

哈尔滨工业大学

信息安全概论课程实验报告

实验三

## 学院：计算机学院

## 班级：11403202

## 学号：1140320206

## 姓名：霍峻杰

1. 实验目的

理解路由器的防火墙工作原理，掌握路由器的防火墙功能配置方法，主要包括网络地址转换功能和数据包过滤功能的配置。

培养根据具体环境与实际需求进行网络地址转换及数据包过滤的能力。

1. 实验内容及实现

# 任务一：使用Packet Tracer构建网络拓扑

      实验内容：使用Packet Tracer构建如下所示网络：

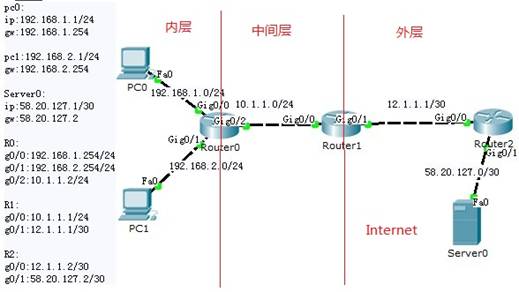


图1实验网络拓扑图

      说明：

      1）本网络拓扑为多层网络架构：内层网络、中层网络、外层网络、Internet网络。其边界的主要标志是路由器。

      2）内层网络为连接在路由器Router0的G0/0和G0/1端口的网络，有两个网络段：192.168.1.0/24和192.168.2.0/24；网络出口为Router0的G0/2端口，与中层Router1相连。

      3）中层网络为连接在路由器Router1的G0/0端口的网络，网络段为10.1.1.0/24；网络出口为Router1的G0/1端口，与外层Router2相连。

      4）外层网络为连接在路由器Router2的G0/0端口的网络，网络段为12.1.1.0/30；网络出口为Router2的G0/1端口，与外部网络（如Internet网络）相连。

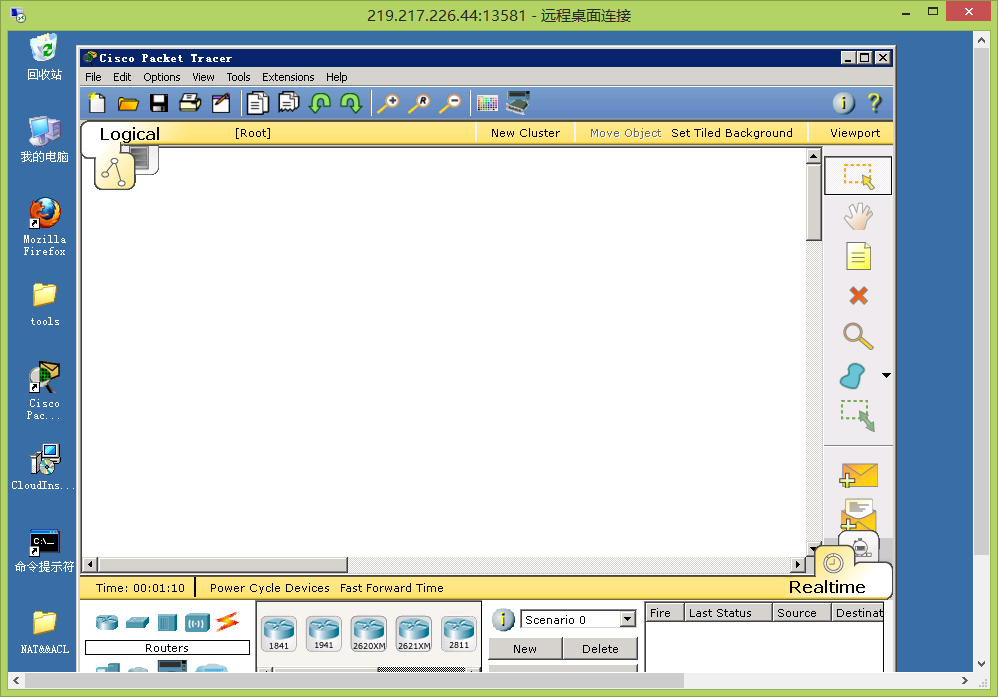
      5）图示PC1与PC2为内层网络所接主机，server0为Internet网络所接服务器。

      6）各网络中网络设备的端口及IP、主机和服务器的IP见图中左边列表。

      实验步骤如下：

      步骤一：运行Packet Tracer。

      登录到实验机后，在桌面上有一个Packet Tracer图标，双击该图标运行Packet Tracer软件。界面如下：



      界面主要部分说明：

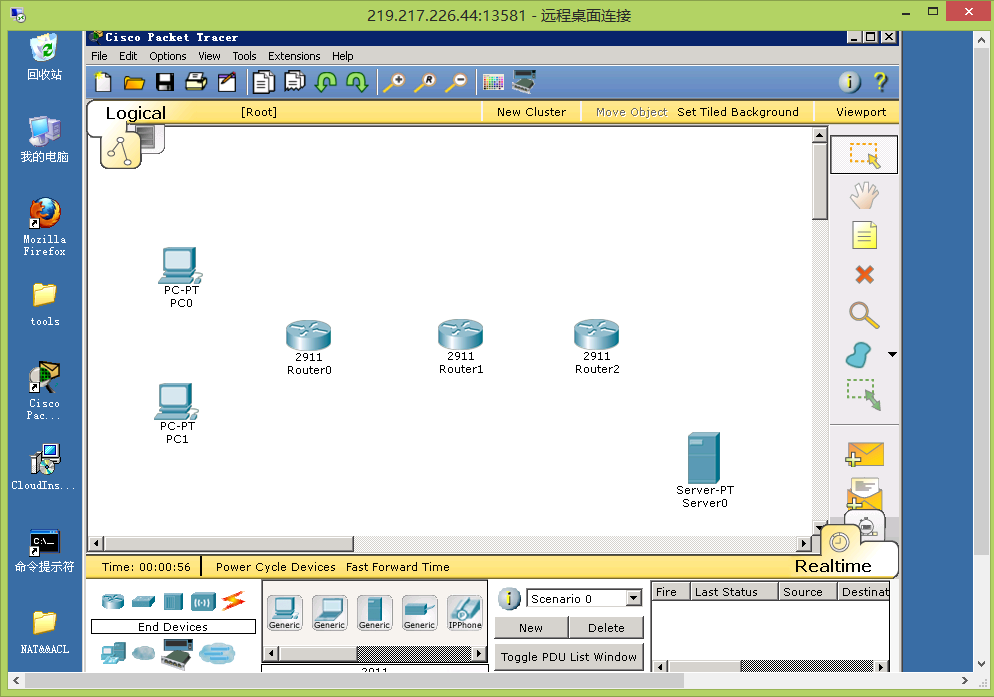
      1）界面中间大空白部分，为设计拓扑图工作区；

      2）界面左下角，为设备种类选择区；

      3）界面中下角，为设备型号区，与设备类型相关；

      步骤二：向网络拓扑中添加物理设备（包括路由器、主机、服务器等）。

      在软件界面下，通过拖拽的方式，将路由器、主机、服务器等拖到工作界面下，按拓扑图所示位置摆放。并按拓扑图中要求修改各设备的名称。



\\n\\n\\n

\n\n\n

\n\n\n

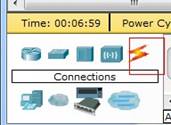
\n\n\n

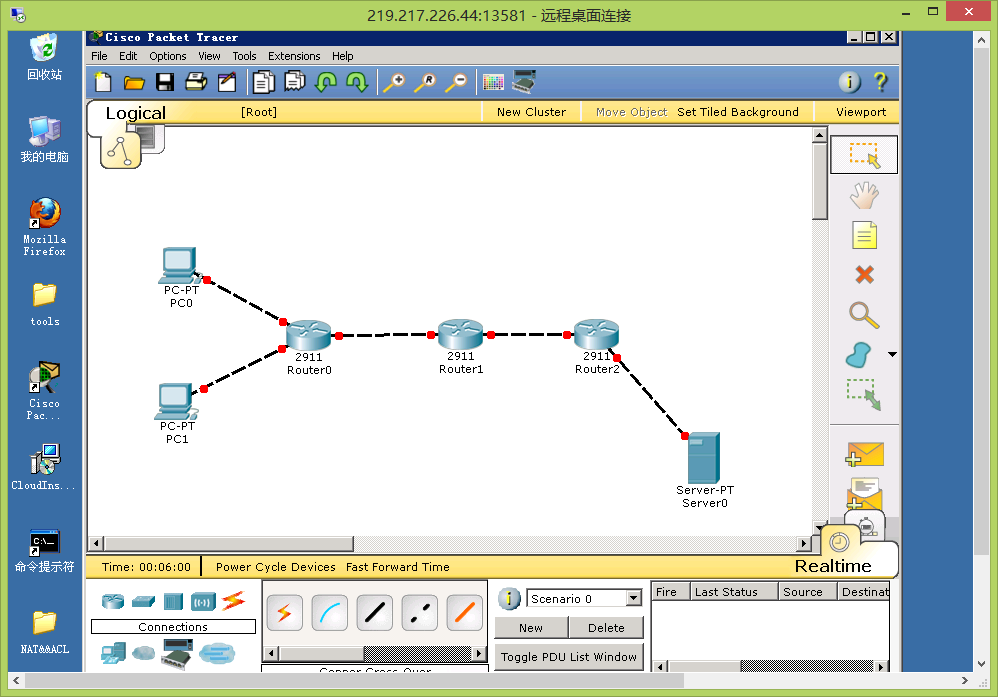
\n\n\n

|  |
| --- |
| 注：  1）添加设备方式为，首先选中所要选的“设备种类”，再到“设备型号“中选择具体的设备，拖拽到工作区即可）。  2）路由器的型号选择2911，其他所有设备型号都选择“Generic”（即通用设备）。 |

      步骤三：设置设备之间的物理连接。

      点击下左图的连线图标（见红框标注），可以看到有各种连线，见下右图。





      我们选择右图中红框所示连线类型，即交叉线。

\\n\\n\\n

\n\n\n

\n\n\n

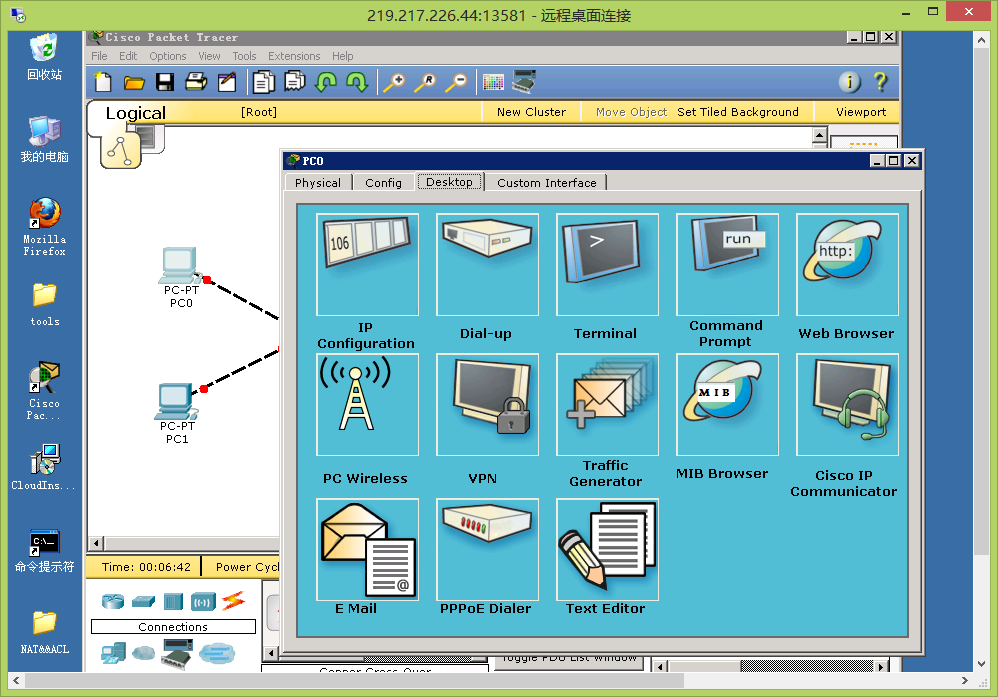
\n\n\n

\n\n\n

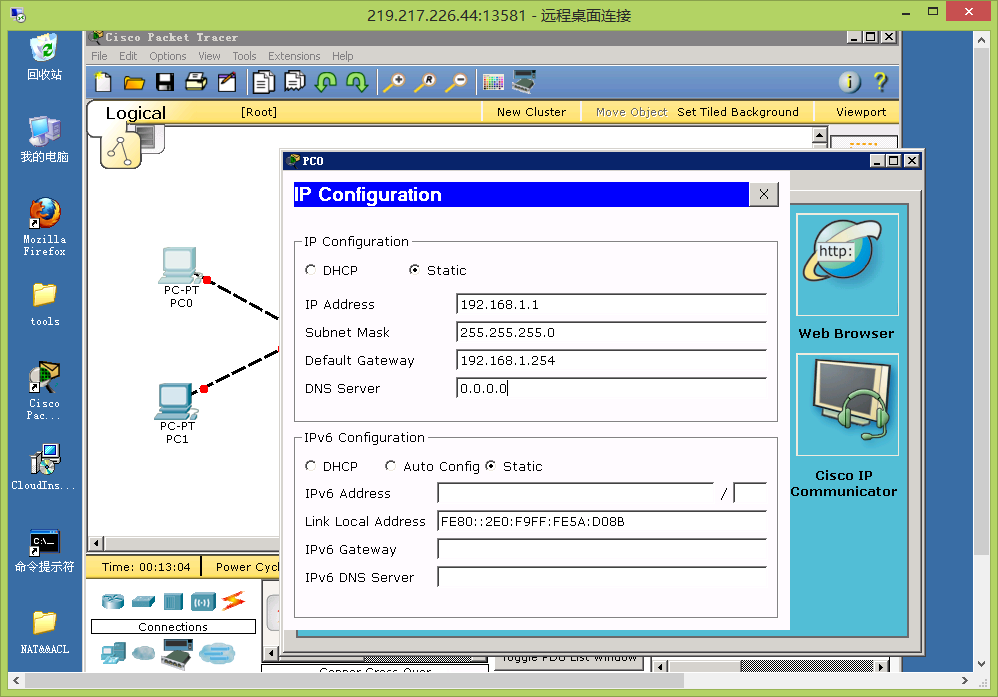
|  |
| --- |
| 注：关于网络设备的定义和设备间连接方式一般规律：  1）两类设备定义：DTE类设备和DCE类设备。  Ø DTE 类设备：PC、路由器、交换机uplink口、HUB级联口；  Ø DCE 类设备：交换机普通口、HUB普通口；  2）同类设备间相连使用交叉线方式；异类设备间相连使用直通线方式。 |

      步骤四：设置各物理设备的IP地址。

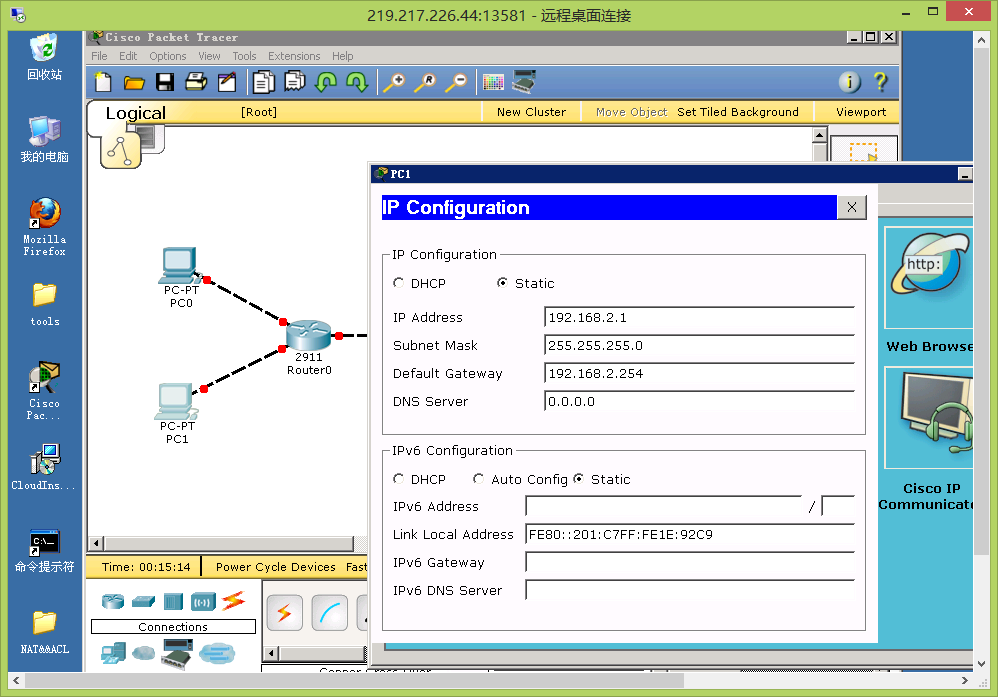
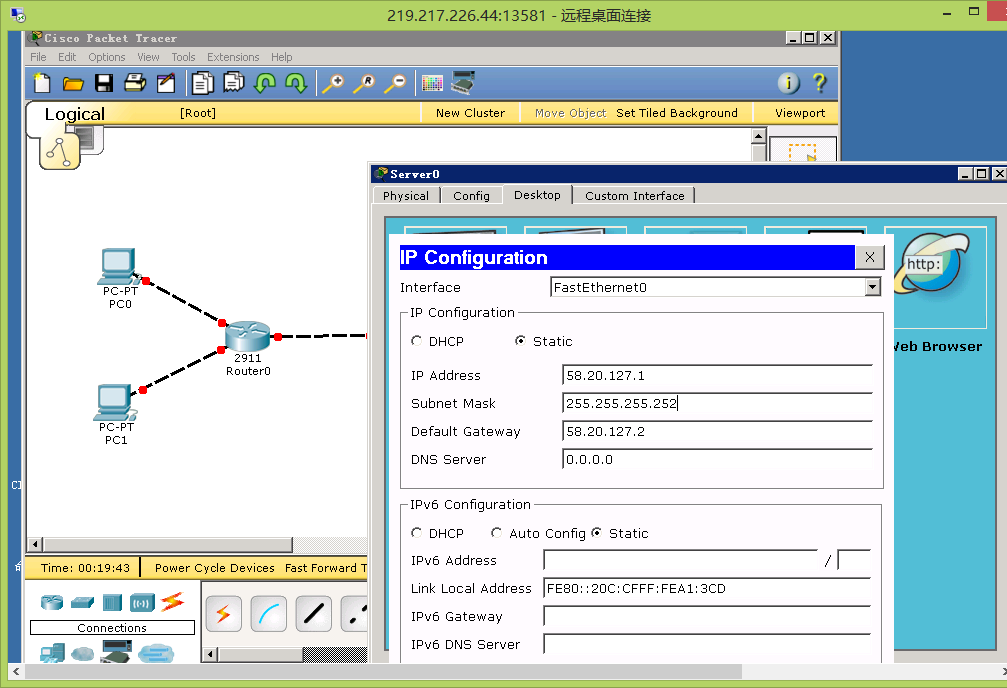
      1）pc0设置IP地址：双击pc0设备图标出现如下界面：



      点击上面红框配置IP地址。按前面拓扑图所规定IP地址进行设置。

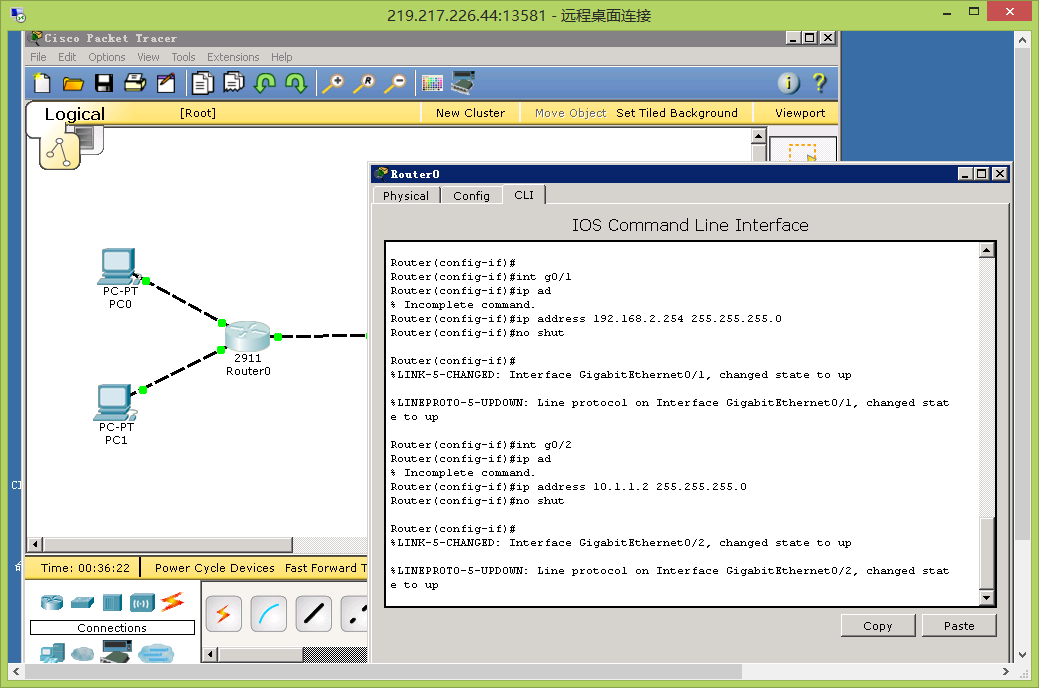


      其他的pc机和server都可以按照上述方式配置。

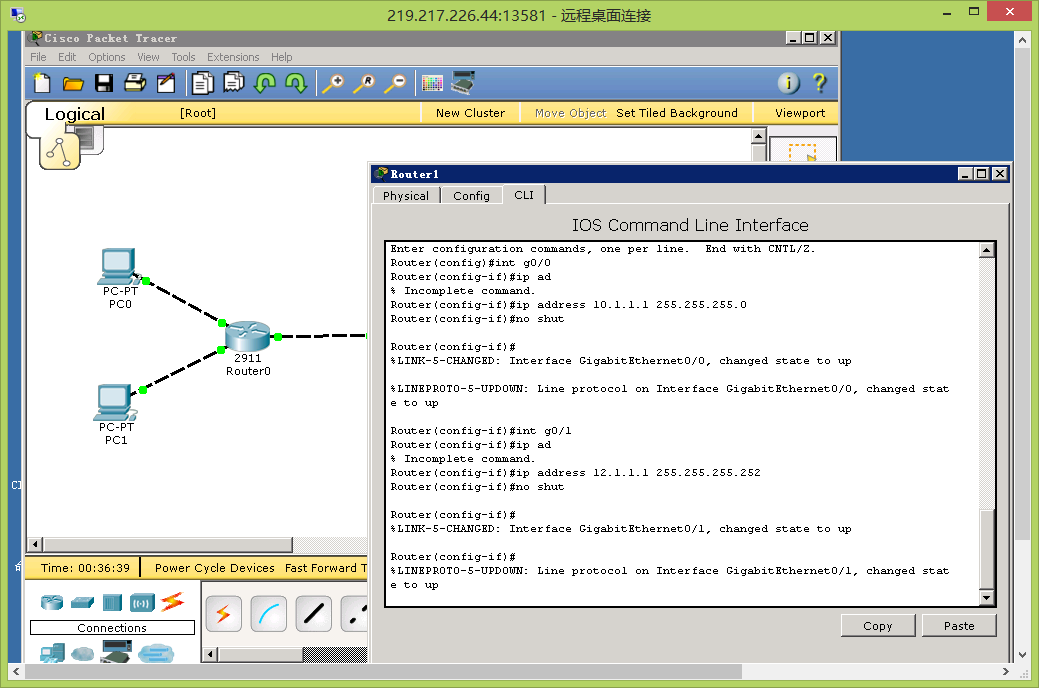
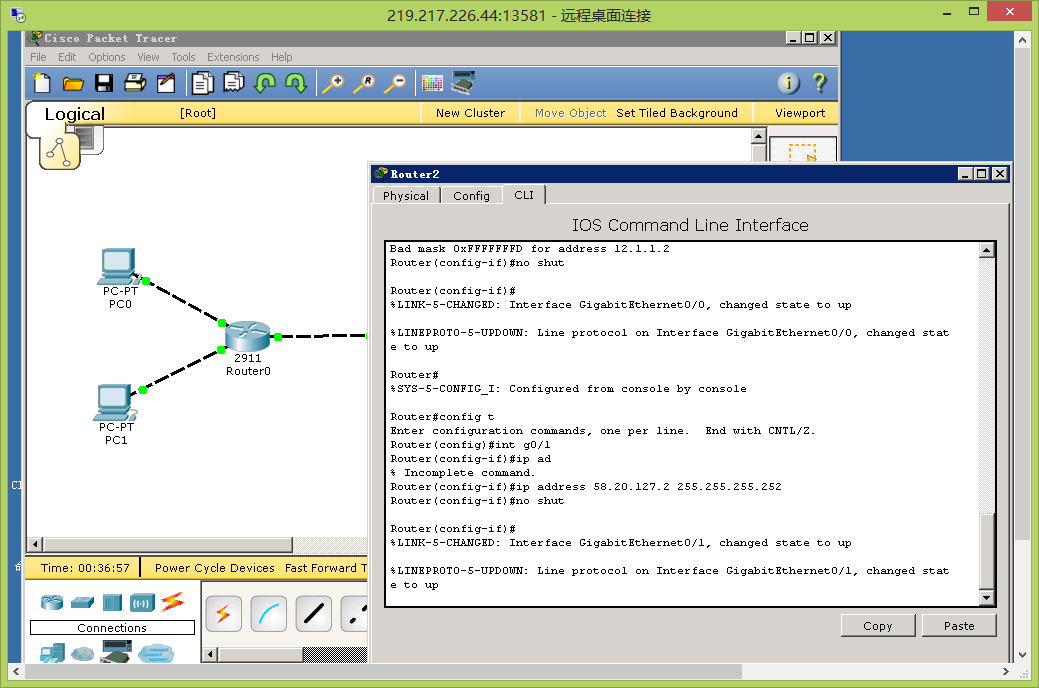


      2）配置路由器接口及IP

      双击路由器Router0图标，出现路由器详情界面，点击“CLI”菜单，即出现路由器的“用户模式”。此时可进入“特权模式”，并进行相关配置，见下图：



      其他接口也按照这种方式配置。同样，按这种方式配置好Router1和Router2。



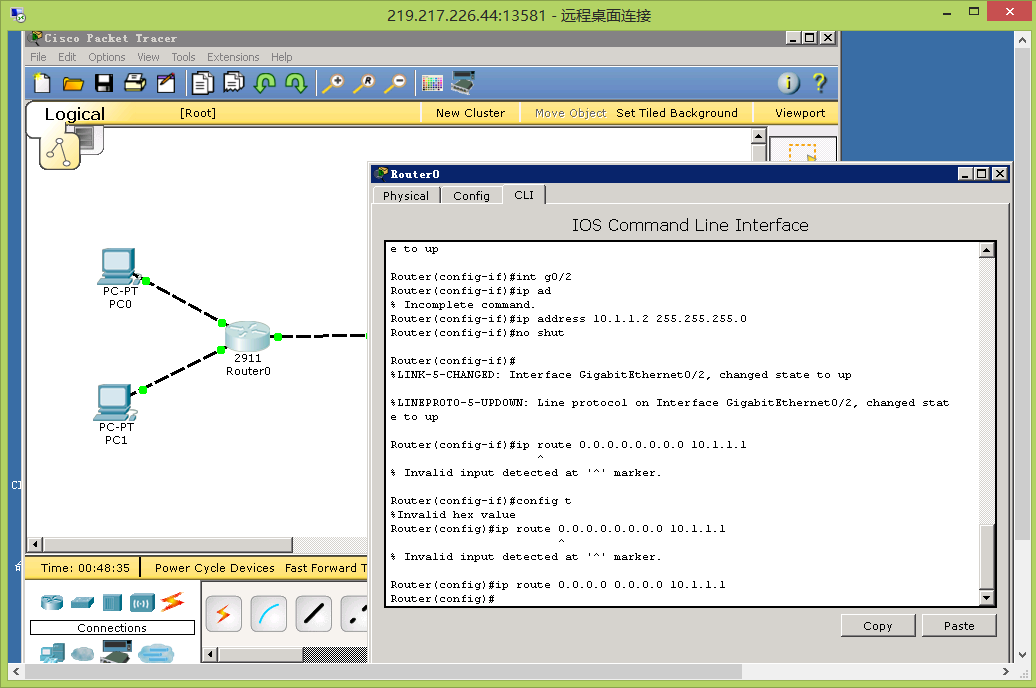
      步骤五：配置路由。

      1）配置R0的路由：

\n\n\n

\n\n\n

|  |
| --- |
| R0(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1 |

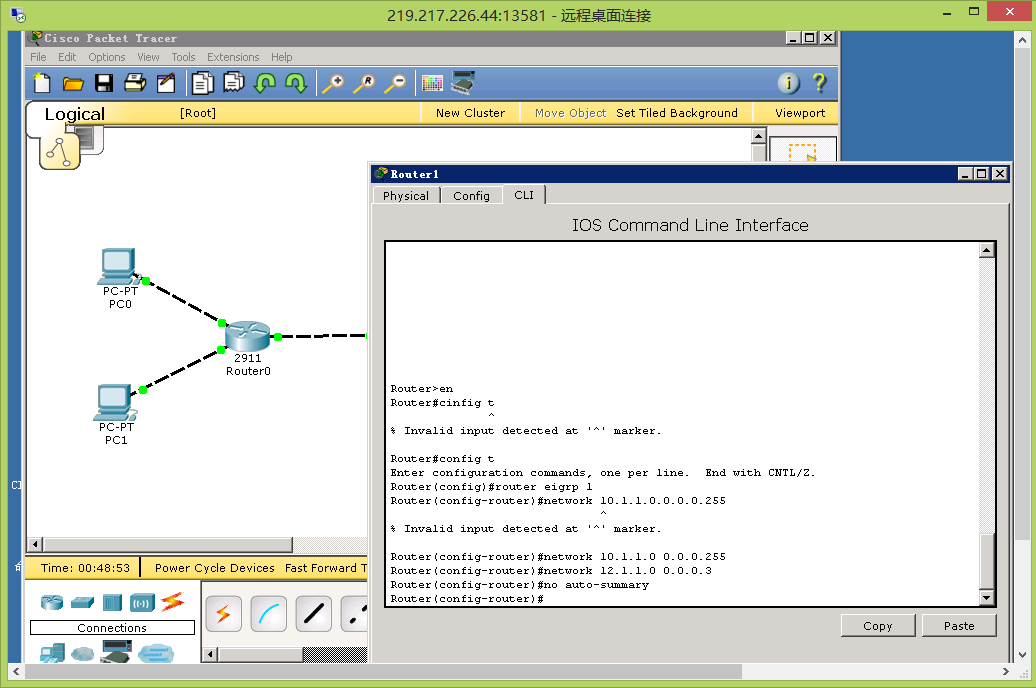


2）配置R1的路由：

\n\n\n

\n\n\n

|  |
| --- |
| R1(config)#router eigrp 1  R1(config-router)#network 10.1.1.0 0.0.0.255  R1(config-router)#network 12.1.1.0 0.0.0.3  R1(config-router)#no auto-summary |

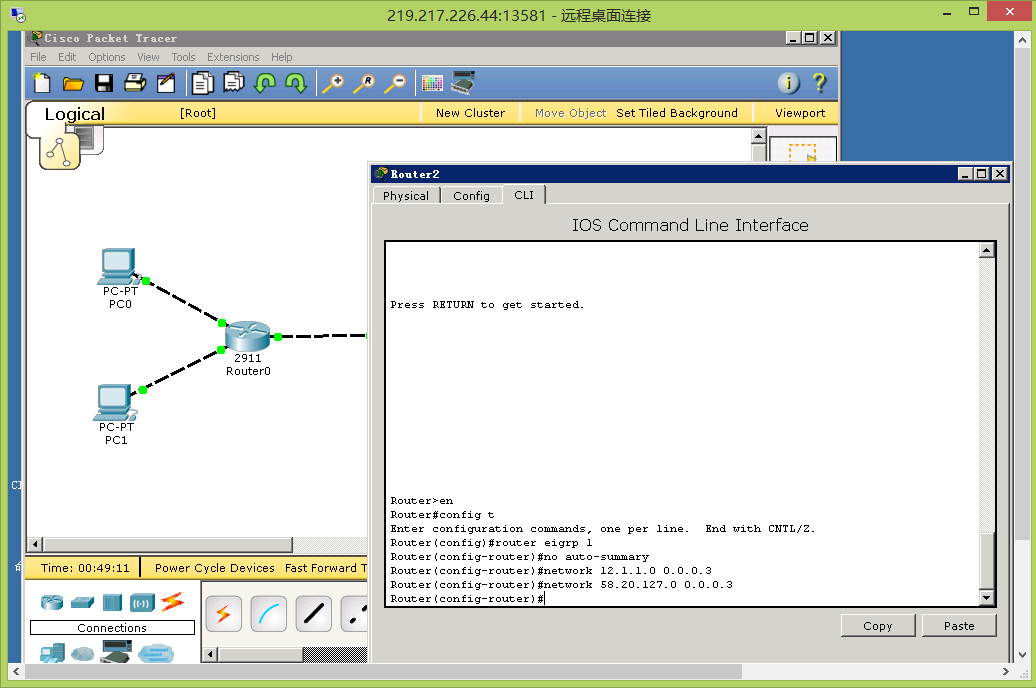


3）配置R2的路由

\n\n\n

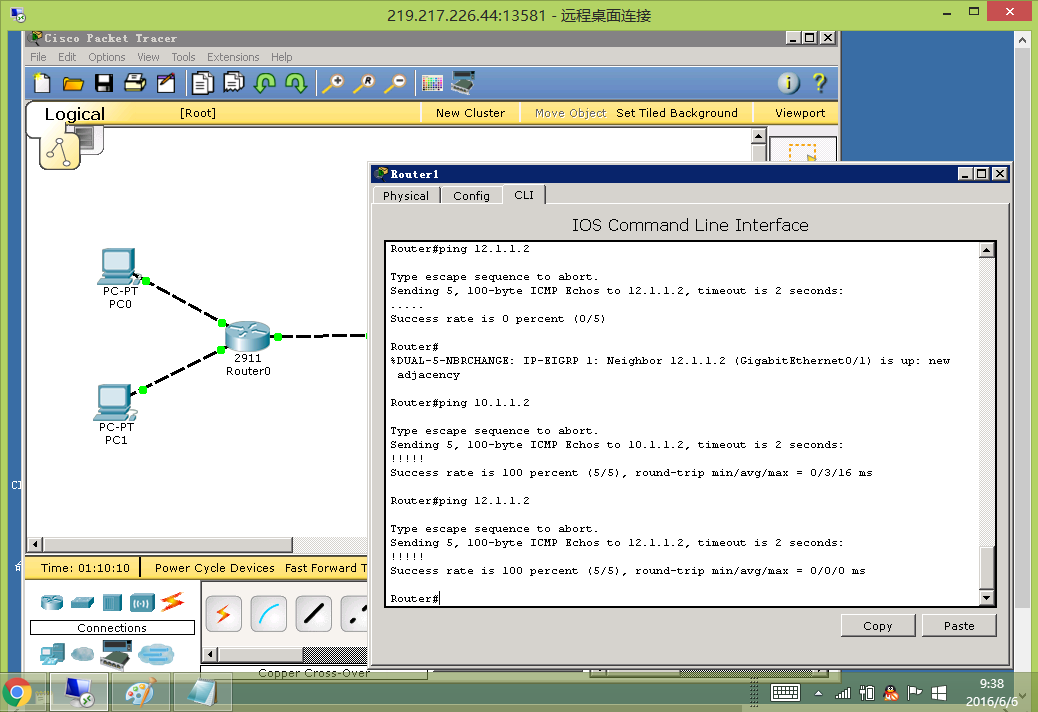
\n\n\n

|  |
| --- |
| R2(config)#router eigrp 1  R2(config-router)#no auto-summary  R2(config-router)#network 12.1.1.0 0.0.0.3  R2(config-router)#network 58.20.127.0 0.0.0.3 |

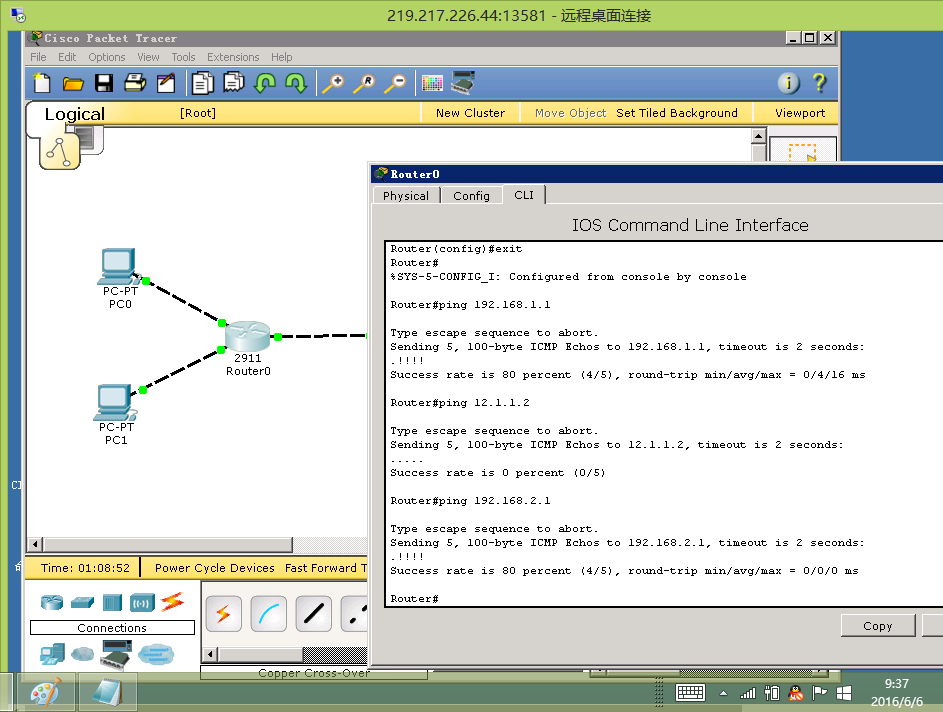
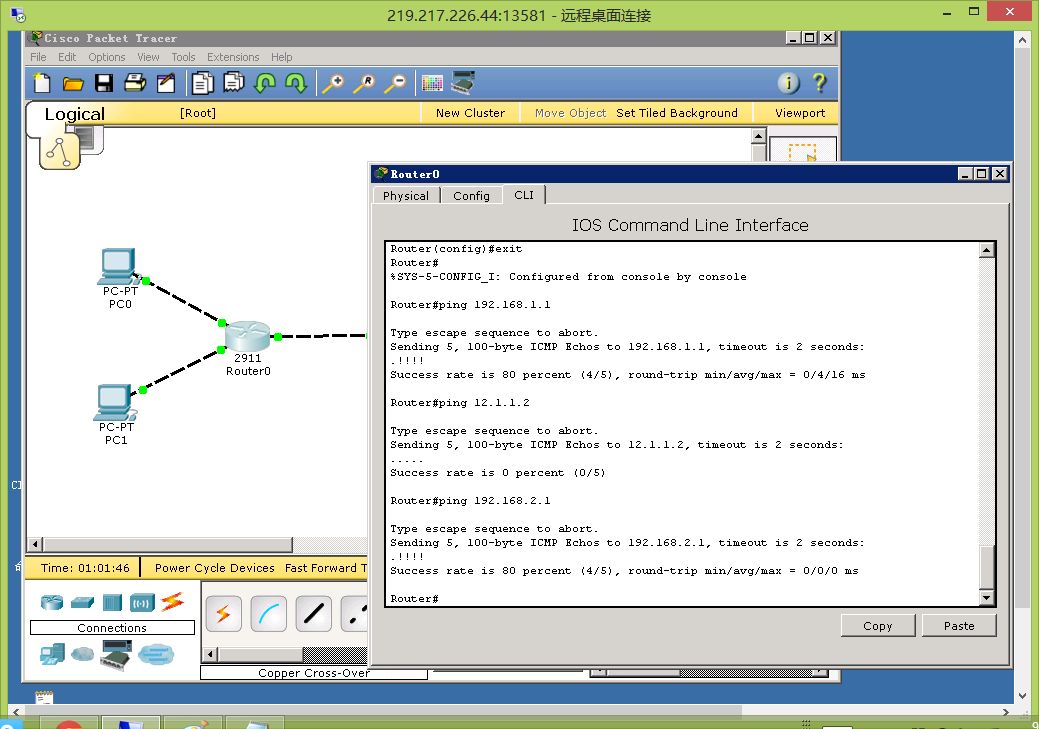


步骤六：测试。

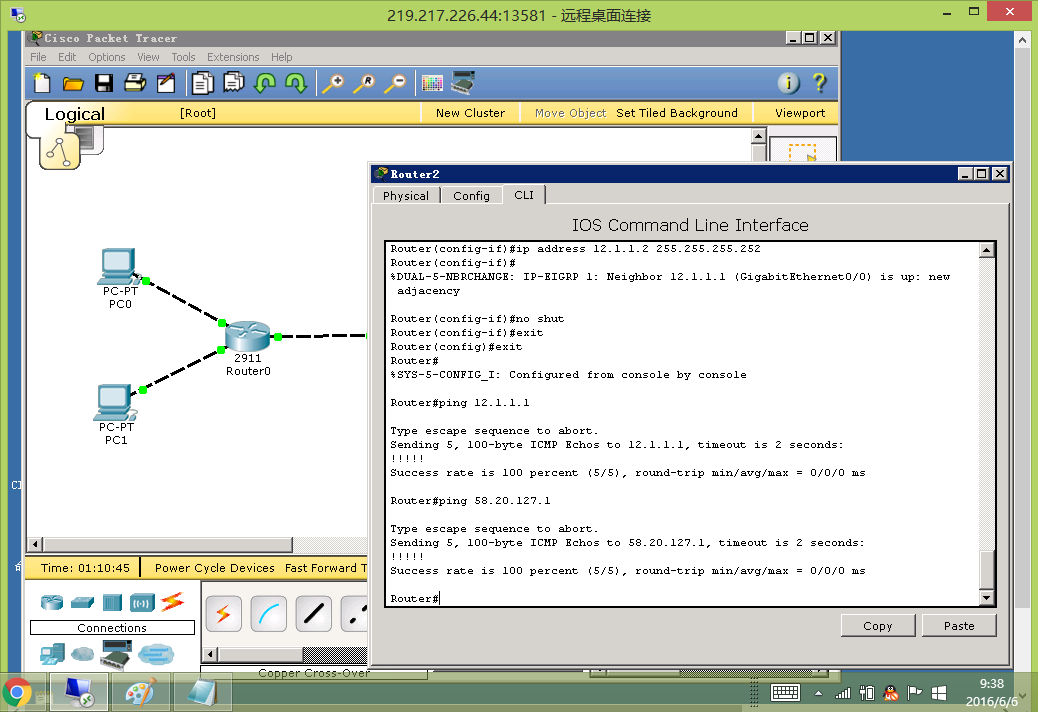
      1）在路由器R1上分别ping 与之相邻的R0和R2的IP地址，能ping通了说明配置好了。



      2）在R0上分别ping两台pc机。



      3）在R2上ping Server0。



 \n\n\n

\n\n\n

|  |
| --- |
| 注意：如果在测试环节中有任何一步测试不成功，必须先把问题解决才能做后面的实验（后面实验任务都将以本任务成功完成为前提），解决的办法是，仔细查看每台设备的IP地址配置、以及路由器的路由设置。 |

# 任务二：NAT设置

      本任务包括三个子任务：

      子任务一：静态NAT配置；

      子任务二：动态NAT配置；

      子任务三：PAT配置。

\n\n\n

\n\n\n

|  |
| --- |
| **注：**  **1、本任务的前提，是已经完成了任务一的实验操作且结果正确。**  **2、为了方便大家实验，在做实验时可以打开桌面上的NAT&&ACL文件夹，里面有对应实验的的pkt文件。** |

### ****子任务一：静态NAT****

      本子任务将实现内网IP与外网IP的静态NAT。包含两个映射：

      1）内网IP为192.168.1.1映射到外网IP为10.1.1.10。

      2）内网IP为192.168.2.1映射到外网IP为10.1.1.20。

      拓扑和路由设置（即完成任务一）。

      实验步骤如下：

      步骤一：打开桌面NAT&&ACL文件夹中的“静态nat.pkt”文件，加载初始化网络环境；

      步骤二：在R0上配置静态NAT

\n\n\n

\n\n\n

|  |
| --- |
| //配置静态nat映射  R0(config)#ipnat inside source static 192.168.1.1 10.1.1.10  R0(config)#ipnat inside source static 192.168.2.1 10.1.1.20    //配置nat内部接口  R0(config)#int g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside    //配置nat外部接口  R0(config-if)#int g0/2  R0(config-if)#ipnat outside |

      步骤三：测试

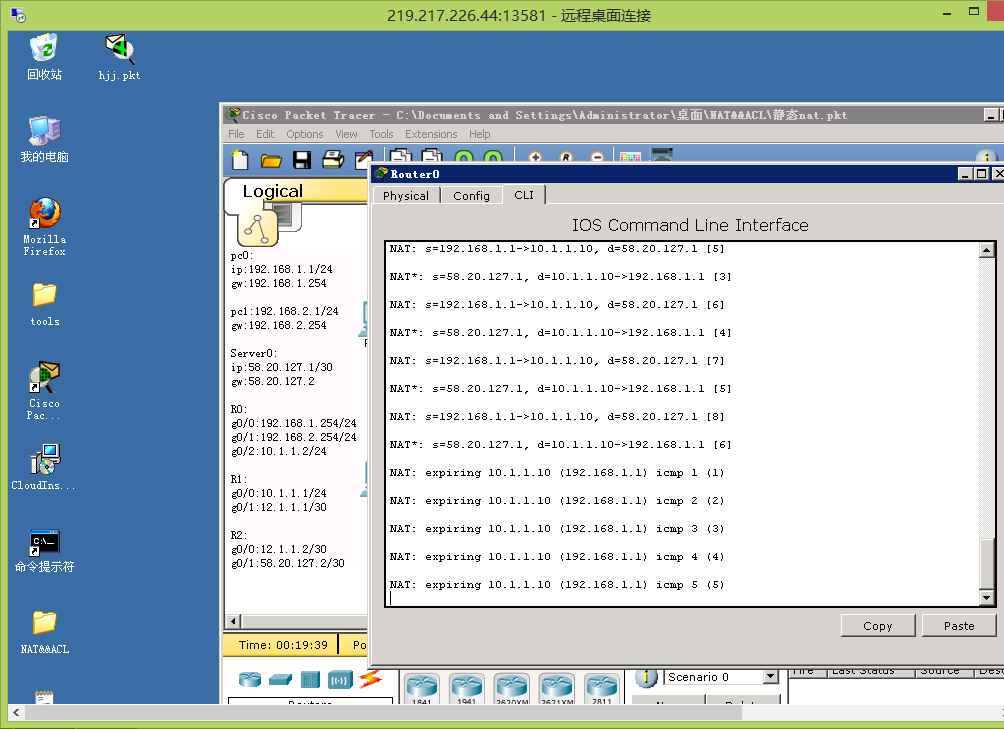
      1）在R0上，打开debug，查看地址翻译的过程：

\n\n\n

\n\n\n

|  |
| --- |
| R0#debug ipnat |

      在pc0或者pc1上ping 58.20.127.1 (Server0的地址)，此时路由器将输出如下内容：



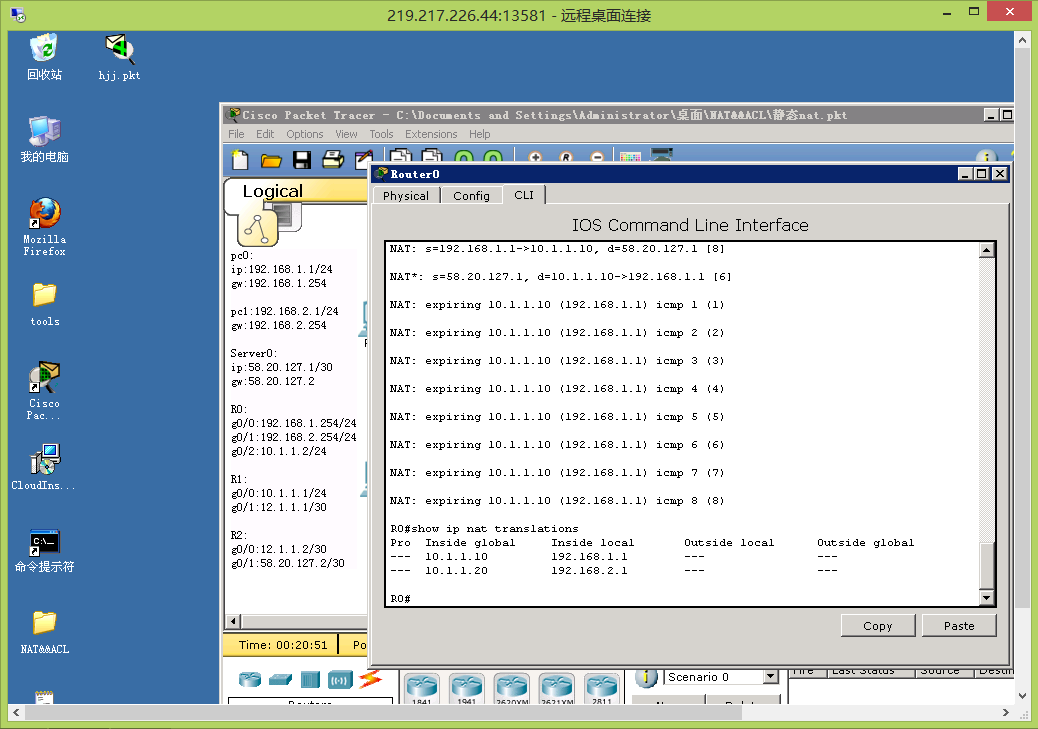
      2）在R0上查看NAT映射表

\n\n\n

\n\n\n

|  |
| --- |
| R0#show ipnat translations |

      可以看到静态时映射表一直存在，表明了内部全局地址和内部局部地址的对应关系。如下图：



      说明：

      内部局部（inside local）地址：在内部网络使用的地址；

      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；

      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；

      外部全局（outside global）地址：外部网络主机的合法 IP 地址。

### ****子任务二：动态NAT****

      本子任务将实现内网IP与外网IP的动态NAT。其中：

      1）内网需要映射的IP有两个：192.168.1.1和192.168.2.1。

      2）外网可以用于映射的IP段：10.1.1.10至10.1.1.15。

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成网络拓扑和路由设置（即完成任务一）。 |

      实验步骤：

      步骤一：打开桌面NAT&&ACL文件夹中的“动态nat.pkt”文件，加载初始化网络环境；

      步骤二：在路由器R0上配置动态NAT。

\n\n\n

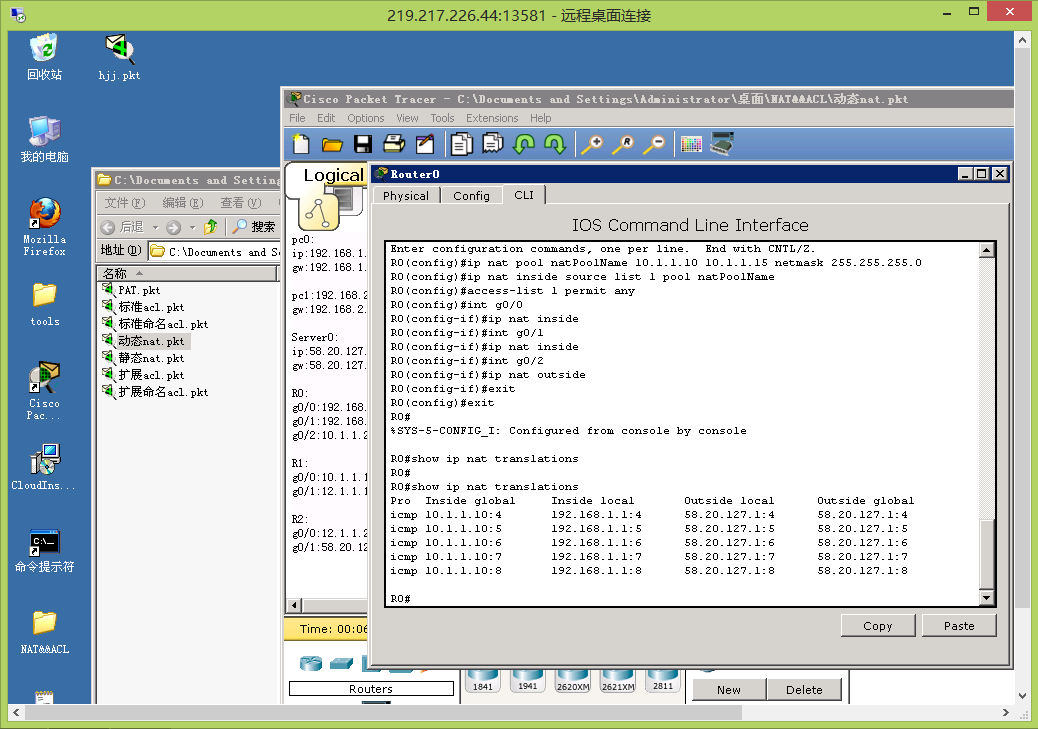
\n\n\n

|  |
| --- |
| //配置动态nat转换的地址池  R0(config)#ipnat pool natPoolName 10.1.1.10 10.1.1.15 netmask 255.255.255.0    //配置动态的nat映射，以及访问控制  R0(config)#ipnat inside source list 1 pool natPoolName    //访问控制列表1，允许全部访问  R0(config)#access-list 1 permit any    R0(config)#int g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside    R0(config-if)#int g0/2  R0(config-if)#ipnat outside |

      步骤三：测试。

      （同子任务一）

      可以看到动态映射关系只有在发起访问时存在，内网192.168.1.1映射到了10.1.1.10，而192.168.2.1则映射到了10.1.1.11。如下图：



### ****子任务三：端口PAT****

      本子任务将实现内网到外网的PAT转换。

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成网络拓扑和路由设置（即完成任务一）。 |

      实验步骤：

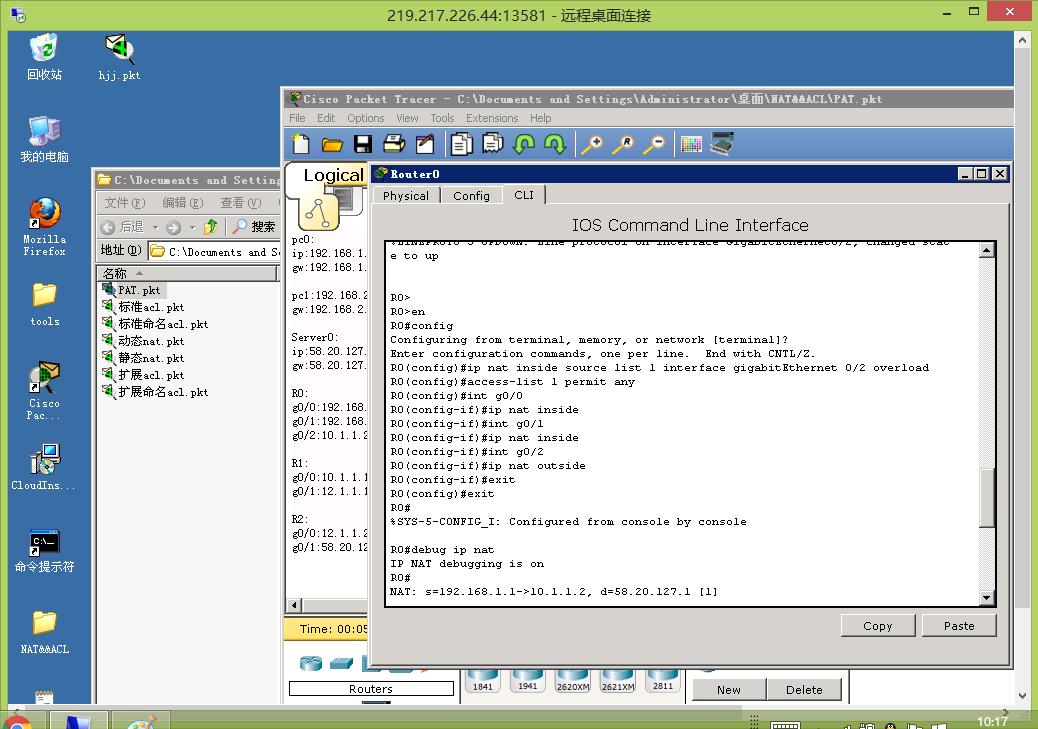
      步骤一：打开桌面NAT&&ACL文件夹中的“PAT.pkt”文件，加载初始化网络环境；

      步骤二：在R0路由器上配置PAT

\n\n\n

\n\n\n

|  |
| --- |
| //配置pat，以及访问控制  R0(config)#ipnat inside source list 1 interface gigabitEthernet 0/2 overload    // 访问控制列表1，允许全部访问  R0(config)#access-list 1 permit any    R0(config)#int g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside    R0(config-if)#int g0/2  R0(config-if)#ipnat outside |



      步骤三：测试。

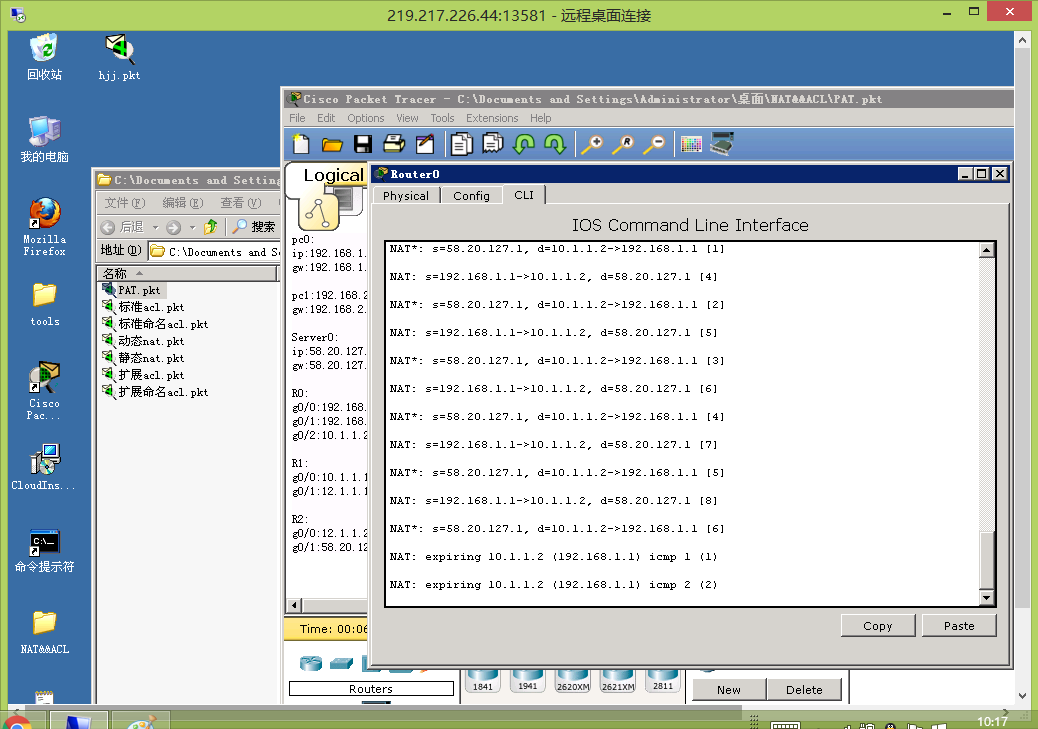
      1）在R0上，打开debug，查看地址翻译的过程

\n\n\n

\n\n\n

|  |
| --- |
| R0#debug ipnat |

      在pc0或者pc1上ping 58.20.127.1 (Server0的地址)，此时路由器将输出如下内容：



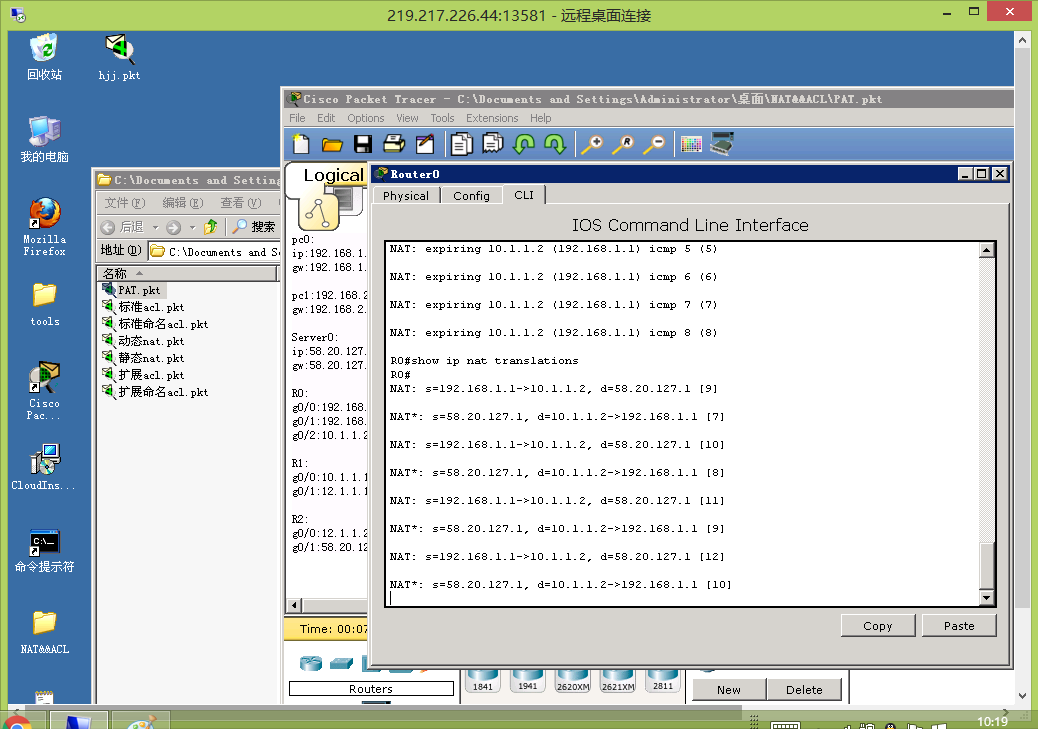
      2）在R0上查看NAT映射表

\n\n\n

\n\n\n

|  |
| --- |
| R0#show ipnat translations |

      可以看到静态时映射表一直存在，表明了内部全局地址和内部局部地址的对应关系。大体如下图：

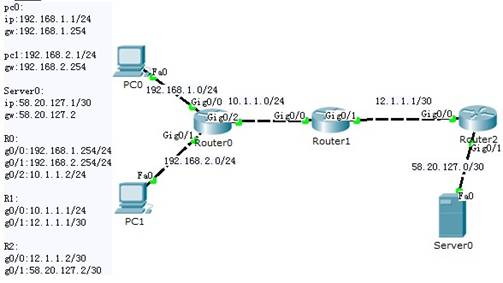


      上图可见，内部IP经PAT转换后，使用同一个IP的不同端口。

# 任务三：使用访问控制列表进行数据包过滤

      本任务，将使用ACL完成数据包过滤。

      实验拓扑图：



      说明：拓扑图与本实验的任务一相同。

      本任务包括三个子任务：

      子任务一：标准ACL；

      子任务二：扩展ACL；

      子任务三：命名ACL。

### ****准备工作：构建网络拓扑图****

      本实现各子任务前，先把网络拓扑图构建好，包括网络设备、接口及IP地址，参见前面的任务一：使用Packet Tracer构建网络拓扑图。

\n\n\n

\n\n\n

|  |
| --- |
| 注：只需把网络拓扑图、接口及IP地址做好，**不要配置路由**。 |

### ****子任务一：标准ACL****

      通过标准ACL实现如下要求：

      192.168.1.0/24网段不能访问Server0

      192.168.2.0/24网段可以访问Server0

      只允许Server0可以telnet R1

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成了网络拓扑构建（即准备工作）。 |

       实验步骤如下：

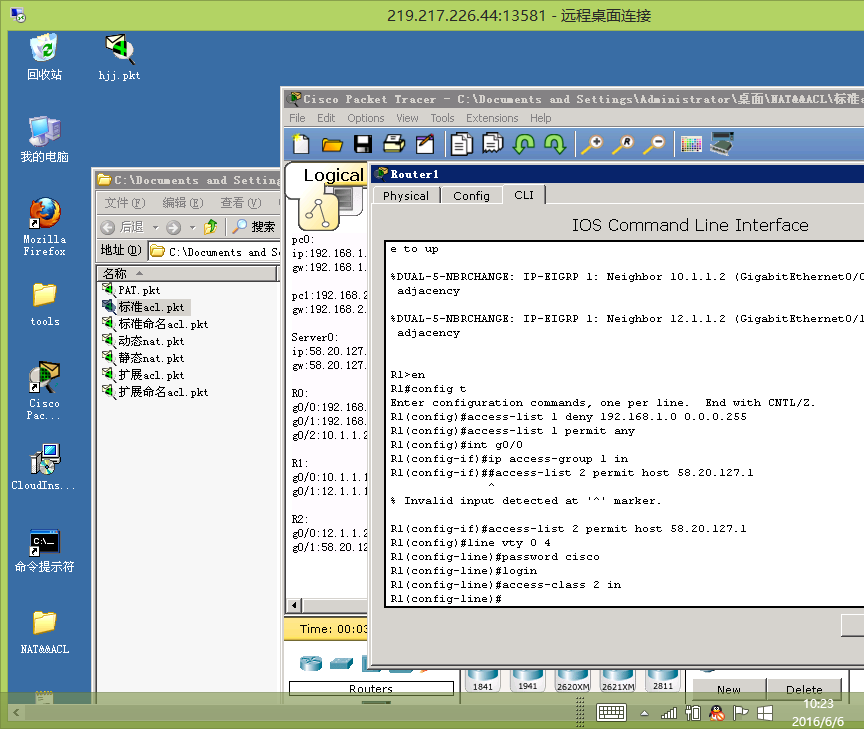
      步骤一：打开桌面NAT&&ACL文件夹中的“标准acl.pkt”文件，加载初始化网络环境；

      步骤二：配置ACL

\n\n\n

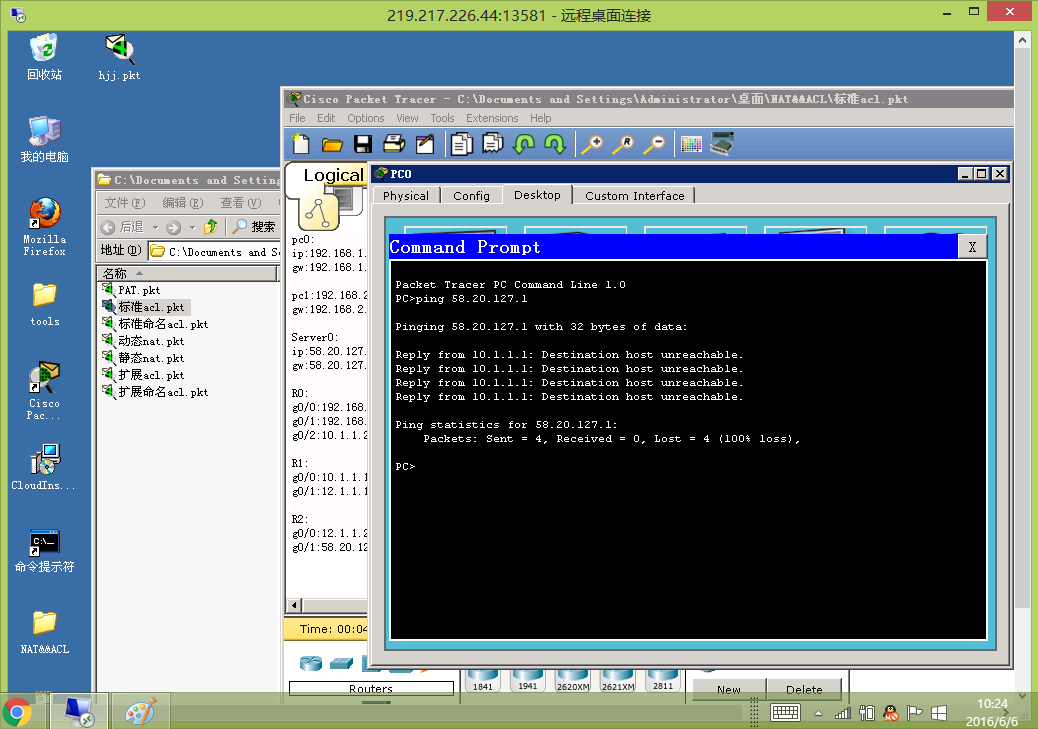
\n\n\n

|  |
| --- |
| //拒绝192.168.1.0/24网段的主机接入  R1(config)#access-list 1 deny 192.168.1.0 0.0.0.255  R1(config)#access-list 1 permit any    //将访问控制列表应用与接口  R1(config)#int g0/0  R1(config-if)#ip access-group 1 in    //允许主机58.20.127.1接入  R1(config)#access-list 2 permit host 58.20.127.1  R1(config)#line vty 0 4  R1(config-line)#password cisco  R1(config-line)#login  //应用访问控制列表2  R1(config-line)#access-class 2 in |

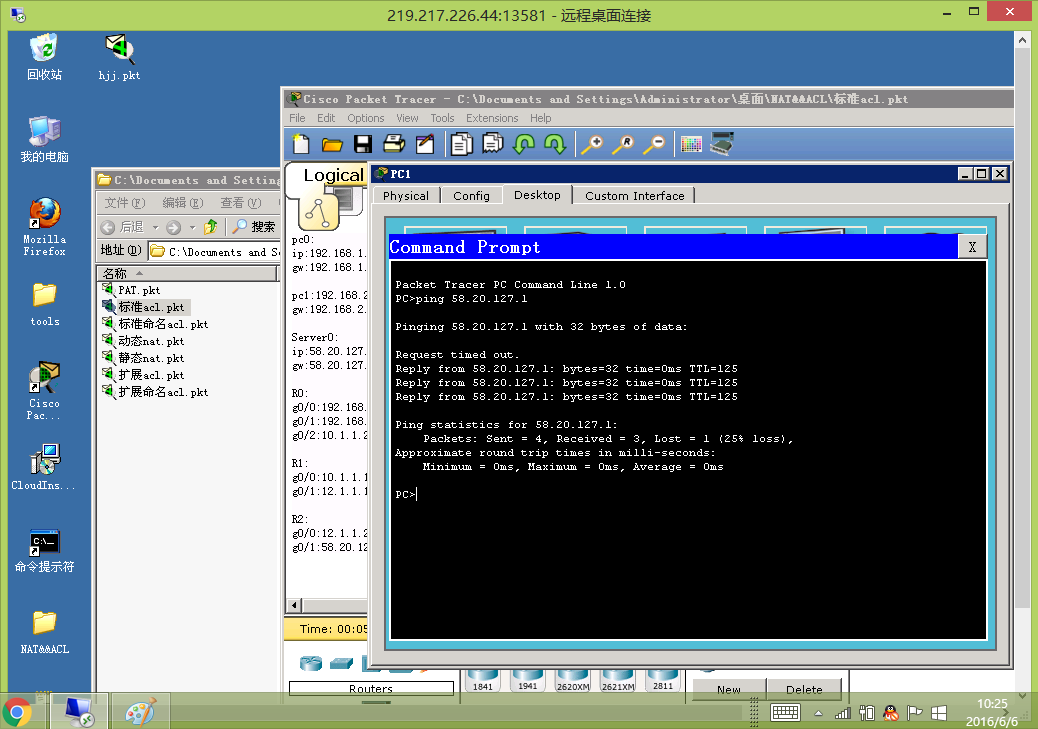


      步骤三：实验测试，达到如下效果即表明实验成功。

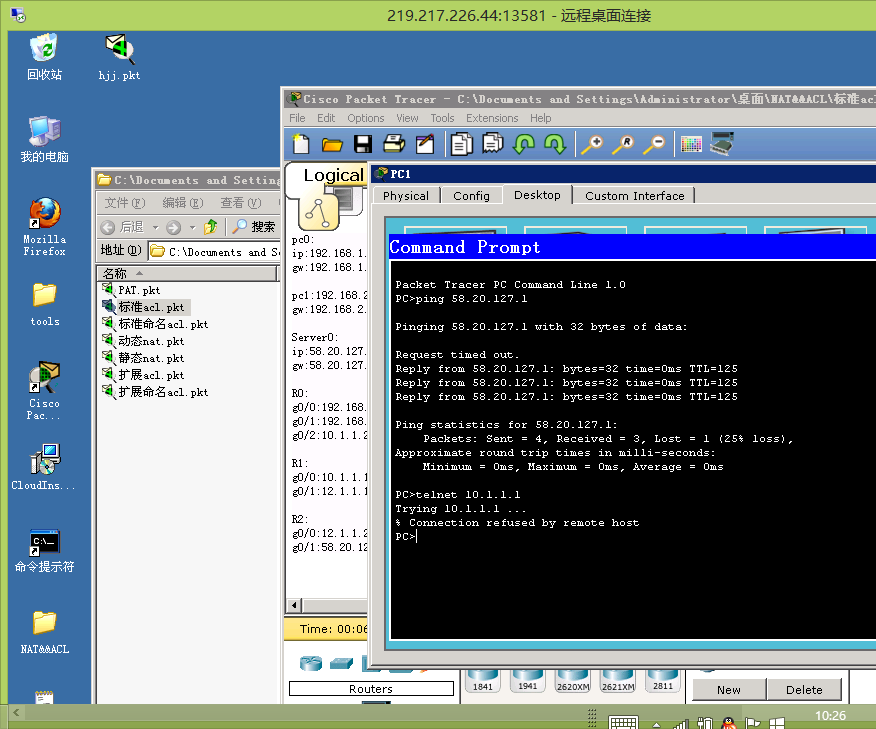
      1）在pc0上ping Server0(58.20.127.1)



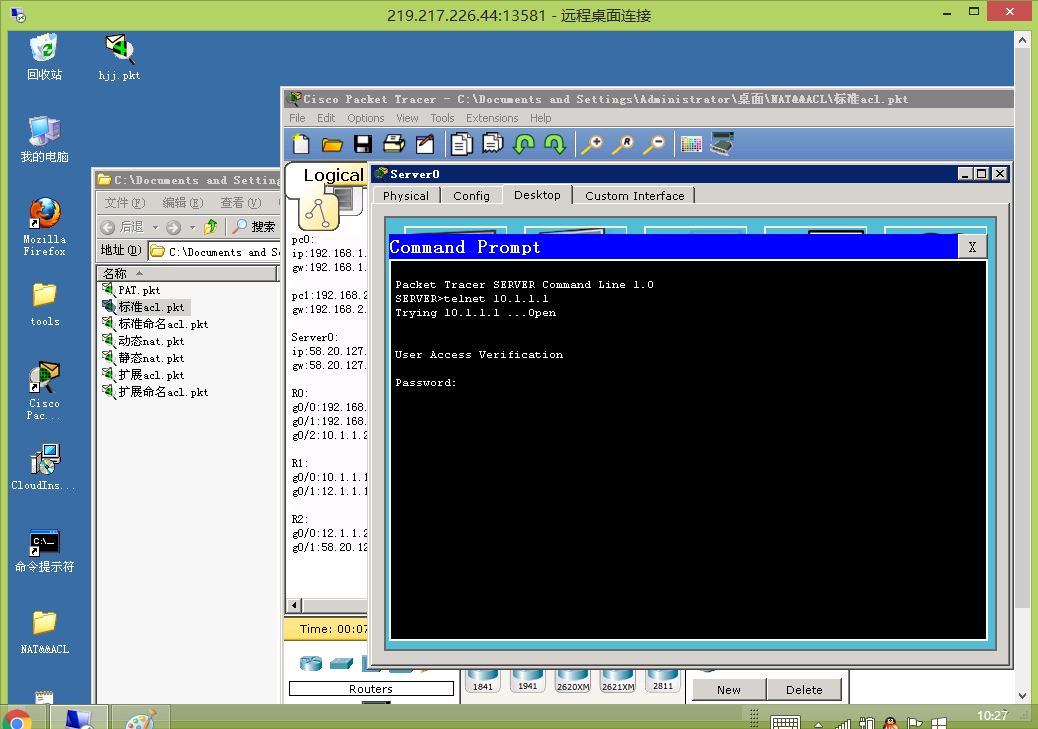
       2）在pc1上ping Server0(58.20.127.1)



       3）在pc1上telnet R1



       4）在Server0上telnet R1



### ****子任务二：扩展ACL****

      通过扩展ACL实现如下要求：

      实现192.168.1.0/24网段访问Server0的www和ftp；

      实现192.168.2.0/24网段访问Server0的ftp

      禁止ping Server0

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成了网络拓扑构建（即准备工作）。 |

      实验步骤如下：

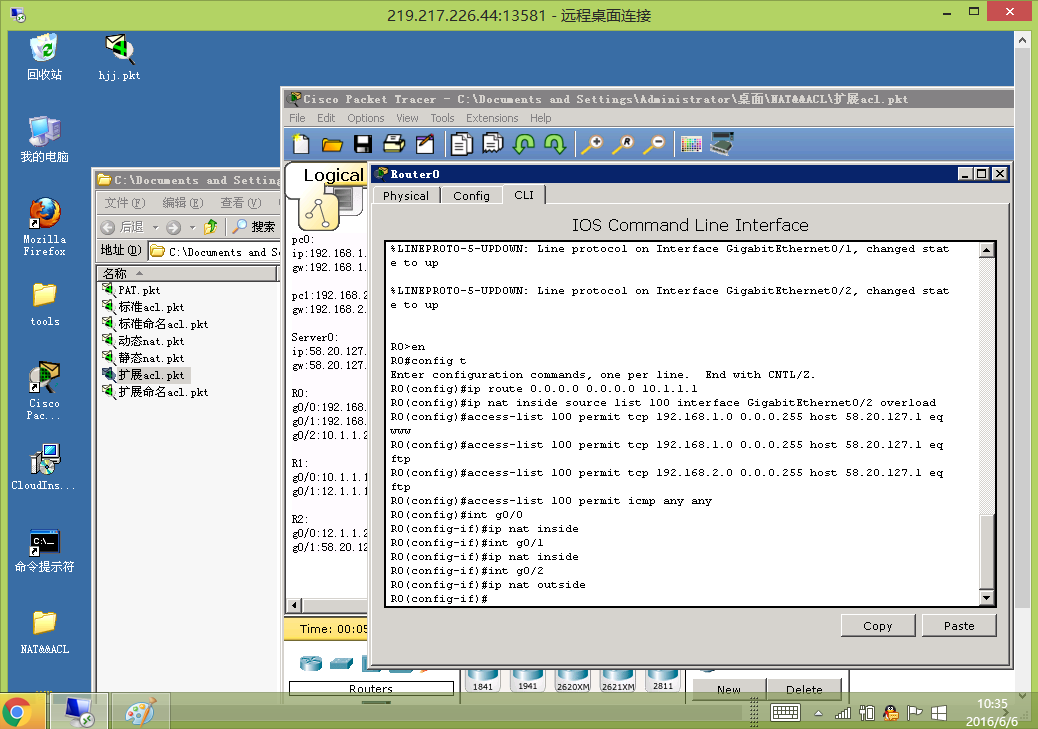
      步骤一：打开桌面NAT&&ACL文件夹中的“扩展acl.pkt”文件，加载初始化网络环境；

      步骤二：配置路由器R0

\n\n\n

\n\n\n

|  |
| --- |
| R0(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1  R0(config)#ipnat inside source list 100 interface GigabitEthernet0/2 overload  R0(config)#access-list 100 permit tcp 192.168.1.0 0.0.0.255 host 58.20.127.1 eq www  R0(config)#access-list 100 permit tcp 192.168.1.0 0.0.0.255 host 58.20.127.1 eq ftp  R0(config)#access-list 100 permit tcp 192.168.2.0 0.0.0.255 host 58.20.127.1 eq ftp  R0(config)#access-list 100 permit icmp any any  R0(config)#int g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside  R0(config-if)#int g0/2  R0(config-if)#ipnat outside |

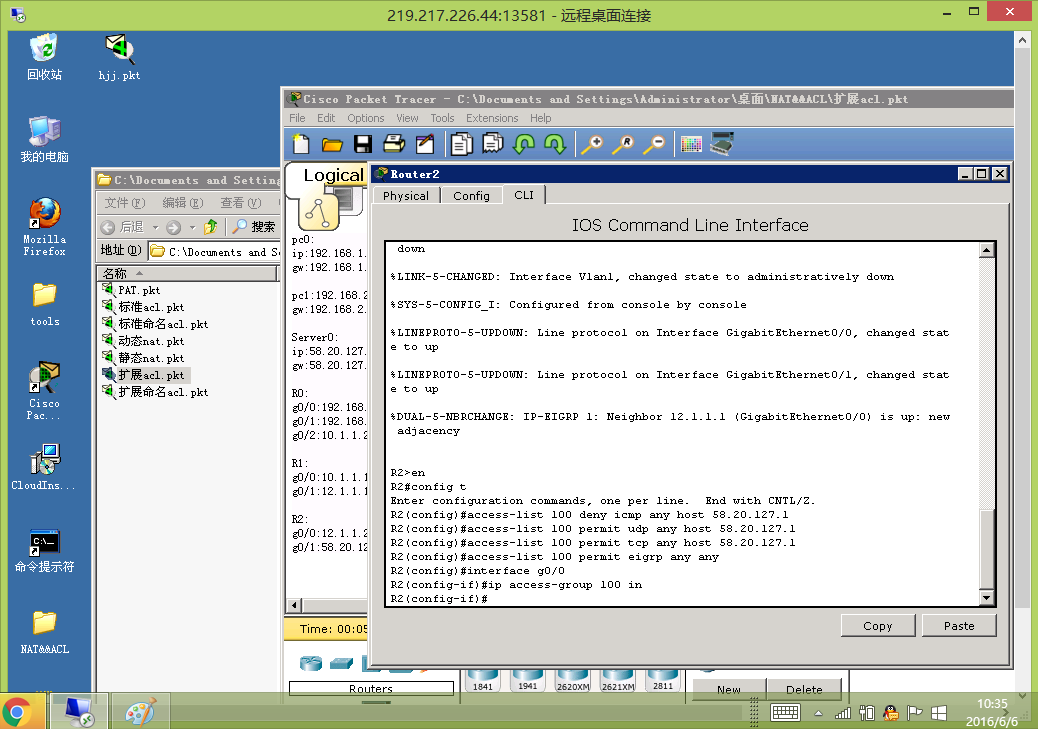


      步骤三：配置路由器R2

\n\n\n

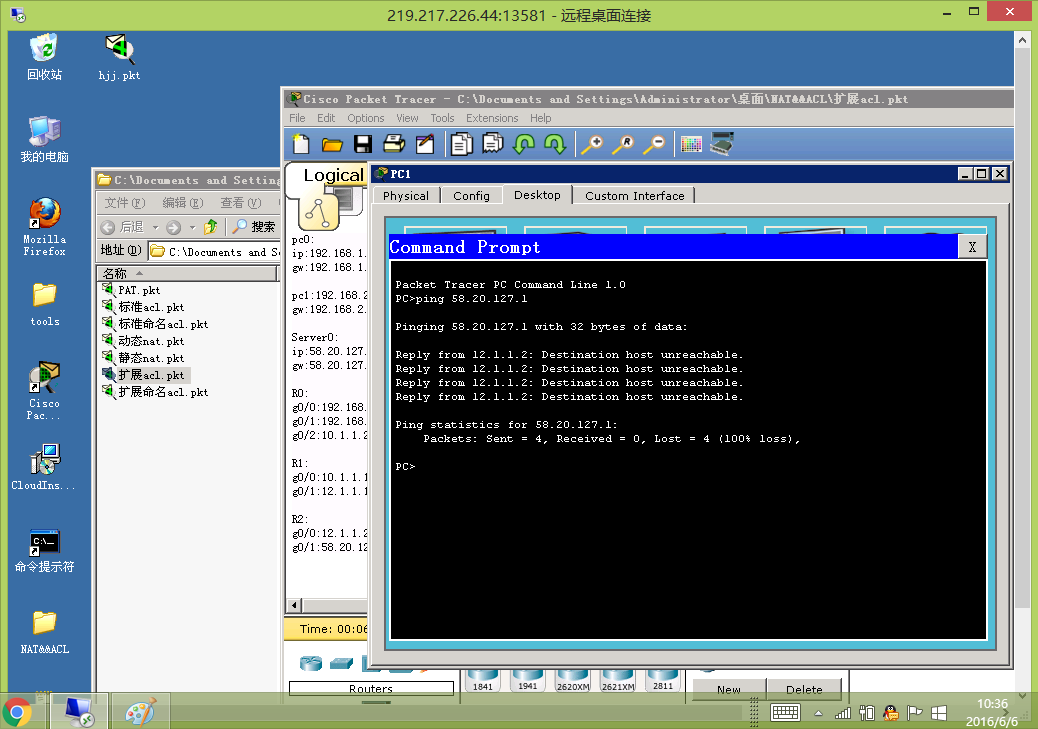
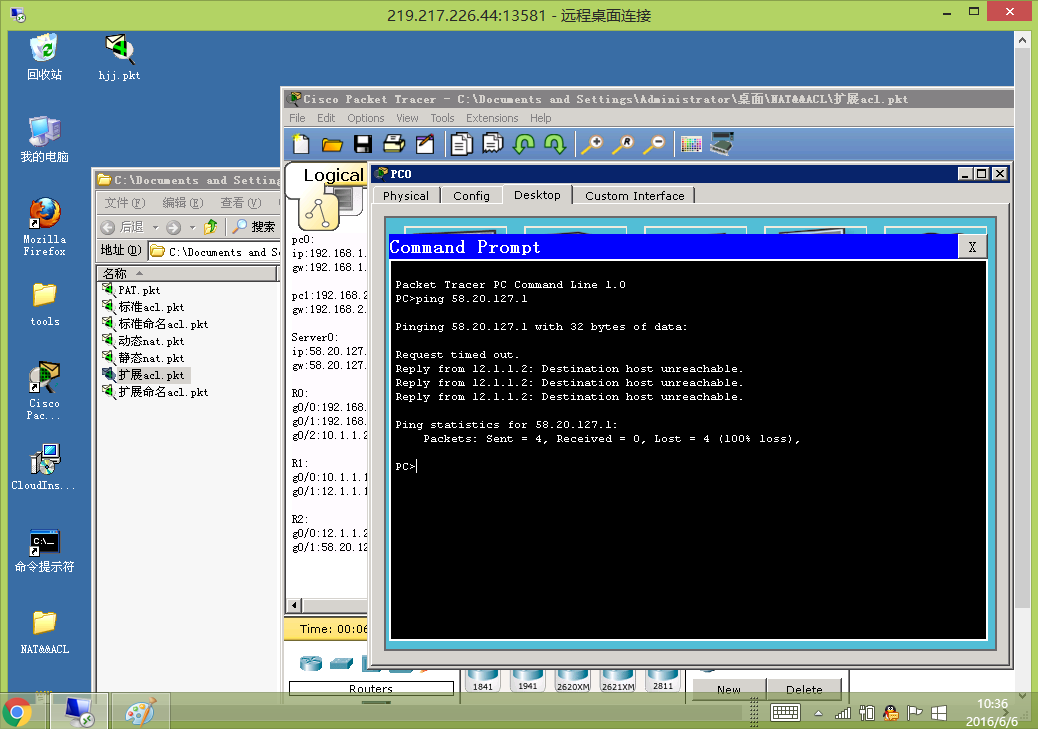
\n\n\n

|  |
| --- |
| R2(config)#access-list 100 deny icmp any host 58.20.127.1  R2(config)#access-list 100 permit udp any host 58.20.127.1  R2(config)#access-list 100 permit tcp any host 58.20.127.1  R2(config)#access-list 100 permit eigrp any any  R2(config)#interface g0/0  R2(config-if)#ip access-group 100 in |

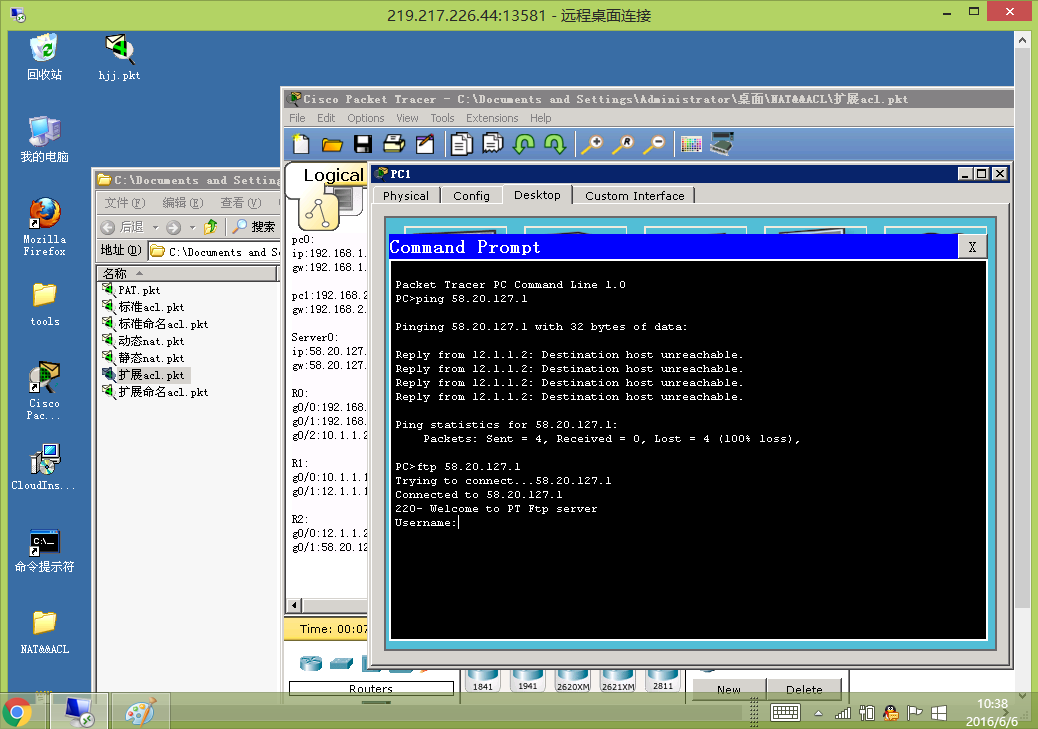
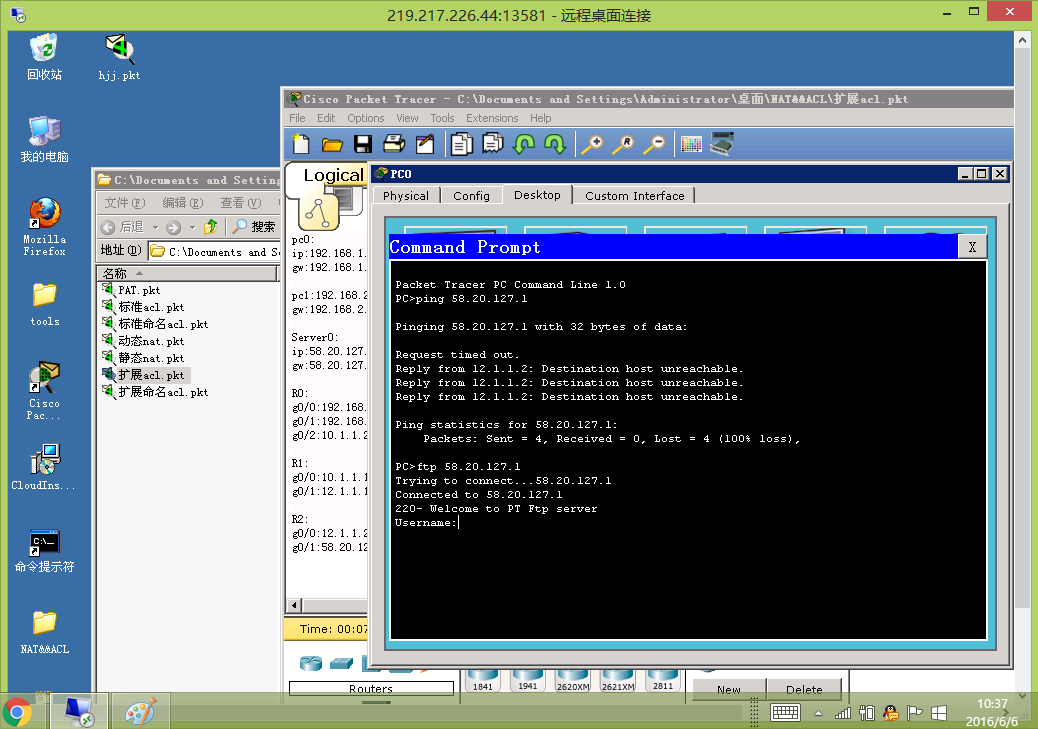


      步骤四：实验测试

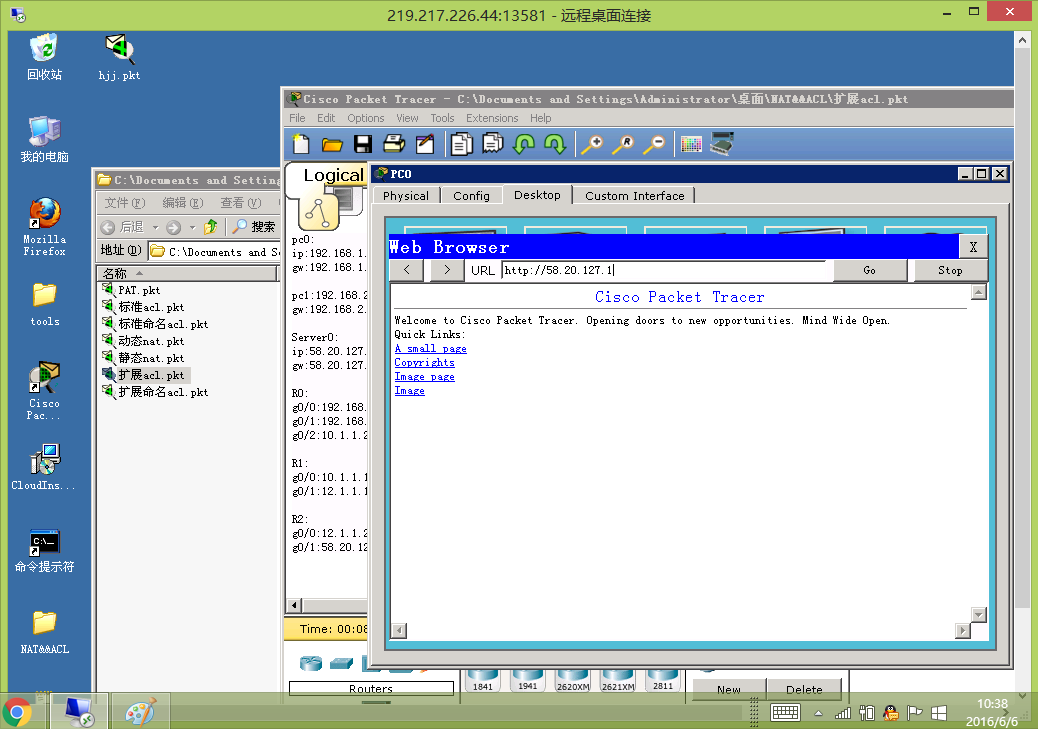
      1）在pc0和pc1上分别ping Server0(58.20.127.1)



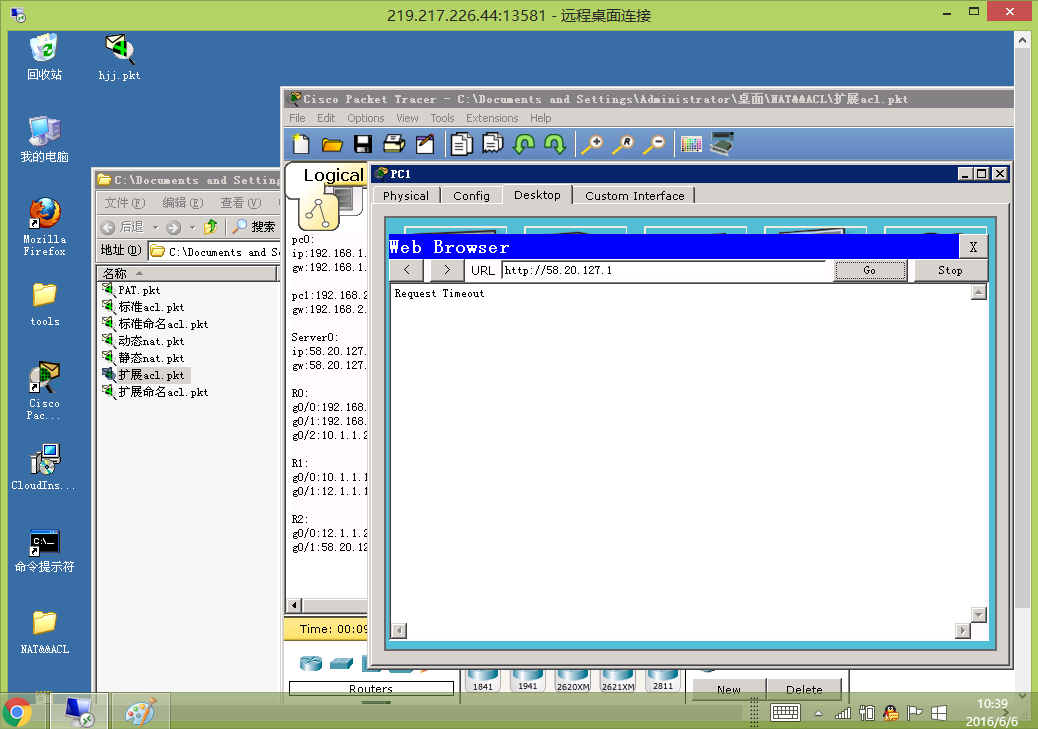
      2）在pc0和pc1上分别ftp登陆Server0



      3）在pc0上访问Server0的web服务，允许访问。



       4）在pc1上访问Server0 的web服务，不允许访问（请求超时）。



### ****子任务三：命名ACL****

      本子任务包括两个部分：标准命名ACL和扩展命名ACL。

**标准命名ACL**

      实现效果：

      实现192.168.1.0/24网段(pc0所在网段)访问互联网（Server0）。

      实现192.168.2.0/24网段（pc1所在网段）禁止访问互联网（Server0）。

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成了网络拓扑构建（即准备工作）。 |

      实验步骤：

      步骤一：打开桌面NAT&&ACL文件夹中的“标准命名acl.pkt”文件，加载初始化网络环境；

      步骤二：配置路由器R0

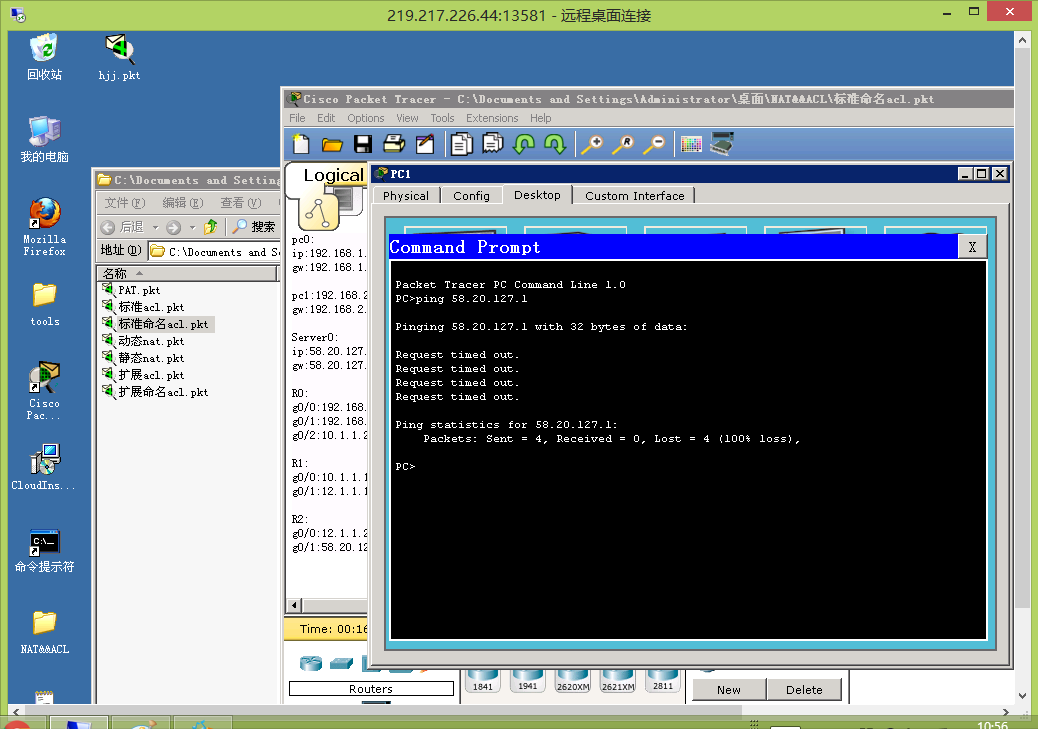
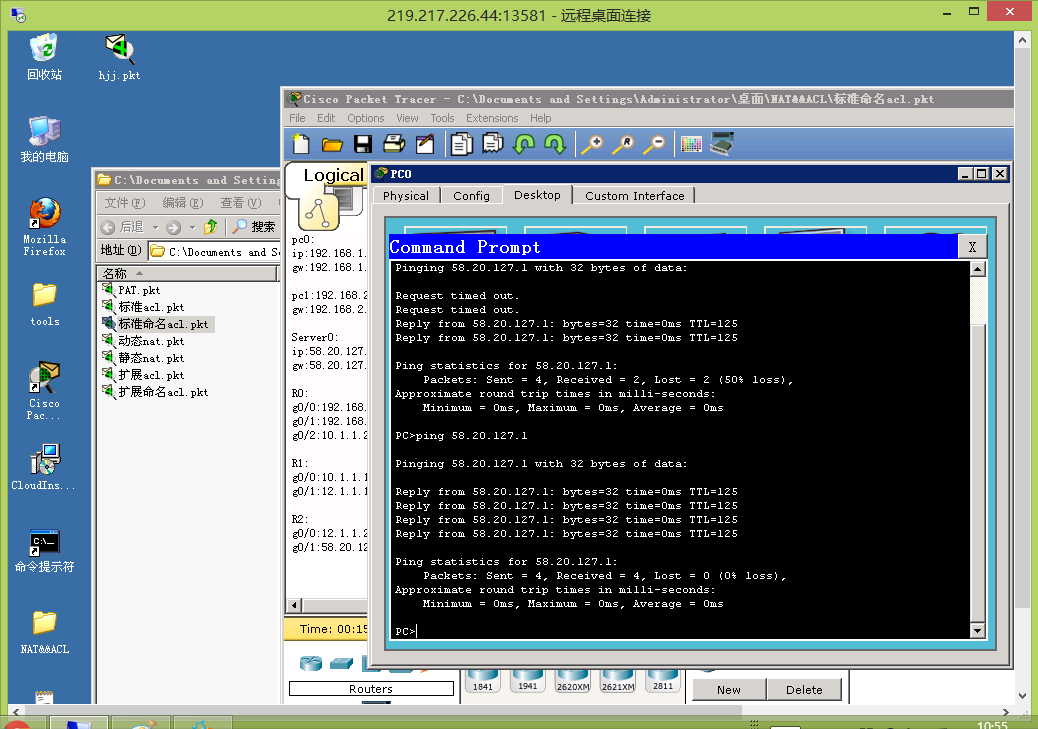
\n\n\n

\n\n\n

|  |
| --- |
| R0(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1  R0(config)#ipnat inside source list stand interface gigabitEthernet 0/2 overload  R0(config)#ip access-list standard stand  R0(config-std-nacl)#permit 192.168.1.0 0.0.0.255  R0(config-std-nacl)#deny 192.168.2.0 0.0.0.255  R0(config)#interface g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside  R0(config-if)#int g0/2  R0(config-if)#ipnat outside |

      步骤三：实验测试

      在pc0、pc1上ping Server0，测试连通性。



**扩展命名ACL**

      实现效果：

      实现192.168.1.0/24网段（pc0所在网段）访问互联网Server0的www；

      实现192.168.2.0/24网段（pc1所在网段）访问互联网Server0的ftp；

      禁止icmp数据包通过

\n\n\n

\n\n\n

|  |
| --- |
| 前提：完成了网络拓扑构建（即准备工作）。 |

      实验步骤：

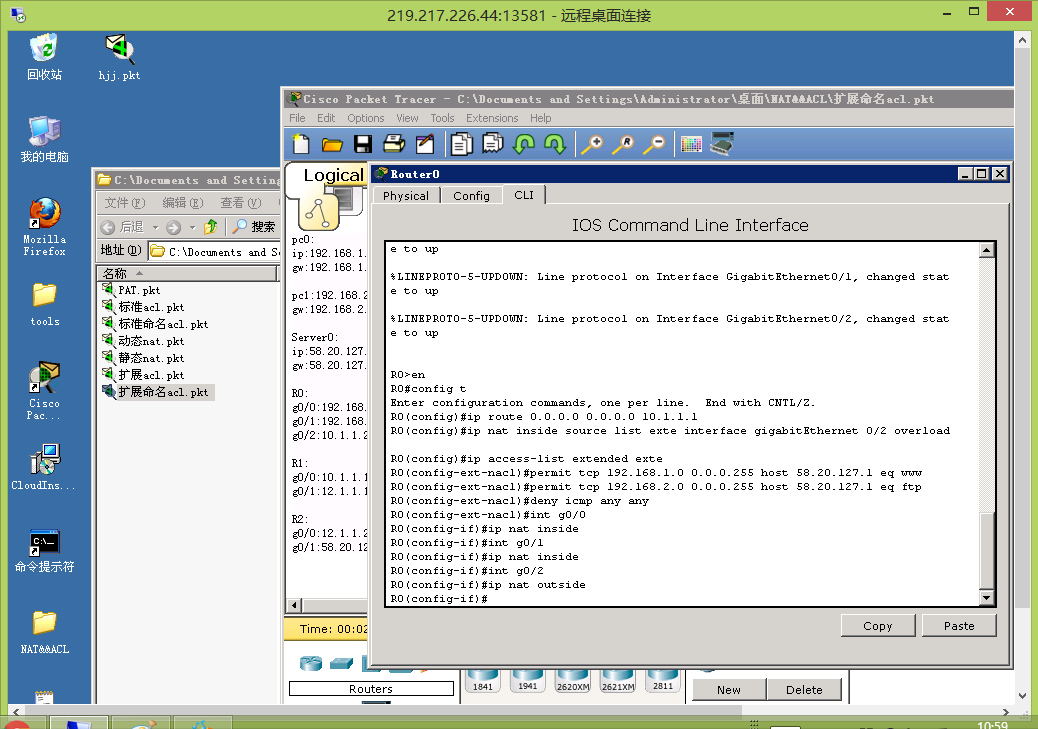
      步骤一：打开桌面NAT&&ACL文件夹中的“扩展命名acl.pkt”文件，加载初始化网络环境；

      步骤二：配置路由器R0

\n\n\n

\n\n\n

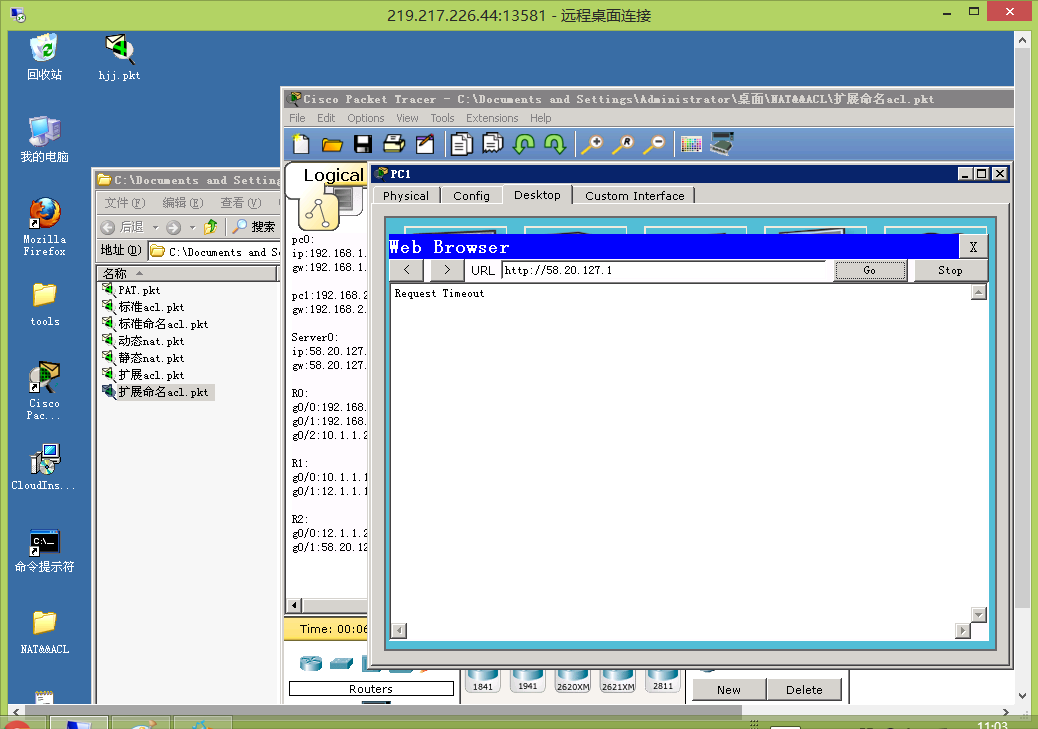
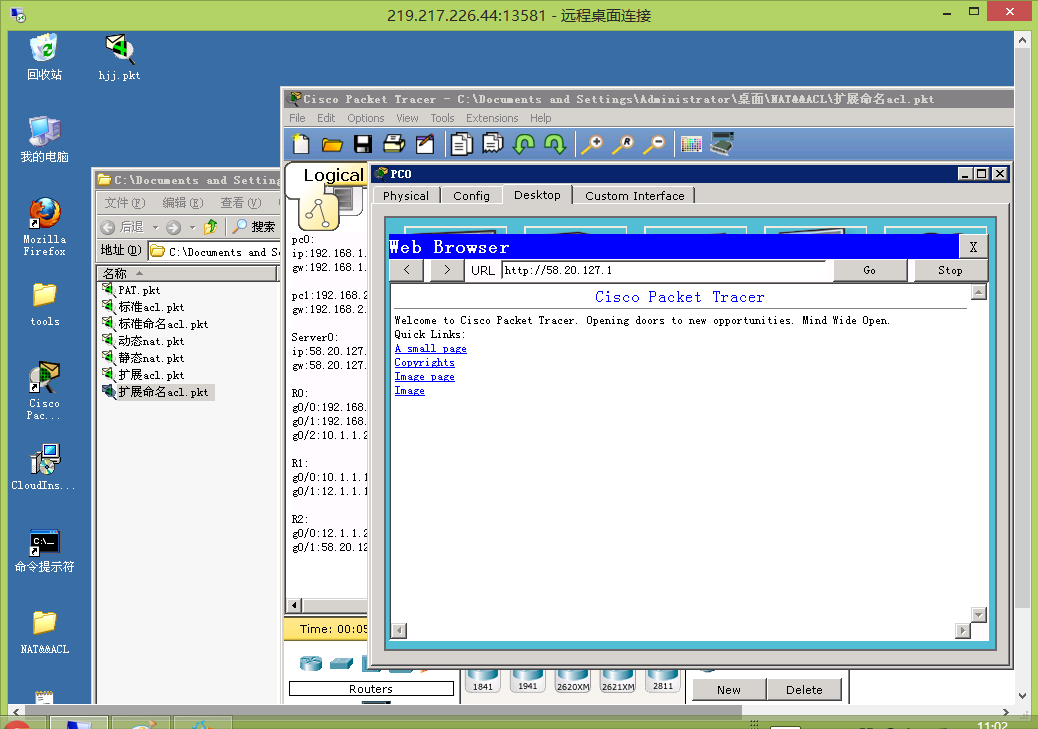
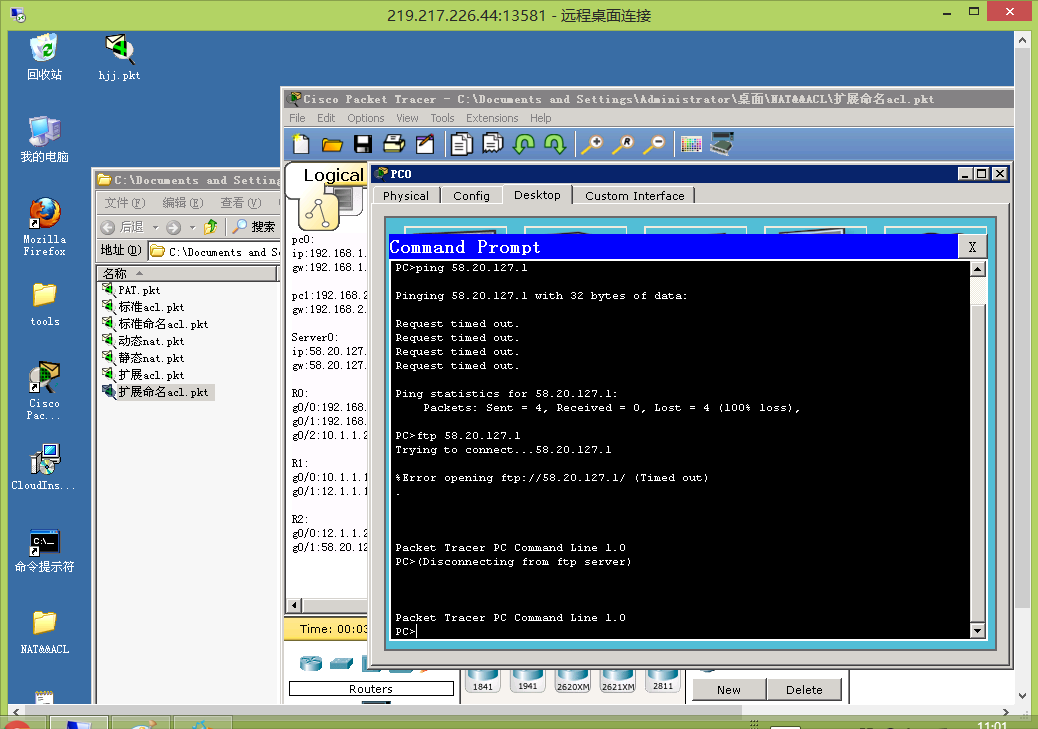
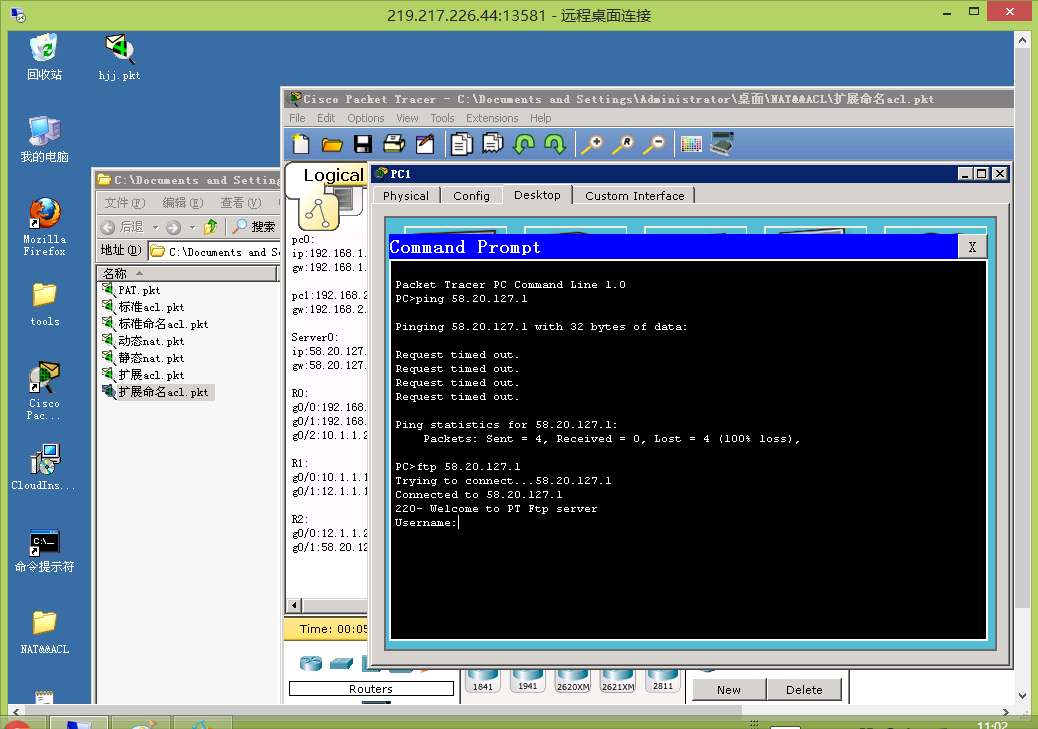
|  |
| --- |
| R0(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.1  R0(config)#ipnat inside source list exte interface gigabitEthernet 0/2 overload  R0(config)#ip access-list extended exte  R0(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 host 58.20.127.1 eq www  R0(config-ext-nacl)#permit tcp 192.168.2.0 0.0.0.255 host 58.20.127.1 eq ftp  R0(config-ext-nacl)#deny icmp any any  R0(config)#int g0/0  R0(config-if)#ipnat inside  R0(config-if)#int g0/1  R0(config-if)#ipnat inside  R0(config-if)#int g0/2  R0(config-if)#ipnat outside |



      步骤三：实验测试

      在pc0（192.168.1.1）和pc1(192.168.2.1)上测试分别测试，都不能ping通Server0，但pc1可以用ftp登陆，pc0不能。

      pc0可以访问web，pc1不能访问web。



1. 实验总结

分析思考题：

1. 静态NAT与动态NAT有什么区别？。

  静态NAT是将内部网络的私有IP地址转换为公有IP地址，IP地址对固定不变，某个私有IP地址只转换为某个公有IP地址。借助于静态转换，可以实现外部网络对内部网络中某些特定设备（如WWW、FTP）的访问。

动态转换是将内部网络的私有IP地址转换为公用IP地址时，IP地址对不确定，是随机的，所有被授权访问Internet的私有IP地址可随机转换为任何指定的合法IP地址。也就是说，只要指定哪些内部地址可以进行转换，以及用哪些合法地址作为外部地址时，就可以进行动态转换。动态转换可以使用多个合法外部地址集。当合法IP地址略少于网络内部的计算机数量时。可以采用动态转换的方式。

通过上面的实验我们也能看出ip地址的固定（静态）与随机（动态）。

静态的NAT 只能是固定的映射 不能动态更新   
  
动态NAT 可以根据你写的规则 进行自动的映射

1. 某单位内网有一内部服务器，现因工作需要让外网访问，需要做静态NAT还是动态NAT，为什么？

静态NAT，因为这个单位只有一个内部服务器，其ip地址很方便的一一对应一个公网ip，由外向内访问时很便捷。

1. 标准ACL与扩展ACL区别是什么？

标准 ACL 最简单，是通过使用 IP 包中的源 IP 地址进行过滤，表号范围 1-99 或 1300-1999；

扩展 ACL 比标准 ACL 具有更多的匹配项，功能更加强大和细化，可以针对包括协议类型、源地址、目的地址、源端口、目的端口、TCP 连接建立等进行过滤，表号范围 100-199 或 2000-2699；

4）某WEB应用服务器被攻击，查看IDS信息后发现有一IP（202.102.3.103）可疑，为防止再次攻击，需要禁止该IP访问该服务器，使用标准ACL与扩展ACL需要如何做？

标准ALC是对此源ip地址进行过滤，deny掉此ip地址发送的数据包，使其不能到达web应用服务器。

扩展ALC指出了过滤的源地址和目的地址及其他已知的信息，比标准ALC更加具体，防护更加到位详细。

本次实验，我们通过实践，进一步对各种NAT有了更进一步的了解，也通过亲自动手，明白了静态，动态，端口NAT之间的差别。除此之外，我们还学习了各种ALC，标准ALC，扩展ALC，命名标准ALC，命名扩展ALC，学习了他们的差别和各自的优缺点。通过这次实验，我们还复习了关于ip的知识，重拾了掩码等基本网络概念。

除此之外，我们还学习了如何上手使用思科Packet tracer，跟着实验做，并且在实验过程中查阅一些资料，为以后的软件使用和学习奠定了良好的基础。