11 网络安全

随着网络技术的不断发展和网络服务的不断丰富，网络安全成为我们越来越关注的话题。网络服务和资源经常面临不同的网络攻击的威胁，所以网络安全的部署和实施也是网络发展的必然要求。

作为一个系统管理员，保护敏感重要的数据和网络资源，防止可能的恶意入侵，是最优先考虑的事情。网络安全的范畴很广泛，包括物理层安全、数据链路层安全、网络层安全、传输层安全以及应用层安全，本章的重点在于网络层安全。

本章将介绍网络安全的基本知识，重点介绍 ACL 访问控制列表。

本章主要内容：

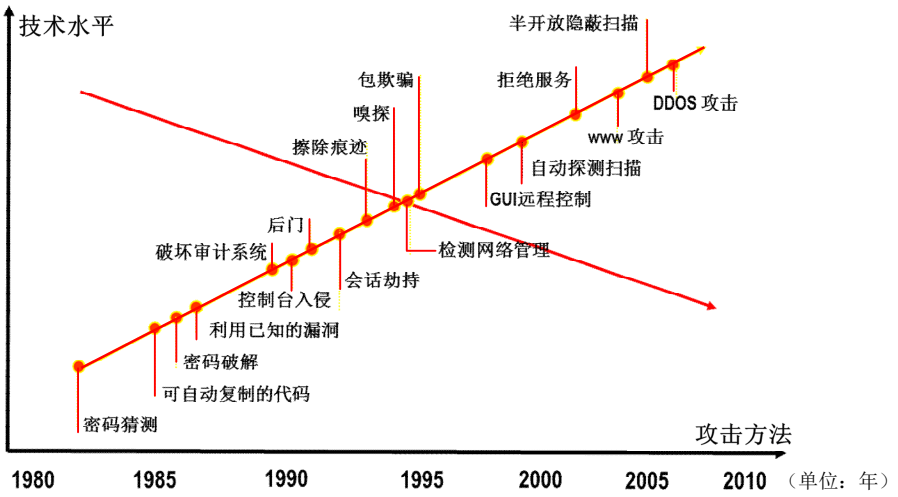
- 网络安全概述
- 安全网络架构
- Cisco IOS 安全特性
- ACL 访问控制列表
- 网络地址转换 NAT

11.1 网络安全简介

在过去的短短数年内，计算机网络不仅规模显著增长，其重要性也与日俱增。如果网络安全受到危害，可能会导致非常严重的后果，如隐私丧失、信息失窃，有的甚至需要追究法律责任。随着网络威胁的种类日渐增多，安全环境所面临的挑战也日趋严峻。

11.1.1 网络中的安全隐患

随着时间的推移，网络中出现的攻击方式越来越多，但需要的技术水平却越来越低，现在有各式各样的网络攻击软件，不需要自己编译，直接使用即可，所以现在网络存在的安全隐患更多。在前些年，攻击者必须具备高深的计算机、编程和网络知识才能利用基本的工具进行简单的攻击，如图 11-1 所示。现在随着攻击者的方法和工具的不断改进，他们不再需要高深的知识即可进行攻击。



▲图 11-1　网络攻击方法及入侵技术的发展

说到网络攻击，我们通常认为这种攻击行为是黑客实施的，那么到底黑客指的是哪类人呢？事实上目前人们都有一种误解，将黑客直接和电脑犯罪者联系在一起，这种看法是错误的。而 hacker 的原意是用斧头做家具的能工巧匠，麻省理工的学生们重新赋予这个极具挑战性的古老职业新的含义：能够面对挑战、创造技术、解决问题的人。电脑犯罪者更应该被称为黑帽子或骇客。这个分类是非常不容易讲清楚的一个问题，因为没有统一的标准，分类方式差异非常大，按照我们的习惯可以简单分为以下 3 种类型：

- 黑客（Hacker）：所做的不是恶意破坏，他们是一群纵横于网络上的技术人员，热衷于科技探索和计算机科学研究。
- 骇客（Cracker）：闯入计算机系统或软件，从事恶意破解商业软件、恶意入侵别人的网站等事务的一类人。他们未必具有很高的技术，通常用一些简单的攻击手段去达到炫耀、恶作剧、搞破坏的目的。
- 红客（Honcker）：中国特色的产物，最早出现于北约轰炸中国驻南联盟大使馆之后。他们用自己的技术维护国内网络安全，并对外来的一切进攻进行还击。他们有爱国、正义、进取的精神，是网络安全时代英雄的代表。

11.1.2　常见的网络攻击方式

网络攻击就是对网络隐患的具体利用，一般来讲网络攻击有以下 3 种方式：

- 被动攻击：对想窃取的信息进行侦听，以获取机密信息。而数据的拥有者或合法用户对此类活动无法得知，所以被动攻击主要关注防范，而非检测。目前针对此类攻击行为，一般都是采用加密技术来保护信息的机密性。
- 主动攻击：对业务数据流报文首部或数据载荷部分进行假冒或篡改，以达到冒充合法用户对业务资源的非授权访问或对业务资源进行破坏的目的。对于此类攻击可以通过对数据流进行分析检测以给出技术解决措施，最终保障业务的正常运行。
- 中间人攻击：一种“间接”类型的攻击方式，根据攻击者对信息不同的攻击行为（信息窃取攻击、信息篡改攻击），将会有被动攻击和主动攻击的特征。通过中间人攻击能达到

以下两个目的：

- 信息窃取：当主机 A 和主机 B 进行数据交互时，攻击者对信息进行截取备份一份，并进行数据转发（可能只是进行侦听，不对其进行转发）。这样攻击者很容易获取主机 A 和主机 B 的机密信息，而主机 A 和主机 B 对其一无所知。
- 信息篡改：攻击者作为主机 A 和主机 B 数据交互的中介，可能主机 A 和主机 B 以为它们之间是直接通信的，其实它们之间的通信有个中转器——攻击者。此类攻击，攻击者一般会往主机 A 和主机 B 之间的数据流中插入或更改相应信息，以达到其攻击的目标。

11.1.3 网络安全概述

网络安全的定义是通过采用各种管理措施和技术手段，使网络系统的硬件、软件及其系统中的数据受到保护，不因偶然或恶意的原因而遭到破坏、更改、泄露，使系统连续、可靠、正常地运行，网络服务不中断。

网络安全的概念很长，它主要包括 4 个方面的要点：

（1）网络安全实施的方法：需要管理和技术结合起来，不能单纯地依赖于技术，管理网络时的责任心对网络安全同样重要。

（2）网络安全保护的对象：网络系统中的硬件资源、软件资源及数据信息，尤其是数据信息被破坏后很难恢复。

（3）网络受到威胁的类型：可能是恶意的也可能是偶然的，多数情况下是用户在下载数据时感染木马或病毒引起的，所以计算机上网时一定要及时修补系统的漏洞，并且要安装杀毒软件。

（4）网络安全实施的目的：是为了保护系统正常的运行服务不中断，也就是说要保证网络资源的可用性。

11.1.4 网络安全的实施目标

我们知道了网络安全的概念，那么把网络建设到什么程度就是安全的呢？一般来讲我们可以将网络安全的实施目标归纳为以下几个方面：

- 系统的可靠性：保证网络系统不因各种因素的影响而中断正常工作。
- 系统的可控性：能够对各种资源访问和信息传播进行控制和日志记录。
- 数据的完整性：保护网络系统中存储和传输的软件（程序）与数据不被非法改变。
- 数据的可用性：保证软件（程序）和数据能被合法用户访问和正常利用。
- 数据的保密性：利用密码技术对数据进行加密处理，保证在系统中存储和网络上传输的数据不被无关人员识别。

需要说明的是，网络安全是一个相对的概念，运行的服务越多，访问的数据量越大，面临网络攻击的可能性就越大，并且整体的网络由多个设备互联而成，每个设备工作时都可能面临不同类型的网络威胁，所以网络安全的实施需要整体规划、综合部署。后面的章节将会介绍一些常见的技术手段用于保护企业网络的安全。

11.1.5 从 OSI 参考模型来看网络安全

在工作中可能会听到这样的词“物理层安全”“数据链路层安全”“网络层安全”“应用层安

全”，这些都是根据 OSI 参考模型的分层来说的。OSI 参考模型将数据通信分为 7 层：应用层、表示层、会话层、传输层、网络层、数据链路层和物理层。网络安全也可以从这个角度来分类。下面针对 OSI 参考模型的层列举一些安全的例子。

1. 物理层安全

通过网络设备进行攻击：Hub 和无线 AP 进行攻击。攻击者将计算机连接到使用 Hub 组建的网络中就可以捕获其他用户通信的数据包。无线 AP 如果没有安全措施，攻击者可以捕获无线 AP 通信。例如，某公司的办公大楼其中一层租给了保险公司，这一层的办公室的网线还在该公司的交换机上连接，并且没有禁用这些端口，保险公司就可以将计算机轻易接入到该公司的网络，这就是物理层不安全。

物理层安全措施：使用交换机替代 Hub，为无线 AP 配置密码实现无线设备的接入保护和实现数据加密通信。

2. 数据链路层安全

数据链路层攻击：恶意获取数据或 MAC 地址，由于大多数 IDS 和操作系统对网络层以下的防御很弱，因此很危险。攻击方式有 ARP 欺骗、ARP 广播、同一网段有重复的 MAC 地址。

数据链路层安全措施：在交换机的端口上控制连接计算机的数量或绑定 MAC 地址，在交换机上划分 VLAN 都属于数据链路层安全。在计算机和路由器上添加 IP 地址和 MAC 地址绑定可以防止 ARP 欺骗。ADSL 拨号上网的账号和密码实现的是数据链路层安全。

3. 网络层安全

网络层攻击：IP Spoofing（IP 欺骗）、Fragmentation Attack（碎片攻击）、Reassembly Attack（重组攻击）、Ping of Death（ping 死攻击）。

网络层安全措施：在路由器上设置访问控制列表和 IPSec，在 Windows 上实现 Windows 防火墙和 IPSec，这些都属于网络层安全。

4. 传输层安全

传输层攻击：Port Scan（端口扫描）、TCP Reset Attack（TCP 重置攻击）、SYN DoS Flood（SYN 拒绝服务攻击）、LAND Attack（LAND 攻击）、Session Hijacking（会话劫持）。

传输层安全措施：使用基于网络的入侵监测系统 IPS。

5. 应用层安全

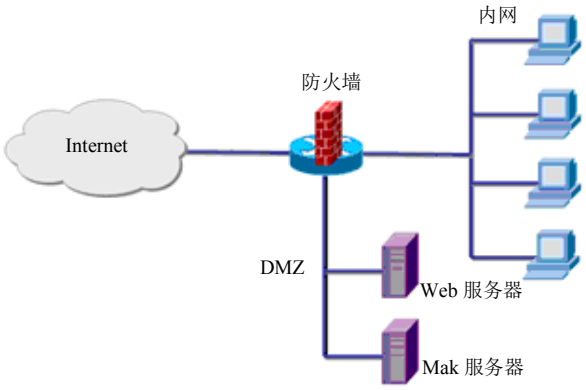
应用层攻击：MS-SQL Slammer Worm 缓冲区溢出、IIS 红色警报、E-mail 蠕虫、蠕虫、病毒、木马、垃圾邮件、IE 漏洞。

应用层安全措施：安装杀毒软件，更新操作系统。

11.1.6 典型的安全网络架构

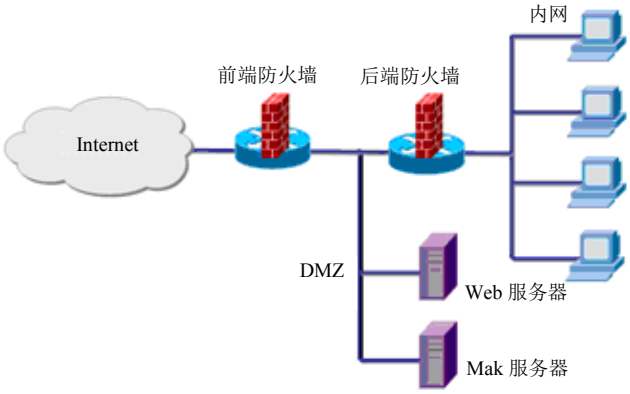
许多大中型企业网络中，各种各样的安全策略都是基于内网、非军事区（DMZ）路由器以及防火墙设备。防火墙通过屏蔽各部分的网络流量来提供附加的安全保障，而进行这些工作需要使用访问控制列表。

典型的网络架构如图 11-2 所示的三向外围网，防火墙设备连接 Internet、内网和 DMZ 区。DMZ 区部署了公司对外的 Web 和 Mail 服务器，一般是公网 IP 地址。内网是私网 IP 地址，一般不对 Internet 用户提供服务，但是需要访问 Internet。如果入侵者突破了该防火墙，就威胁到了 DMZ 和内网的安全。



▲图 11-2　三向外围网

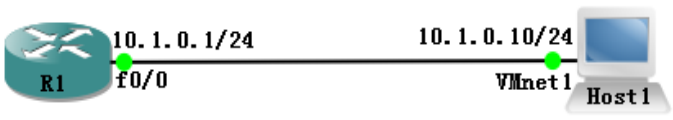
另一种典型的网络架构是背靠背防火墙，如图 11-3 所示，两个防火墙之间是 DMZ 区，内网在后端防火墙后面。建议这两个防火墙最好不是同一家公司的产品，如前端使用 Cisco 公司的 PIX 防火墙，后端使用微软的软件防火墙 ISA 2006。这样入侵者要想入侵内网，就需要突破两个不同厂商的防火墙，增加了难度。



▲图 11-3　背靠背防火墙

11.2　保护 Cisco IOS 路由器的安全

因为路由器是通往其他网络的网关，所以它们是明显的攻击目标，容易遭到各种各样的攻击。攻击者可以使用不同的方法破坏路由器，因此网络管理员无法仅靠单一的方法对抗攻击者。本节重点介绍如何保护路由器的安全，所用的拓扑环境如图 11-4 所示。



▲图 11-4　本节使用的拓扑环境

11.2.1 路由器的密码安全

确保路由器基本安全的方法是配置口令密钥，强口令密钥是控制安全访问路由器最基本的要素，因此应该始终配置强口令密钥，强口令密钥的相关设置包括以下几项内容：

- 由大写字母、小写字母、阿拉伯数字、特殊字符 4 部分组成。
- 设置尽可能长的口令，最佳做法是设置至少 8 个字符的口令。
- 请勿将口令记在明显的地方，如办公桌或显示器上。
- 避免使用字典中可以查到的单词、姓名、电话号码和日期，避免口令遭到字典攻击。
- 故意将口令中的词拼错，如 Smith 可以拼成 Smyth。
- 尽可能经常更改口令，降低破解口令的可能性，同时避免口令被破解后信息暴露。

注意，口令开头的空格会被忽略，但第一个字符之后的空格都不会被忽略。

默认情况下，当在路由器中输入口令时，Cisco IOS 软件会以明文的形式保存口令，而这样很不安全，因为当查看运行配置时，使用 enable password 命令或 username *username* password *password* 命令即会以明文的方式显示出这些口令，如下：

```
R1(config)#username user password cisco
R1(config)#end
R1#show run | include username
username user password 0 cisco      ---运行配置中显示的 0 表示口令没有被隐藏---
```

为了安全配置文件，所有口令都应该加密，Cisco IOS 提供了两种保护口令的方法：

- 7 类方案的简单加密：它使用 Cisco 定义的简单加密算法隐藏口令，很容易破解。
- 5 类方案的复杂加密：它使用安全的 MD5 哈希算法加密口令，理论上不能破解。

注意：7 类加密可以用于 enable password、username 和 line password 命令（包括 vty、线路控制台和辅助端口），此方法提供的防护较为有限，虽然不像 5 类加密那样安全，但比不加密强。

要为口令使用 7 类加密，请使用全局配置命令 service password-encryption，该命令使配置文件中的口令难以辨认，过程如下：

```
R1(config)#service password-encryption
R1(config)#end
R1#show run | include username
username user password 7 121A0C041104   ---7 表示口令隐藏，同样的方式可用于 line---
```

Cisco 推荐尽可能使用 5 类加密代替 7 类加密，要使用此加密方法，将关键字 password 替换为 secret 即可，如将上面 user 的 7 类加密方式删除并使用 5 类加密方式重建该信息，操作过程如下：

```
R1(config)#no username user
R1(config)#username user secret cisco
R1(config)#^Z
R1#show run | include username
username user secret 5 $1$wrQY$qi6chlccr1y4U96S5sBcG0
```

同样应该使用 enable secret 命令设置特权口令提高特权执行级别的安全性，而不要使用 enable password，因为它的安全性较低，操作过程如下：

```
R1(config)#enable secret cisco123
```

```
R1(config)#^Z
R1#show run | include enable
enable secret 5 $1$7RNE$Br.EfO8bVkG6sFJeUOo5d.
```

注意：某些过程可能无法使用 5 类加密口令，如 PAP 和 CHAP 要求使用明文口令，这些过程不能使用 MD5 加密口令。

11.2.2 路由器的安全访问管理

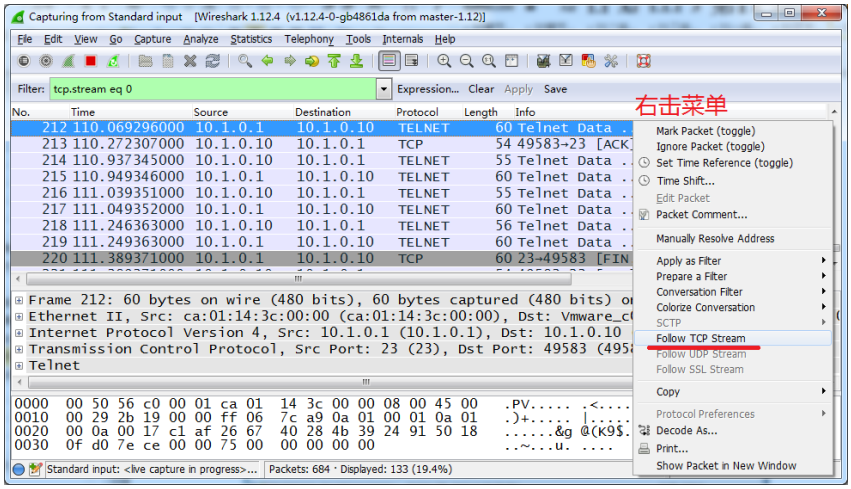
随着网络规模的扩大和网络设备数量的增加，通过本地的终端接口管理设备的工作量将变得非常大，对于需要管理许多设备的管理员来说，远程管理访问比本地访问更加方便。但是，我们平时使用 Telnet 执行远程管理访问的方式很不安全，因为 Telnet 以明文的方式发送所有网络流量，攻击者可以在管理员远程登录到路由器时捕获网络流量，并嗅探到管理员口令或路由器配置信息。我们设置 R1 使用本地数据库对远程用户进行认证，操作过程如下：

```
R1(config)#line vty 0 4
R1(config-line)#login local
R1(config-line)#end
```

在 Host1 上使用 Telnet 对 R1 进行远程管理并开启抓包软件，过程如下：

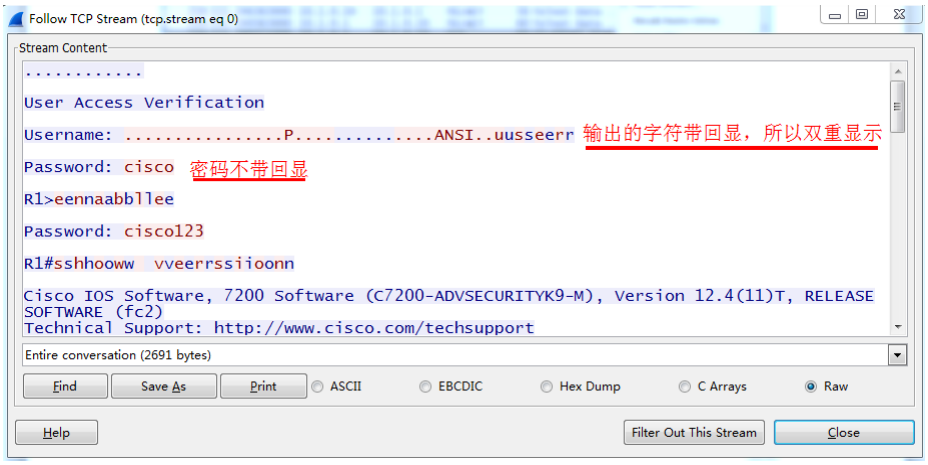
```
C:\Users\Administrator>telnet 10.1.0.1
User Access Verification
Username: user
Password:
R1>enable
Password:
R1#show version
Cisco IOS Software, 7200 Software (C7200-ADVSECURITYK9-M), Version 12.4(11)T, RE
---输出部分省略---
Configuration register is 0x2102
R1#quit
```

Telnet 过程的抓包结果如图 11-5 所示。



▲图 11-5 Telnet 的数据包

右击抓获的 Telnet 数据包并选择 Follow TCP Stream 选项，即可显示 Telnet 明文的数据包内容，如图 11-6 所示。

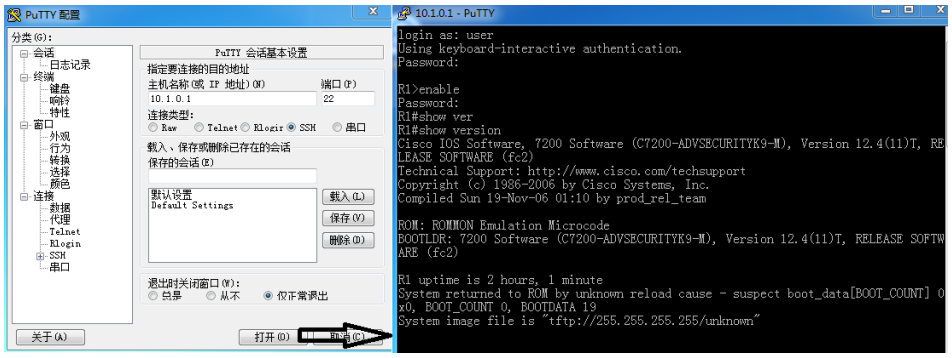


▲图 11-6　Telnet 的明文内容

SSH 为 Secure Shell 的缩写，由 IETF 的网络工作小组（Network Working Group）所制定。SSH 为建立在应用层和传输层基础上的安全协议，已经取代 Telnet 成为了执行远程路由器管理的最佳做法。要让路由器支持 SSH 的方式进行管理，需要进行如下设置：

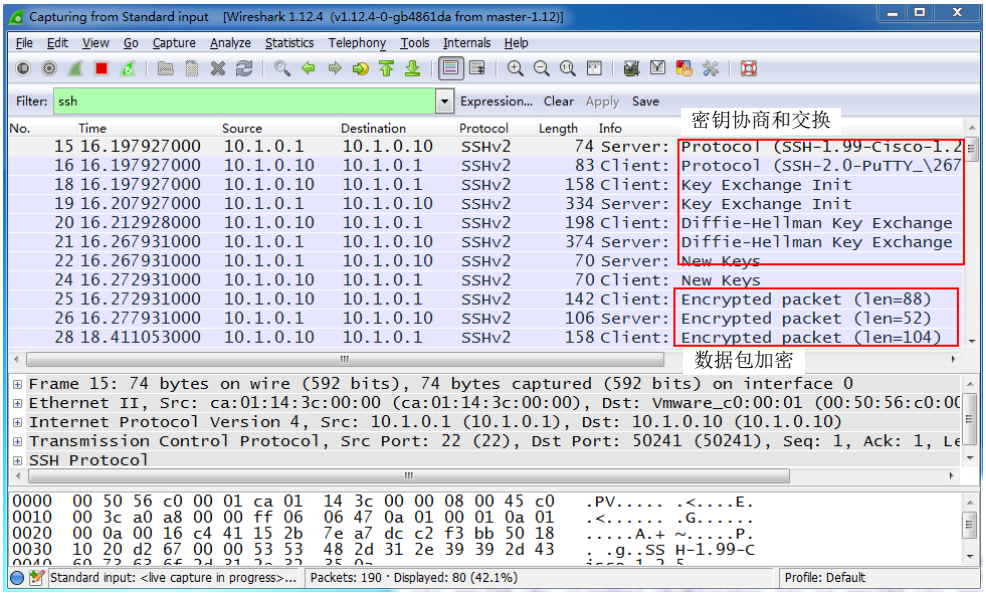
```
R1(config)#ip domain-name study.net          ---设置域名，路由器产生自签名密钥时使用---
R1(config)#crypto key generate rsa           ---设置路由器产生自签名密钥---
The name for the keys will be: R1.study.net
Choose the size of the key modulus in the range of 360 to 2048 for your
  General Purpose Keys. Choosing a key modulus greater than 512 may take
  a few minutes.
How many bits in the modulus [512]:
% Generating 512 bit RSA keys, keys will be non-exportable...[OK]
*Apr 23 18:29:23.935: %SSH-5-ENABLED: SSH 1.99 has been enabled
R1(config)#line vty 0 4
R1(config-line)#transport input ssh          ---设置路由器仅支持 SSH 访问---
R1(config-line)#login local
R1(config-line)#end
```

使用 PuTTY 软件通过 SSH 访问路由器并启用抓包软件，操作过程如图 11-7 所示。



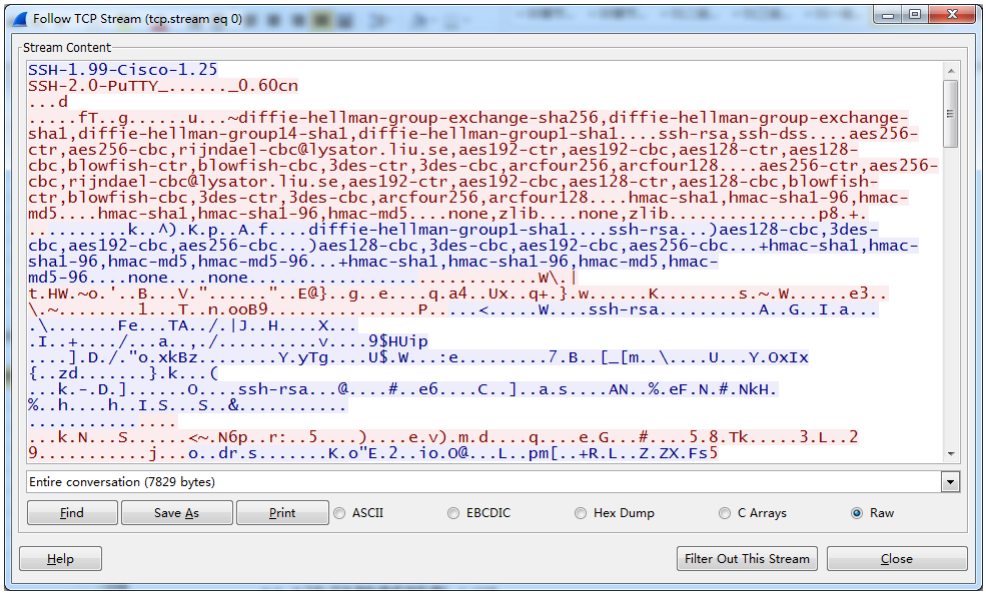
▲图 11-7　使用 SSH 管理路由器

SSH 过程的抓包结果如图 11-8 所示。



▲图 11-8　SSH 的数据包

右击抓获的 SSH 数据包并选择 Follow TCP Stream 选项，即可显示 SSH 数据包的内容，我们可以看到这些内容已经进行了加密处理，如图 11-9 所示。



▲图 11-9　SSH 加密的数据内容

11.2.3　路由器的安全加固

Cisco 路由器默认运行着大量的网络服务，有些服务通常是不必要的，它们不仅耗用了系统资

源，还具有潜在的安全风险，表 11-1 介绍了通常易受攻击的路由器服务，并列出了与这些服务相关的最佳做法。

▲表 11-1　易受攻击的路由器服务

功能	描述	默认	建议
Cisco 发现协议（CDP）	运行在 Cisco 设备之间的第二层专有协议	启用	CDP 很少用到，将其禁用
TCP 小型服务器	标准 TCP 网络服务 echo、chargen 等	依设备版本而定	这是一项较旧的功能，将其明确禁用
UDP 小型服务器	标准 UDP 网络服务 echo、discard 等	依设备版本而定	这是一项较旧的功能，将其明确禁用
Finger	UNIX 用户查找服务，允许远程列出用户列表	启用	未授权用户不需要知道此信息，将其禁用
HTTP 服务器	某些 Cisco IOS 设备允许通过 Web 进行配置	依设备版本而定	若未使用，则明确禁用此功能；否则需限制访问权
BOOTP 服务器	允许其他路由器从此设备启动的一项服务	启用	此功能很少使用，而且可能带来安全隐患，将其禁用
IP 源路由	一项 IP 功能，允许数据包指明自己的路由	启用	此功能很少使用，而且容易被攻击者利用，将其禁用
代理 ARP	路由器会作为第二层地址解析的代理	启用	除非路由器用作 LAN 网桥，否则禁用此服务
IP 定向广播	数据包可以识别广播的目标 LAN	依设备版本而定	定向广播可能被用于攻击，将其禁用
IP 重定向	对于所路由的某些 IP 数据包，路由器会发出 ICMP 重定向消息	启用	可能被用于网络映射，在不受信任的接口上禁用此功能
简单网络管理协议	路由器支持 SNMP 远程查询和配置	启用	若未使用，则明确禁用此功能；否则需限制访问权

网络管理员不需要知道如何利用这些服务进行攻击，但是他们必须知道如何禁用这些服务。如果在路由器上分别禁用这些服务是非常繁琐的工作，管理员就需要一种自动的方法来加快设备安全加固的过程。

Cisco 的 AutoSecure 特性可以帮助管理员精确控制各种服务的启用和禁用功能，即使管理员不了解 Cisco IOS 软件的所有功能也可以配置安全策略，提高 Cisco IOS 网络的安全性。使用 AutoSecure 时可以采用以下两种运作模式：

- 交互模式（Interactive Mode）：在禁用或启用服务以及设置其他安全相关特性时将提示用户，与用户交互。
- 非交互模式（Noninteractive Mode）：使用推荐的设置自动执行 auto secure 命令。

使用交互（interactive）选项时的主要步骤依次是：确定外网（outside）接口、管理平面的安全设置、创建安全 banner、配置密码、AAA 和 SSH、接口的安全设置、转发平面的安全设置，操作过程如下：

```
R1#auto secure
                    ---AutoSecure Configuration---
*** AutoSecure configuration enhances the security of
```

```
the router, but it will not make it absolutely resistant
to all security attacks ***

AutoSecure will modify the configuration of your device.
All configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
Autosecure documentation.
At any prompt you may enter '?' for help.
Use ctrl-c to abort this session at any prompt.

Gathering information about the router for AutoSecure

Is this router connected to internet? [no]:
```

使用非交互模式时，在特权模式中执行如下命令：

```
R1#auto secure no-interact
                    ---AutoSecure Configuration---

*** AutoSecure configuration enhances the security of
the router, but it will not make it absolutely resistant
to all security attacks ***

AutoSecure will modify the configuration of your device.
All configuration changes will be shown. For a detailed
explanation of how the configuration changes enhance security
and any possible side effects, please refer to Cisco.com for
Autosecure documentation.

Securing Management plane services...

Disabling service finger
Disabling service pad
Disabling udp & tcp small servers
Enabling service password encryption
Enabling service tcp-keepalives-in
Enabling service tcp-keepalives-out
Disabling the cdp protocol

Disabling the bootp server
Disabling the http server
Disabling the finger service
Disabling source routing
Disabling gratuitous arp

Configuring interface specific AutoSecure services
Disabling the following ip services on all interfaces:

 no ip redirects
 no ip proxy-arp
 no ip unreachables
```

```
 no ip directed-broadcast
 no ip mask-reply
Disabling mop on Ethernet interfaces

Securing Forwarding plane services...

Enabling CEF (This might impact the memory requirements for your platform)

This is the configuration generated:

no service finger
no service pad
no service udp-small-servers
no service tcp-small-servers
service password-encryption
service tcp-keepalives-in
service tcp-keepalives-out
no cdp run
no ip bootp server
no ip http server
no ip finger
no ip source-route
no ip gratuitous-arps
no ip identd
security passwords min-length 6
security authentication failure rate 10 log
service timestamps debug datetime msec localtime show-timezone
service timestamps log datetime msec localtime show-timezone
logging facility local2
logging trap debugging
service sequence-numbers
logging console critical
logging buffered
interface FastEthernet0/0
 no ip redirects
 no ip proxy-arp
 no ip unreachables
 no ip directed-broadcast
 no ip mask-reply
 no mop enabled
ip cef
!
end

Applying the config generated to running-config

R1#
```

我们以前介绍过使用 SDM 软件管理路由器，SDM 中提供的安全审核（Security Audit）和一步式锁定（One-Step Lockdown）也可以完成路由器的自动安全加固。

11.3 访问控制列表 ACL

作为网络管理员，所需掌握的最重要的技能之一便是访问控制列表（ACL）。管理员使用 ACL 来阻止流量，或者仅允许特定流量的同时阻止网络中的所有其他流量。本节将介绍如何在安全解决方案中使用标准 ACL 和扩展 ACL，并提供在 Cisco 路由器上配置 ACL 的方法，其中包括 ACL 的使用技巧、注意事项、建议和一般指导原则。

11.3.1 防火墙的分类

网络设计师使用防火墙来防止网络被未授权用户使用。防火墙是强制执行网络安全策略的硬件或软件解决方案。可以想象建筑物内一间房间的门锁，该锁仅允许拥有钥匙或门卡的授权用户进入。类似地，防火墙过滤未经授权或可能存在危险的数据包，防止其进入网络。下面是常见的几种防火墙类型。

1. 数据包过滤

数据包过滤（Packet Filtering）技术是在网络层对数据包进行选择，选择的依据是系统内设置的过滤逻辑，称为访问控制列表（Access Control List，ACL）。通过检查数据流中每个数据包的源地址、目的地址、所用的端口号、协议状态等因素或它们的组合来确定是否允许该数据包通过。数据包过滤防火墙逻辑简单、价格便宜、易于安装和使用、网络性能和透明性好，它通常安装在路由器上。路由器是内部网络与 Internet 连接必不可少的设备，因此在原有网络上增加这样的防火墙几乎不需要任何额外的费用。

数据包过滤又称为网络级别的防火墙，网络级别的防火墙很快，在今天仍然可以在许多网络设施上找到它们的身影，特别是在路由器上，但是不能基于数据包的内容过滤数据。

2. 应用级网关

应用级网关（Application Level Gateways）是在网络应用层上建立协议过滤和转发功能的。它针对特定的网络应用服务协议使用指定的数据过滤逻辑，并在过滤的同时对数据包进行必要的分析、登记和统计，形成报告。实际中的应用网关通常安装在专用工作站的系统上。

数据包过滤和应用网关防火墙有一个共同的特点，就是它们仅仅依靠特定的逻辑判定是否允许数据包通过。一旦满足逻辑，防火墙内外的计算机系统则建立直接联系，防火墙外部的用户便有可能直接了解防火墙内部的网络结构和运行状态。

3. 代理服务

代理服务（Proxy Service）也称为链路级网关（Circuit Level Gateways）或 TCP 通道（TCP Tunnels），也有人将它归为应用级网关。它是针对数据包过滤和应用网关技术存在的缺点而引入的防火墙技术，其特点是将所有跨越防火墙的网络通信链路分为两段。防火墙内外计算机系统间应用层的“链接”由两个终止代理服务器上的“链接”来实现，外部计算机的网络链路只能到达代理服务器，从而起到隔离防火墙内外计算机系统的作用。此外，代理服务也对过往的数据包进行分析、注册登记，形成报告，同时当发现被攻击的迹象时会向网络管理员发出警报，并保留攻击痕迹。国内代理服务器软件有 CCProxy，微软的代理服务器软件有 ISA2006。

4. 状态化防火墙

普通的数据包过滤防火墙功能简单，只检测数据包的头部字段，不检测负载，没有应用识别的

能力；应用级网关或代理服务具有应用识别的能力，但需要对负载部分做深度的数据包检测，这样会影响防火墙的转发性能。状态检测防火墙虽然继承了数据包过滤防火墙和应用网关防火墙的优点，它在网络层有一个检查引擎截获数据包并抽取出与应用层状态有关的信息，并以此为依据决定对该连接是接受还是拒绝。这种技术提供了高度安全的解决方案，同时具有较好的适应性和扩展性。

5. 下一代防火墙 NG Firewall

下一代防火墙是在状态化防火墙的基础上提出的可以全面应对应用层威胁的高性能防火墙。通过深入洞察网络流量中的用户、应用和内容，并借助全新的高性能单路径异构并行处理引擎，NGFW 能够为用户提供有效的应用层一体化安全防护，帮助用户安全地开展业务并简化用户的网络安全架构。

不管是什么类型的防火墙，基本的安全策略需要通过 ACL 来实现，ACL 是实施网络安全最基本的工具。

11.3.2 访问控制列表 ACL 简介

ACL 是一种路由器配置脚本，它是由一系列 permit 或 deny 语句组成的顺序列表，执行时从上到下逐条行进行比对，通过比较数据包报头中各字段与设定的条件是否匹配来控制路由器应该允许还是拒绝数据包通过，也就是说 ACL 每一条语句的工作过程都是匹配了相应的条件去执行对应的允许或拒绝动作，如果数据包与所有的语句都不匹配，最后一条隐含的语句适用于不满足之前任何条件的所有数据包并默认发出“拒绝”指令。

创建访问控制列表相当于编写一系列 if-then 语句，如果满足给定的条件，就采取给定的措施；如果不满足，则不采取任何措施，而继续比对下一条语句。访问控制列表语句相当于分组过滤器，根据它对分组进行比对分类，执行相应的允许或拒绝动作。

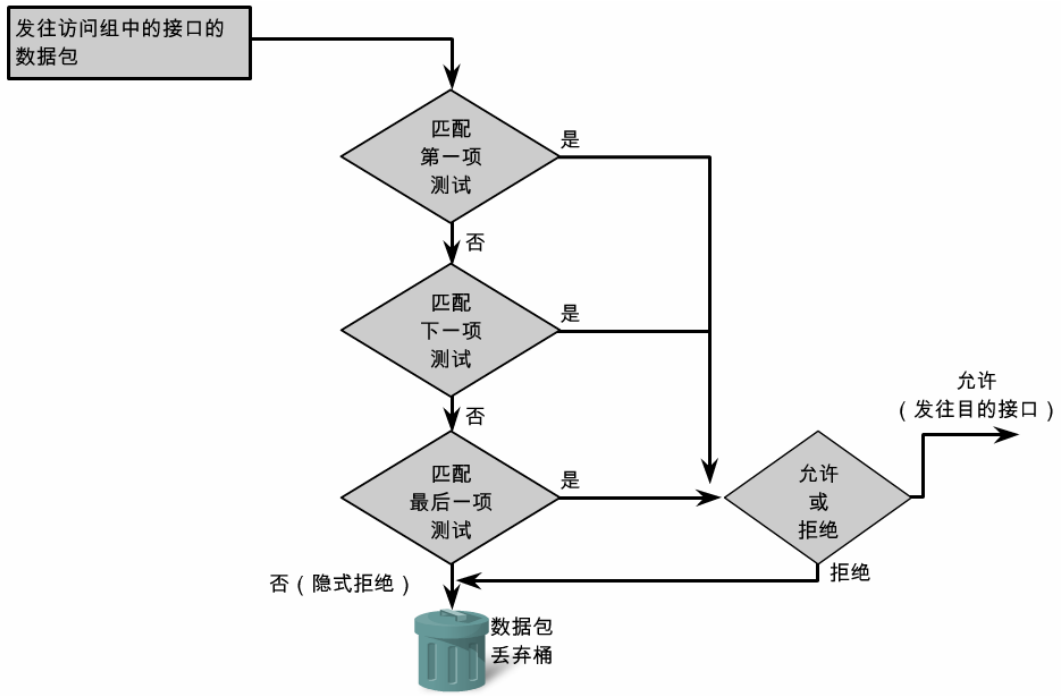
访问控制列表 ACL 分为以下两大类：

- 标准 ACL：只基于 IP 数据包的源 IP 地址作为转发或拒绝的条件，所有决定是基于源 IP 地址的。这意味着标准的访问控制列表基本上允许或拒绝整个协议组，它们不区分 IP 流量类型，如 Telnet、UDP 等服务。
- 扩展 ACL：以基于 IP 数据包的第三层和第四层信息作为数据包是否转发的条件，也就是能够基于数据包的源地址、目标地址、协议和目标端口这些条件来决定是否转发数据包。这使得扩展访问控制列表比标准访问控制列表的控制粒度更细。

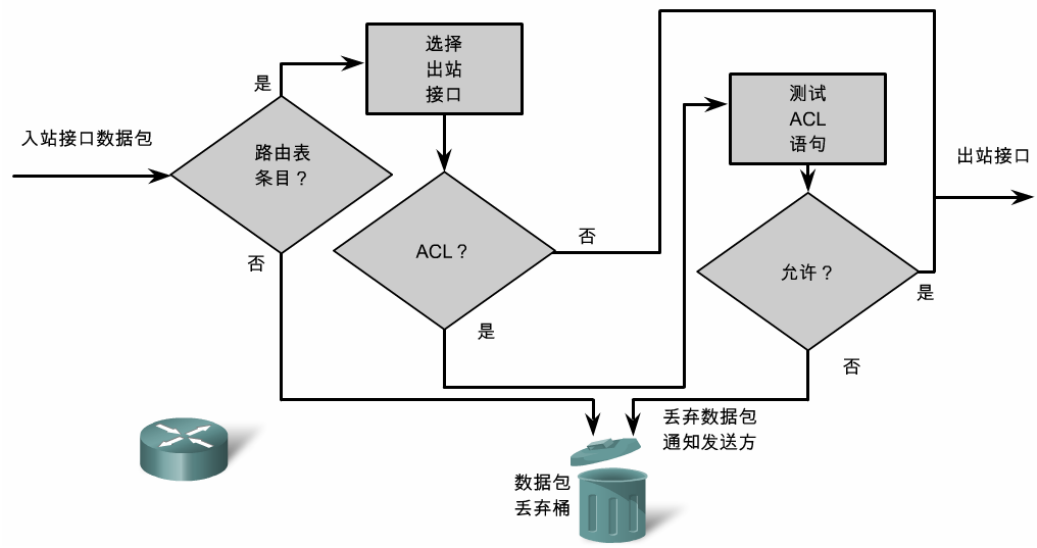
创建访问控制列表 ACL 后，就可以将其应用于任何接口的入站或出站数据流，之后就可以对穿越该接口的数据流进行控制了。ACL 要么用于入站流量，要么用于出站流量：

- 入站 ACL：数据包经过 ACL 处理之后才会被路由到出站接口。入站 ACL 非常高效，如果数据包被丢弃，则节省了执行路由查找的开销。当测试表明应允许该数据包后，路由器才会处理路由工作，如图 11-10 所示。
- 出站 ACL：数据包路由到出站接口后才由 ACL 进行处理，如图 11-11 所示。

注意：如果创建了 ACL 而没有应用它，ACL 会处于非活动状态。



▲图 11-10　入站 ACL 的执行流程



▲图 11-11　出站 ACL 的执行流程

每个 ACL 都有唯一的编号或字符名称，标准的 ACL 使用 1～99 和 1300～1999 之间的数字作为表号，扩展的 ACL 使用 100～199 和 2000～2699 之间的数字作为表号；命名 ACL 是以列表名称代替列表编号来定义 ACL 的，用字符可以很便捷地代表特定的意义，同样包括标准和扩展两种列表，需要在定义时用关键字标注。

11.3.3 通配符掩码

ACL 语句包含的掩码称为通配符掩码。通配符掩码是一串二进制数字，在访问控制列表的语句中，可以使用通配符来指定特定主机、特定网络或网络的一部分作为比对的条件，所以要通过结合使用通配符和主机（网络）地址来告诉路由器要比对的地址范围。通配符掩码为 32 位，它使用以下规则匹配二进制 1 和 0：

- 通配符掩码位 0：匹配地址中对应位的值。
- 通配符掩码位 1：忽略地址中对应位的值。

如果要指定一台主机，可以使用类似下面的组合：

```
172.16.30.5  0.0.0.0   （等效于 host 172.16.30.5）
```

其中通配符掩码中 0 表示地址的相应字节必须与指定的地址相同，1 表示任意匹配。

如果要指定某个字节为任意值，可以使用 255。例如，下面的示例演示了如何使用通配符掩码指定一个/24 子网：

```
172.16.30.0  0.0.0.255
```

这告诉路由器前 3 个字节必须完全相同，而第 4 个字节可以为任意值。

如果想表示所有的地址都匹配，可以使用下面的表示方式：

```
0.0.0.0 255.255.255.255  （等效于 any）
```

如果要匹配从 192.168.16.0 到 192.168.31.0 的网络，可以这样表示：

```
192.168.16.0 0.0.15.255  ---只检测前两个字节和第三个字节的前四个比特位---
```

提示：通配符掩码对创建 IP 访问控制列表来说很重要，必须掌握。在标准 IP 访问控制列表和扩展 IP 访问控制列表中，其用法完全相同，思路和配置 OSPF 时的反掩码一致。

11.3.4 标准访问控制列表

要在 Cisco 路由器上使用 access-list 全局配置命令以 1～99 内的数字定义标准 ACL。Cisco IOS 12.0.1 版扩大了编号的范围，允许使用 1300～1999 的编号，从而可以定义最多 799 个标准 ACL。标准 ACL 命令的完整语法如下：

```
Router(config)#access-list access-list-number  {deny|permit}  [remark] source [source-wildcard] [log]
```

各个参数的意义和作用如下：

- access-list-number：ACL 的编号，这是一个十进制数，值在 1～99 或 1300～1999 之间（适用于标准 ACL）。
- deny|permit：匹配条件时执行的动作。
- remark：可选参数，在 IP 访问列表中添加备注，增强列表的可读性。
- source：发送数据包的网络号或主机号。
- source-wildcard：可选参数，要对源应用的通配符位。
- log：可选参数，对匹配条目的数据包生成信息性日志消息并发送到控制台。

配置标准 ACL 之后，可以使用 ip access-group 命令将其关联到接口，如下：

```
Router(config-if)#ip access-group {access-list-number } {in | out}
```

- access-list-number：调用的 ACL 编号。

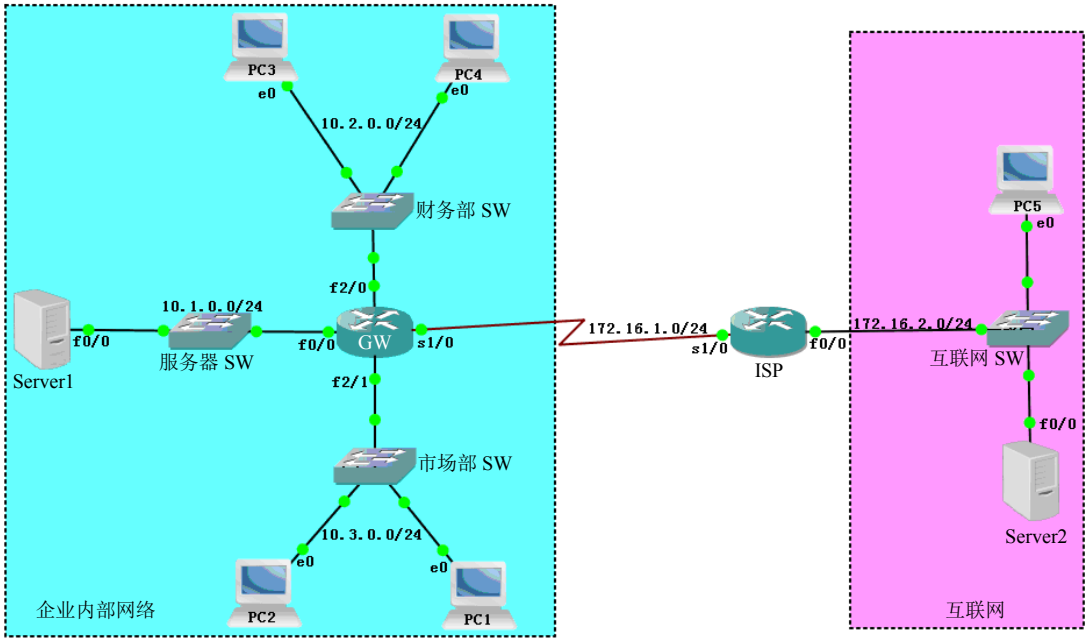
- in：对进入路由器接口的数据进行匹配。
- out：对离开路由器接口的数据进行匹配。

要删除 ACL，首先在接口上输入 no ip access-group 命令，然后输入全局命令 no access-list 删除整个 ACL。

标准 ACL 的应用实例：

如图 11-12 所示，不应该让市场部 LAN 的用户访问财务部 LAN，但应该允许它们访问因特网，可以在 GW 上设置下列 ACL 并应用：

```
GW(config)#access-list 10 deny 10.3.0.0 0.0.0.255
GW(config)#access-list 10 permit any
GW(config)#interface fastEthernet 2/0
GW(config-if)#ip access-group 10 out
GW(config-if)#exit
```



▲图 11-12　标准访问控制列表的应用实例

可以使用 show access-lists 命令验证访问控制列表的设置，操作过程如下：

```
GW#show access-lists
Standard IP access list 10
    10 deny    10.2.0.0, wildcard bits 0.0.0.255
    20 permit any
```

删除 ACL 时首先需要从接口移除相应的关联，再从全局模式中删除，操作如下：

```
GW(config)#interface fastEthernet 2/0
GW(config-if)#no ip access-group 10 out
GW(config-if)#exit
GW(config)#no access-list 10
```

用命名访问控制列表设置如下：

```
GW(config)#ip access-list standard ACL1
```

```
GW(config-std-nacl)#deny 10.3.0.0 0.0.0.255
GW(config-std-nacl)#permit any
GW(config-std-nacl)#exit
GW(config)#interface fastEthernet 2/0
GW(config-if)#ip access-group ACL1 out
GW(config-if)#end
GW#show access-lists
Standard IP access list ACL1
    10 deny    10.3.0.0, wildcard bits 0.0.0.255
    20 permit any
GW#
```

如果要拒绝市场部的特定主机（如 PC2，IP:10.3.0.2）访问财务处，可以使用下面的方式定义 ACL：

```
GW(config)#access-list 10 deny 10.3.0.2 0.0.0.0
GW(config)#access-list 10 permit any
```

或

```
GW(config)#access-list 10 host 10.3.0.2
GW(config)#access-list 10 permit any
```

11.3.5 扩展访问控制列表

为了更加精确地控制流量过滤，可以使用编号在 100～199 和 2000～2699 的扩展 ACL（最多可使用 800 个扩展 ACL），当然也可以对扩展 ACL 命名。扩展 ACL 比标准 ACL 更常用，因为其控制范围更广，可以提升安全性。与标准 ACL 类似，扩展 ACL 可以检查数据包源地址，除此之外，它们还可以检查目的地址、协议和端口号（或服务），这样就可以基于更多的因素来构建 ACL。

配置扩展 ACL 的操作步骤与配置标准 ACL 的步骤相同，只不过用于支持扩展 ACL 所提供的附加功能的命令语法和参数较为复杂，其语法格式如下：

```
Router(config)#access-list access-list-number {permit | deny [remark] } protocol source [ source-wildcard ] [operator operand] [port port-number or name] destination [destination-wildcard] [operator operand] [port port-number or name] [established]
```

各个参数的意义和作用如下：

- access-list-number：使用 100～199（扩展 IP ACL）或 2000～2699（扩充 IP ACL）之间的数字标识访问列表。
- deny|permit：匹配条件时执行的动作。
- remark：可选参数，在 IP 访问列表中添加备注，增强列表的可读性。
- protocol：Internet 协议的名称或编号。常见的关键字包括 icmp、ip、tcp 或 udp，要匹配所有 Internet 协议（包括 ICMP、TCP 和 UDP）等。
- source：发送数据包的网络号或主机号。
- source-wildcard：可选参数，要对源应用的通配符位。
- destination：数据包发往的网络号或主机号。
- destination-wildcard：要对目的地应用的通配符位。
- operator：（可选）对比源或目的端口，可用的操作符包括 lt（小于）、gt（大于）、eq（等于）、neq（不等于）和 range（范围）。
- port：（可选）TCP 或 UDP 端口的十进制编号或名称。

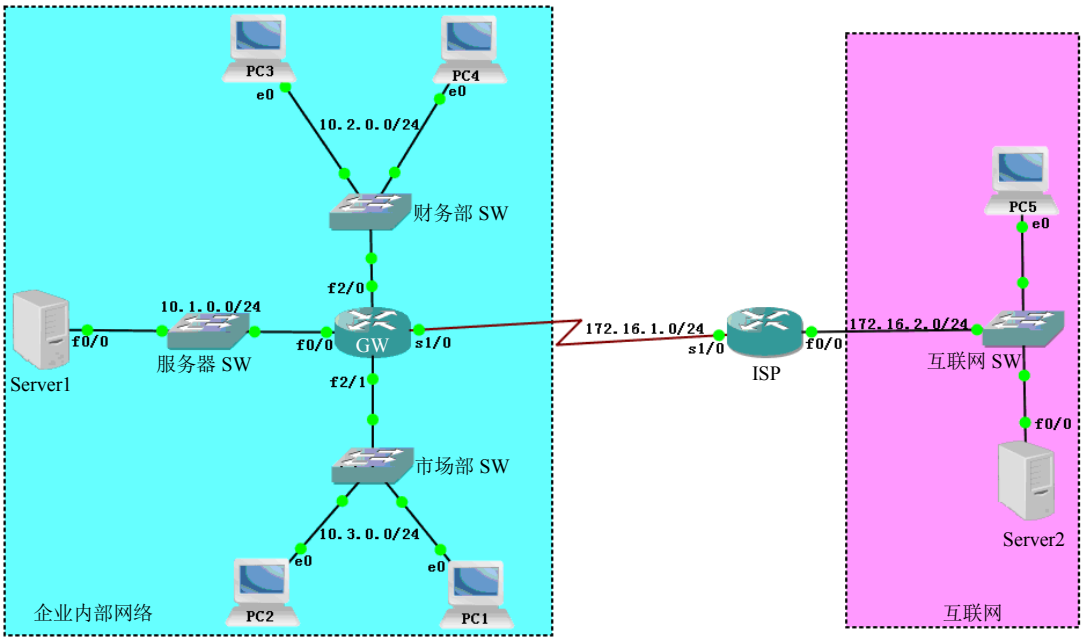
- established：（可选）仅用于 TCP 协议，指示已建立的连接。

配置扩展 ACL 之后，可以使用 ip access-group 命令将其关联到接口，使用方式和标准 ACL 一致。同样，要删除 ACL，首先在接口上输入 no ip access-group 命令，然后输入全局命令 no access-list 删除整个 ACL。

扩展 ACL 的应用实例：

如图 11-13 所示，不应该让市场部 LAN 的用户访问内外服务器的 FTP 和 Telnet 服务，但允许内外的其他服务，可以在 GW 上设置下列 ACL 并应用：

```
GW(config)#access-list 101 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq 21
GW(config)#access-list 101 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq 23
GW(config)#access-list 101 permit ip any any
GW(config)#interface fastEthernet 2/1
GW(config-if)#ip access-group 101 in
GW(config-if)#exit
```



▲图 11-13　扩展访问控制列表的应用实例

可以使用 show access-lists 命令验证访问控制列表的设置，操作过程如下：

```
GW#show access-lists
Extended IP access list 101
    10 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq ftp
    20 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq telnet
    30 permit ip any any
GW#
```

可以使用 show ip interface 命令验证接口上控制列表的应用情况，操作过程如下：

```
GW#show ip interface fastEthernet 2/1
FastEthernet2/1 is up, line protocol is up
  Internet address is 10.3.0.1/24
  Broadcast address is 255.255.255.255
```

```
    Address determined by setup command
    MTU is 1500 bytes
    Helper address is not set
    Directed broadcast forwarding is disabled
    Outgoing access list is not set
    Inbound access list is 101                    ---ACL 的编号和应用方向---
    Proxy ARP is enabled
    …
GW#
```

删除 ACL 时首先需要从接口移除相应的关联，再从全局模式中删除，操作如下：

```
GW(config)#interface fastEthernet 2/1
GW(config-if)#no ip access-group 101 in
GW(config-if)#exit
GW(config)#no access-list 101
```

用命名访问控制列表的设置如下：

```
GW(config)#ip access-list extended ACL101
GW(config-ext-nacl)#deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq ftp
GW(config-ext-nacl)#deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq telnet
GW(config-ext-nacl)#permit ip any any
GW(config-ext-nacl)#exit
GW(config)#interface fastEthernet 2/1
GW(config-if)#ip access-group ACL 101 in
GW(config-if)#end
GW#show access-lists
Extended IP access list ACL101
    10 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq ftp
    20 deny tcp 10.3.0.0 0.0.0.255 10.1.0.0 0.0.0.255 eq telnet
    30 permit ip any any
GW#
```

11.3.6 ACL 的语句顺序和放置位置

路由器执行 ACL 定义的策略时，是按照由上到下的顺序依次检查执行的，所以语句的顺序非常重要，我们来看下面的例子：

```
Router#show access-lists 10                          ---查看 ACL---
Standard IP access list 10
    permit 192.168.2.0 0.0.0.255 （4 match（es））    ---第 1 条---
    permit 192.168.1.0 0.0.0.255                     ---第 2 条---
    deny host 192.168.2.2                            ---第 3 条---
```

路由器在应用访问控制列表时，会逐一从上到下检查，如果发现匹配的就不再检查 ACL 中后面的设置。拒绝主机 192.168.2.2 的第 3 条不会用上，因为第 1 条就已经允许了。因此需要将第 3 条的设置放到第 1 条的位置，但是路由器没有调整顺序的功能，需要删除 ACL 重新创建，操作过程如下：

```
Router(config)#no access-list 10      ---删除 ACL 10 的所有设置---
Router(config)#access-list 10 deny host 192.168.2.2
Router(config)#access-list 10 permit 192.168.2.0 0.0.0.255
Router(config)#access-list 10 permit 192.168.1.0 0.0.0.255
```

所以在安排 ACL 具体的语句顺序时，应该将严格限定条件的语句放在访问控制列表的最前面。

为了方便我们可以使用文本编辑器编辑访问列表，调整好顺序后再粘贴到路由器中。

在适当的位置放置 ACL 可以过滤掉不必要的流量，使网络更加高效。ACL 可以充当防火墙来过滤数据包并去除不必要的流量，它的放置位置决定了是否能有效减少不必要的流量。例如，会被远程目的地拒绝的流量不应该消耗通往该目的地的路径上的网络资源，所以每个 ACL 只有放置在合适的位置才能很好地发挥作用。基本规则如下：

- 将扩展 ACL 尽可能靠近要拒绝流量的源。
- 将标准 ACL 尽可能靠近目的地。

11.3.7 使用访问控制列表保护路由器的安全访问

为了远程配置路由器方便，路由器一般都开启了 Telnet 功能，如何保护路由器的安全呢？例如，我们只想让网络管理员所在网段的主机访问路由器，如果创建访问控制列表只允许特定的计算机能够 Telnet 路由器，并且将该访问控制列表应用到路由器每个接口的入口方向上，这对一个具有多个接口的大型路由器来说就太麻烦了。这时我们可以将 ACL 应用于 VTY 线路，来控制那些地址远程地访问路由器，操作过程如下：

```
Router(config)#access-list 12 permit 192.168.1.0 0.0.0.255      ---定义 ACL---
Router(config)#line vty 0 15                                    ---进入 VTY 虚接口---
Router(config-line)#access-class 12 in                          ---将 ACL 应用于 Line---
```

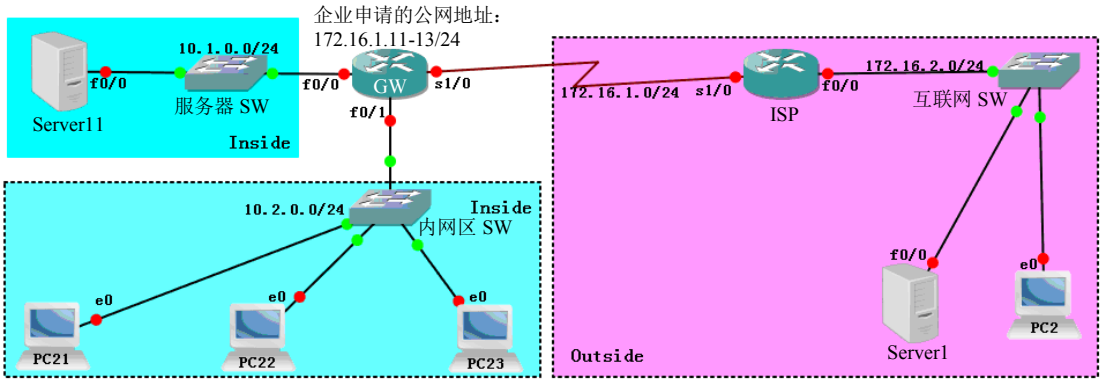
这样路由器就只能通过 192.168.1.0/24 这一网络的主机进行访问了。

11.4 网络地址转换 NAT

Internet 上的每台主机都需要有一个唯一合法的 IP 地址，随着网络技术的发展，接入 Internet 主机的数量越来越多，而可分配的 IP 地址越来越少，日益增多的 Internet 主机和日益减少的可分配 IP 地址之间的矛盾越来越突出。

NAT 技术解决了这些矛盾，NAT 全称是 Network Address Translation，中文意思是“网络地址转换”。它解决问题的办法是在内部网络中使用内部地址（私有地址），通过 NAT 把内部地址翻译成合法的 IP 地址（公网地址）在 Internet 上使用。

本节的拓扑环境如图 11-14 所示，10.x.x.x 为企业内部私有地址，172.16.x.x 为公网地址，企业从 ISP 申请的公网地址范围是 172.16.1.11-13/24。



▲图 11-14　本节的拓扑环境

11.4.1 NAT 概述

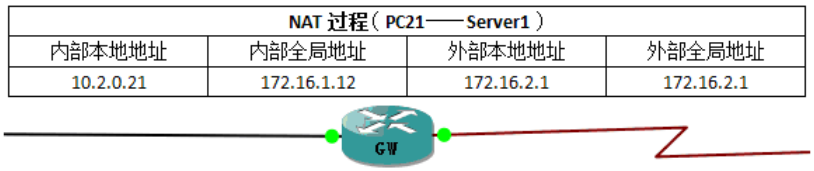
NAT 就像办公室中的前台接待员，客户拨打你单位的总机号码给前台的接待员并告诉接待员他找谁时，接待员会检查分机列表，该列表显示了与你的名字相对应的分机号，之后将它转接到你的分机。当你使用分机打电话给外面的客户时，经过前台由单位的总机号码将呼叫信息传送出去，呼叫接通后客户的电话也只显示你单位的总机号码。

启用 NAT 的设备保留一个或多个有效公网地址供网络外部访问使用。当客户端发送数据包到网络外部时，NAT 将客户端的内部私有 IP 地址转换为外部公网 IP 地址。对于外部用户来说，所有进出网络的流量均具有相同的 IP 地址或地址池。NAT 一般部署在企业网络边界的设备上，如路由器或防火墙等。

NAT 有很多用途，但最主要的用途是让网络能使用私有 IP 地址以节省地址。NAT 将不可路由的私有内部地址转换成可路由的公有地址。NAT 还能在一定程度上增加网络的私密性和安全性，因为它对外部网络隐藏了内部地址。

11.4.2 NAT 术语

结合如图 11-15 所示的通信过程，本节在讨论 NAT 时使用下列术语：



▲图 11-15 NAT 通信过程

- 内部本地地址（Inside Local）：企业内部主机使用的 IP 地址，公网上不可路由，一般使用 RFC 1918 中定义的私有地址，图中 PC21 使用的是私有地址。
- 内部全局地址（Inside Global）：当内部主机流量流出 NAT 路由器时分配给内部主机的公有地址。当来自 PC21 的流量发往服务器 Server1 172.16.2.1 时，路由器 GW 必须进行地址转换，GW 的公网地址池是 172.16.1.11-13，GW 会将其源地址转换为公网地址池中的一个地址，如 172.16.1.12，这就是内部全局地址。
- 外部全局地址（Outside Global）：分配给 Internet 上主机的可达 IP 地址，也是内部主机访问 Internet 设备的目的地址，如图中 Server1 从内部网络的角度来看，外部全局地址为 172.16.2.1，它收到企业内部主机发送数据包的源地址是 172.16.1.12。
- 外部本地地址（Outside Local）：分配给外部网络上主机的本地 IP 地址。大多数情况下，此地址与外部设备的外部全局地址相同。

注意：本节中将使用内部本地地址、内部全局地址和外部全局地址，外部本地地址的使用超出了本书的范围。

11.4.3 NAT 的类型和设置

下面介绍 NAT 的三种类型：静态 NAT、动态 NAT 和 PAT。

1. 静态 NAT

内部主机地址被一对一映射到外部主机地址，这种类型的 NAT 需要在 NAT 设备上一一建立内部本地地址到内部全局地址的静态映射，这样网络中的每台主机都拥有一个真实的因特网 IP 地址，一般用于外网对企业内部服务器的访问。

如果在规划网络时想要将 172.16.1.11 作为企业服务器 Server11（IP:10.1.0.11）对外提供服务的公网地址，则需要在 GW 上设置静态 NAT，配置过程如下：

```
GW(config)#ip nat inside source static 10.1.0.11 172.16.1.11
---建立内部本地地址与内部全局地址之间的静态映射---
GW(config)#interface fastEthernet 0/0
GW(config-if)#ip nat inside              ---指定该接口为 NAT 的内网接口---
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside             ---指定该接口为 NAT 的内网接口---
GW(config-if)#end
```

可以通过命令 show ip nat translations 来验证 NAT 的设置，操作过程如下：

```
GW#show ip nat translations
Pro Inside Global      Inside Local       Outside Local      Outside Global
--- 172.16.1.11        10.1.0.11          ---                ---
```

也可以在 GW 上执行 debug ip nat 命令来观察 NAT 的转换过程，然后在 PC2 上 ping 172.16.1.11（Server11 的内部全局地址），可以看到如下信息：

```
GW#debug ip nat
IP NAT debugging is on
*May 18 20:16:21.355: NAT*: s=172.16.2.2, d=172.16.1.11->10.1.0.11 [23887]
---对 ICMP 请求报文的源地址进行 NAT 转换---
*May 18 20:16:21.415: NAT*: s=10.1.0.11->172.16.1.11, d=172.16.2.2 [23887]
---对 ICMP 响应报文的目的地址进行 NAT 转换---
```

同样，也可以从 PC2 通过地址 172.16.1.11 访问 Server11 提供的网络服务。

2. 动态 NAT

这种类型的 NAT 可以实现映射一个内部本地地址到公网地址池中的一个内部全局地址的映射，这个映射过程是动态分配的，所以这种类型的 NAT 不像静态 NAT 需要手工地在路由器上建立映射，可以用于企业内部主机对外部网络的访问。

假设使用地址池（172.16.1.12-13）作为内部主机（IP：10.2.0.0/24）访问外部网络的公网地址，则需要在 GW 上设置动态的 NAT，配置过程如下：

```
GW(config)#access-list 2 permit 10.2.0.0 0.0.0.255
  ---通过 ACL 定义哪些内部本地地址可以用于 NAT 的转换---
GW(config)#ip nat pool TEST 172.16.1.12 172.16.1.13 netmask 255.255.255.0
  ---用于 NAT 的公网地址池，即内部全局地址---
GW(config)#ip nat inside source list 2 pool TEST
  ---用于将内部本地地址和内部全局地址关联起来---
GW(config)#interface fastEthernet 0/1
GW(config-if)#ip nat inside              ---指定该接口为 NAT 的内网接口---
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside             ---指定该接口为 NAT 的内网接口---
GW(config-if)#end
```

我们依次从 PC21、PC22、PC23 去访问 Internet 上的服务器 Server1（IP：172.16.2.1），会发现 PC21 和 PC22 能够正常访问，PC23 访问超时，使用 show ip nat translations 来验证 NAT 的表项，显示如下：

```
GW#show ip nat translations
Pro Inside Global      Inside Local       Outside Local      Outside Global
--- 172.16.1.11        10.1.0.11          ---                ---
--- 172.16.1.12        10.2.0.21          ---                ---
--- 172.16.1.13        10.2.0.22          ---                ---
```

这是因为公网地址池只有两个地址，PC21 和 PC22 先于 PC23 访问已经占用了地址池中所有可用的地址，PC23 访问时已经没有可用的公网地址了，所以访问超时，可见这种动态 NAT 并没有真正地解决地址不够用的问题，如果想让内部成百上千的主机都能同时使用内部全局地址池中有限数量的公网地址访问外面的网络，需要部署 PAT。

3. PAT

PAT 实际上是动态 NAT 的一种高级形式，它映射多个私网 IP 地址（内部本地地址）到一个公网 IP 地址（内部全局地址），并通过记录和跟踪每个不同会话的端口来区分内网主机，也被称为端口复用的地址转换 PAT，广泛用于企业、园区、学校的网络中，从而解决了网络地址不够用的问题。

假设使用地址池（172.16.1.12-13）作为内部主机（IP：10.2.0.0/24）访问外部网络的公网地址，并且要实现所有的主机都可以同时使用地址池中的公网地址访问外网，则需要在 GW 上设置 PAT，配置过程如下：

```
GW(config)#access-list 2 permit 10.2.0.0 0.0.0.255
  ---通过 ACL 定义哪些内部本地地址可以用于 NAT 的转换---
GW(config)#ip nat pool TEST 172.16.1.12 172.16.1.13 netmask 255.255.255.0
  ---用于 NAT 的公网地址池，即内部全局地址---
GW(config)#ip nat inside source list 2 pool TEST overload
  ---将内部本地地址和内部全局地址关联起来，并在转换时记录会话端口---
GW(config)#interface fastEthernet 0/1
GW(config-if)#ip nat inside          ---指定该接口为 NAT 的内网接口---
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside         ---指定该接口为 NAT 的内网接口---
GW(config-if)#end
```

我们依次从 PC21、PC22、PC23 去访问 Internet 上的服务器 Server1（IP：172.16.2.1），会发现所有的主机都能正常访问，使用 show ip nat translations 来验证 NAT，显示如下：

```
GW#show ip nat translations
Pro Inside Global      Inside Local       Outside Local      Outside Global
--- 172.16.1.11        10.1.0.11          ---                ---
icmp 172.16.1.13:1026  10.2.0.21:54122    172.16.2.1:54122   172.16.2.1:1026
icmp 172.16.1.13:1029  10.2.0.22:54890    172.16.2.1:54890   172.16.2.1:1029
icmp 172.16.1.13:1038  10.2.0.23:56170    172.16.2.1:56170   172.16.2.1:1038
```

通过上面的显示可以看出，PAT 通过记录端口号来区分不同的网络会话，实现了多个内部本地地址对一个公网地址的复用。在有些情况下 NAT 设备连接外面网络的接口是动态获取地址的，每次获取的地址信息不一致，无法确定地址池的范围，这时可以设置让内部的主机将该接口的公网地址作为内部全局地址来访问外网，操作过程如下：

```
GW(config)#access-list 2 permit 10.2.0.0 0.0.0.255
```

```
  ---通过 ACL 定义哪些内部本地地址可以用于 NAT 的转换---
GW(config)#ip nat inside source list 2 interface serial 1/0 overload
  ---将内部本地地址和外部接口关联，并将该接口地址作为 PAT 时的全局地址，同时在转换时记录会话端口---
GW(config)#interface fastEthernet 0/1
GW(config-if)#ip nat inside          ---指定该接口为 NAT 的内网接口---
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside         ---指定该接口为 NAT 的内网接口---
GW(config-if)#end
```

我们依次从 PC21、PC22、PC23 去访问 Internet 上的服务器 Server1（IP：172.16.2.1），会发现所有的主机都能正常访问，使用 show ip nat translations 来验证 NAT，显示如下：

```
GW#show ip nat translations
Pro Inside Global      Inside Local       Outside Local      Outside Global
--- 172.16.1.11        10.1.0.11          ---                ---
icmp 172.16.1.1:1024   10.2.0.21:1135     172.16.2.1:1135    172.16.2.1:1024
icmp 172.16.1.1:1135   10.2.0.22:1135     172.16.2.1:1135    172.16.2.1:1135
icmp 172.16.1.1:1903   10.2.0.23:1903     172.16.2.1:1903    172.16.2.1:1903
GW#
```

11.4.4 使用 NAT 实现网络安全

NAT 不仅可以解决 IP 地址紧缺的问题，同时也能将内部网络和外部网络隔离，为网络提供一定的安全保障，具体体现在以下两个方面：

- 静态 NAT 的安全特性：可以限定只有特定的服务才能进行 NAT 的转换，从外网来看只能访问特定的网络服务，而内网访问不受限制，效果上相当于基本的防火墙。
- PAT 的安全特性：对外屏蔽了内部网络，使外部网络不能直接对内部网络进行访问，起到了安全隔离的作用。

如果在规划网络时想要将 172.16.1.11 作为企业服务器 Server11（IP:10.1.0.11）对外提供的 Web 服务，则需要在 GW 上设置静态 NAT，配置过程如下：

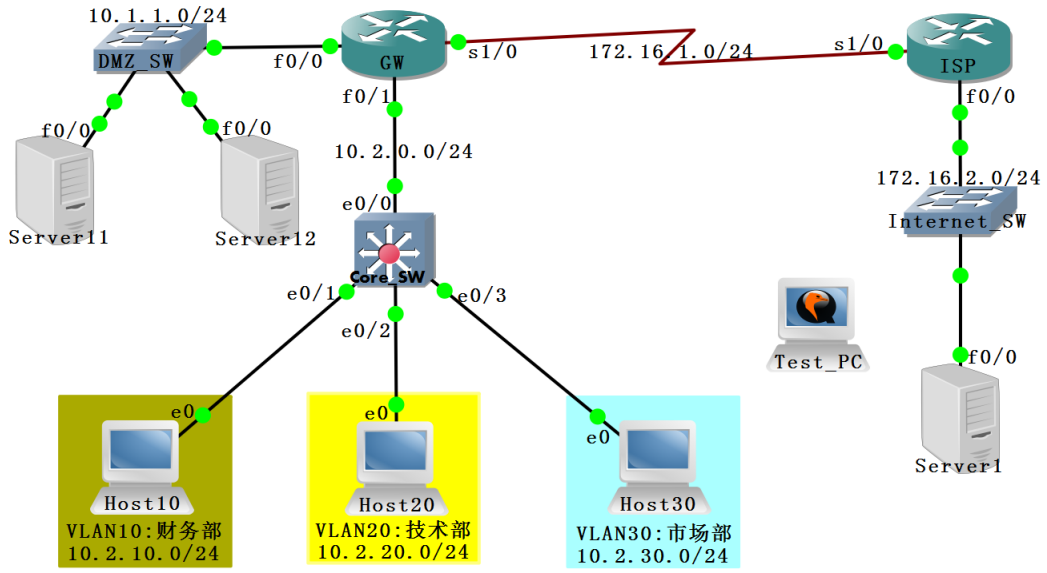
```
GW(config)#ip nat inside source static tcp 10.1.0.11 80 172.16.1.11 80
---建立内部本地地址与内部全局地址之间的 Web 访问的静态映射---
GW(config)#interface fastEthernet 0/0
GW(config-if)#ip nat inside          ---指定该接口为 NAT 的内网接口---
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside         ---指定该接口为 NAT 的内网接口---
GW(config-if)#end
```

通过上面的配置，使得从 Internet 上的主机仅能通过 172.16.1.11 访问 Server11 提供的 Web 服务，而从内网访问 Server11 的一切服务均可自由访问，保证了从外网访问时的安全。

11.5 实训案例

11.5.1 实验环境

实验拓扑：本次实验使用的拓扑通过 GNS3 搭建，如图 11-16 所示。



▲图 11-16　实验拓扑

实验说明：Server11 模拟服务器对外提供 Web 服务，同时对内提供 Web 和 FTP 服务；Server12 为财务服务器，只能由 VLAN10 财务部的用户访问；Server1 模拟 Internet 公网 Web 服务器；Test_PC 为测试机，根据测试需要随时调整地址信息完成实验的测试任务。本次实验中所有的设备均已经配置了地址信息。

实验设备：本次实验的设备如表 11-2 所示。

▲表 11-2　实验设备

设备名称	设备类型	平台版本	实现方式
ISP	路由器	C7200-JK9O3S-M，Version 12.4（25g）	GNS3 1.3.9
GW	路由器	C7200-JK9O3S-M，Version 12.4（25g）	GNS3 1.3.9
Core_SW	交换机	I86BI_LINUXL2-ADVENTERPRISEK9-M，V15.1	IOU
Internet_SW	交换机	普通非网管	GNS3 1.3.9
DMZ_SW	交换机	普通非网管	GNS3 1.3.9
Server1	服务器	C7200-JK9O3S-M，Version 12.4（25g）	GNS3 1.3.9
Server11	服务器	C7200-JK9O3S-M，Version 12.4（25g）	GNS3 1.3.9
Server12	服务器	C7200-JK9O3S-M，Version 12.4（25g）	GNS3 1.3.9
Host10	PC 机	VPCS （Version 0.6.1）	GNS3 1.3.9
Host20	PC 机	VPCS （Version 0.6.1）	GNS3 1.3.9
Host30	PC 机	VPCS （Version 0.6.1）	GNS3 1.3.9
Test_PC	PC 机	物理机（通过 VMnet1 桥接）	本地主机

地址分配：本次实验的地址分配如表 11-3 所示。

▲表 11-3　地址分配

设备	接口	IP 地址	子网掩码	网关	备注
ISP	f0/0	172.16.2.254	255.255.255.0	——	
	s1/0	172.16.1.2	255.255.255.0	——	
GW	s1/0	172.16.1.1	255.255.255.0	——	
	f0/0	10.1.0.254	255.255.255.0	——	
	f0/1	10.2.0.1	255.255.255.0	——	
Core_SW	e0/0	10.2.0.2	255.255.255.0	——	
	VLAN10	10.2.10.254	255.255.255.0	——	
	VLAN20	10.2.20.254	255.255.255.0	——	
	VLAN30	10.2.30.254	255.255.255.0	——	
Server1	e0	172.16.2.1	255.255.255.0	172.16.2.254	由路由器模拟服务器
Server11	e0	10.1.1.11	255.255.255.0	10.1.1.254	
Server12	e0	10.1.1.12	255.255.255.0	10.1.1.254	
Host10	e0	10.2.10.10	255.255.255.0	10.2.10.254	
Host20	e0	10.2.20.20	255.255.255.0	10.2.20.254	
Host30	e0	10.2.30.30	255.255.255.0	10.2.30.254	
Test_PC	VMnet1	在实验步骤中根据任务进行调整			

11.5.2　实验目的

- 掌握使用标准 ACL 限制路由器的安全管理。
- 掌握使用扩展 ACL 限制网络的安全访问。
- 掌握使用静态 NAT 对外提供安全的网络服务。
- 掌握使用 PAT 实现多个内外主机对外网的同时访问。

11.5.3　实验过程

任务一：设备的基础设置（路由器的地址信息已做了预设置）

Step 1　关闭 Server11 的路由功能并设置网关，使其成为一台服务器并启用 Web 服务。

```
Server11(config)#no ip routing                          ---关闭路由功能---
Server11(config)#interface fastEthernet 0/0
Server11(config-if)#ip address 10.1.1.11 255.255.255.0  ---设置接口地址---
Server11(config-if)#no shut
Server11(config-if)#exit
Server11(config)#ip default-gateway 10.1.1.254          ---设置默认网关---
Server11(config)#ip http server                         ---用 Web 功能模拟 Web 服务器---
```

Step 2　用同样的方式将 Server12 和 Server 配置成一台 Web 服务器（操作过程略）。

Step 3　在 GW 上设置。

```
GW(config)#ip route 10.2.0.0 255.255.0.0 fastEthernet 0/1 10.2.0.2
```

```
GW(config)#ip route 0.0.0.0 0.0.0.0 serial 1/0
```

Step 4 按照规划配置 Core_SW 接口所属的 VLAN 和对于 SVI 接口的地址。

```
Core_SW(config)#vlan 10,20,30
Core_SW(config-vlan)#exit
Core_SW(config)#interface ethernet 0/1
Core_SW(config-if)#switchport mode access
Core_SW(config-if)#switchport access vlan 10
Core_SW(config-if)#exit
Core_SW(config)#interface ethernet 0/2
Core_SW(config-if)#switchport mode access
Core_SW(config-if)#switchport access vlan 20
Core_SW(config-if)#exit
Core_SW(config)#interface ethernet 0/3
Core_SW(config-if)#switchport mode access
Core_SW(config-if)#switchport access vlan 30
Core_SW(config-if)#exit
Core_SW(config)#interface vlan 10
Core_SW(config-if)#ip address 10.2.10.254 255.255.255.0
Core_SW(config-if)#no shutdown
Core_SW(config-if)#exit
Core_SW(config)#interface vlan 20
Core_SW(config-if)#ip address 10.2.20.254 255.255.255.0
Core_SW(config-if)#no shutdown
Core_SW(config-if)#exit
Core_SW(config)#interface vlan 30
Core_SW(config-if)#ip address 10.2.30.254 255.255.255.0
Core_SW(config-if)#no shutdown
Core_SW(config-if)#exit
```

Step 5 设置 Core_SW 的 e0/0 接口地址和默认路由。

```
Core_SW(config)#interface ethernet 0/0
Core_SW(config-if)#no switchport
Core_SW(config-if)#ip address 10.2.0.2 255.255.255.0
Core_SW(config-if)#no shutdown
Core_SW(config-if)#exit
Core_SW(config)#ip route 0.0.0.0 0.0.0.0 ethernet 0/0 10.2.0.1
```

Step 6 测试从内部主机（如 Host20）到 Server11 的连通性。

```
Host20> ping 10.2.20.254
84 bytes from 10.2.20.254 icmp_seq=1 ttl=255 time=0.000 ms
84 bytes from 10.2.20.254 icmp_seq=2 ttl=255 time=12.500 ms
84 bytes from 10.2.20.254 icmp_seq=3 ttl=255 time=0.000 ms
84 bytes from 10.2.20.254 icmp_seq=4 ttl=255 time=0.000 ms
84 bytes from 10.2.20.254 icmp_seq=5 ttl=255 time=0.000 ms
Host20>
```

任务二：使用标准 ACL 限制路由器的安全管理

Step 1 在 GW 的 VTY 终端上启用 Telnet 远程管理的功能，Telnet 密码为 cisco。

```
GW(config)#line vty 0 9
GW(config-line)#password cisco
GW(config-line)#login
```

Step 2 分别从 Server11 和 Core_SW 上 Telnet 到 GW，之后在 GW 查看用户登录的情况。

```
Server11#telnet 10.1.1.254
Trying 10.1.1.254 ... Open
User Access Verification
Password:
GW>

Core_SW#telnet 10.2.0.1
Trying 10.2.0.1 ... Open
User Access Verification
Password:
GW>

GW#show users
      Line          User          Host(s)          Idle          Location
*   0 con 0                     idle             00:00:00
    2 vty 0                     idle             00:00:49 10.1.1.11
    3 vty 1                     idle             00:00:05 10.2.0.2
  Interface       User          Mode             Idle        Peer Address
GW#
```

Step 3 设置标准的 ACL（名称为 ACL1），并应用于 GW 的 line 模式中，使路由器 GW 仅能通过技术部所在的子网进行远程管理。

```
GW(config)#ip access-list standard ACL1
GW(config-std-nacl)#permit 10.2.20.0 0.0.0.255
GW(config-std-nacl)#exit
GW(config)#line vty 0 9
GW(config-line)#access-class ACL1 in
GW(config-line)#end
```

Step 4 从 GW 清除所有远程登录的用户，再次分别从 Server11 和 Core_SW 上 Telnet 到 GW，观察信息提示。

```
GW#clear line vty 0
[confirm]
 [OK]
GW#clear line vty 1
[confirm]
 [OK]
GW#

Server11#telnet 10.1.1.254
Trying 10.1.1.254 ...
% Connection refused by remote host          ---提示拒绝访问---
Server11#

Core_SW#telnet 10.2.0.1
Trying 10.2.0.1 ...
% Connection refused by remote host          ---提示拒绝访问---

Core_SW#
```

Step 5 从 Core_SW 上使用 VLAN20 的 SVI 接口地址登录 GW（模拟技术部的主机，因为 VPCS 不支持 Telnet 命令，也可以将 Test_PC 连接到 Core_SW 的 VLAN20 中进行测试）。

```
Core_SW#telnet 10.2.0.1 /source-interface vlan 20
Trying 10.2.0.1 ... Open
User Access Verification
Password:
GW>
```

任务三：使用扩展 ACL 实现仅财务部的主机能访问 Server12 服务器

Step 1 配置扩展 ACL（名称为 ACL2），并应用于 GW 的 F0/0 接口的出口方向。

```
GW(config)#ip access-list extended ACL2
GW(config-ext-nacl)#permit ip 10.2.10.0 0.0.0.255 host 10.1.1.12
GW(config-ext-nacl)#deny ip any host 10.1.1.12
GW(config-ext-nacl)#permit ip any any
GW(config-ext-nacl)#exit
GW(config)#interface fastEthernet 0/0
GW(config-if)#ip access-group ACL2 out
GW(config-if)#end
```

Step 2 在 GW 上验证 ACL 的配置及应用情况。

```
GW#show access-lists
Standard IP access list ACL1
    10 permit 10.2.20.0, wildcard bits 0.0.0.255 (2 matches)
Extended IP access list ACL2
    10 permit ip 10.2.10.0 0.0.0.255 host 10.1.1.12
    20 deny ip any host 10.1.1.12
    30 permit ip any any

GW#show ip interface
FastEthernet0/0 is up, line protocol is up
  Internet address is 10.1.1.254/24
  Broadcast address is 255.255.255.255
  Address determined by non-volatile memory
  MTU is 1500 bytes
  Helper address is not set
  Directed broadcast forwarding is disabled
  Outgoing access list is ACL2                    ---ACL2 在接口上的应用---
  Inbound access list is not set
  --More—
```

Step 3 从 Host10 分别测试到 Server11 和 Server12 的连通性。

```
Host10> ping 10.1.1.11
84 bytes from 10.1.1.11 icmp_seq=1 ttl=253 time=22.500 ms
84 bytes from 10.1.1.11 icmp_seq=3 ttl=253 time=17.500 ms
Host10> ping 10.1.1.12
84 bytes from 10.1.1.12 icmp_seq=1 ttl=253 time=31.200 ms
84 bytes from 10.1.1.12 icmp_seq=3 ttl=253 time=25.000 ms
```

Step 4 从 Host20 分别测试到 Server11 和 Server12 的连通性。

```
Host20> ping 10.1.1.11
84 bytes from 10.1.1.11 icmp_seq=1 ttl=253 time=25.000 ms
```

```
84 bytes from 10.1.1.11 icmp_seq=2 ttl=253 time=27.500 ms
Host20> ping 10.1.1.12
10.1.1.12 icmp_seq=1 timeout
10.1.1.12 icmp_seq=1 timeout
```

Step 5 从 Host30 分别测试到 Server11 和 Server12 的连通性。

```
Host30> ping 10.1.1.11
84 bytes from 10.1.1.11 icmp_seq=1 ttl=253 time=18.001 ms
84 bytes from 10.1.1.11 icmp_seq=2 ttl=253 time=20.001 ms
Host30> ping 10.1.1.12
10.1.1.12 icmp_seq=1 timeout
10.1.1.12 icmp_seq=1 timeout
```

任务四：设置静态 NAT 使外网可以访问 Server11 的 Web 服务

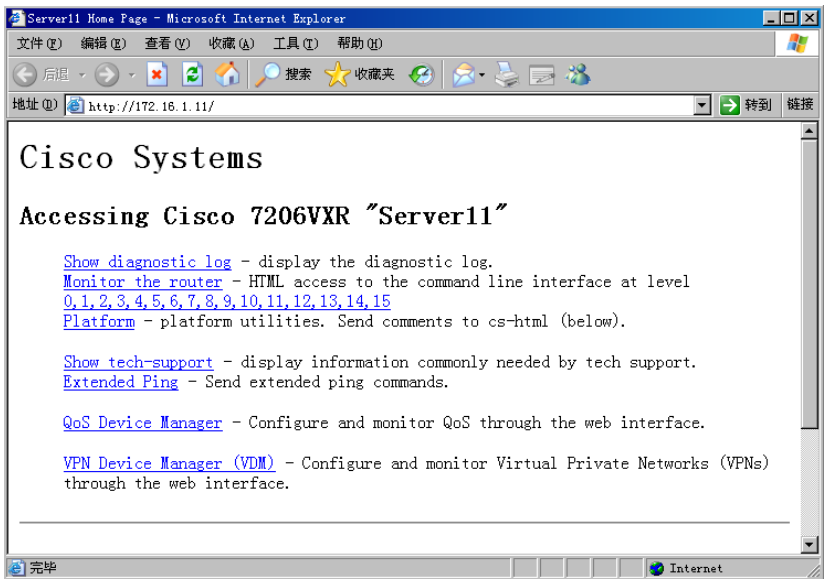
Step 1 在 GW 上建立内部本地地址（10.1.1.11）到内部全局地址（172.16.1.11）Web 服务的映射。

```
GW(config)#ip nat inside source static tcp 10.1.1.11 80 172.16.1.11 80
```

Step 2 在 GW 上设置静态 NAT 的内部接口和外部接口并验证设置。

```
GW(config)#interface fastEthernet 0/0
GW(config-if)#ip nat inside
GW(config-if)#exit
GW(config)#interface serial 1/0
GW(config-if)#ip nat outside
GW(config-if)#end
GW#show ip nat translations
Pro Inside Global     Inside Local       Outside Local      Outside Global
tcp 172.16.1.11:80    10.1.1.11:80       ---                ---
GW#
```

Step 3 将 Test_PC 连接到 Internet_SW，并设置其网络参数（IP：172.16.2.20；MASK：255.255.255.0；GW：172.16.2.254），之后访问 Server11 的公网地址（172.16.1.11），如图 11-17 所示。



▲图 11-17　访问 Server11 的公网地址

Step 4 在 Server11 上开启 Telnet 服务，不用验证，在 Core_SW 上进行测试。

```
Server11(config)#line vty 0 9
Server11(config-line)#no login
Server11(config-line)#^Z

Core_SW>telnet 10.1.1.11
Trying 10.1.1.11 ... Open
Server11>
```

Step 5 从 Test_PC 远程 Telnet 到 Server11 响应超时（效果上相当于 Server11 对外仅提供 Web 服务）。

```
C:\Documents and Settings\Administrator>telnet 172.16.1.11
---正在连接到 172.16.1.11……不能打开到主机的连接，在端口 23: 连接失败---
```

任务五：设置 PAT 使内部主机能够访问 Internet

Step 1 定义内部主机访问外网时使用的公网地址池（PATPOOL：172.16.1.12-13，即内部全局地址）。

```
GW(config)#ip nat pool PATPOOL 172.16.1.12 172.16.1.13 netmask 255.255.255.0
```

Step 2 通过 ACL（名称为 PATACL）限定可以使用公网地址的内部主机。

```
GW(config)#ip access-list standard PATACL
GW(config-std-nacl)#permit 10.2.10.0 0.0.0.255
GW(config-std-nacl)#permit 10.2.20.0 0.0.0.255
GW(config-std-nacl)#permit 10.2.30.0 0.0.0.255
GW(config-std-nacl)#end
```

Step 3 将内部主机与对应的公网地址池进行关联并启用端口复用特性，之后标注接口。

```
GW(config)#ip nat inside source list PATACL pool PATPOOL overload
GW(config)#interface fastEthernet 0/1
GW(config-if)#ip nat inside
GW(config-if)#end
```

Step 4 分别从 Host10、Host20、Host30 使用 ping 命令测试到 Server1（172.16.2.1）的网络连通性（PAT 的效果）。

```
Host10> ping 172.16.2.1
84 bytes from 172.16.2.1 icmp_seq=1 ttl=252 time=37.500 ms
84 bytes from 172.16.2.1 icmp_seq=2 ttl=252 time=50.001 ms
Host20> ping 172.16.2.1
84 bytes from 172.16.2.1 icmp_seq=1 ttl=252 time=93.600 ms
84 bytes from 172.16.2.1 icmp_seq=2 ttl=252 time=37.501 ms
Host30> ping 172.16.2.1
84 bytes from 172.16.2.1 icmp_seq=1 ttl=252 time=35.001 ms
84 bytes from 172.16.2.1 icmp_seq=2 ttl=252 time=37.002 ms
```

Step 5 在 GW 上查看 NAT 的允许情况。

```
GW#show ip nat translations
Pro Inside Global      Inside Local       Outside Local      Outside Global
tcp 172.16.1.11:80     10.1.1.11:80       ---                ---
icmp 172.16.1.12:469   10.2.10.10:469     172.16.2.1:469     172.16.2.1:469
icmp 172.16.1.12:725   10.2.10.10:725     172.16.2.1:725     172.16.2.1:725
…
icmp 172.16.1.12:5333  10.2.20.20:5333    172.16.2.1:5333    172.16.2.1:5333
icmp 172.16.1.12:5589  10.2.20.20:5589    172.16.2.1:5589    172.16.2.1:5589
```

```
…
icmp 172.16.1.12:2005    10.2.30.30:2005    172.16.2.1:2005    172.16.2.1:2005
icmp 172.16.1.12:2261    10.2.30.30:2261    172.16.2.1:2261    172.16.2.1:2261
GW#
```

本次实验任务结束。

11.6 习题

1．下面各选项中标准访问控制列表定义正确的是_____。

A．access-list 110 permit host 1.1.1.1

B．access-list 1 deny 172.16.10.1 0.0.0.0

C．access-list 1 permit 172.16.10.1 255.255.0.0

D．access-list standard 1.1.1.1

2．要禁止来自网络 192.168.160.0-192.168.191.0 的数据流通过，应该使用的 ACL 是_____。

A．access-list 10 deny 192.168.160.0 255.255.224.0

B．access-list 10 deny 192.168.160.0 0.0.191.255

C．access-list 10 deny 192.168.160.0 0.0.31.255

D．access-list 10 deny 192.168.0.0 0.0.31.255

3．将命名的访问控制列表 Blocksales 应用于路由器接口 S0 的入站方向，正确的指令是_____。

A．(config)#ip access-group 110 in

B．(config-if)#ip access-group 110 in

C．(config-if)#ip access-group Blocksales in

D．(config-if)#Blocksales ip access-list in

4．下面各选项中只允许 HTTP 数据流进入网络 196.15.7.0 的访问控制列表是_____。

A．access-list 100 permit tcp any 196.15.7.0 0.0.0.255 eq www

B．access-list 10 deny tcp any 196.15.7.0 eq www

C．access-list 100 permit 196.15.7.0 0.0.0.255 eq www

D．access-list 110 permit ip any 196.15.7.0 0.0.0.255

5．能够确定在特定接口上是否应用了 IP 访问控制列表路由器的命令是_____。

A．show ip port　　B．show access-lists

C．show ip interface　　D．show access-lists interface

6．要禁止从网络 200.200.10.0/24 以 FTP 方式访问网络 200.199.11.0/24，但允许其他所有数据流通过，应该使用的命令是_____。

A．access-list 110 deny 200.200.10.0 to network 200.199.11.0 eq ftp
access-list 111 permit ip any 0.0.0.0 255.255.255.255

B．access-list 1 deny ftp 200.200.10.0 200.199.11.0 any any

C．access-list 100 deny tcp 200.200.10.0 0.0.0.255 200.199.11.0 0.0.0.255 eq ftp

D．access-list 198 deny tcp 200.200.10.0 0.0.0.255 200.199.11.0 0.0.0.255 eq ftp
access-list 198 permit ip any 0.0.0.0 255.255.255.255

7．要创建一个标准访问控制列表，要禁止来自主机 172.16.50.172/20 所属子网的数据流通过，首先应该创建的语句是_____。

A．access-list 10 deny 172.16.48.0 255.255.240.0

B．access-list 10 deny 172.16.0.0 0.0.255.255

C．access-list 10 deny 172.16.64.0 0.0.31.255

D．access-list 10 deny 172.16.48.0 0.0.15.255

8．下面各选项中能显示路由器所有 NAT 活动转换条目的命令是_____。

A．show ip nat translations　　B．show ip nat statistics

C．debug ip nat　　D．clear ip nat translations *

9. 下面各选项中能创建一个名为 Todd 且包含 30 个全局地址的动态地址池的命令是_____。

A．ip nat pool Todd 171.16.10.65 171.16.10.94 net 255.255.255.240

B．ip nat pool Todd 171.16.10.65 171.16.10.94 net 255.255.255.224

C．ip nat pool Todd 171.16.10.65 171.16.10.94 net 255.255.255.224

D．ip nat pool Todd 171.16.10.1 171.16.10.254 net 255.255.255.0

10．经过转换后的内部主机地址是_____。

A．内部本地地址　　B．外部本地地址

C．内部全局地址　　D．外部全局地址

习题答案

1．B　2．C　3．C　4．A　5．C　6．D　7．D　8．A　9．B　10．C

12

IPv6

我们使用的第二代互联网 IPv4 技术，核心技术属于美国，它最大的问题是网络地址资源有限。从理论上讲，编址 1600 万个网络 40 亿台主机，但采用 A、B、C 三类编址方式后，可用的网络地址和主机地址的数目大打折扣，以至 IP 地址已于 2011 年 2 月 3 日分配完毕，随着物联网技术的出现，计算机网络将进入人们的日常生活，地址资源的不足已经严重地制约了网络的应用和发展。

IPv6 是 Internet Protocol Version 6 的缩写，其中 Internet Protocol 译为“互联网协议”。IPv6 是 IETF（Internet Engineering Task Force，互联网工程任务组）设计的用于替代现行版本 IP 协议（IPv4）的下一代 IP 协议，它由 128 位二进制数码表示，单从数量级上来说，IPv6 所拥有的地址容量是 IPv4 的约 8×10^{28} 倍，达到 2^{128}（算上全零的）个。有一个形象的描述说，IPv6 会使地球上的每一粒沙子都有一个 IP 地址。

本章将介绍 IPv6 比现在的 IP 有哪些方面的改进，具体介绍 IPv6 的地址体系、IPv6 下的计算机地址配置方式、IPv6 的静态路由和动态路由、支持 IPv6 的动态路由协议 RIPng 和 EIGRPv6 以及 OSPFv3 的配置、IPv6 和 IPv4 共存技术、双协议栈技术、6to4 的隧道技术、ISATAP 隧道和 NAT-PT 技术。

本章主要内容：

- IPv6 概述
- IPv6 的地址配置
- IPv6 的路由配置
- IPv6 和 IPv4 的共存技术

12.1 IPv6 概述

如果不是人们意识到 IPv4 地址即将耗竭，IPv6 便不会出现。不过除了能提供更大的地址空间外，IPv6 的制定也使人们有机会应用从 IPv4 发展中吸取的经验，从而创造一种更新、功能更强的

协议。

IPv6 使用简化的报头结构和协议运作，这意味着运行成本的降低。其固有的安全功能使安全管理更加容易，配置和部署会更加便捷。

12.1.1 IPv4 的不足之处

IPv4 的不足主要体现在以下几个方面：

（1）地址空间的不足。

在 Internet 发展的初期，人们认为网络地址是不可能分配完的，这就导致网络地址分配的随意性，其结果就是 IP 地址的利用率较低。由于组织的存在，IP 地址不是一个接一个地分配，而且由于缺乏经验的地址分类，造成了大量的地址浪费。

分配的过程是按时间顺序进行的，刚开始的时候一个学校可以拥有一个 A 类网络，后来一个国家可能只能拥有一个 C 类网络。A 类网络的数目并不多，因此问题的焦点就集中在 B 类和 C 类网络地址上，A 类的网络太大，而 C 类的网络太小，因此后来的几乎所有的申请者都愿意申请一个 B 类网络。一个 B 类网络可以拥有 65534 个主机地址，而实际上根本用不了这么多的地址，由于这样低效率的分配方法，导致 B 类地址消耗得特别快，也就导致对现有的 IP 地址的分配速率快，造成 IP 地址即将被分配完的局面。

（2）对现有路由技术的支持不够。

由于历史原因，今天的 IP 地址空间的拓扑结构都只有两层或三层，这在路由选择上是非常糟糕的。各级路由器中路由表的数目过度增长，最终的结果是使路由器不堪重负，Internet 的路由选择机制因此崩溃。

当前，Internet 发展的瓶颈已经不再是物理线路的速率，ATM 技术、百兆/千兆以太网技术的出现使得物理线路的速率有了显著的改善。现在路由器的处理速度成为阻碍 Internet 发展的主要因素，而 IPv4 天生设计上的缺陷更大大加重了路由器的负担。

首先，IPv4 的分组报头的长度是不固定的，这样不利于在路由器中直接利用硬件来实现分组中路由信息的提取、分析和选择。

其次，目前的路由选择机制仍然不够灵活，对每个分组都进行同样过程的路由选择，没有充分利用分组间的相关性。

最后，由于 IPv4 设计时未能完全遵循端到端通信的原则，加上当时物理线路的误码率比较高，使得路由器还要具备以下两个功能：

- 根据线路的 MTU 来分段和重组过大的 IP 分组。
- 逐段进行数据校验。

但这同样会造成路由器处理速度的降低。

（3）无法提供多样的 QoS。

随着 Internet 的成功和发展，商家已经将更多的关注投向了 Internet，他们意识到其中蕴含着巨大的商机，今天乃至将来，有很多的业务应用都会在互联网上进行。这些业务中包括对时间和带宽要求很高的实时多媒体业务，如语音、图像等；对安全性要求很高的电子商务业务以及发展越来越迅猛的移动 IP 业务等。这些业务对网络 QoS 的要求各不相同，但 IPv4 在设计时没有引入 QoS 这样的概念，设计上的不足使得它很难相应地提供丰富的灵活的 QoS 选项。

虽然人们提出了一系列的技术，如 NAT、CIDR、VLSM、RSVP 等来缓解这些问题，但这些

方法都只是权宜之计，解决不了因地址不多及地址结构不合理而导致的地址短缺的根本问题，最终 IPv6 应运而生。

12.1.2 IPv6 的改进

IPv6 相对于 IPv4 来说有以下几个方面的改进：

（1）扩展的地址空间和结构化的路由层次。

IPv6 地址长度由 IPv4 的 32 位扩展到 128 位，全局单点地址采用支持无分类域间路由的地址聚类机制，可以支持更多的地址层次和更多的节点数目，并且使得自动配置地址更加简单。

（2）简化了报头格式。

IPv4 报头中的一些字段被取消或变成可选项，尽管 IPv6 的地址长度是 IPv4 的 4 倍，但是 IPv6 基本报头的字段类型比 IPv4 报头的字段类型多。取消了对报头中可选项长度的严格限制，增加了灵活性。

（3）简单的管理：即插即用。

IPv6 通过实现一系列的自动发现和自动配置功能，简化网络节点的管理和维护。已实现的典型技术包括最大传输单元发现（MTU Discovery）、邻接节点发现（Neighbor Discovery）、路由器通告（Router Advertisement）、路由器请求（Router Solicitation）、节点自动配置（Auto-configuration）等。

（4）安全性。

在制定 IPv6 技术规范的同时，产生了 IPSec（IP Security），用于提供 IP 层的安全性。目前，IPv6 实现了认证头（Authentication Header，AH）和封装安全载荷（Encapsulated Security Payload，ESP）两种机制。前者实现数据的完整性及对 IP 包来源的认证，保证分组确实来自源地址所标记的节点；后者提供数据加密功能，实现端到端的加密。

（5）QoS 能力。

报头中的“标签”字段允许鉴别属于同一数据流的所有报文，因此路径上所有路由器可以鉴别一个流的所有报文，实现非默认的服务质量或实时服务等特殊处理。

（6）改进的多点寻址方案。

通过在组播地址中增加“范围”字段，允许将组播的路由限定在正确的范围之内；另一个“标志”字段允许 Internet 区分永久性的多点地址和临时性的多点地址。

（7）定义了一种新的群通信地址方式：Anycast。

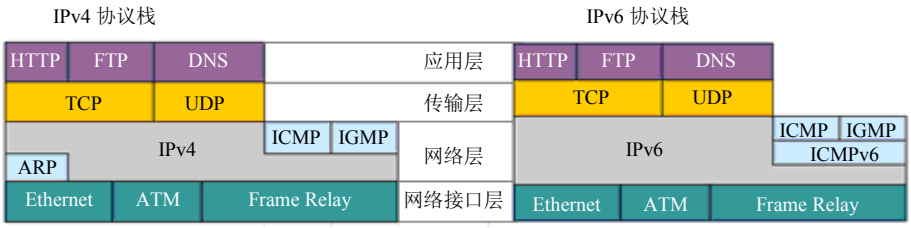
在点到多点的通信中，将报文传递到一组节点中的一个（通常是最近的一个），从而在源点路由中允许节点控制传递路径。

（8）可移动性。

IPv6 协议设计的若干技术有利于移动计算的实现，包括信宿选项头（Destination Options Header）、路由选项头（Routing Header）、自动配置、安全机制以及 Anycast 技术。将 QoS 技术同移动节点相结合还将强化 IPv6 对移动计算的支持。

12.1.3 IPv6 协议栈

IPv4 和 IPv6 协议栈的比较如图 12-1 所示。



▲图 12-1　IPv4 和 IPv6 协议栈的比较

可以看到，IPv6 协议栈与 IPv4 协议栈相比较在网络层变化最大，IPv6 的网络层没有 ARP 协议和 IGMP 协议，ICMP 协议功能作了很大的扩展，ICMP 在 IPv6 定义中重新修订。此外，IPv4 组成员协议（IGMP）的多点传送控制功能和 ARP 协议的功能也嵌入到 ICMPv6 中，分别是邻居发现（ND）协议和多播侦听器发现（MLD）协议。

IPv6 网络层的核心协议包括以下几种：

- IPv6：取代 IPv4，它是一个可路由协议，为数据包进行寻址、路由、分段和重组。
- Internet 控制消息协议 IPv6 版（ICMPv6）：取代 ICMP，报告错误和其他信息以帮助诊断不成功的数据包传送。
- 邻居发现（ND）协议：ND 取代 ARP，管理相邻 IPv6 节点间的交互，包括自动配置地址和将下一跃点 IPv6 地址解析为 MAC 地址。
- 多播侦听器发现（MLD）协议：MLD 取代 IGMP，管理 IPv6 多播组成员身份。

12.1.4　IPv6 的地址格式和层次结构

IPv6 的主要改变就是地址的长度，变为了 128 位。IPv6 地址一共有 2^{128} 个，书写时使用冒号将 128 个二进制分割成 8 个 16 比特的数组，再将每个数组表示成 4 位十六进制数，如 11010001.11011100.11001001.01110001.11011100.11001100.01110001.11010001.11011100.11001001.11010001.11011100.11001001.01110001 可以表示为 A524:72D3:2C80:DD02:0029:EC7A:002B:EA73，尽管可以用十六进制表示 IPv6 的地址，但记录起来依然不太方便，我们还可以用下面的方法简化 IPv6 的书写：

- 字段中的前导零可以省略。例如，字段 09C0 等效于 9C0，字段 0000 等效于 0，2031:0000:130F:0000:0000:09C0:876A:130B 等效于 2031:0:130F:0:0:9C0:876A:130B。
- 连续的零字段可以用两个冒号“::”表示，但这种缩写方法在一个地址中只能使用一次。例如，2031:0:130F:0000:0000:9C0:876A:130B 等效于 2031:0:130F::9C0:876A:130B。

下面是一些 IPv6 地址的书写实例：

- FF01:0:0:0:0:0:0:1 可以表示为 FF01::1。
- 0:0:0:0:0:0:0:1 可以表示为::1。
- 0:0:0:0:0:0:0:0 可以表示为::。
- E3D7:0000:0000:0000:51F4:00C8:C0A8:6420 可以表示为 E3D7::51F4:C8:C0A8:6420。

当使用 Web 浏览器向一台 IPv6 设备发起 HTTP 连接时，必须将 IPv6 地址输入浏览器，而且要用方括号将 IPv6 地址括起来。为什么呢？这是因为浏览器在指定端口号时，已经使用了一个冒号。如果不用方括号将 IPv6 地址括起来，浏览器将无法识别出信息。

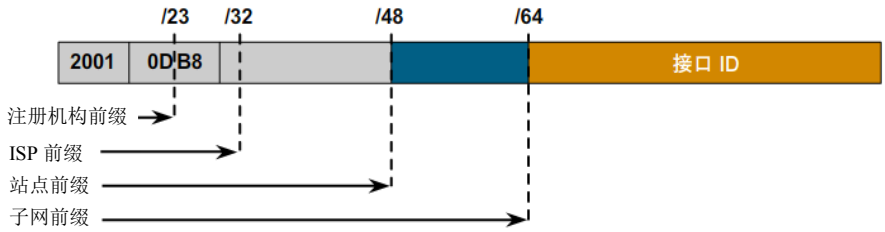
下面是这种情况的一个例子。

```
http://[2001:0db8:3c4d:0012:0000:0000:1234:56ab]/default.html
```

显然，使用网站的域名来访问站点更便捷，因此在 IPv6 的网络中 DNS 变得尤为重要。

我们知道 IP 地址由网络部分和主机部分组成，IPv6 用前缀表示网络部分，表示形式是“IPv6 地址/前缀长度”，其中“前缀长度”是一个十进制数，表示该地址的前多少位是网络部分。例如，F00D:4598:7304:3210:FEDC:BA98:7654:3210，如果前 64 位是该地址的网络部分，就可以表示为 F00D:4598:7304:3210:FEDC:BA98:7654:3210/64。

IPv6 地址的前缀表示方法特别便于地址的向上聚合，最终到达 ISP。IPv6 的全球单播地址（相当于 IPv4 的公网地址）通常由 48 位全球路由前缀和 16 位子网 ID 组成。各组织可以使用 16 位子网字段创建自己的本地编址架构，此字段允许组织使用最多 65535 个子网，如图 12-2 所示。



▲图 12-2　IPv6 的全局单播地址

图 12-2 的上部可以看出，如何使用注册机构前缀、ISP 前缀和站点前缀将附加架构添加到 48 位全球路由前缀中。第 49 位至 64 位为子网前缀，共有 16 个比特位组成，一般可用于企业网络各个网段的标识，接口 ID 代表一个网络中的节点标识，类似于 IPv4 地址的主机部分。

目前的全球单播地址由 IANA 分配，使用的地址范围是从二进制值 001（2000::/3）开始，它占全部 IPv6 地址空间的 1/8，是最大的一块分配地址。IANA 将 2001::/16 范围内的 IPv6 地址空间分配给 5 家 RIR 注册机构（ARIN、RIPE、APNIC、LACNIC 和 AfriNIC）。

有关更多信息请参考 RFC 3587“IPv6 全球单播地址格式”，它取代了 RFC 2374。

12.1.5　IPv6 的地址类型

RFE 2373 中定义了三种 IPv6 的地址类型：单播地址（Unicast）、多播地址（Multicast）和任播地址（Anycast）。

1. 单播地址（Unicast）

和 IPv4 的单播地址一致，是在点对点通信时使用的地址，此地址仅标识一个网络接口，是接口在 IPv6 网络中的逻辑标识。单播地址有以下几种形式：

- 全球单播地址（Global Unicast Address）：相当于 IPv4 的公网地址，其结构在图 12-2 中已经介绍过，目前有一小部分全球单播地址已经由 IANA（互联网名称与数字地址分配机构 ICANN 的一个分支）分配给了用户。全球单播地址的前缀是“2000::/3”，代表公共 IP 网络上任意可及的地址。IANA 负责将该段地址范围内的地址分配给多个区域互联网注册管理机构（RIR），它负责全球 5 个区域的地址分配，其中已经分配的地址块包括 2400::/12（APNIC）、2600::/12（ARIN）、2800::/12（LACNIC）、2A00::/12（RIPE NCC）和 2C00::/12（AfriNIC），这些地址符合典型的分层结构，便于地址信息的聚合。

- 链路本地单播地址（Link-Local Unicast Address）：只能在连接到同一本地链路的节点之间使用，以链路的本地地址为源地址或目的地址的 IPv6 报文不会被路由器转发到其他链路。可以在自动地址分配、邻居发现和链路上没有路由器的情况下使用链路本地地址。链路本地地址的前缀是“FE80::/10”，对应第 3 个十六进制数字是 8 到 B 之间的值，因此这些地址以“FE8”“FE9”“FEA”或“FEB”开始。
- 站点本地单播地址（Site-Local Unicast Address）：与当今 IPv4 中 RFC 1918“私有 Internet 地址分配”规定的地址相似，这些地址的使用范围是整个站点或组织内部。站点本地地址的前缀是“FEC0::/10”，对应第 3 个十六进制数字是 C 到 F 之间的值，因此这些地址以“FEC”“FED”“FEE”或“FEF”开始。不过由于 IPv6 的地址资源极其丰富，2003 年发布的 RFC 3879 已经不支持使用此类地址。

2. 多播地址（Multicast）

多播地址也叫做组播地址，是为特定应用组播组保留的，如路由协议和一些常见的语音视频应用。IPv6 中没有广播地址，使用组播地址替代广播地址可以确保报文只发送给特定的组播组而不是 IPv6 网络中的任意终端，组播地址的前缀是“FF00::/8”。

3. 任播地址（Anycast）

将一个单播地址分配给一组提供统一网络服务的服务器的形式，当一个单播地址被分配给多个服务器时，向任播地址发送的数据包被发往离源地址最近的服务器，并由最近的服务器响应源地址发送的请求，目前任播地址的部署和应用还处于实验阶段。

12.1.6 IPv6 中特殊的地址

环回地址：与 IPv4 一样，IPv6 也提供了特殊环回地址以供测试使用，发送到此地址的数据报文会环回到发送设备。不过 IPv6 中用于此功能的地址只有一个，而不是一个地址块。环回地址为 0:0:0:0:0:0:0:1，一般用零的压缩形式表示为“::1”。

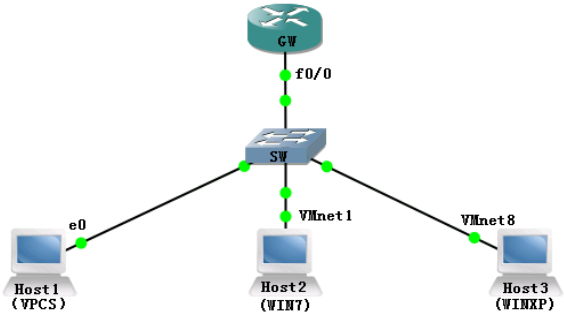
不特定地址：IPv4 中，全零 IP 地址有特殊的意义，它指主机本身，当设备不知道其自身地址时使用。IPv6 已将此概念规范化，将全零地址（0:0:0:0:0:0:0:0）命名为“不特定”地址。当设备要求配置自身 IP 地址时，该地址将用在所发送数据报的源地址字段中。对此地址可以应用地址压缩，因为该地址全为 0，所以可以简单地记为“::”。

12.2 IPv6 的地址配置

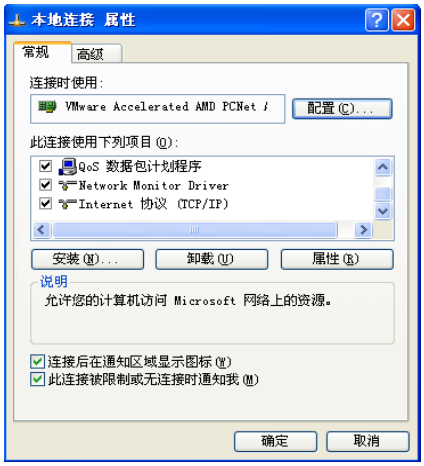
IPv6 地址是网络设备或计算机主机接口的逻辑标识，IPv6 协议的一个突出特点是支持网络节点地址自动配置，极大地简化了网络管理者的工作，本节以图 12-3 为例介绍各种情况下 IPv6 地址的分配方法。

12.2.1 启用 IPv6 协议

我们给某个网络设备或主机的接口分配 IPv6 的地址，首先需要该设备支持 IPv6 的协议，Windows Server 2003 和 Windows 7 及以上版本默认已经启用了 IPv6，Windows XP 默认没有启用 IPv6 协议，如图 12-4 所示。



▲图 12-3　IPv6 地址分配环境



▲图 12-4　Windows XP 默认的协议栈

如果需要在 Host3 安装 IPv6 协议，在 CMD 窗口中执行 ipv6 install 命令，操作如下：

```
C:\Documents and Settings\Administrator>ipv6 install
```

在 Host3 的命令提示符下，输入 ipconfig 能够看到 IPv6 的链路本地地址，操作如下：

```
Ethernet adapter 本地连接:
        Connection-specific DNS Suffix. :
        Description . . . . . . . . . . . :    VMware Accelerated AMD PCNet Adapter
        Physical Address. . . . . . . . . :    00-0C-29-7E-E2-57
        Dhcp Enabled. . . . . . . . . . . :    Yes
        Autoconfiguration Enabled . . . . :    Yes
        Autoconfiguration IP Address. . . :    169.254.72.231
        Subnet Mask . . . . . . . . . . . :    255.255.0.0
        IP Address. . . . . . . . . . . . :    fe80::20c:29ff:fe7e:e257%5    ---链路本地地址---
        Default Gateway . . . . . . . . . :
        DNS Servers . . . . . . . . . . . :    fec0:0:0:ffff::1%1
```

注意：系统启用 IPv6 后所有的接口都会自动获得一个都是以“FE80::/10”为前缀的链路本地地址，所以在使用链路本地地址进行通信时还必须指明相应的接口名称或编号。上面的链路本地地址后面的“%5”标识系统 ID 为 5 的接口对应的链路本地地址，其他的接口有不同的 ID 标识，这个 ID 是系统分配的，只有本地意义。

Windows 7 系统已经默认启用了 IPv6 的协议，我们可以在 Host2 上查看其 IPv6 的信息。

```
C:\Users\Administrator>ipconfig
以太网适配器 VMware Network Adapter VMnet1:
   连接特定的 DNS 后缀 . . . . . :
   本地连接 IPv6 地址. . . . . :    fe80::dad:ffb9:a9f3:d9fe%14    ---链路本地地址---
   IPv4 地址 . . . . . . . . . . . . :    10.1.0.100
   子网掩码 . . . . . . . . . . . . :    255.255.255.0
   默认网关. . . . . . . . . . . . . :
```

如果从 Host3 上通过 ping 命令测试到 Host2 链路本地地址的连通性，需要在 ping 命令后面标识出接口 ID，操作过程如下：

```
C:\Documents and Settings\Administrator>ping fe80::dad:ffb9:a9f3:d9fe%5
Pinging fe80::dad:ffb9:a9f3:d9fe%5 with 32 bytes of data:
Reply from fe80::dad:ffb9:a9f3:d9fe%5: time<1ms
Reply from fe80::dad:ffb9:a9f3:d9fe%5: time<1ms
Reply from fe80::dad:ffb9:a9f3:d9fe%5: time<1ms
Reply from fe80::dad:ffb9:a9f3:d9fe%5: time<1ms
Ping statistics for fe80::dad:ffb9:a9f3:d9fe%5:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
C:\Documents and Settings\Administrator>
```

注意：使用 Windows 7 系统的 ping 命令测试到 Windows XP 系统 IPv6 接口的连通性时，Windows 7 能够自动识别相应的接口，因此可以在 ping 命令后面不指明对应的接口 ID，我们可以对比下面的结果。

```
C:\Users\Administrator>ping fe80::20c:29ff:fe7e:e257%14          ---指明接口 ID---
正在 ping fe80::20c:29ff:fe7e:e257%14 具有 32 字节的数据:
来自 fe80::20c:29ff:fe7e:e257%14 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257%14 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257%14 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257%14 的回复: 时间<1ms
fe80::20c:29ff:fe7e:e257%14 的 ping 统计信息:
    数据包: 已发送= 4，已接收= 4，丢失= 0（0%丢失）
往返行程的估计时间（以毫秒为单位）:
    最短= 0ms，最长= 0ms，平均= 0ms

C:\Users\Administrator>ping fe80::20c:29ff:fe7e:e257              ---未指明接口 ID---
正在 ping fe80::20c:29ff:fe7e:e257 具有 32 字节的数据:
来自 fe80::20c:29ff:fe7e:e257 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257 的回复: 时间<1ms
来自 fe80::20c:29ff:fe7e:e257 的回复: 时间<1ms
fe80::20c:29ff:fe7e:e257 的 ping 统计信息:
    数据包: 已发送= 4，已接收= 4，丢失= 0（0%丢失）
往返行程的估计时间（以毫秒为单位）:
    最短= 0ms，最长= 0ms，平均= 0ms
C:\Users\Administrator>
```

GNS3 的 VPCS 默认也启用了 IPv6 协议，可以使用下面的命令查看其地址信息。

```
Host1> show
NAME        IP/MASK              GATEWAY         MAC                LPORT   RHOST:PORT
Host1       0.0.0.0/0            0.0.0.0         00:50:79:66:68:00  10000   10.8.0.100:10001
            fe80::250:79ff:fe66:6800/64
Host1>
```

和 Windows 7 一致，VPCS 使用 IPv6 的链路本地地址通信时也能自动识别接口 ID，我们从 Host1 使用 ping 命令测试到 Host2 和 Host3 链路本地地址的连通性，操作如下：

```
Host1> ping fe80::dad:ffb9:a9f3:d9fe
fe80::dad:ffb9:a9f3:d9fe icmp6_seq=1 ttl=128 time=0.000 ms
fe80::dad:ffb9:a9f3:d9fe icmp6_seq=2 ttl=128 time=0.000 ms
fe80::dad:ffb9:a9f3:d9fe icmp6_seq=3 ttl=128 time=1.000 ms
fe80::dad:ffb9:a9f3:d9fe icmp6_seq=4 ttl=128 time=2.000 ms
fe80::dad:ffb9:a9f3:d9fe icmp6_seq=5 ttl=128 time=1.000 ms

Host1> ping fe80::20c:29ff:fe7e:e257
fe80::20c:29ff:fe7e:e257 icmp6_seq=1 ttl=128 time=135.007 ms
fe80::20c:29ff:fe7e:e257 icmp6_seq=2 ttl=128 time=4.001 ms
fe80::20c:29ff:fe7e:e257 icmp6_seq=3 ttl=128 time=0.000 ms
fe80::20c:29ff:fe7e:e257 icmp6_seq=4 ttl=128 time=8.000 ms
fe80::20c:29ff:fe7e:e257 icmp6_seq=5 ttl=128 time=4.000 ms
Host1>
```

在思科路由器上启用 IPv6 协议，需要在全局配置模式中执行 ipv6 unicast-routing 命令，操作如下：

```
GW(config)#ipv6 unicast-routing                  ---在路由器上激活 IPv6 协议并启用路由转发---
GW(config)#interface fastEthernet 0/0
GW(config-if)#ipv6 enable                        ---在接口上启用 IPv6---
```

注意：路由器的接口明确启用了 IPv6 协议或配置了单播 IPv6 地址后才有链路本地地址。

```
GW#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::C801:9FF:FE14:0
GW#
```

可以在子路由器上使用命令 show ipv6 interface [brief]查看接口 IPv6 的设置情况。

```
GW#show ipv6 interface
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C801:9FF:FE14:0
  No global unicast address is configured
---省略部分输出---
GW#show ipv6 interface brief
FastEthernet0/0            [up/up]
    FE80::C801:9FF:FE14:0
GW#
```

如果从路由器上通过 ping 测试到 Host2 链路本地地址的连通性，需要指出相应的接口 fas0/0，操作过程如下：

```
GW#ping fe80::dad:ffb9:a9f3:d9fe          ---测试到 Host2 接口链路本地地址的连通性---
Output Interface: FastEthernet0/0         ---指明出口---
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FE80::DAD:FFB9:A9F3:D9FE, timeout is 2 seconds:
Packet sent with a source address of FE80::C801:9FF:FE14:0
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/24/28 ms
GW#
```

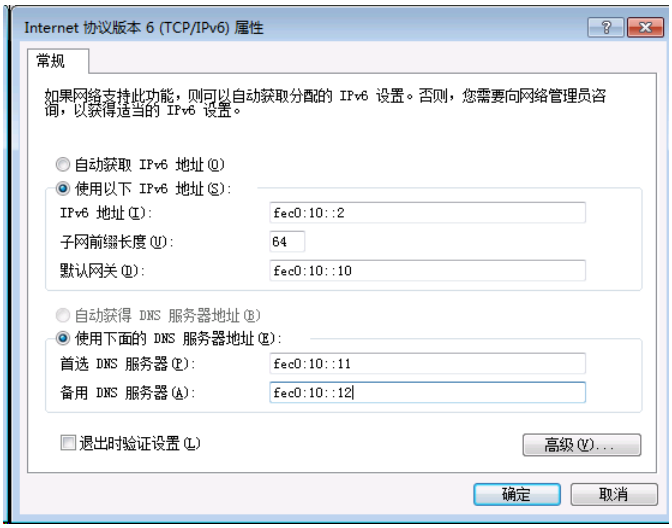
12.2.2 手工配置 IPv6 地址

Windows XP 和 Windows Server 2003 没有提供图形界面配置 IPv6 的地址、网关及 DNS 等，这些网络信息可以通过 netsh 命令设置，以 Host3 为例，设置过程如下：

```
C:\Documents and Settings\Administrator>netsh      ---执行 netsh 命令---
netsh>interface                                    ---进入接口视图---
netsh interface>ipv6                               ---进入 IPv6 视图，在此视图中可以执行 IPv6 相关的命令---
netsh interface ipv6>reset                         ---重置 IPv6 协议栈，清除 IPv6 相关的参数、复位设置---
netsh interface ipv6>show interface                ---查看接口的索引、状态、名称等信息---
正在查询活动状态……
索引    Met    MTU    状态          名称
----    ----   -----  ------------  -----
5       0      1500   已连接        本地连接
4       2      1280   已断开        Teredo Tunneling Pseudo-Interface
3       1      1280   已连接        6to4 Pseudo-Interface
2       1      1280   已连接        Automatic Tunneling Pseudo-Interface
1       0      1500   已连接        Loopback Pseudo-Interface
netsh interface ipv6>add address 5 fec0:10::3      ---索引号为 5 的接口设置 IPv6 地址---
netsh interface ipv6>show address                  ---查看本机所有接口的地址信息---
正在查询活动状态……
接口 5：本地连接
地址类型    DAD 状态   有效寿命    首选寿命    地址
---------   ---------  ----------  ----------  ----------------------------
手动        首选项     infinite    infinite    fec0:10::3
链接        首选项     infinite    infinite    fe80::20c:29ff:fe7e:e257

接口 4：Teredo Tunneling Pseudo-Interface
地址类型    DAD 状态   有效寿命    首选寿命    地址
---------   ---------  ----------  ----------  ----------------------------
链接        首选项     infinite    infinite    fe80::ffff:ffff:fffd
---省略部分输出---
netsh interface ipv6>add route ::/0 5 fec0:10::10  ---设置默认路由，相当于网关---
netsh interface ipv6>show route                    ---查看路由（网关）信息---
正在查询活动状态……
发行        类型       Met        前缀              索引    网关/接口名
-------     --------   ----       ----------------  ---     --------------------
no          手动       0          ::/0              5       fec0:10::10
netsh interface ipv6>add dns   "本地连接" fec0:10::11    ---设置 DNS 信息---
netsh interface ipv6>add dns   "本地连接" fec0:10::12    ---注意设置 DNS 时使用的接口名称---
netsh interface ipv6>show dns                      ---查看 DNS 信息---
DNS 在接口上的 DNS 服务器：本地连接
```

```
索引    DNS 服务器
-----   ------------------------------------------------
 1      fec0:10::11
 2      fec0:10::12
netsh interface ipv6>exit                          ---退出 netsh---
C:\Documents and Settings\Administrator>ipconfig   ---查看网络参数的设置---
Windows IP Configuration
Ethernet adapter 本地连接：
        Connection-specific DNS Suffix  . :
        Autoconfiguration IP Address. . .  :169.254.72.231
        Subnet Mask . . . . . . . . . . .  :255.255.0.0
        IP Address. . . . . . . . . . . .  :fec0:10::3%1
        IP Address. . . . . . . . . . . .  :fe80::20c:29ff:fe7e:e257%5
        Default Gateway . . . . . . . . .  :fec0:10::10%1
```

Windows 7 的 IPv6 信息设置和 IPv4 类似，除了使用 netsh 还可以通过窗口直接设定，以 Host2 为例，设置方式如图 12-5 所示。



▲图 12-5　Windows 7 中设置 IPv6 的相关信息

在 VPCS 中设置 IPv6 信息的方式如下（以 Host1 为例）：

```
Host1> ip fec0:10::1/64 fec0:10::10          ---设置 IPv6 地址信息---
PC1 : fec0:10::1/64
Host1> show                                  ---验证 IPv6 地址信息---
NAME      IP/MASK                GATEWAY       MAC               LPORT    RHOST:PORT
Host1     0.0.0.0/0              0.0.0.0       00:50:79:66:68:00 10000    10.8.0.100:10001
          fe80::250:79ff:fe66:6800/64
          fec0:10::1/64
Host1>
```

在路由器 GW 中手动设置接口 Fas0/0 的 IPv6 地址，操作如下：

```
GW(config)#interface fastEthernet 0/0
GW(config-if)#ipv6 address fec0:10::10/64
GW(config-if)#end
GW#show ipv6 interface brief
```

```
FastEthernet0/0                [up/up]
    FE80::C801:9FF:FE14:0
    FEC0:10::10
GW#
```

从路由器使用 ping 命令测试到每个主机的连通性，操作过程如下：

```
GW#ping fec0:10::1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/28/32 ms

GW#ping fec0:10::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/34/72 ms

GW#ping fec0:10::3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 12/25/36 ms
GW#
```

12.2.3 无状态自动配置

前面我们介绍过，一个典型的 IPv6 主机单播地址由 3 部分组成：全局路由前缀、子网 ID 和接口 ID（64 位）。全局路由前缀用来识别分配给一个站点或企业的一个地址范围；子网 ID 也称为子网前缀，一个子网 ID 与一个链接相关联，以识别站点中的某个链接或网段；接口 ID 用来识别链接上的某个接口，在该链接上是唯一的，接口 ID 的配置方式有以下 3 种：

- 网络管理员手动配置。
- 通过系统软件生成。
- 采用扩展唯一标识符（EUI-64）格式生成。

就实用性而言，EUI-64 格式是 IPv6 生成接口 ID 最常用的方式。IEEE EUI-64 标准采用接口的 MAC 地址生成 IPv6 接口 ID。MAC 地址只有 48 位，而接口 ID 却要求 64 位。MAC 地址的前 24 位代表厂商 ID，后 24 位代表制造商分配的唯一扩展标识。MAC 地址的第七高位是一个 U/L 位，值为 1 时表示 MAC 地址全局唯一，值为 0 时表示 MAC 地址本地唯一。在 MAC 地址向 EUI-64 格式的转换过程中，在 MAC 地址的前 24 位和后 24 位之间插入了 16 比特的 FFFE，并将 U/L 位的值从 0 变成了 1，这样就生成了一个 64 比特的接口 ID，且接口 ID 的值全局唯一，如图 12-6 所示。

IPv6 支持无状态地址自动配置，无需使用如 DHCP 之类的辅助协议，只要在路由器的接口配置一个 64 位的网络前缀，该网络的每个节点即可获取该 IPv6 的前缀并结合 64 位的接口 ID 形成一个 128 位的 IPv6 全球单播地址。

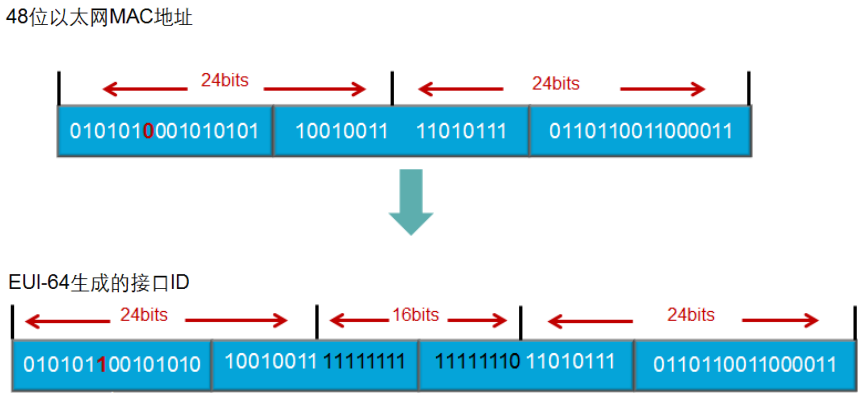
路由器发现功能是 IPv6 地址自动配置功能的基础，其工作过程如图 12-7 所示。

IPv6 地址自动配置主要通过以下两种报文实现：

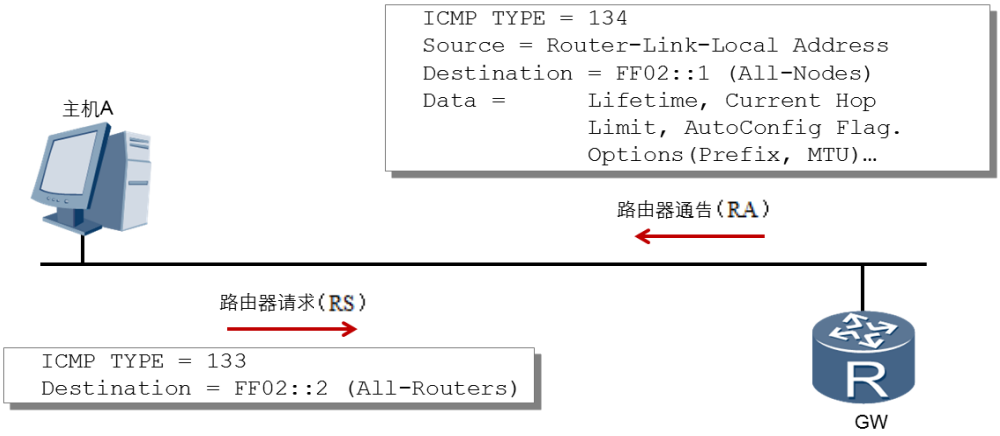
- RA 报文：每台路由器为了让二层网络上的主机和其他路由器知道自己的存在，定期以组

播方式（FF02::1）发送携带网络配置参数的 RA 报文。

- RS 报文：主机接入网络后可以主动发送 RS 报文，RS 报文是由路由器定期发送的，但是如果主机希望能够尽快收到 RS 报文，它可以立刻主动通过组播（FF02::2）发送 RS 报文给路由器。网络上的路由器收到该 RS 报文后会立即向相应的主机单播回应 RA 报文，告知主机该网段的默认路由器和相关配置参数。



▲图 12-6　EUI-64 接口 ID 的形成



▲图 12-7　IPv6 地址自动配置

网关设备在给接口分配 IPv6 单播地址之前会进行重复地址检测（DAD），确认是否有其他的节点使用了该地址，其作用和 IPv4 中的免费 ARP 类似，用于地址分配或主机连接网络时检测重复的主机地址。如果网关收到某个其他站点回应的邻居通告（NA）报文，就证明该地址已被网络使用，节点将不能使用它进行通信，这时网络管理员需要手动为该节点分配另外一个地址。尽管在自动配置的情况下出现地址重复的概率非常小，但进行 DAD 检测是很必要的，尤其是在地址自动配置的时候。

在路由器 GW 上通过无状态自动配置设置接口 Fas0/0 的 IPv6 地址并验证，操作如下：

```
GW(config)#interface fastEthernet 0/0
GW(config-if)#ipv6 address fec0:10::/64 eui-64
GW(config-if)#no shut
GW(config-if)#end
```

```
GW#show interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is DEC21140, address is ca01.0914.0000 (bia ca01.0914.0000)     ---MAC---
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
  ---省略部分输出---
GW#show ipv6 interface fastEthernet 0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::C801:9FF:FE14:0              ---链路本地地址---
  Global unicast address(es):
    FEC0:10::C801:9FF:FE14:0, subnet is FEC0:10::/64 [EUI]                  ---全球单播地址---
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF14:0
  MTU is 1500 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 200 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses.      ---主机地址采用无状态自动配置方式---
```

从上面的输出可以看出，只要在接口分配一个 64 位的网络前缀，该接口就会以 EUI-64 接口 ID 为基础自动形成一个全球单播地址和一个链路本地地址，同样的结论适用于 Windows XP 和 Windows 7 客户端，这些客户端只要在 IPv6 中设置自动获取地址即可，Host3 验证过程如下：

```
C:\Documents and Settings\Administrator>ipconfig
Windows IP Configuration
Ethernet adapter 本地连接：
  Connection-specific DNS Suffix  . :
  Description . . . . . .         :VMware Accelerated AMD PCNet Adapter
  Physical Address. . . . . .     :00-0C-29-7E-E2-57
  Dhcp Enabled. . . . . .         : Yes
  Autoconfiguration Enabled . . .  . : Yes
  Autoconfiguration IP Address. . .  : 169.254.72.231
  Subnet Mask . . . . . .         : 255.255.0.0
  IP Address. . . . . .           : fec0:10::20c:29ff:fe7e:e257%1     ---全球单播地址---
  IP Address. . . . . .           : fe80::20c:29ff:fe7e:e257%5        ---链路本地地址---
  Default Gateway . . . . . .     : fe80::c801:9ff:fe14:0%5           ---网关（GW 的链路本地地址）---
```

注意：自动配置中所有设备的网关地址都使用网关设备的链路本地地址，而不使用网关设备的全球单播地址，在 IPv6 路由中表示下一跳时同样使用邻居的链路本地地址。

Host2 的 IPv6 地址设置成自动获取之后，验证如下：

```
C:\Users\Administrator>ipconfig /all
```

```
Windows IP 配置
以太网适配器 VMware Network Adapter VMnet8：
描述. . . . . . . . . . . . . . .    :VMware Virtual Ethernet Adapter for VMnet8
物理地址. . . . . . . . . . . . .    :00-50-56-C0-00-08
DHCP 已启用 . . . . . . . . . . .    :否
自动配置已启用. . . . . . . . . .    :是
本地站点的 IPv6 地址. . . . . . .    :fec0:10::b087:c227:596f:362f%1（首选）
本地链接 IPv6 地址. . . . . . . .    :fe80::b087:c227:596f:362f%15（首选）
默认网关. . . . . . . . . . . . .    : fe80::c801:9ff:fe14:0%15
```

在 Host1 上设置 IPv6 地址通过无状态自动配置获取，操作和验证过程如下：

```
Host1> ip auto      ---设置 VPCS 通过无状态自动配置获取地址---
Host1> show
NAME    IP/MASK      GATEWAY       MAC                LPORT     RHOST:PORT
Host1   0.0.0.0/0    0.0.0.0       00:50:79:66:68:00  10000     10.8.0.100:10001
        fe80::250:79ff:fe66:6800/64              ---链路本地地址---
        fec0:10::2050:79ff:fe66:6800/64 eui-64   ---全局单播地址---
```

在 GW 上通过 ping 命令分别测试到每个终端设备的连通性，操作过程如下：

```
GW#ping fec0:10::2050:79ff:fe66:6800              ---测试到 Host1 的连通性---
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::2050:79FF:FE66:6800, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/30/44 ms
GW#ping fec0:10::b087:c227:596f:362f               ---测试到 Host2 的连通性---
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::B087:C227:596F:362F, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/28/40 ms
GW#ping fec0:10::20c:29ff:fe7e:e257                ---测试到 Host3 的连通性---
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::20C:29FF:FE7E:E257, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/34/64 ms
GW#
```

注意：和链路本地地址不同，通过全球单播地址进行通信时不需要指明接口。

12.2.4 有状态自动配置

主机使用无状态地址自动配置方案来获取 IPv6 地址时，路由器并不记录主机的 IPv6 地址信息，可管理性差。另外，IPv6 主机无法获取 DNS 服务器地址等网络配置信息，在可用性上也存在一定的缺陷。DHCPv6 是一种有状态地址自动配置协议，在有状态地址配置过程中，DHCPv6 服务器为主机分配一个完整的 IPv6 地址，并提供 DNS 服务器地址等其他配置信息。此外，DHCPv6 服务器还可以对已经分配的 IPv6 地址和客户端进行集中管理。

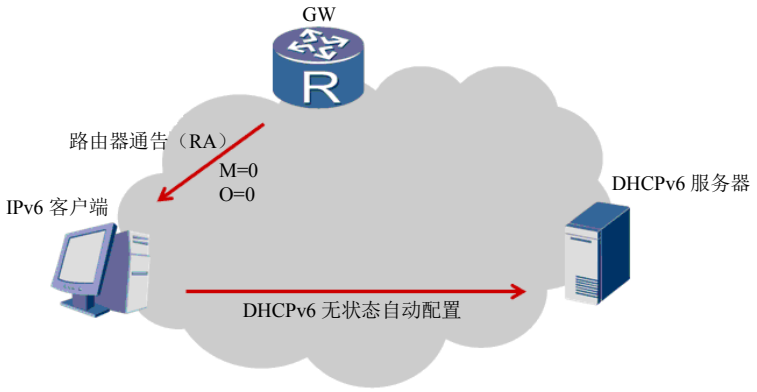
DHCPv6 分配地址时又分为以下两种情况：

- DHCPv6 有状态自动分配：DHCPv6 服务器为客户端分配 IPv6 地址及其他网络配置参数（如 DNS、NIS、SNTP 服务器地址等）。
- DHCPv6 无状态自动分配：主机的 IPv6 地址仍然通过路由通告的方式自动生成，DHCPv6

服务器只分配除 IPv6 地址以外的配置参数。

DHCPv6 客户端在向 DHCPv6 服务器发送请求报文之前会发送 RS 报文，在同一链路范围的路由器接收到此报文后会回复 RA 报文，在 RA 报文中包括管理地址配置标记（M）和有状态配置标记（O）。当 M 取值为 1 时，启用 DHCPv6 有状态地址配置，即 DHCPv6 客户端需要从 DHCPv6 服务器获取 IPv6 地址；取值为 0 时，则启用 IPv6 无状态地址自动分配方案。当 O 取值为 1 时，定义客户端需要通过有状态的 DHCPv6 来获取其他网络配置参数，如 DNS、NIS、SNTP 服务器地址等；取值为 0 时，则启用 IPv6 无状态地址自动分配方案。

默认情况下思科路由器发送的 RA 信息中 M=0/O=0，客户端向服务器请求地址时使用的是无状态自动配置的方式，过程如图 12-8 所示。



▲图 12-8　DHCPv6 无状态自动配置

所以在默认情况下，IPv6 客户端如果设置成自动获取地址，不会获得 DNS 信息，如果让客户端使用 DHCPv6 提供的地址和 DNS，需要在路由器发出的 RA 报文中将 M 和 O 都标识为 1，这可以通过路由的接口配置命令来实现。

DHCP 设备的唯一标识符 DUID（DHCPv6 Unique Identifier）用来标识一台 DHCPv6 服务器或客户端，每个 DHCPv6 节点有且只有一个 DUID，DUID 采用以下两种方式生成：

- 基于链路层地址（LL）：采用链路层地址方式生成 DUID。
- 基于链路层地址与时间组合（LLT）：采用链路层地址和时间组合方式生成 DUID。

在 GW 上配置 DHCPv6 的过程如下：

```
GW(config)#ipv6 dhcp pool IPV6POOL              ---设置 IPv6 的本地地址池，名字为 IPV6POOL---
GW(config-dhcp)#address prefix fec0:10::/64     ---地址池的前缀---
GW(config-dhcp)#dns-server fec0:10::1
GW(config-dhcp)#exit
GW(config)#interface fastEthernet 0/0
GW(config-if)#ipv6 address fec0:10::10/64
GW(config-if)#ipv6 dhcp server IPV6POOL         ---使能 IPv6 的 DHCP 功能并绑定地址池---
GW(config-if)#ipv6 nd managed-config-flag       ---将 RA 报文的 M 位设置为 1---
GW(config-if)#ipv6 nd other-config-flag         ---将 RA 报文的 O 位设置为 1---
GW(config-if)#end
```

Host2 的 IPv6 地址设置成自动获取之后进行验证，信息显示如下：

```
C:\Users\Administrator>ipconfig /all
以太网适配器  VMware Network Adapter VMnet8：
```

```
连接特定的 DNS 后缀 . . . . . :
描述. . . . . . . . . . . . . .  :VMware Virtual Ethernet Adapter for VMnet8
物理地址. . . . . . . . . . . .  :00-50-56-C0-00-08
DHCP 已启用 . . . . . . . . . .  :是
自动配置已启用. . . . . . . . .  :是
本地站点的 IPv6 地址. . . . . .  :fec0:10::3deb:cbed:3ed4:ca55%1（首选）
获得租约的时间 . . . . . . . .  :2016 年 7 月 8 日 15:26:14
租约过期的时间 . . . . . . . .  :2016 年 7 月 10 日 15:26:13
本地链接 IPv6 地址. . . . . . .  :fe80::b087:c227:596f:362f%15（首选）
IPv4 地址 . . . . . . . . . . .  :10.8.0.100（首选）
子网掩码 . . . . . . . . . . . .  :255.255.255.0
默认网关. . . . . . . . . . . .  :fe80::c801:9ff:fe14:0%15
DHCPv6 IAID . . . . . . . . . . .  :385896534
DHCPv6 客户端 DUID . . . . . .  :00-01-00-01-1B-2A-B1-BA-44-87-FC-4D-F9-B2
DNS 服务器 . . . . . . . . . . .  :fec0:10::1%1
TCP/IP 上的 NetBIOS . . . . . .  :已启用
```

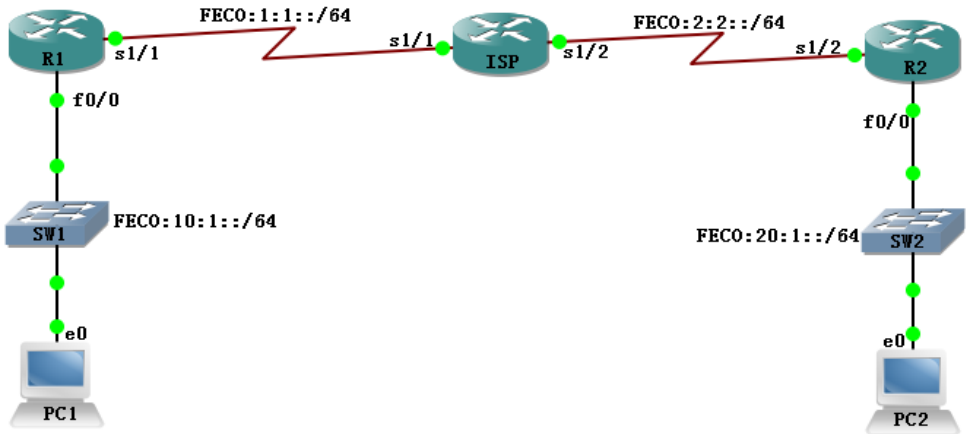
注意：由于 Windows XP 本身对 IPv6 协议支持不够完善，所以 Host3 设置成通过 DHCPv6 获取地址时可能无法获得 DNS 信息，VPCS 也不支持有状态自动配置。

从 GW 通过 ping 命令测试到 Host2 设备的连通性，操作过程如下：

```
GW#ping fec0:10::3deb:cbed:3ed4:ca55
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:10::3DEB:CBED:3ED4:CA55, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/13/24 ms
```

12.3 配置 IPv6 路由

和 IPv4 的网络一样，为了使数据包能在 IPv6 网络中正常地传输，需要部署 IPv6 的路由。前面讨论过的大多数路由协议在版本升级后都能在 IPv6 的网络中使用，虽然名称有了相应的变化，但基本的部署思路和工作原理都是相似的。本节以图 12-9 为例介绍各种情况下 IPv6 路由的实施方法，图中的所有设备均已启用 IPv6 协议。



▲图 12-9 IPv6 路由

注意：所有路由器的串口 IPv6 地址采用手动的方式分配，路由器和主机的以太口采用无状态自动配置的方式分配，PC1 和 PC2 使用 VPCS，各设备接口 IPv6 地址信息如下：

```
R1#show ipv6 interface brief
FastEthernet0/0                [up/up]
    FE80::C801:16FF:FE6C:8
    FEC0:10:1:0:C801:16FF:FE6C:8
…
Serial1/1                      [up/up]
    FE80::C801:16FF:FE6C:8
    FEC0:1:1::1
…
R1#

R2#show ipv6 interface brief
FastEthernet0/0                [up/up]
    FE80::C802:17FF:FE74:8
    FEC0:20:1:0:C802:17FF:FE74:8
…
Serial1/2                      [up/up]
    FE80::C802:17FF:FE74:8
    FEC0:2:2::2
Serial1/3                      [administratively down/down]
R2#

ISP#show ipv6 interface brief
…
Serial1/1                      [up/up]
    FE80::C803:16FF:FE64:0
    FEC0:1:1::3
Serial1/2                      [up/up]
    FE80::C803:16FF:FE64:0
    FEC0:2:2::3
Serial1/3                      [administratively down/down]
ISP#

PC1> show
NAME     IP/MASK     GATEWAY         MAC               LPORT   RHOST:PORT
PC1      0.0.0.0/0   0.0.0.0         00:50:79:66:68:00 10002   127.0.0.1:10003
         fe80::250:79ff:fe66:6800/64
         fec0:10:1:0:2050:79ff:fe66:6800/64 eui-64

PC2> show
NAME     IP/MASK     GATEWAY         MAC               LPORT   RHOST:PORT
PC2      0.0.0.0/0   0.0.0.0         00:50:79:66:68:01 10001   127.0.0.1:10000
         fe80::250:79ff:fe66:6801/64
         fec0:20:1:0:2050:79ff:fe66:6801/64 eui-64
```

12.3.1 配置 IPv6 静态路由

参照拓扑如图 12-9 所示，在 ISP 分别设置两条到 R1 和 R2 以太口的静态路由，配置过程如下：

```
ISP(config)#ipv6 route fec0:10:1::/64 serial 1/1 FE80::C801:16FF:FE6C:8
ISP(config)#ipv6 route fec0:20:1::/64 serial 1/2 FE80::C802:17FF:FE74:8
```

注意：在设置下一条地址时建议使用邻居设备的链路本地地址，这样做的好处是当邻居或目标网络的全球单播地址发生变动时，路由信息可以不做对应的调整。

验证 ISP 的 IPv6 路由表，操作过程如下：

```
ISP#show ipv6 route
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
L   FE80::/10 [0/0]                ---链路本地地址段，每一个接口启动 IPv6 后都有一个链路本地地址---
      via ::, Null0
C   FEC0:1:1::/64 [0/0]            ---直连的网段---
      via ::, Serial1/1
L   FEC0:1:1::3/128 [0/0]          ---接口的地址，128 位---
      via ::, Serial1/1
C   FEC0:2:2::/64 [0/0]
      via ::, Serial1/2
L   FEC0:2:2::3/128 [0/0]
      via ::, Serial1/2
S   FEC0:10:1::/64 [1/0]           ---静态路由---
      via FE80::C801:16FF:FE6C:8, Serial1/1
S   FEC0:20:1::/64 [1/0]
      via FE80::C802:17FF:FE74:8, Serial1/2
L   FF00::/8 [0/0]                 ---组播地址段，每一个接口启动 IPv6 后都会加入一些组播组完成相应的功能---
      via ::, Null0
ISP#
```

在 R1 和 R2 分别设置两条默认路由，设置过程如下：

```
R1(config)#ipv6 route ::/0 serial 1/1 FE80::C803:16FF:FE64:0
R2(config)#ipv6 route ::/0 serial 1/2 FE80::C803:16FF:FE64:0
```

以 R1 为例验证设置的静态路由或默认路由信息，操作过程如下：

```
R1#show ipv6 route static
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S   ::/0 [1/0]
      via FE80::C803:16FF:FE64:0, Serial1/1
R1#
```

分别使用 ping 和 trace 命令测试 PC1 到 PC2 的连通性，操作过程如下：

```
PC1> ping fec0:20:1:0:2050:79ff:fe66:6801
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=1 ttl=58 time=67.004 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=2 ttl=58 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=3 ttl=58 time=39.002 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=4 ttl=58 time=39.002 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=5 ttl=58 time=39.002 ms

PC1> trace fec0:20:1:0:2050:79ff:fe66:6801
trace to fec0:20:1:0:2050:79ff:fe66:6801, 64 hops max
 1 fec0:10:1:0:c801:16ff:fe6c:8          5.000 ms    9.001 ms    9.000 ms
 2 fec0:1:1::3                          19.001 ms   20.002 ms   21.001 ms
 3 fec0:2:2::2                          29.002 ms   30.001 ms   39.003 ms
 4 fec0:20:1:0:2050:79ff:fe66:6801      49.002 ms   50.003 ms   39.003 ms
PC1>
```

12.3.2 配置 RIPng

RIPng 是专为 IPv6 网络设计的距离矢量的路由协议，也称为下一代 RIP，主要特性与 RIPv2 一样，如最大跳数为 15，使用水平分割、毒性逆转和其他的环路避免机制，但它现在使用 UDP 端口 521。RIPng 仍然使用组播来发送更新信息，但在 IPv6 中，它使用 FF02::9 为传输地址，并且在路由表中使用链路本地地址来跟踪下一跳地址。

要在路由器上启用 RIPng 路由，首先使用 ipv6 router rip *name* 全局配置命令启动 RIPng 进程，然后在接口配置模式下使用 ipv6 rip *name* enable 命令将接口加入该 RIPng 进程，name 参数表示 RIP 进程，所有的 IPv6 路由协议都使用接口下的命令将该接口所在的网段加入 IPv6 的路由进程。

参照拓扑如图 12-9 所示，在网络中部署 RIPng 路由协议，配置过程如下：

```
R1(config)#ipv6 router rip LAB
R1(config-rtr)#exit
R1(config)#interface fastEthernet 0/0
R1(config-if)#ipv6 rip LAB enable
R1(config-if)#exit
R1(config)#interface serial 1/1
R1(config-if)#ipv6 rip LAB enable
R1(config-if)#exit
ISP(config)#ipv6 router rip LAB
ISP(config-rtr)#exit
ISP(config)#interface serial 1/1
ISP(config-if)#ipv6 rip LAB enable
ISP(config-if)#exit
ISP(config)#interface serial 1/2
ISP(config-if)#ipv6 rip LAB enable
ISP(config-if)#end
R2(config)#ipv6 router rip LAB
R2(config-rtr)#exit
R2(config)#interface fastEthernet 0/0
R2(config-if)#ipv6 rip LAB enable
R2(config-if)#exit
```

```
R2(config)#interface serial 1/2
R2(config-if)#ipv6 rip LAB enable
R2(config-if)#exit
```

注意：上面的配置中，LAB 表示 RIPng 的路由进程，不同的路由器可以使用不同的进程标识，建议自定义的进程名称、策略名称等都使用大写字母，以便和命令关键字区分开。

以 ISP 为例验证 IPv6 的路由配置和路由表，操作过程如下：

```
ISP#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "rip LAB"
  Interfaces:
    Serial1/2
    Serial1/1
  Redistribution:
    None
ISP#show ipv6 route rip
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
R     FEC0:10:1::/64 [120/2]
       via FE80::C801:16FF:FE6C:8, Serial1/1
R     FEC0:20:1::/64 [120/2]
       via FE80::C802:17FF:FE74:8, Serial1/2
ISP#
```

使用 ping 命令测试 PC1 到 PC2 的连通性，操作过程如下：

```
PC1> ping fec0:20:1:0:2050:79ff:fe66:6801
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=1 ttl=58 time=93.005 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=2 ttl=58 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=3 ttl=58 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=4 ttl=58 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=5 ttl=58 time=49.002 ms
PC1>
```

12.3.3 配置 OSPFv3

OSPFv3 是运行在 IPv6 网络的 OSPF 协议。运行 OSPFv3 的路由器使用物理接口的链路本地单播地址作为源地址来发送 OSPF 报文。相同链路上的路由器互相学习与之相连的其他路由器的链路本地地址，并在报文转发的过程中将这些地址当成下一跳信息使用。

Router ID 在 OSPFv3 中也用于标识路由器。与 OSPFv2 的 Router ID 不同，OSPFv3 的 Router ID 必须手工配置；如果没有手工配置 Router ID，OSPFv3 将无法正常运行。OSPFv3 在广播型网络和 NBMA 网络中选举 DR 和 BDR 的过程与 OSPFv2 相似。IPv6 使用组播地址 FF02::6 表示 AllDRouters，

而 OSPFv2 中使用的是组播地址 224.0.0.6。

OSPFv3 是基于链路而不是网段的。在配置 OSPFv3 时，不需要考虑路由器的接口是否配置在同一网段，只要路由器的接口连接在同一链路上，就可以不配置 IPv6 全局地址而直接建立联系。这一变化影响了 OSPFv3 协议报文的接收、Hello 报文的内容以及网络 LSA 的内容。

OSPFv3 的配置步骤和 RIPng 基本一致，参照拓扑如图 12-9 所示，将路由器之间互联的串口链路划分到 area0、R1 的以太口划分到 area1、R2 的以太口划分到 area2，R1、R2、R3 的 RID 分别为 1.1.1.1、2.2.2.2、3.3.3.3，在网络中部署 RIPng 路由协议，配置过程如下：

```
R1(config)#ipv6 router ospf 10                  ---10 是 OSPF 的进程 ID---
R1(config-rtr)#router-id 1.1.1.1                ---设置 OSPF 的 RID---
R1(config-rtr)#exit
R1(config)#interface fastEthernet 0/0
R1(config-if)#ipv6 ospf 10 area 1               ---将相应的接口加入 OSPF 进程并标注区域---
R1(config-if)#exit
R1(config)#interface serial 1/1
R1(config-if)#ipv6 ospf 10 area 0
R1(config-if)#exit
ISP(config)#ipv6 router ospf 10
ISP(config-rtr)#router-id 3.3.3.3
ISP(config-rtr)#exit
ISP(config)#interface serial 1/1
ISP(config-if)#ipv6 ospf 10 area 0
ISP(config-if)#exit
ISP(config)#interface serial 1/2
ISP(config-if)#ipv6 ospf 10 area 0
ISP(config-if)#exit
R2(config)#ipv6 router ospf 10
R2(config-rtr)#router-id 2.2.2.2
R2(config-rtr)#exit
R2(config)#interface serial 1/2
R2(config-if)#ipv6 ospf 10 area 0
R2(config-if)#exit
R2(config)#interface fastEthernet 0/0
R2(config-if)#ipv6 ospf 10 area 2
R2(config-if)#exit
```

以 ISP 为例验证 IPv6 的路由配置、OSPFv3 邻居关系和路由表，操作过程如下：

```
ISP#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "static"
IPv6 Routing Protocol is "ospf 10"
  Interfaces (Area 0):
    Serial1/2
    Serial1/1
  Redistribution:
    None
ISP#show ipv6 ospf neighbor
Neighbor ID     Pri   State           Dead Time   Interface ID    Interface
2.2.2.2           1   FULL/  -        00:00:37    8               Serial1/2
1.1.1.1           1   FULL/  -        00:00:39    7               Serial1/1
```

```
ISP#show ipv6 route ospf
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI   FEC0:10:1::/64 [110/65]
       via FE80::C801:16FF:FE6C:8, Serial1/1
OI   FEC0:20:1::/64 [110/65]
       via FE80::C802:17FF:FE74:8, Serial1/2
ISP#
```

使用 ping 命令测试 PC1 到 PC2 的连通性，操作过程如下：

```
PC1> ping fec0:20:1:0:2050:79ff:fe66:6801
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=1 ttl=58 time=124.800 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=2 ttl=58 time=93.600 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=3 ttl=58 time=93.601 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=4 ttl=58 time=93.600 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=5 ttl=58 time=93.600 ms
PC1>
```

12.3.4 配置 EIGRPv6

EIGRPv6 是在 IPv6 网络中应用的 EIGRP 协议，EIGRP 的大多数特性在 EIGRPv6 中都保留了，比如它仍然是高级距离矢量路由协议，并且有一些链路状态路由协议的特征。邻居发现的过程仍然使用 hello 来进行，它仍然使用可靠的传输协议来提供可靠的通信，并使用弥散更新算法（DUAL）实现无环路的快速收敛。EIGRPv6 使用组播地址 FF02::A 传输来发送 hello 包和更新信息。

在 EIGRPv6 中，仍然使用路由器配置模式来启用路由协议，不同的是路由进程必须指定 RID 并用命令 no shutdown 激活，其他的配置方式和思路与 RIPng 一致。参照拓扑如图 12-9 所示，R1、R2、R3 的 RID 分别为 1.1.1.1、2.2.2.2、3.3.3.3，在网络中部署 EIGRPv6 路由协议，配置过程如下：

```
R1(config)#ipv6 router eigrp 10                  ---10 是 AS 号---
R1(config-rtr)#eigrp router-id 1.1.1.1           ---指明 EIGRP 的 RID---
R1(config-rtr)#no shutdown                       ---激活 EIGRPv6 进程---
R1(config-rtr)#exit
R1(config)#interface serial 1/1
R1(config-if)#ipv6 eigrp 10
R1(config-if)#exit
R1(config)#interface fastEthernet 0/0
R1(config-if)#ipv6 eigrp 10
R1(config-if)#exit
ISP(config)#ipv6 router eigrp 10
ISP(config-rtr)#eigrp router-id 3.3.3.3
ISP(config-rtr)#no shutdown
ISP(config-rtr)#exit
ISP(config)#interface serial 1/1
ISP(config-if)#ipv6 eigrp 10
ISP(config-if)#exit
ISP(config)#interface serial 1/2
```

```
ISP(config-if)#ipv6 eigrp 10
ISP(config-if)#exit
R2(config)#ipv6 router eigrp 10
R2(config-rtr)#eigrp router-id 2.2.2.2
R2(config-rtr)#no shutdown
R2(config-rtr)#exit
R2(config)#interface serial 1/2
R2(config-if)#ipv6 eigrp 10
R2(config-if)#exit
R2(config)#interface fastEthernet 0/0
R2(config-if)#ipv6 eigrp 10
R2(config-if)#exit
```

以 ISP 为例验证 IPv6 的路由配置、EIGRPv6 邻居关系和路由表，操作过程如下：

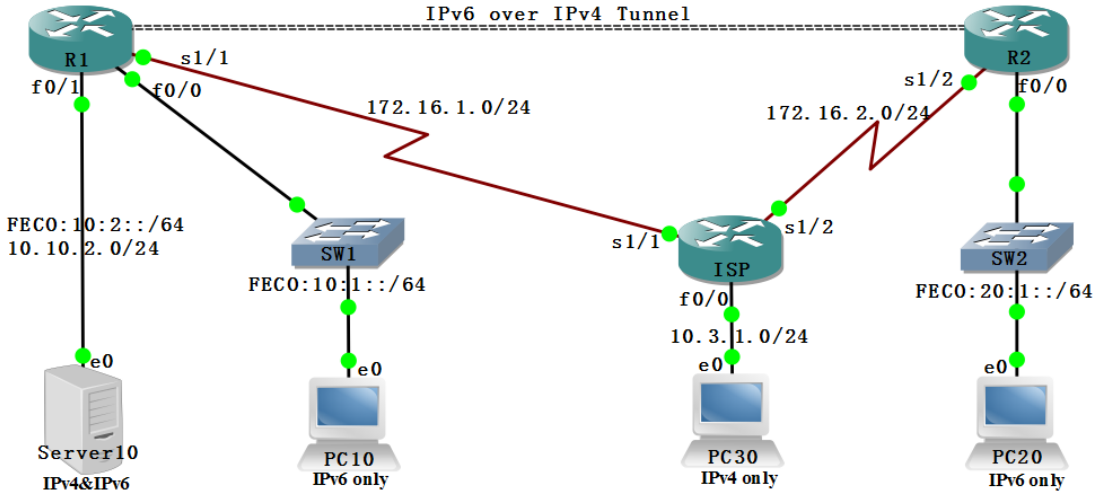
```
ISP#show ipv6 protocols
IPv6 Routing Protocol is "connected"
IPv6 Routing Protocol is "ND"
IPv6 Routing Protocol is "eigrp 10"
EIGRP-IPv6 Protocol for AS(10)
  Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  NSF-aware route hold timer is 240
  Router-ID: 3.3.3.3
  Topology : 0 (base)
    Active Timer: 3 min
    Distance: internal 90 external 170
    Maximum path: 16
    Maximum hopcount 100
    Maximum metric variance 1
  Interfaces:
    Serial1/1
    Serial1/2
  Redistribution:
    None
ISP#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(10)
H   Address                 Interface   Hold Uptime   SRTT   RTO   Q   Seq
                                        (sec)         (ms)       Cnt Num
1   Link-local address:     Se1/2       11 00:07:15   100    600   0   4
    FE80::C802:17FF:FE74:8
0   Link-local address:     Se1/1       11 00:08:52   57     342   0   5
    FE80::C801:16FF:FE6C:8
```

使用 ping 命令测试 PC1 到 PC2 的连通性，操作过程如下：

```
PC1> ping fec0:20:1:0:2050:79ff:fe66:6801
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=1 ttl=58 time=124.800 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=2 ttl=58 time=93.600 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=3 ttl=58 time=93.601 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=4 ttl=58 time=93.600 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=5 ttl=58 time=93.600 ms
PC1>
```

12.4 IPv6 和 IPv4 共存

在目前以 IPv4 为基础的网络技术如此成熟的情况下，不可能马上抛开原有的 IPv4 网络来创建 IPv6 网络，只能通过分步实施的方法来逐步过渡。因此，在今后相当长的一段时间内，IPv6 网络将和 IPv4 网络共存。如何以合理的代价逐步地将 IPv4 网络过渡到 IPv6、解决好 IPv4 与 IPv6 互相共存将是我们需要迫切考虑的。针对以上问题，目前提出了三种主要的过渡技术：双协议栈（DualStack）、隧道技术（Tunnel）、地址协议转换（NAT-PT）。当然，这些过渡技术都不是普遍适用的，每一种技术都只适用于某种或几种特定的网络情况，在实际应用时需要综合考虑各方面的现实情况，然后选择合适的转换机制进行设计和实施。本节以图 12-10 为例介绍各种情况下 IPv4 到 IPv6 的过渡方案。



▲图 12-10　IPv4 到 IPv6 的过渡方案

12.4.1 双协议栈技术

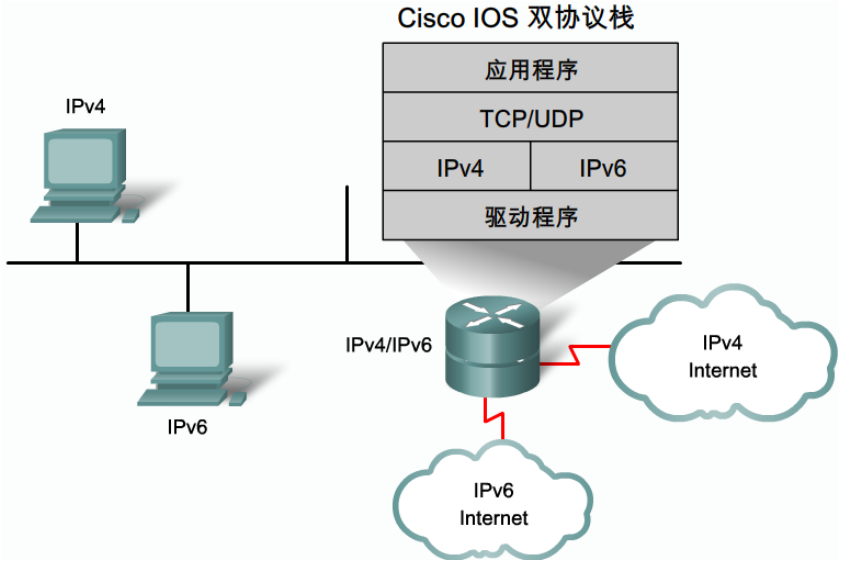
双协议栈是指在单个节点同时支持 IPv4 和 IPv6 两种协议。由于 IPv6 和 IPv4 是功能相近的网络层协议，两者都基于相同的物理平台，而且加载于其上的传输层协议 TCP 和 UDP 也没有区别，所以可以在一台主机上同时支持 IPv4 协议和 IPv6 协议。

双协议栈技术的工作原理是一台主机同时支持 IPv6 和 IPv4 两种协议，该主机既能与支持 IPv4 协议的主机通信，又能与支持 IPv6 协议的主机通信。双协议栈是其他 IPv4 或 IPv6 互通技术的基础，当 IPv6 可用时，双协议栈节点将优先使用 IPv6，双协议栈主机的协议结构如表 12-1 所示。

▲表 12-1　双协议栈主机的协议结构

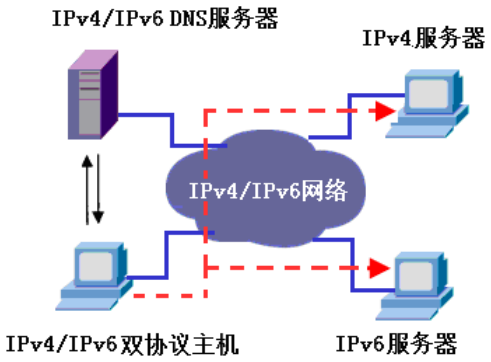
应用程序	
TCP/UDP 协议	
IPv6 协议	IPv4 协议
接入网络	

同样，双栈主机相应的网关也支持双栈协议，如图 12-11 所示。



▲图 12-11　双栈路由器

双协议栈主机在通信时首先通过支持双协议的 DNS 服务器查询与目的主机名对应的 IP 地址，然后根据指定的 IPv6 或 IPv4 地址开始通信，双协议栈通信方式如图 12-12 所示。



▲图 12-12　双协议栈通信示意图

将 IPv4 应用程序过渡到 IPv6 的经验表明，对于大多数应用程序，只需要对源代码内的某些局部地方做极少量的修改，该应用程序可以逐步升级到 IPv6。Windows Server 2008 默认是双协议栈，它的 DNS 服务器支持 IPv4 和 IPv6 的名称解析。

参照拓扑如图 12-10 所示，我们可以将路由器 R1 的 Fas0/1 和服务器 Server10 的 E0 口设置为双栈模式，配置和验证过程如下：

```
R1(config)#ipv6 unicast-routing
R1(config)#interface fastEthernet 0/1
R1(config-if)#ip address 10.10.2.1 255.255.255.0
R1(config-if)#ipv6 address fec0:10:2::/64 eui-64
R1(config-if)#end
```

```
R1#show running-config interface fastEthernet 0/1
Building configuration...
Current configuration : 130 bytes
!
interface FastEthernet0/1
ip address 10.10.2.1 255.255.255.0
ipv6 address FEC0:10:2::/64 eui-64
end
R1#
R1#show ipv6 interface brief
FastEthernet0/1                [up/up]
    FE80::C801:16FF:FE6C:6
    FEC0:10:2:0:C801:16FF:FE6C:6

Ser_10> ip 10.10.2.10/24 10.10.2.1                    ---设置节点的 IPv4 地址和网关---
Checking for duplicate address...
PC1 : 10.10.2.10 255.255.255.0 gateway 10.10.2.1
Ser_10> ip auto                                       ---设置节点通过自动的方式获得 IPv6 地址---
GLOBAL SCOPE         : fec0:10:2:0:2050:79ff:fe66:6803/64
ROUTER LINK-LAYER : ca:01:16:6c:00:06
Ser_10> show                                          ---验证节点的 IPv4 和 IPv6 地址信息---
NAME      IP/MASK             GATEWAY        MAC                LPORT   RHOST:PORT
Ser_10    10.10.2.10/24       10.10.2.1      00:50:79:66:68:03  10007   127.0.0.1:10006
          fe80::250:79ff:fe66:6803/64
          fec0:10:2:0:2050:79ff:fe66:6803/64
```

注意：本例中 Server10 使用 VPCS 代替，下面验证双协议栈的连通性，过程如下：

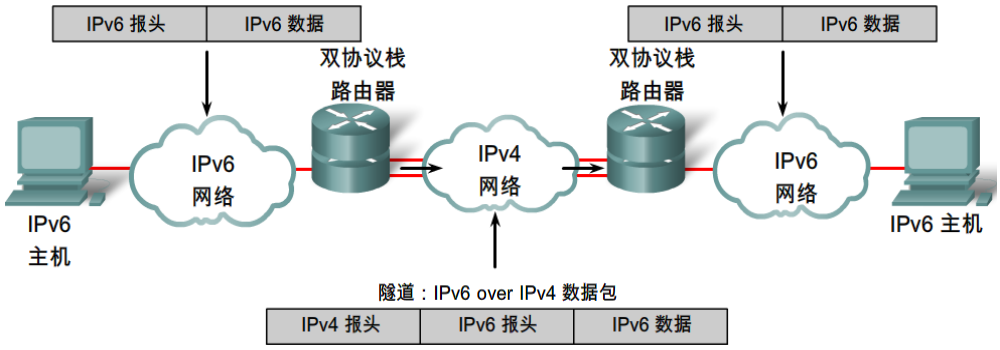
```
Ser_10> ping 10.10.2.1
84 bytes from 10.10.2.1 icmp_seq=1 ttl=255 time=19.001 ms
84 bytes from 10.10.2.1 icmp_seq=2 ttl=255 time=5.001 ms
84 bytes from 10.10.2.1 icmp_seq=3 ttl=255 time=9.000 ms
84 bytes from 10.10.2.1 icmp_seq=4 ttl=255 time=9.000 ms
84 bytes from 10.10.2.1 icmp_seq=5 ttl=255 time=9.001 ms
Ser_10> ping FEC0:10:2:0:C801:16FF:FE6C:6
FEC0:10:2:0:C801:16FF:FE6C:6 icmp6_seq=1 ttl=64 time=21.001 ms
FEC0:10:2:0:C801:16FF:FE6C:6 icmp6_seq=2 ttl=64 time=9.000 ms
FEC0:10:2:0:C801:16FF:FE6C:6 icmp6_seq=3 ttl=64 time=9.001 ms
FEC0:10:2:0:C801:16FF:FE6C:6 icmp6_seq=4 ttl=64 time=9.000 ms
FEC0:10:2:0:C801:16FF:FE6C:6 icmp6_seq=5 ttl=64 time=9.000 ms
Ser_10>
```

12.4.2 手工隧道

随着 IPv6 网络的发展，将会出现许多局部的 IPv6 网络，但是这些 IPv6 网络被运行 IPv4 协议的主干网络所分隔开来。IPv6 网络就像是处于 IPv4“海洋”中的“孤岛”，为了使这些 IPv6“孤岛”互通，可以使用 IPv6 over IPv4 的隧道技术，基本的思路是将 IPv6 的数据包封装到 IPv4 的数据包中再通过网络传递。

手工隧道模拟两个 IPv6 域之间穿越 IPv4 骨干的永久性链路，在隧道两端创建逻辑的隧道接口，并手工给每个隧道接口配置 IPv6 地址。工作时路由器将 IPv6 的数据包封装入 IPv4 中，IPv4 数据

包的源地址和目的地址分别是隧道入口和出口的 IPv4 地址。在隧道的出口处再将 IPv6 数据包取出转发给目的站点。隧道技术只要求隧道两端的节点（路由器）都支持 IPv4 或 IPv6 两种协议，而无需将中间网络转换成 IPv6。当使用 IPv4 封装 IPv6 数据包时，IPv4 的协议类型为 41，其通信方式如图 12-13 所示。



▲图 12-13　IPv6 over IPv4 手工隧道

参照拓扑如图 12-10 所示，我们可以在路由器 R1 和 R2 之间做 IPv6 over IPv4 手工隧道，前提需要保证 R1 的 Serial1/1（IP:172.16.1.1）和 R2 的 Serial1/2（IP:172.16.12.2）接口之间可以相互通信。R1、R2、ISP 串口地址的配置不再详细介绍，隧道的配置和验证过程如下：

```
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 1/1
R1(config)#ipv6 unicast-routing
R1(config)#interface tunnel 12                  ---配置隧道接口---
R1(config-if)#ipv6 address fec0:12:12::1/64     ---隧道的 IPv6 地址---
R1(config-if)#tunnel source 172.16.1.1          ---隧道的源---
R1(config-if)#tunnel destination 172.16.2.2     ---隧道的目的地---
R1(config-if)#tunnel mode ipv6ip                ---隧道的模式---
R1(config-if)#end
R1#show ipv6 interface brief
---省略部分输出---
Tunnel12                    [up/up]
    FE80::AC10:101
    FEC0:12:12::1
R1#
R2(config)#ip route 0.0.0.0 0.0.0.0 serial 1/2
R2(config)#ipv6 unicast-routing
R2(config)#interface tunnel 12
R2(config-if)#ipv6 address fec0:12:12::2/64
R2(config-if)#tunnel source 172.16.2.2
R2(config-if)#tunnel destination 172.16.1.1
R2(config-if)#tunnel mode ipv6ip
R2(config-if)#end
R2#show ipv6 interface brief
---省略部分输出---
Tunnel12                    [up/up]
    FE80::AC10:202
    FEC0:12:12::2
R2#
```

在 R1 上通过 ping 命令测试隧道是否成功建立，操作过程如下：

```
R1#ping fec0:12:12::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:1:1::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 16/26/32 ms
R1#
```

如需 PC10 和 PC20 这两个 IPv6 主机直接通过建立的隧道通信，在网络中部署路由即可。

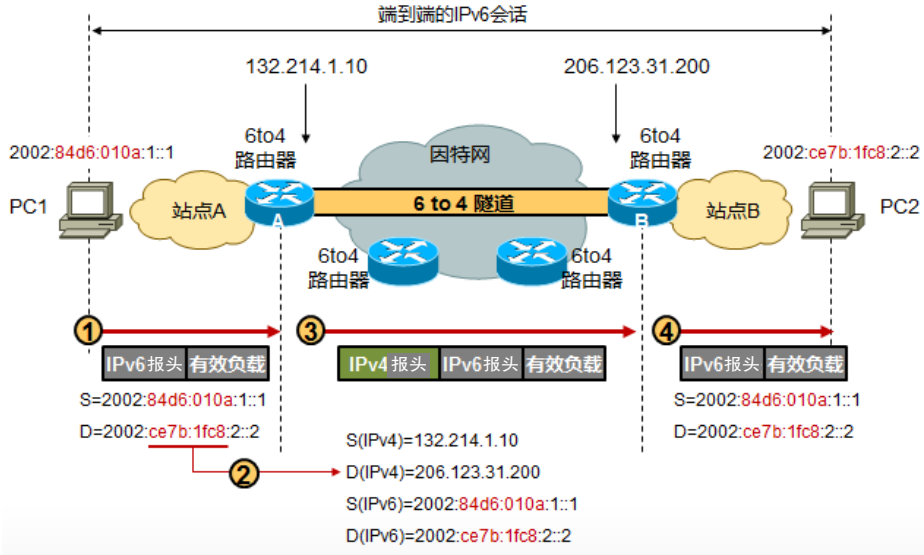
12.4.3 6to4 隧道

手工隧道虽然简单，但它需要为每对路由器都配置一条独立的隧道，如果企业路由器的数量比较多，手工配置的管理成本将会非常大，因此这种方式难以满足大量隧道扩展性的需求。IETF 定义了一种称为 6to4 的隧道机制，这是一种自动构造隧道的方式，它的好处在于只需要一个全球唯一的 IPv4 地址便可以使得整个站点获得 IPv6 的连接。

6to4 隧道不是点到点的，而是点到多点的。国际地址分配机构 IANA 专门为 6to4 隧道分配了一个永久性的 IPv6 的地址范围，对应的网络前缀是 2002::/16，称为 6to4 地址，其结构如图 12-14 所示。

▲图 12-14　6to4 地址结构

6to4 地址的网络前缀有 64 位，前 48 位（2002:a.b.c.d）由分配给路由器上的 IPv4 地址决定，用户不能改变；后 16 位 SLA ID 是子网标识，由用户自己定义，PortID 是接口标识。利用 6to4 地址，隧道的 IPv4 地址可以从相应的 IPv6 地址的 48 比特前缀中自动提取出来，假设隧道的 IPv4 地址为 136.214.1.10，转换成十六进制 84d6:010a，则 6to4 地址前缀为 2002: 84d6:010a::/48，其隧道的数据交互如图 12-15 所示。



▲图 12-15　6to4 隧道数据通信过程

参照拓扑如图 12-10 所示，我们可以在路由器 R1 和 R2 之间做 6to4 隧道，前提需要保证 R1 的 Serial1/1 和 R2 的 Serial1/2 接口之间可以相互通信，配置过程如下：

```
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 1/1
R1(config)#ipv6 unicast-routing
R1(config)#interface tunnel 12
R1(config-if)#ipv6 address 2002:ac10:101:12::/128
R1(config-if)#tunnel source 172.16.1.1      ---172.16.1.1 对应的十六进制值为 ac10:0101---
R1(config-if)#tunnel mode ipv6ip 6to4
R2(config)#ip route 0.0.0.0 0.0.0.0 serial 1/2
R2(config)#ipv6 unicast-routing
R2(config)#interface tunnel 12
R2(config-if)#ipv6 address 2002:ac10:202:12::/128
R2(config-if)#tunnel source 172.16.2.2
R2(config-if)#tunnel mode ipv6ip 6to4
```

由于隧道连接的每个节点地址都属于不同的子网，所以需要在每个路由器上设置到 6to4 网络的路由和到远端 IPv6 网络的路由：

```
R1(config)#ipv6 route 2002::/16 tunnel 12                    ---设置 6to4 网络的静态路由---
R1(config)#ipv6 route fec0:20:1::/64 2002:ac10:202:12::    ---下一跳为 R2 的 Tunnel 地址---
R2(config)#ipv6 route 2002::/16 tunnel 12                    ---设置 6to4 网络的静态路由---
R2(config)#ipv6 route fec0:10:1::/64 2002:ac10:101:12::    ---下一跳为 R1 的 Tunnel 地址---
```

验证 R1 的路由表并使用 ping 命令从 Host10 测试到 Host20 的连通性：

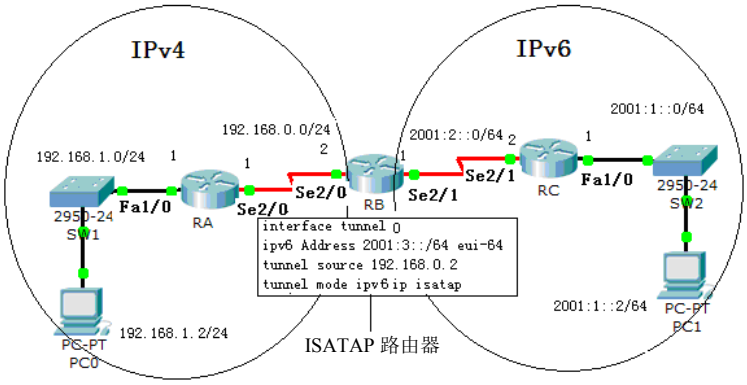
```
R1#show ipv6 route static
IPv6 Routing Table - 9 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S     2002::/16 [1/0]
       via ::, tunnel 12
S     FEC0:20:1::/64 [1/0]
       via 2002:AC10:202:12::
R1#

PC10> ping fec0:20:1:0:2050:79ff:fe66:6801
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=1 ttl=60 time=73.004 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=2 ttl=60 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=3 ttl=60 time=39.002 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=4 ttl=60 time=49.003 ms
fec0:20:1:0:2050:79ff:fe66:6801 icmp6_seq=5 ttl=60 time=49.003 ms
PC10>
```

12.4.4 ISATAP 隧道

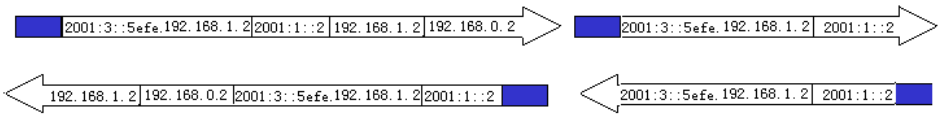
ISATAP（站内自动隧道寻址协议）是一种地址分配和主机到主机、主机到路由器和路由器到主机的自动隧道技术，它为 IPv6 主机之间提供了跨越 IPv4 内部网络的单播 IPv6 连通性，ISATAP 一般用于 IPv4 网络中的 IPv6/IPv4 节点间的通信。

ISATAP 是一种非常容易部署和使用的 IPv6 过渡机制。部署时首先需要终端 PC 支持双栈协议，

然后需要有一台支持 ISATAP 的路由器，该路由器可以在网络中的任何位置，只要终端 PC 能够通过 IPv4 的网络与它通信即可。当使用 PC 访问 IPv6 资源时，可以与 ISATAP 路由器建立起 ISATAP 隧道，PC 根据 ISATAP 路由器下发的 IPv6 前缀构造自己的 IPv6 地址，并且将这台 ISATAP 路由器设置为自己的 IPv6 默认网关，如此一来，后续的这台主机就能够通过这台 ISATAP 路由器去访问 IPv6 的资源，ISATAP 通信过程如图 12-16 所示。



PC0 到 PC1 使用 PC0 的 IPv4 地址作为源 IP 地址，ISATAP 路由器的 IPv4 地址作为目标 IP 地址去掉 IPv4 的数据报头进入 IPv6 网络

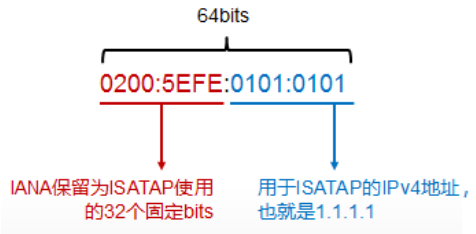


从 IPv6 的数据包提取出 IPv4 的地址作为目标地址，Tunnel Source 地址作为 IPv4 的源地址将 IPv6 的数据包封装到 IPv4 的数据包中进入 IPv4 的网络

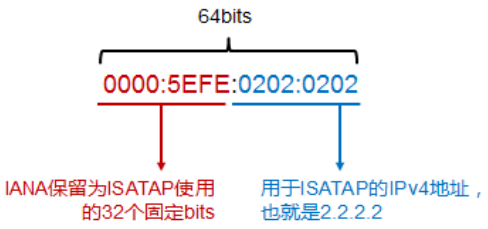
▲图 12-16　ISATAP 隧道

ISATAP 在主机上启用后，会产生一个 ISATAP 虚拟网卡，该虚拟网卡会产生一个 64bits 的特殊接口标识，有点类似 EUI-64，但是产生机制不同，它是由专为 ISATAP 保留的 32 位的 0200:5EFE 加上主机上配置的 IPv4 地址构成，如图 12-17 所示，其中假设 ISATAP 主机配置的 IPv4 地址为 1.1.1.1。

另一方面，在路由器上部署 ISATAP 后，路由器也会产生一个 tunnel 接口，用于响应 ISATAP 主机的隧道建立请求，这个 tunnel 接口同样会产生接口标识。地址的格式是 IANA 保留给 ISATAP 的 32bits 的 0000:5EFE 后追加 32bits 的 IPv4 地址，如图 12-18 所示，其中假设 ISATAP 路由器配置的 IPv4 地址是 2.2.2.2。



▲图 12-17　ISATAP 主机接口标识



▲图 12-18　ISATAP 路由器接口标识

参照拓扑如图 12-10 所示，我们在路由器 R1 的 Serial1/1 接口部署 ISATAP 隧道（IPv6 网络前缀为 FEC0:13:13::/64），使 ISATAP 主机 Host30（Windows XP 系统）能通过 ISATAP 路由器 R1 与 Host10 或 Server10 通信，ISATAP 路由器的配置和验证过程如下：

```
R1(config)#ip route 0.0.0.0 0.0.0.0 serial 1/1
R1(config)#interface tunnel 13
R1(config-if)#ipv6 address fec0:13:13::/64 eui-64          ---ISATAP 必须使用 eui-64 方式---
R1(config-if)#tunnel source 172.16.1.1
R1(config-if)#no ipv6 nd suppress-ra                       ---允许路由器通过 ISATAP 隧道发送 RA 信息---
R1(config-if)#tunnel mode ipv6ip isatap                    ---设置隧道的模式---
R1(config-if)#^Z
R1#show ipv6 interface brief
---省略部分输出---
Tunnel 13                       [up/up]
    FE80::5EFE:AC10:101
    FEC0:13:13::5EFE:AC10:101
```

ISATAP 主机 Host30 的设置过程如下（首先要确保主机能和路由器通信）：

```
C:\Documents and Settings\Administrator>ipconfig
Ethernet adapter 本地连接：
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . .  : 10.3.1.30
        Subnet Mask . . . . . . . . . . .  : 255.255.255.0
        Default Gateway . . . . . . . . .  : 10.3.1.3
C:\Documents and Settings\Administrator>ping 172.16.1.1
Pinging 172.16.1.1 with 32 bytes of data:
Reply from 172.16.1.1: bytes=32 time=44ms TTL=254
Reply from 172.16.1.1: bytes=32 time=18ms TTL=254
Reply from 172.16.1.1: bytes=32 time=22ms TTL=254
Reply from 172.16.1.1: bytes=32 time=13ms TTL=254
Ping statistics for 172.16.1.1:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=13ms, Maximum=44ms, Average=24ms
C:\Documents and Settings\Administrator>
```

在 ISATAP 主机 Host30 上启用 IPv6 协议并设置 ISATAP 隧道，操作过程如下：

```
C:\Documents and Settings\Administrator>ipv6 install        ---启用 IPv6 协议---
Installing...
Succeeded.
C:\Documents and Settings\Administrator>netsh interface ipv6 ISATAP set router 172.16.1.1     ---建立 ISATAP 隧道---
```

验证获取的 IPv6 地址信息，操作过程如下：

```
C:\Documents and Settings\Administrator>ipconfig
---省略部分输出---
Tunnel adapter Automatic Tunneling Pseudo-Interface:
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . .  : fec0:13:13::5efe:10.3.1.30%1
        IP Address. . . . . . . . . . . .  : fe80::5efe:10.3.1.30%2
        Default Gateway . . . . . . . . .  : fe80::5efe:172.16.1.1%2
```

使用 ping 命令测试 Host30 到 Server10（fec0:10:2:0:2050:79ff:fe66:6803）的连通性：

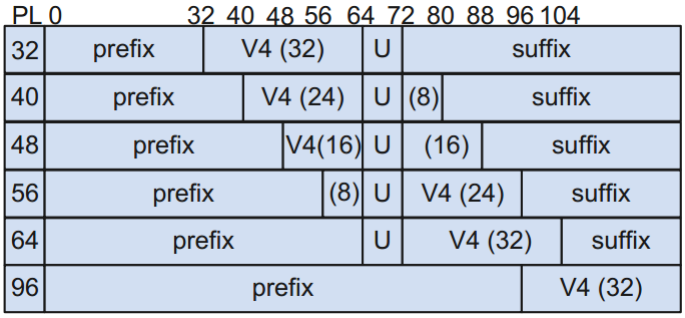
```
C:\Documents and Settings\Administrator>ping fec0:10:2:0:2050:79ff:fe66:6803
```

```
Pinging fec0:10:2:0:2050:79ff:fe66:6803 with 32 bytes of data:
Reply from fec0:10:2:0:2050:79ff:fe66:6803: time=46ms
Reply from fec0:10:2:0:2050:79ff:fe66:6803: time=22ms
Reply from fec0:10:2:0:2050:79ff:fe66:6803: time=22ms
Reply from fec0:10:2:0:2050:79ff:fe66:6803: time=23ms
Ping statistics for fec0:10:2:0:2050:79ff:fe66:6803:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=22ms, Maximum=46ms, Average=28ms
C:\Documents and Settings\Administrator>
```

12.4.5 NAT64

在 IPv6 网络的发展过程中，面临的最大问题应该是 IPv6 与 IPv4 的不兼容性，因此无法实现两种不兼容网络之间的互访。为了实现 IPv6 与 IPv4 的互访，IETF（互联网工程任务组）在早期设计了 NAT-PT 的解决方案 RFC2766。NAT-PT 通过 IPv6 与 IPv4 的网络地址与协议转换，实现了 IPv6 网络与 IPv4 网络的双向互访。但 NAT-PT 在实际网络应用中面临各种缺陷，IETF 建议不再使用，因此已被 RFC4966 废除。为了解决 NAT-PT 中的各种问题，同时实现 IPv6 与 IPv4 之间的网络地址与协议转换技术，IETF（互联网工程任务组）重新设计了一项新的解决方案：NAT64 与 DNS64 技术。

NAT64 是一种有状态的网络地址与协议转换技术，一般用于 IPv6 网络侧用户发起连接访问 IPv4 侧网络资源，但也支持手工配置静态映射关系，实现 IPv4 网络主动发起连接访问 IPv6 网络。DNS64 主要是配合 NAT64 的工作，将 DNS 查询信息中的 A 记录（IPv4 地址）合成到 AAAA 记录（IPv6 地址）中，返回合成的 AAAA 记录用户给 IPv6 侧用户。64:FF9B::/96 为 DNS64 的知名前缀，DNS64 一般默认使用此前缀进行 IPv4 地址到 IPv6 地址的合成，当然，NAT64 中也可以使用长度为 32、40、48、56、64 或 96 的自定义前缀，根据前缀长度的不同，IPv4 地址嵌入 IPv6 地址时嵌入的位置存在差异，具体差异如图 12-19 所示，其中 PL（Prefix Length）表示前缀长度，suffix 表示后缀，U 为保留位，NAT64 设备不处理 suffix 和 U 这两个字段，可以任意取值。

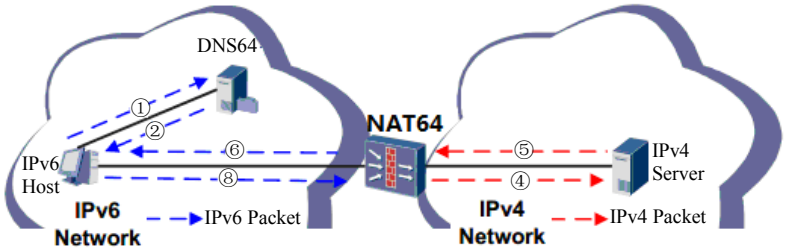


▲图 12-19　NAT64 中 IPv4 地址嵌入 IPv6 地址的各种情况

以 IPv4 地址为 192.168.0.1 为例，如果 NAT 前缀为 3000::/64，则该 IPv4 地址对应的 IPv6 地址为 3000:0000:0000:0000:00C0:A800:0100:0000，即 3000::C0:A800:100:0。

在 NAT64 和 DNS64 部署时需要特别注意的是，DNS64 地址合成所使用的 IPv6 前缀要和 NAT64 配置的 IPv6 前缀一致。

当 IPv6 的主机发起连接访问普通 IPv6 网站时，流量将会匹配 IPv6 默认路由而直接转发至 IPv6 Router 处理；而访问 IPv4 单协议栈的服务器时，将经 DNS64 Server 进行前缀合成，之后的流量将转发至 NAT64 设备上，从而实现 IPv6 与 IPv4 地址和协议的转换，访问 IPv4 网络中的资源，其工作过程如图 12-20 所示。



▲图 12-20　NAT64 的工作过程

（1）IPv6 主机直接使用域名访问 IPv4 网络中的服务器时会向 DNS64 服务器请求该域名对应的 IPv6 地址。

（2）DNS64 查询后确定该域名映射的仅有对应的 IPv4 地址，则自动使用 NAT64 前缀与服务器的 IPv4 地址组合成一个 IPv6 地址，并将该地址返回给 IPv6 主机。

（3）IPv6 主机将收到的 IPv6 地址作为目的地址发起连接请求，报文经过 NAT64 进行转发。

（4）NAT64 网关收到 IPv6 主机发出的 IPv6 报文后，发现报文的目的地址中包含特定的 NAT64 前缀（该前缀在 NAT64 上事先设置），则从 IPv6 报文的目的地址中提取 IPv4 地址，并从配置的 NAT 地址池中选择一个地址作为 IPv4 报文的源地址，并将 IPv6 报文转换为 IPv4 报文发送出去，同时生成会话表。

（5）IPv4 服务器收到请求报文后完成响应。

（6）NAT64 设备收到 IPv4 网络中服务器的响应报文后，根据会话表将 IPv4 报文转换为 IPv6 报文，然后发送至 Host。

根据业务流程可以清楚地了解到，DNS64 是将 IPv4 服务器的 IPv4 地址嵌入到 IPv6 地址中，这个过程叫做合成。通常现网的 DNS4 服务器不能直接升级支持 DNS6，则需要单独部署一台 DNS6 服务器并同时配置 DNS64 软件模块，用于进行 IPv6 服务器的域名解析以及 AAAA 记录的合成和维护。DNS64 服务器的架设知识不在本书的讨论范围中，公网上的 DNS64 服务器有 2001:778::37/128、2001:DF8:0:7::1/128 等。

参照拓扑如图 12-10 所示，假设该网络 NAT64 的前缀是 FEC0:10:3::/96，NAT64 转换的 IPv4 地址范围是 172.16.1.11-12，在路由器 R1 部署 NAT64 的转换使 PC10 能和 PC30 通信，过程如下：

```
R1(config)#ipv6 unicast-routing
R1(config)#interface serial 1/1
R1(config-if)#ip address 172.16.1.1 255.255.255.0
R1(config-if)#nat64 enable                    ---在接口上启用 NAT64 转换的功能---
R1(config-if)#exit
R1(config)#interface fastEthernet 0/0
R1(config-if)#ipv6 address fec0:10:1::/64 eui-64
R1(config-if)#nat64 enable
R1(config-if)#exit
R1(config)#nat64 prefix stateful fec0:10:3::/96    ---状态化 NAT64 的前缀和长度---
```

```
R1(config)#nat64 v4 pool IPV4POOL 172.16.1.11 172.16.1.12    ---IPv4 地址池---
R1(config)#ipv6 access-list MYLIST                           ---定义 IPv6 的 ACL，确定执行 NAT64 的范围---
R1(config-ipv6-acl)#permit ipv6 fec0:10:1::/48 any
R1(config-ipv6-acl)#exit
R1(config)#nat64 v6v4 list MYLIST pool IPV4POOL overload     ---将 ACL 和前缀进行关联---
```

除了上述动态 NAT64 映射外，还可以通过配置 NAT64 静态映射，将特定的 IPv4 地址和 IPv6 地址进行互相转换，实现内部的 IPv6 Server 提供给 IPv4 网络服务的场景。如果我们将 PC20 当成一台 IPv6 服务器（本例中 IPv6 地址为 fec0:20:1:0:2050:79ff:fe66:6801），使其可以通过 172.16.2.20 为 IPv4 的网络提供服务，可以在 R2 上进行下面的设置：

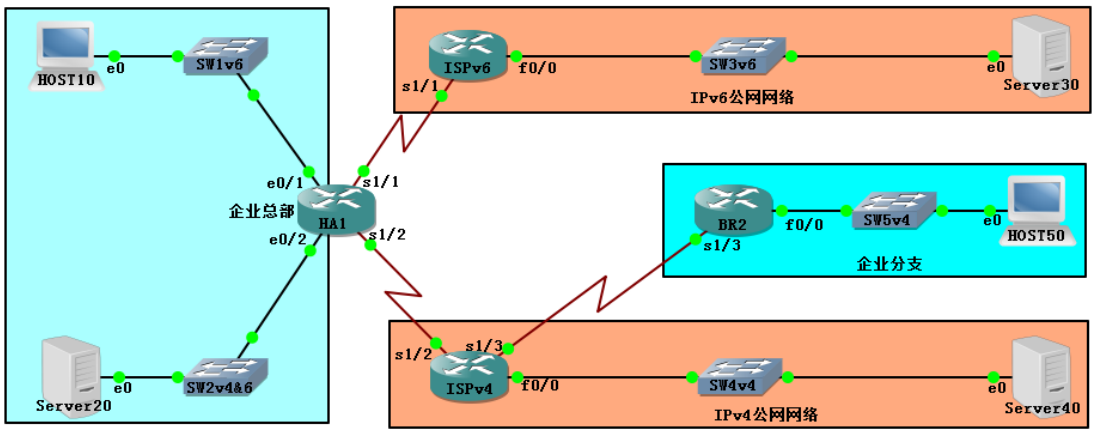
```
R2(config)#interface serial 1/2
R2(config-if)#nat64 enable        ---在接口上启用 NAT64 转换的功能---
R2(config-if)#exit
R2(config)#interface fastEthernet 0/0
R2(config-if)#nat64 enable
R2(config-if)#exit
R2(config)#nat64 v6v4 static fec0:20:1:0:2050:79ff:fe66:6801 172.16.2.20
```

由于 IOU 对 NAT64 的特性支持受限且 NAT6 的工作需要 DNS64 的配合，DNS64 服务器的搭建超出了本书的介绍范围，故本书中只描述了 NAT64 的配置过程，对其效果不再进行测试。如有兴趣，请读者自行阅读 DNS64 相关的书籍和案例。

12.5 实训案例

12.5.1 实验环境

实验拓扑：本次实验使用的拓扑通过 GNS3 搭建，如图 12-21 所示。



▲图 12-21　实验拓扑

实验说明：HA1 为企业总部的网络出口设备，分别通过 IPv6 和 IPv4 连接到不同的 ISP；HOST10 为纯 IPv6 的客户端，通过 DHCPv6 的方式获取 IPv6 的地址，之后可以通过 IPv6 公网访问 IPv6 服务器 Server30；Server20 为 IPv4 和 IPv6 双栈服务器，可以为企业提供 DNS64 服务；当 Host10 访问公网 IPv4 服务器 Server40 时，HA1 提供 NAT64 转换；BR2 为企业分支机构的出口设备，连接

ISP 的 IPv4 网络，当主机 Host50 访问 IPv6 服务时，首先通过 BR2 的 ISATAP 服务获取 IPv6 的地址，再由 HA1 和 BR2 之间建立的 IPv6 over IPv4 的 Tunnel 连接到 IPv6 的网络；总部和分支机构之间运行 OSPFv3 路由协议。本实验中所有设备的 IPv4 地址及路由信息已经预先进行了配置。

实验设备：本实验的设备如表 12-2 所示。

▲表 12-2　实验设备

设备名称	设备类型	平台版本	实现方式
HA1	路由器	I86BI_LINUX-ADVENTERPRISEK9-M，V15.4	IOU
BR2	路由器	C7200-JK9O3S-M , Version 12.4（25g）	GNS3 1.3.9
ISPv4	路由器	C7200-JK9O3S-M , Version 12.4（25g）	GNS3 1.3.9
ISPv6	路由器	C7200-JK9O3S-M , Version 12.4（25g）	GNS3 1.3.9
SW1v6	交换机	普通非网管	GNS3 1.3.9
SW2v4&v6	交换机	普通非网管	GNS3 1.3.9
SW3v6	交换机	普通非网管	GNS3 1.3.9
SW4v4	交换机	普通非网管	GNS3 1.3.9
Server20	PC 机	VPCS（version 0.6.1）	GNS3 1.3.9
Server30	PC 机	VPCS（version 0.6.1）	GNS3 1.3.9
Server40	PC 机	VPCS（version 0.6.1）	GNS3 1.3.9
Host10	PC 机	VPCS（version 0.6.1）	GNS3 1.3.9
Host50	PC 机	物理机（Windows XP）	本地主机

地址分配：本实验的地址分配如表 12-3 所示。

▲表 12-3　地址分配

设备	接口	IPv4 地址信息	IPv6 地址信息	备注
HA1	s1/1	——	FEC0:10::1/64	
	s1/2	172.16.1.1/24	——	
	e0/1	——	FEC0:11:1:10::/64 EUI-64	
	e0/2	10.1.20.1/24	FEC0:11:1:20::/64 EUI-64	
	Tunnel12	——	FEC0:11:12::1/64	
BR2	s1/3	172.16.2.2/24	——	
	f0/0	10.1.50.2/24	——	
	Tunnel12	——	FEC0:11:12::2/64	连接 HA1
	Tunnel50	——	FEC0:11:2:50::/64 EUI-64	连接 SW5v4
ISPv6	f0/0	——	FEC0:30: /64 EUI-64	
	s1/1	——	FEC0:10::6/64	
ISPv4	f0/0	172.16.40.4/24	——	
	s1/2	172.16.1.4/24	——	
	s1/3	172.16.2.4/24	——	

续表

设备	接口	IPv4 地址信息	IPv6 地址信息	备注
Server20	e0	10.1.20.20/24	FEC0:11:20::20/64	
Server30	e0	——	FEC0:30::30/64	
Server40	e0	172.16.40.40/24	——	
Host10	e0	——	DHCPv6 CLIENT	
Host50	e0	10.1.50.50/24	——	

12.5.2 实验目的

- 掌握 IPv6 各种地址的基本配置。
- 掌握 IPv6 各种隧道的部署方式。
- 掌握 IPv6 路由的部署方式。
- 掌握 NAT64 的部署方式。

12.5.3 实验过程

任务一：设备 IPv6 地址的基础设置（路由器的 IPv4 地址信息已做预设置）

Step 1 参照地址表设置 HA1 各物理接口的 IPv6 地址。

```
HA1(config)#ipv6 unicast-routing
HA1(config)#interface serial 1/1
HA1(config-if)#ipv6 address fec0:10::1/64
HA1(config-if)#exit
HA1(config)#interface ethernet 0/1
HA1(config-if)#ipv6 address fec0:11:1:10::/64 eui-64
HA1(config-if)#exit
HA1(config)#interface ethernet 0/2
HA1(config-if)#ipv6 address fec0:11:1:20::/64 eui-64
HA1(config-if)#end
HA1#show ipv6 interface brief
Ethernet0/0             [administratively down/down]
    unassigned
Ethernet0/1             [up/up]
    FE80::A8BB:CCFF:FE00:110
    FEC0:11:1:10:A8BB:CCFF:FE00:110
Ethernet0/2             [up/up]
    FE80::A8BB:CCFF:FE00:120
    FEC0:11:1:20:A8BB:CCFF:FE00:120
Ethernet0/3             [administratively down/down]
    unassigned
Serial1/0               [administratively down/down]
    unassigned
Serial1/1               [up/up]
    FE80::A8BB:CCFF:FE00:100
```

```
        FEC0:10::1
Serial1/2                [up/up]
        unassigned
Serial1/3                [administratively down/down]
        unassigned
Tunnel12                 [up/up]
        FE80::AC10:101
        FEC0:11:12::1HA1#
```

Step 2 在 HA1 上部署 DHCPv6，为 Host10 所在网段的设备分配地址信息，相应 DNSv6 的地址为 FEC0:11:1:20::20，过程如下：

```
HA1(config)#ipv6 dhcp pool IPV6POOL
HA1(config-dhcpv6)#address prefix FEC0:11:1:10::/64
HA1(config-dhcpv6)#dns-server FEC0:11:1:120::20
HA1(config-dhcpv6)#exit
HA1(config)#interface ethernet 0/1
HA1(config-if)#ipv6 dhcp server IPV6POOL
HA1(config-if)#ipv6 nd managed-config-flag
HA1(config-if)#ipv6 nd other-config-flag
HA1(config-if)#end
HA1#
```

Step 3 设置 Host10 自动获取地址并验证（注意，VPCS 只能使用无状态自动地址配置的方式，不能验证 DNS 的信息）。

```
Host10> ip auto
GLOBAL SCOPE       : fec0:11:1:10:2050:79ff:fe66:6802/64
ROUTER LINK-LAYER : aa:bb:cc:00:01:10
Host10> show
NAME     IP/MASK          GATEWAY          MAC                LPORT   RHOST:PORT
Host10   0.0.0.0/0        0.0.0.0          00:50:79:66:68:02  10003   10.10.91.100:10004
         fe80::250:79ff:fe66:6802/64
         fec0:11:1:10:2050:79ff:fe66:6802/64 eui-64
```

Step 4 设置 ISPv6 和 Server30 的 IPv6 地址信息并验证。

```
ISPv6(config)#ipv6 unicast-routing
ISPv6(config)#interface fastEthernet 0/0
ISPv6(config-if)#no shutdown
ISPv6(config-if)#ipv6 address fec0:30::/64 eui-64
ISPv6(config-if)#exit
ISPv6(config)#interface serial 1/1
ISPv6(config-if)#no shut
ISPv6(config-if)#ipv6 address fec0:10::6/64
ISPv6(config-if)#end
ISPv6#show ipv6 interface brief
FastEthernet0/0          [up/up]
        FE80::C801:5FF:FEA0:0
        FEC0:30::C801:5FF:FEA0:0
Serial1/0                [administratively down/down]
Serial1/1                [up/up]
        FE80::C801:5FF:FEA0:0
        FEC0:10::6
Serial1/2                [administratively down/down]
```

```
Serial1/3                          [administratively down/down]
ISPv6#
Ser_30> ip auto
GLOBAL SCOPE          : fec0:30::2050:79ff:fe66:6801/64
ROUTER LINK-LAYER : ca:01:05:a0:00:00
Ser_30> show
NAME       IP/MASK       GATEWAY       MAC                LPORT   RHOST:PORT
Ser_30     0.0.0.0/0     0.0.0.0       00:50:79:66:68:01  10002   10.80.0.100:10001
           fe80::250:79ff:fe66:6801/64
           fec0:30::2050:79ff:fe66:6801/64 eui-64
Ser_30>
```

任务二：在 HA1 和 BR2 之间部署 IPv6 over IPv4 的隧道

Step 1 测试 HA1 和 BR2 之间的连通性。

```
HA1#ping 172.16.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/20/23 ms
HA1#
```

Step 2 在 HA1 上建立 IPv6 over IPv4 的隧道并验证。

```
HA1(config)#interface tunnel 12
HA1(config-if)#ipv6 address fec0:11:12::1/64
HA1(config-if)#tunnel source 172.16.1.1
HA1(config-if)#tunnel destination 172.16.2.2
HA1(config-if)#tunnel mode ipv6ip
HA1(config-if)#end
HA1#show ipv6 interface tunnel 12
Tunnel 12 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::AC10:101
  No Virtual link-local address(es):
  Global unicast address(es):
    FEC0:11:12::1, subnet is FEC0:11:12::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::FB
    FF02::1:FF00:1
    FF02::1:FF10:101
---省略部分输出---
```

Step 3 在 BR2 上建立 IPv6 over IPv4 的隧道并验证。

```
BR2(config)#interface tunnel 12
BR2(config-if)#ipv6 address fec0:11:12::2/64
BR2(config-if)#tunnel source 172.16.2.2
BR2(config-if)#tunnel destination 172.16.1.1
BR2(config-if)#tunnel mode ipv6ip
BR2(config-if)#end
BR2#show ipv6 interface tunnel 12
Tunnel 12 is up, line protocol is up
```

```
    IPv6 is enabled, link-local address is FE80::AC10:202
    Global unicast address(es):
      FEC0:11:12::2, subnet is FEC0:11:12::/64
    Joined group address(es):
      FF02::1
      FF02::2
      FF02::1:FF00:2
      FF02::1:FF10:202
---省略部分输出---
```

Step 4 验证 IPv6 over IPv4 隧道的连通性。

```
HA1#ping FEC0:11:12::2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to FEC0:11:12::2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/24/42 ms
HA1#
```

任务三：在 BR2 上部署 ISATAP 的隧道，为企业分支用户分配 IPv6 地址，使之可以访问企业总部的 IPv6 服务

Step 1 将 BR2 部署为 ISATAP 服务器并验证。

```
BR2(config)#interface tunnel 50
BR2(config-if)#ipv6 address fec0:11:2:50::/64 eui-64
BR2(config-if)#tunnel source fastEthernet 0/0
BR2(config-if)#no ipv6 nd suppress-ra
BR2(config-if)#tunnel mode ipv6ip isatap
BR2(config-if)#end
BR2#show ipv6 interface tunnel 50
Tunnel 50 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::5EFE:A01:3202
  Global unicast address(es):
    FEC0:11:2:50:0:5EFE:A01:3202, subnet is FEC0:11:50::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF01:3202
  MTU is 1480 bytes
  ICMP error messages limited to one every 100 milliseconds
  ICMP redirects are enabled
  ND DAD is not supported
  ND reachable time is 30000 milliseconds
BR2#
```

Step 2 在 Host50 启用 IPv6 后设置 ISATAP 隧道并验证。

```
C:\>ipv6 install
Installing...
Succeeded.
C:\>netsh interface ipv6 ISATAP set router 10.1.50.2
---确定---
C:\Documents and Settings\Administrator>ipconfig
---省略部分输出---
```

```
Tunnel adapter Automatic Tunneling Pseudo-Interface:
        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . .  : fec0:11:2:50:0:5efe:10.1.50.50%1
        IP Address. . . . . . . . . . . .  : fe80::5efe:10.1.50.50%2
        Default Gateway . . . . . . . . .  : fe80::5efe:10.1.50.2%2
```

Step 3 测试 ISATAP 隧道的连通性。

```
C:\ >ping FEC0:11:2:50:0:5EFE:A01:3202
Pinging FEC0:11:2:50:0:5EFE:A01:3202 with 32 bytes of data:
Reply from FEC0:11:2:50:0:5EFE:A01:3202: time=25ms
Reply from FEC0:11:2:50:0:5EFE:A01:3202: time=12ms
Reply from FEC0:11:2:50:0:5EFE:A01:3202: time=10ms
Reply from FEC0:11:2:50:0:5EFE:A01:3202: time=9ms
Ping statistics for FEC0:11:2:50:0:5EFE:A01:3202:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=9ms, Maximum=25ms, Average=14ms
```

任务四：部署 IPv6 的路由信息，使所有的 IPv6 终端能相互访问

Step 1 在 ISPv6 路由器上部署到企业 IPv6 网络的静态路由并验证。

```
ISPv6(config)#ipv6 route fec0:11::/32 serial 1/1
ISPv6(config)#end
ISPv6#show ipv6 route static
IPv6 Routing Table - 7 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S      FEC0:11::/32 [1/0]
        via ::, Serial1/1
ISPv6#
```

Step 2 在 HA1 路由器上部署到公网 IPv6 网络的默认路由并验证。

```
HA1(config)#ipv6 route ::/0 serial 1/1
HA1(config)#end
HA1#show ipv6 route static
IPv6 Routing Table - default - 10 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
       B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
       H - NHRP, I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea
       IS - ISIS summary, D - EIGRP, EX - EIGRP external, NM - NEMO
       ND - ND Default, NDp - ND Prefix, DCE - Destination, NDr - Redirect
       O - OSPF Intra, OI - OSPF Inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2, la - LISP alt
       lr - LISP site-registrations, ld - LISP dyn-eid, a - Application
S      ::/0 [1/0]
        via Serial1/1, directly connected
HQ1#
```

Step 3 在企业网络中部署 OSPFv3，HA1 与 BR2 之间 IPv6 的 Tunnel12 为 OSPF 的骨干区域，

HA1 连接企业总部网络的区域 ID 为 1，BR2 的 Tunnel50 所在区域 ID 为 2，过程如下：

```
HA1(config)#ipv6 router ospf 10
HA1(config-rtr)#router id 1.1.1.1
HA1(config-rtr)#exit
HA1(config)#interface tunnel 12
HA1(config-if)#ipv6 ospf 10 area 0
HA1(config-if)#exit
HA1(config)#interface ethernet 0/1
HA1(config-if)#ipv6 ospf 10 area 1
HA1(config-if)#exit
HA1(config)#interface ethernet 0/2
HA1(config-if)#ipv6 ospf 10 area 1
HA1(config-if)#exit
BR2(config)#ipv6 router ospf 10
BR2(config-rtr)#router id 2.2.2.2
BR2(config-rtr)#exit
BR2(config)#interface tunnel 12
BR2(config-if)#ipv6 ospf 10 area 0
BR2(config-if)#exit
BR2(config)#interface tunnel 50
BR2(config-if)#ipv6 ospf 10 area 2
BR2(config-if)#exit
```

Step 4 在 BR2 上验证 OSPFv3 的邻居关系和路由表。

```
BR2#show ipv6 ospf neighbor

Neighbor ID     Pri    State           Dead Time    Interface ID    Interface
1.1.1.1         1      FULL/  -        00:00:32     14              Tunnel 12
BR2#show ipv6 route ospf
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OI   FEC0:11:1:10::/64 [110/11121]
       via FE80::AC10:101, Tunnel 12
OI   FEC0:11:1:20::/64 [110/11121]
       via FE80::AC10:101, Tunnel 12
BR2#
```

Step 5 在 HA1 上向 IPv6 网络注入默认路由并对区域 1 的路由进行汇总。

```
HA1(config)#ipv6 router ospf 10
HA1(config-rtr)#default-information originate
HA1(config-rtr)#area 1 range fec0:11:1::/56
HA1(config-rtr)#exit
```

Step 6 再次在 BR2 上验证 OSPFv3 的邻居关系和路由表。

```
BR2#show ipv6 route ospf
IPv6 Routing Table - 8 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
```

```
          I1 - ISIS L1, I2 - ISIS L2, IA - ISIS interarea, IS - ISIS summary
          O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
          ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
OE2   ::/0 [110/1], tag 10
        via FE80::AC10:101, Tunnel 12
OI    FEC0:11:1::/56 [110/11121]
        via FE80::AC10:101, Tunnel 12
```

Step 7 从 Host10 测试到 Server20 和 Server30 的 IPv6 的连通性。

```
HOST10> ping fec0:11:1:20:2050:79ff:fe66:6803              ---到 Server20---
fec0:11:1:20:2050:79ff:fe66:6803 icmp6_seq=1 ttl=62 time=15.600 ms
fec0:11:1:20:2050:79ff:fe66:6803 icmp6_seq=2 ttl=62 time=0.000 ms
fec0:11:1:20:2050:79ff:fe66:6803 icmp6_seq=3 ttl=62 time=0.000 ms
fec0:11:1:20:2050:79ff:fe66:6803 icmp6_seq=4 ttl=62 time=0.000 ms
fec0:11:1:20:2050:79ff:fe66:6803 icmp6_seq=5 ttl=62 time=0.000 ms
HOST10> ping fec0:30::2050:79ff:fe66:6800                  ---到 Server30---
fec0:30::2050:79ff:fe66:6800 icmp6_seq=1 ttl=60 time=31.200 ms
fec0:30::2050:79ff:fe66:6800 icmp6_seq=2 ttl=60 time=31.200 ms
fec0:30::2050:79ff:fe66:6800 icmp6_seq=3 ttl=60 time=31.200 ms
fec0:30::2050:79ff:fe66:6800 icmp6_seq=4 ttl=60 time=31.200 ms
fec0:30::2050:79ff:fe66:6800 icmp6_seq=5 ttl=60 time=31.200 ms
```

Step 8 从 Host50 测试到 Server20 和 Server30 的 IPv6 的连通性。

```
C:\>ping fec0:11:1:20:2050:79ff:fe66:6803                  ---到 Server20---
Pinging fec0:11:1:20:2050:79ff:fe66:6803 with 32 bytes of data:
Reply from fec0:11:1:20:2050:79ff:fe66:6803: time=63ms
Reply from fec0:11:1:20:2050:79ff:fe66:6803: time=57ms
Reply from fec0:11:1:20:2050:79ff:fe66:6803: time=56ms
Reply from fec0:11:1:20:2050:79ff:fe66:6803: time=54ms
Ping statistics for fec0:11:1:20:2050:79ff:fe66:6803:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=54ms, Maximum=63ms, Average=57ms

C:\>ping fec0:30::2050:79ff:fe66:6800                      ---到 Server30---
Pinging fec0:30::2050:79ff:fe66:6800 with 32 bytes of data:
Reply from fec0:30::2050:79ff:fe66:6800: time=106ms
Reply from fec0:30::2050:79ff:fe66:6800: time=83ms
Reply from fec0:30::2050:79ff:fe66:6800: time=81ms
Reply from fec0:30::2050:79ff:fe66:6800: time=80ms
Ping statistics for fec0:30::2050:79ff:fe66:6800:
    Packets: Sent=4, Received=4, Lost=0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum=80ms, Maximum=106ms, Average=87ms
```

任务五：部署 NAT64，使 IPv6 和 IPv4 网络能相互访问

Step 1 在 HA1 上部署状态化 NAT64 动态映射，使 SW1v6 连接的终端设备能访问 Server40（NAT64 的前缀使用 FEC0:11:1:30::/96，IPv4 地址池的范围是 172.16.1.11-19）。

```
HA1(config)#interface ethernet 0/1
HA1(config-if)#nat64 enable
```

```
HA1(config-if)#exit
HA1(config)#interface serial 1/2
HA1(config-if)#nat64 enable
HA1(config-if)#exit
HA1(config)#nat64 prefix stateful fec0:11:1:30::/96
HA1(config)#nat64 v4 pool IPV4POOL 172.16.1.11 172.16.1.19
HA1(config)#ipv6 access-list MYLIST
HA1(config-ipv6-acl)#permit ipv6 fec0:11:1:10::/64 any
HA1(config-ipv6-acl)#exit
HA1(config)#nat64 v6v4 list MYLIST pool IPV4POOL overload
```

Step 2 在 HA1 上部署 NAT64 静态映射，ISPv4 连接的外网纯 IPv4 终端能够使用 IPv4 地址 172.16.1.20 访问企业 IPv6 服务器 Server20。

```
HA1(config)#nat64 v6v4 static fec0:11:1:20:2050:79ff:fe66:6803 172.16.1.20
```

Step 3 测试 Host10 与 Server40、Host50 与 Server20 之间的网络连通性。

由于 IOU 对 NAT64 的特性支持受限，本任务中测试过程均省略。若具备条件，请读者自行在真实环境中进行测试。

12.6 习题

1．下面各选项中有关全局单播地址的说法，正确的是_____。

A．目标地址为单播地址的分组被传输到单个接口

B．这是典型的公有可路由地址，就像 IPv4 中的公有可路由地址

C．这些地址类似于 IPv4 私有地址，也不能路由到因特网

D．这些地址不用于路由选择，但是全局唯一的，因此不可能重复使用

2．要 ping IPv6 本地主机的环回地址，可以使用的命令是_____。

A．ping 127.0.0.1　　B．ping 0.0.0.0　　C．ping ::1　　D．trace 0.0::1

3．下面各选项中不属于 IPv6 的迁移机制的是_____。

A．6to4 隧道　　B．GRE 隧道　　C．ISATAP 隧道　　D．Teredo 隧道

4．下列选项中_____是链路本地地址所用的前缀。

A．2001::/10　　B．FE80::/10　　C．FEC0::/10　　D．2002::/10

5．构架在 IPv4 网络上的两个 IPv6“孤岛”互联，一般会使用_____技术解决。

A．IPv6 over IPv4 隧道　　B．ISATAP 隧道

C．双栈　　D．GRE 隧道

6．下面各选项中能让 IPv6 主机获得其所属子网默认网关的是_____。

A．有状态 DHCP　　B．无状态 RS

C．无状态自动配置　　D．邻居发现协议

7．下面各选项中有关 OSPFv2 和 OSPFv3 的说法，正确的是_____。

A．选择路由器 ID 的方法相同　　B．成为邻居前检查的条件相同

C．都是距离矢量的路由协议　　D．都在一个接口上主持多个实例

8．一台客户端主机分别使用 IPv4 和 IPv6 与两台服务器通信，下面各选项中最适合用于该主机的特性是_____。

A．点到点隧道　　B．多点隧道　　C．NAT64　　D．双栈

9．下面是一个作为 IPv6 隧道端点的路由器的配置，该配置创建的隧道类型是_____。

```
interface loopback 1
ip address 1.1.1.1 255.255.255.255
interface tunnel 2
ipv6 address 2000::1/64
tunnel source loopback 1
tunnel destination 2.2.2.2
tunnel mode ipv6ip
```

A．6to4 自动隧道　　B．多点隧道　　C．手工隧道　　D．GRE 隧道

10．下列各选项中不属于 IPv6 地址类型的是_____。

A．单播　　B．组播　　C．广播　　D．任意播

习题答案

1．B　2．C　3．B　4．B　5．A　6．D　7．A　8．D　9．C　10．C