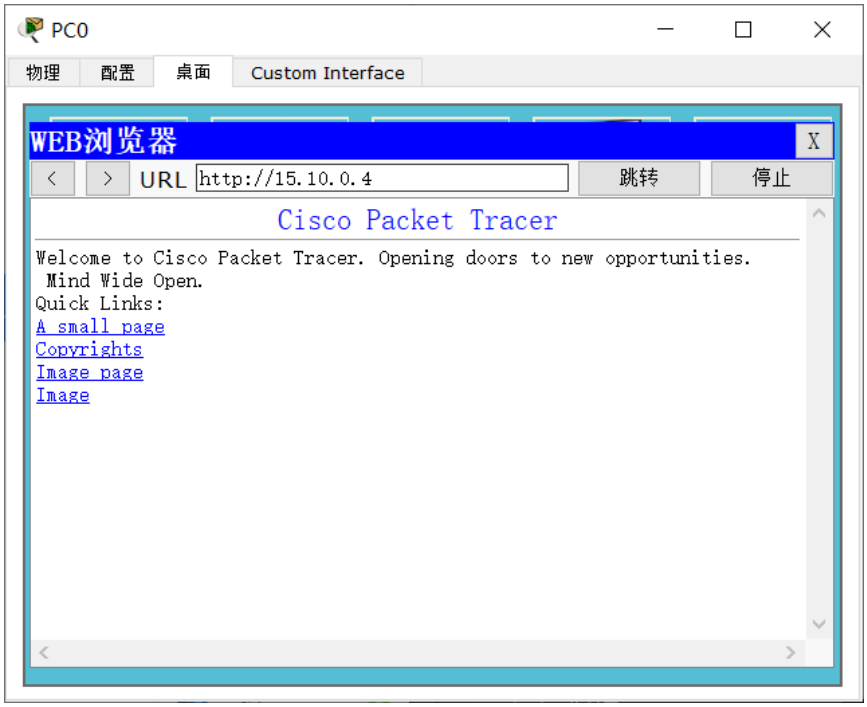
```
R2#ping 192.168.1.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.4, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 0/0/1 ms

R2#ping 15.10.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 15.10.0.4, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

仍然无法ping通修改后的公网Ip。

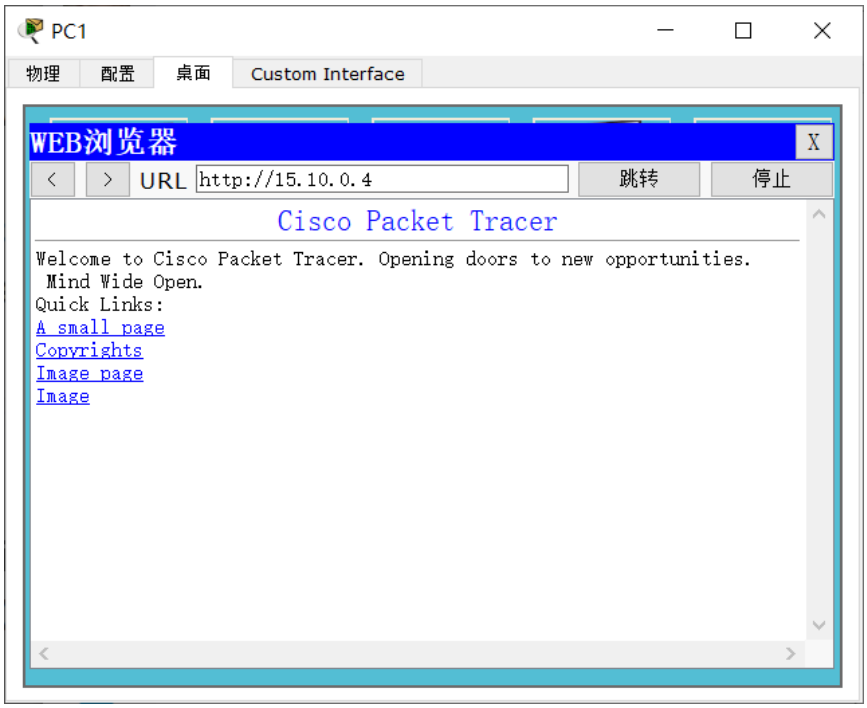
但是，在开始写实验报告的过程中，不知道为什么，仅仅时对两侧路由器各端口的ip进行检查重新赋值之后发现左侧的pc可以ping 通15.10.0.4了



此时进行必要的web访问尝试



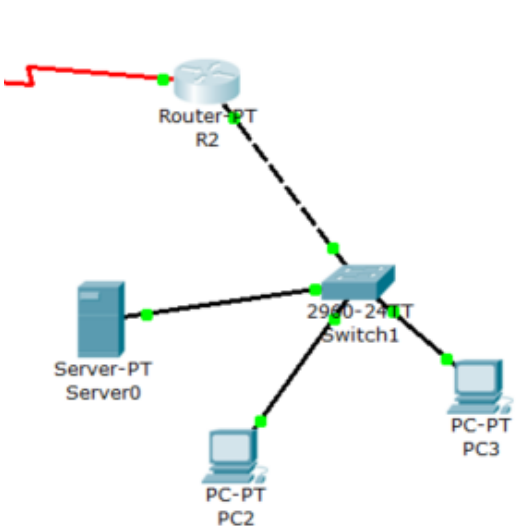
pc0通过！注意要在这里点击停止之后再进行另一台主机的尝试，因为三次握手可能会让其他pc发起的连接失败



pc1通过！

注意到左侧配了公网IP，可以从左侧访问到右侧的信息；但是右侧本身两个pc机并未分配公网ip所以并不能ping到左侧的路由器？

但是



IOS命令行

```
R2(config-if)#exit
R2(config)#
R2(config)#end
R2#
%SYS-5-CONFIG_I: Configured from console by console

R2#ping 15.10.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 15.10.0.4, timeout
.....
Success rate is 0 percent (0/5)

R2#ping 15.10.0.4

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 15.10.0.4, timeout
.....
Success rate is 0 percent (0/5)

R2#
```

在R2上测试ping 15.10.0.4就已经失败了

```
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside glob
---  15.10.0.4         192.168.1.4        ---                ---
tcp 15.10.0.4:80       14.10.1.2:80       ---                ---
tcp 15.10.0.4:80       192.168.1.4:80     ---                ---
```

即使已经配置了静态nat，且已经进行了**端口映射操作。**但是实际上时无效的。（后又做出了更改，将14.10.1.2这一条静态nat删除了）

内部网络的pc机器无法访问内部web服务的原因

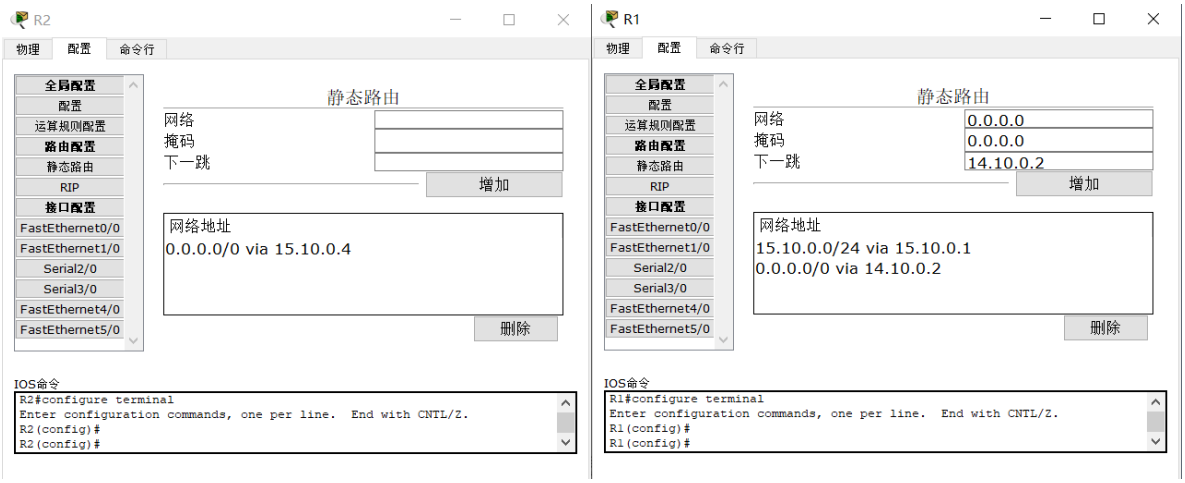
- 原因-路由回流
  当用路由器防火墙等设备将内网服务器发布到公网上，供外网用户访问的过程中出现的一种现象，就是你发现web服务器已经成功发布了，外网用户能够成功访问，但内网用户确无法访问到web服务器，这就是路由回流。造成路由回流的原因主要是出口设备路由器或者是防火墙做了NAT/PAT（也被称作源地址转换）和端口映射（也被称为目标地址转换）造成的。
- 已知一个提示未。server认为自己的ip时192.xxxx.xx.xx,而将15.10.0.4的包丢弃
- 查阅资料后试图采用内部NAT方案进行解决
  - 路由器两侧配置为nat server
  - 利用内网DNS，即内网的所有客户端的DNS的ip都应填写这台内网的DNS服务器的IP地址，还需要再内网DNS服务器上配置转发器
  - 防火墙DNS mapping
  - 路由器DNS mapping

NAT 发生在路由之前还是之后?

> 使用 NAT 处理事务的顺序基于数据包是从内部网络传递到外部网络，还是从外部网络传递到内部网络。内部到外部的转换发生在路由之后，外部到内部的转换发生在路由之前。

但是好像发现自己路由器配置颠倒了

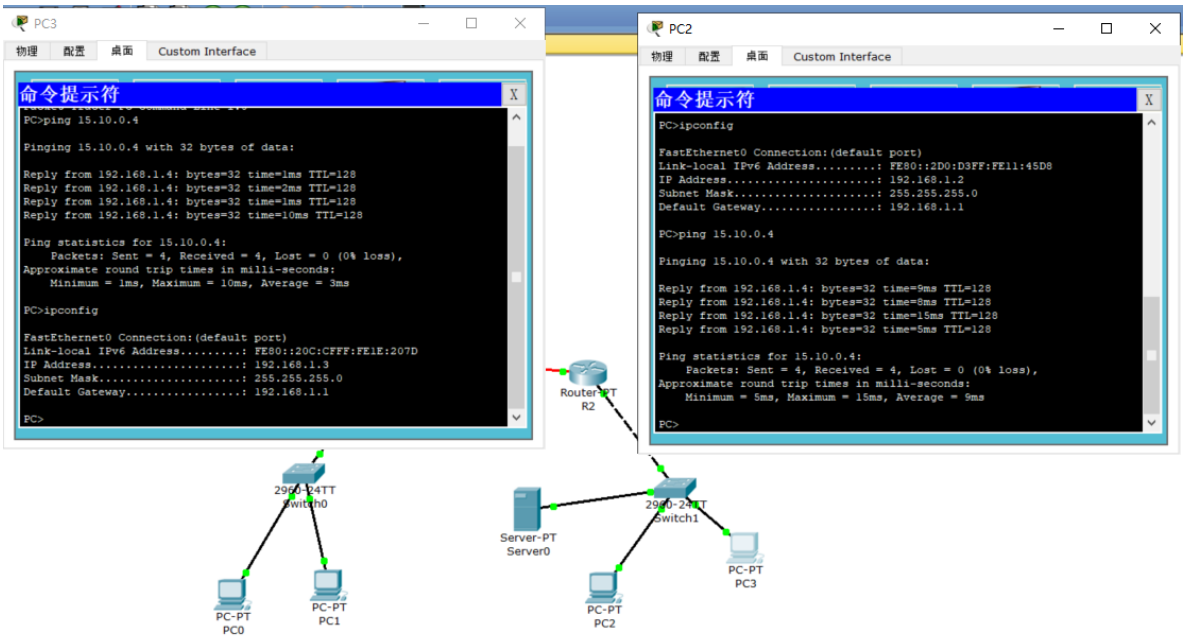
但是修改路由表后实际表明，无论是左侧还是右侧的pc都无法ping 通15.10.0.4



> 继续在反复的重复流程之后，尤其是在**不断的调试静态路由表**（要充分明白静态路由表的目的，从哪一侧到哪一侧，某个时刻谁是内网，谁又是外网），分配tcp端口之后，右侧的两台设备同样可以通过内部网段之外的公网ip进行访问了。静态路由的作用十分关键。

同时也不能忘记配置两个路由器两侧接口的nat功能；这个可能是关键因素，

> 要区分好每个路由器两侧的定位，一侧运用ip nat inside，一侧运用ip nat outside。否则的话，nat的方向错了系统不会提示，很多地方直接请求超时于是也无从debug。

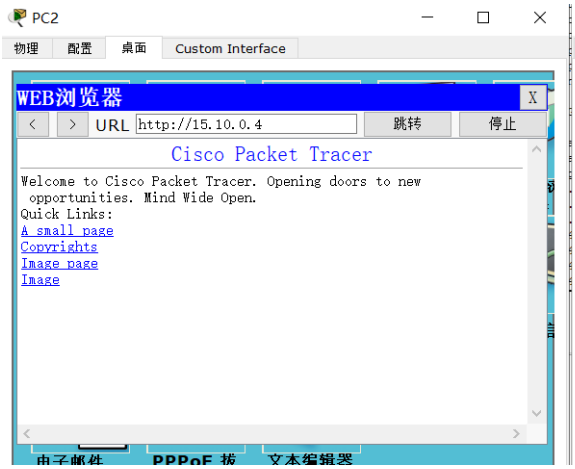


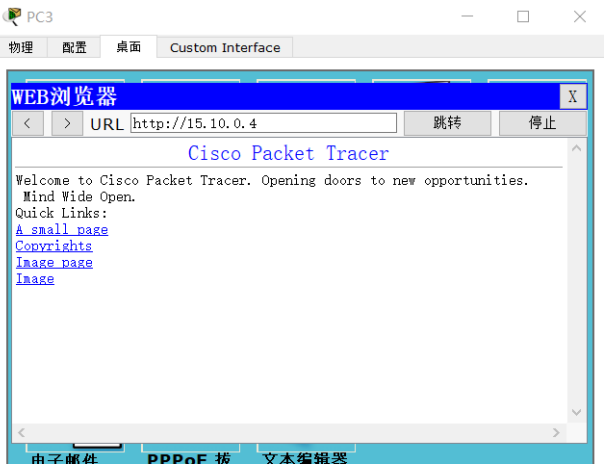
同时，还要保证可以通过web浏览器对15.10.0.4的内容进行访问，在此之前需要对R2的NAT进行更多的配置

```
R2(config)#ip nat inside source static  192.168.1.3  15.10.0.3
R2(config)#ip nat inside source static  192.168.1.2  15.10.0.2
```

```
R2#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside glob
---  15.10.0.2         192.168.1.2        ---                ---
---  15.10.0.3         192.168.1.3        ---                ---
---  15.10.0.4         192.168.1.4        ---                ---
tcp 15.10.0.4:80       192.168.1.4:80     ---                ---
tcp 15.10.0.4:80       192.168.1.4:80     14.10.0.2:1025     14.10.0.2:10
tcp 15.10.0.4:80       192.168.1.4:80     192.168.1.2:1025   192.168.1.2:
```

这样相当于指令先通过R2发到外网然后再绕回来得到结果。





在家如何访问实验室的电脑?

- **teamviewer**软件，等远程控制软件成本低廉
- **需要配置至少两台路由器，一台是实验室中的，一台是家里的电脑。**
-
  - 先明确自己的**IP**，可通过ipconfig进行确认
  - 打开自己的路由器设置，查看上网设置的WAN口的IP地址信息等，以确认外网机器（如实验室电脑）访问通过路由器电脑时通过的ip
  - 进行虚拟服务器设置，包括外网机器访问通过路由器的电脑的端口，远程连接的内网端口，内网ip
  - **利用注册表设置远程桌面默认的**xxxx端口
  - **关闭防火墙，并允许远程连接**

小结

本次实验花费了2天半，在没有理论基础的情况下查阅了大量资料终于实现最终目标。

- 成功调整静态路由表，使得配置公网IP的设备可以从内网ping通外网的设备，如PC0至R2
- 成功在R2上实现静态NAT，使得服务器有一个唯一的外网地址15.10.0.4
- 成功在R1上实现NPAT，让两台左侧pc对外有统一的外网地址14.10.0.2，并分配不同的端口号使得报文一一对应。
- 成功实现端口映射，使得tcp数据访问得以实现
- 四台pc在最后均能在浏览器中访问公网ip下的web服务器中的内容，且右侧内部网段的pc理所当然的可以通过192.x.x.x进行同样内容的访问，但操作时要间隔请求以免因为tcp建立连接而导致的部分请求失败。
- **实验中较为关键的语法**
  - # show ip nat statics
  - ip nat inside source static tcp x.x.x.x 80 x.x.x.x 80 (端口映射)
  - no ip nat inside source list **num** pool **name**
  - access-list **num** permiit 192.168.1.0 0.0.0.255
  - ip nat pool nanpool 14.10.0.2 14.10.0.2 net mask 255.255.255.0
  - ip nat inside source list **num** pool nanpool overload
  - show ip nat translations (观察转换的log)