CCNA 实验手册

Version 3.0

By @红茶三杯

ccietea.com 敏捷的网络工程师之路

发布时间	2014-09-01
文档地址	http://ccietea.com
文档作者	红茶三杯（微博）

修订记录

时间	版本	内容
2011-12-01	V1.0	创建文档。
2012-04-01	V2.0	重新修订各个实验及配置描述。
2014-08-05	V2.6	增加综合实验、三层交换等。
2014-11-25	V3.0	修订上一个版本中的错误；增加综合实验 2；增加 GNS3 等章节。

文档说明

目 录

1 实验准备

1.1 通过 Console 接口登录设备

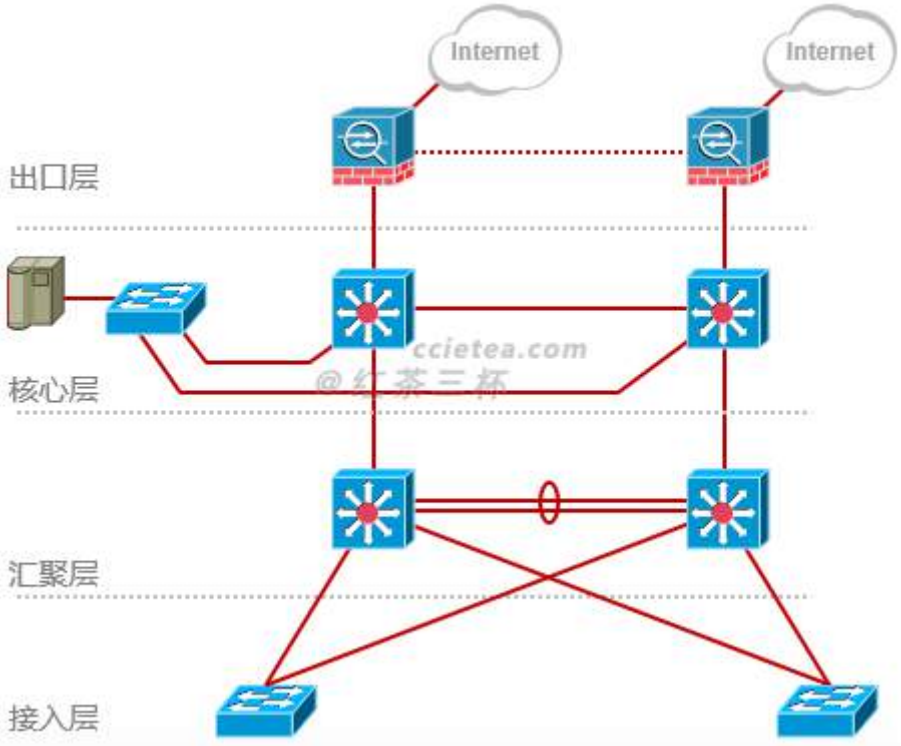

图 1-1 数据网络

在网络工程领域，数据网络（Data Network），如图 1-1 所示，指的是由各种网络设备（例如路由器、交换机、防火墙、负载均衡器）、终端及服务器等搭建而成的一张网。数据网络的基本功能是使得网络上不同节点之间能够相互通信，从而使得各种业务系统能够在网络上正常运行。当然这些设备是需要经过我们的配置及部署之后才能够发挥作用，数通工程师（在许多场合也被称为网络工程师）根据实际的需求，对网络进行规划、设计，并最终将设备调试妥当，把网络搭建起来。

通常对于设备的调试，大多是基于命令行的。网络设备（例如路由器、交换机、防火墙等）在面板上都会有一个用于配置和管理的专用接口 —— Console 口（或 CON 口），如图 1-2 所示，通过这个接口并使用专用的连接线缆将设备与管理 PC 进行连接，即可实现对设备的配置及管理，这是我们在工程实施中最常用的设备配置及管理方法之一。在设备开箱上电后的初次调试，通常都是通过设备的 Console 接口对其进行配置。现在我们通过四个步骤进行讲解：

1. 认识设备的 Console 接口

2. 准备好相关线缆
3. 搭建配置环境
4. 通过终端管理软件登陆设备

下面我们来看看如何使用 PC 通过 console 口对设备进行配置和管理。

1. 认识设备的 Console 接口

图 1-2 设备的 Console 接口

工业级的数据通信设备：路由器、交换机、防火墙等，一般都有配备 Console 口，用于设备的配置及管理，该接口会标记“Console”，或者 CON 字样，如下：

图 1-3 Console 接口

上图所示的是一个 RJ45 的 Console 口，也就是采用水晶头的 Console 线进行连接的接口，大部分数通设备都是 RJ45 的 Console 接口。

2. 准备好相关线缆

图 1-4 USB-RSR232 及 Console 线缆

Console 线缆（上图右）一般会随设备装箱，线缆的一端为 RJ45 水晶头，另一端为 DB9 的串口接头。RJ45 接头用于连接网络设备的 RJ45 标准的 Console 口，线缆另一端的串口用于连接 PC 机，现在大部分台式机都有串口可以直接连接 Console 线缆。遗憾的是，大部分笔记本电脑上并没有配置串口，因此我们需要另一根线缆（上图左）来转接，这就是 USB-RS232 线缆。这根线缆可以说是数通工程师必备的工作工具，各大电子产品商铺均有销售（USB-RS232 需要安装驱动才可使用，驱动程序安装包随线缆附送）。

3. 搭建配置环境

图 1-5 线缆的连接

如图 1-5 所示，将 USB-RS232 线缆的 USB 接口连接到笔记本电脑上，线缆另一头的 RS232 接头连接到 Console 线的串口，Console 线的 RJ45 接头则连接网络设备的 Console 口，配置环境即可搭建完成。

4. 通过终端管理软件登录设备

在管理 PC 上，我们需要准备好终端管理软件用于管理和配置网络设备。常用的终端管理软件有：

- Windows 自带的终端管理工具（WIN7 系统没有自带该工具）
- SecureCRT
- Putty

本文档以 SecureCRT 为例做讲解。请自行下载 SecureCRT 并安装。安装完成之后，打开软件。在自动弹出的 Connect 对话框中选择下图所示的按钮来（Qucik Connect）创建一个连接：

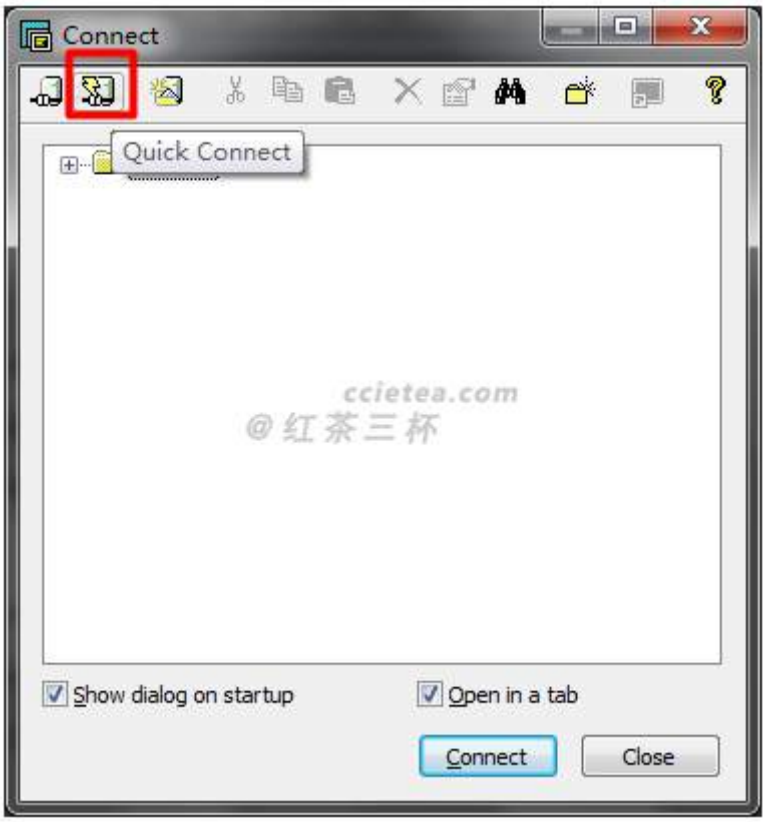

图 1-6 新建一个连接

在弹出的 Quick Connect 对话框中选择“serial”，即使用串行口：

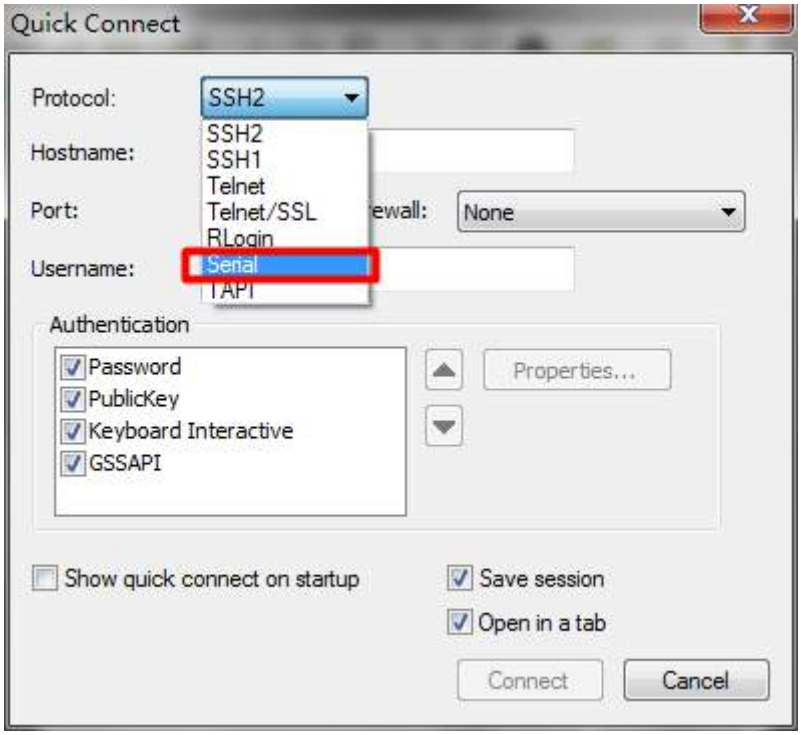

图 1-7 选择连接类型

进一步配置如下，波特率 BaudRate 选择 9600（有些设备可能波特率并非 9600，具体值请参考设备随机文档），Data Bits 选择 8，Stop Bits 选择 1：

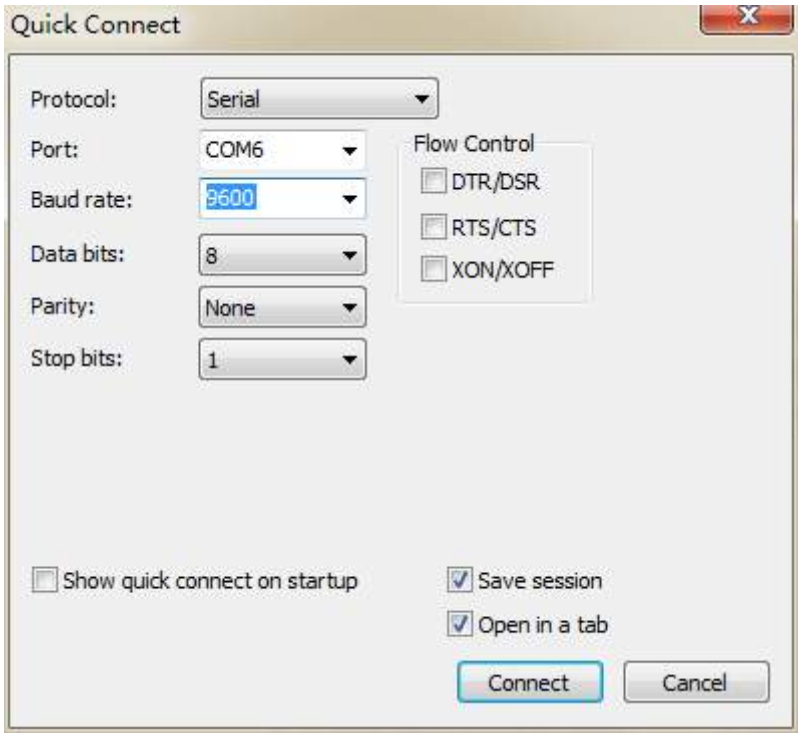

图 1-8 修改连接参数

注意在此处 Port 的选择，视实际情况而定，当用笔记本通过 USB-RS232 线缆转接 Console 线管理设备时，USB-RS232 线缆是需要在 windows 系统上安装驱动程序的，这实际上是在笔记本

电脑上通过 USB 接口来模拟 COM 口，因此安装完成的结果是在系统中会出现一个模拟的 COM 口，该 COM 口编号可在右键“我的电脑”-“计算机管理”-“设备管理”-“端口（COM 和 LPT）”中看到相应的编号，该接口编号要与 SecureCRT 中 Port 选项对应：

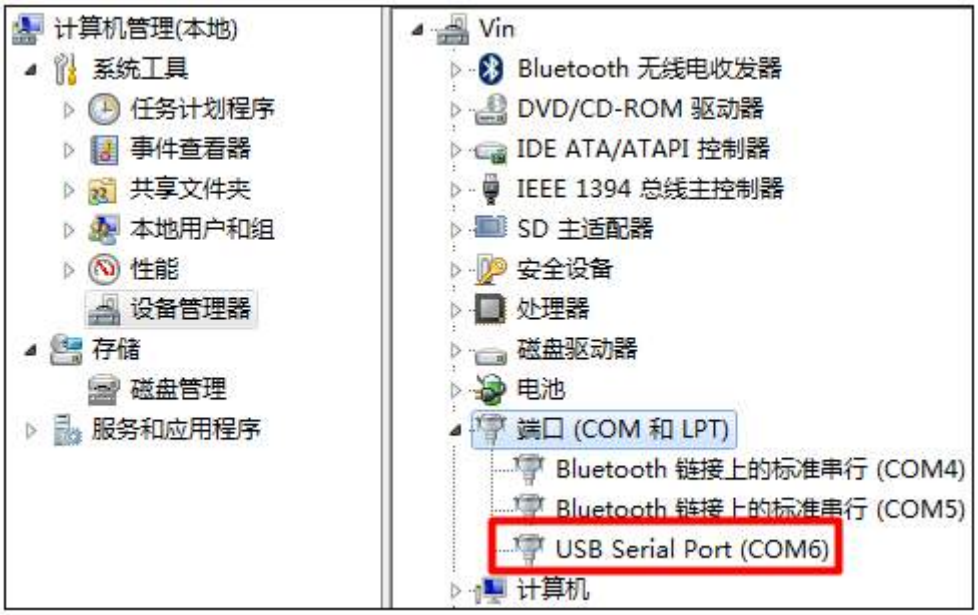

图 1-9 留意 Port 编号

完成后，勾选“save session”将会话保存，然后点击 Connect 按钮，正常的话即可登陆到设备的命令行管理界面，现在就可以开始愉快地配置设备了：

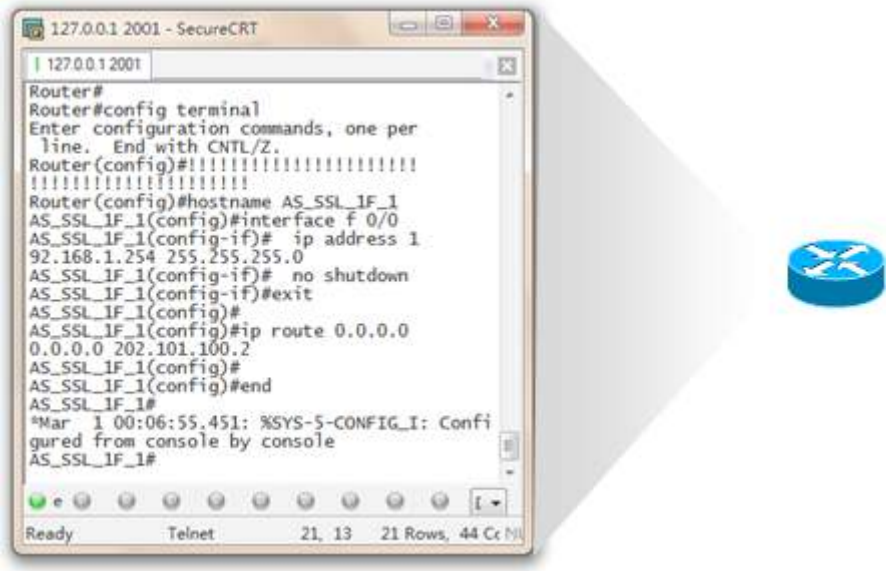

图 1-10 登陆到设备

1.2 Cisco IOS 基础

Cisco IOS(Cisco Internetwork Operating System ,思科网际操作系统)简单的说就是一套运行在 Cisco 网络设备上的操作系统，就像 Windows 是 PC 的操作系统。针对 IOS 的配置通常是通过命令行界面（Command Line Interface，CLI）进行，因此掌握各种常用的 IOS 命令是从事数据网络建设的基础。当我们按照上一个小节讲述的方法搭建好配置环境后 ,即可通过终端管理软件登陆到网络设备上 ,而登录进入后看到的界面，就是 IOS 的 CLI。

1. Cisco IOS 配置模式

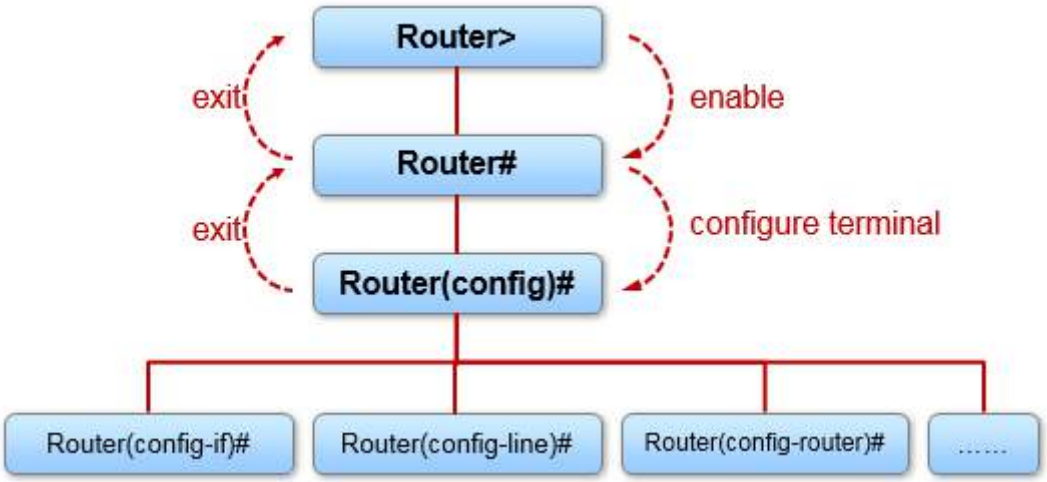

图 1-11 Cisco IOS 配置模式

IOS 定义了各种各样的配置模式（如图 1-11 所示），要想对设备的某个模块进行配置，就要进入相应的配置模式。配置模式的设定使得命令更加模块化和层次化，而且也避免了误操作。当我们登陆一台设备的 IOS 命令行时，可能看到是“Router>”这样的指示符，Router 是设备的名称，这是可以自定义的，而“>”这个符号则表示你当前处于“用户模式”，在用户模式下我们只能够做一些基本的查看及验证命令。在用户模式下使用“enable”命令 ,你会发现“Router>”变成了“Router#”，这表示你已经进入“特权模式”,在这个模式下可以执行更多的命令。而继续使用“configure terminal”命令，则可进入“全局模式”，在该模式下可以对设备的全局特性或功能做一些调试，例如修改设备名称，定义特权密码，或者开启、关闭某个服务等。而在全局模式下使用“interface”关键字加上某个接口，例如 interface serial 0/0，则会进入到 serial0/0 这个接口的配置模式下，就可以针对该接口进行相应的配置，而且此时所做的配置只会影响这个接口。使用“exit”命令可以返回上一个模式。

2. IOS 的命令及关键字

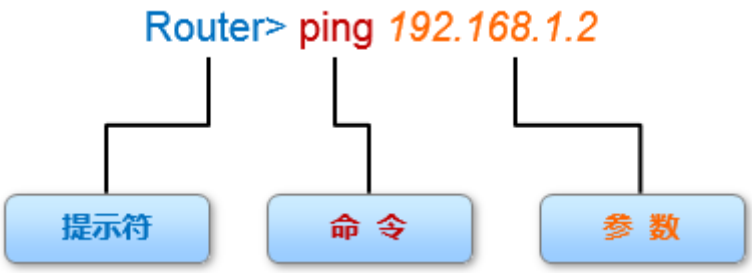

图 1-12 命令

一条典型的 IOS 命令如上图所示，“ping”是关键字，“192.168.1.2”是参数，这条命令用于探测本地到 192.168.1.2 的连通性。再与“hostname R1”这条命令中，“hostname”是关键字，“R1”是参数，这条命令被配置后设备的名称就变成了 R1。当然要注意的是特定的命令需要在正确的配置模式下完成输入才能生效，否则就有可能会报错。

3. 使用 IOS 的帮助功能

初学 IOS 的时候，你可能会感觉非常吃力，因为有太多的命令需要去记，其实不用着急，许多命令和关键字都是在重复使用中自然而然就记住了，而且 IOS 强大的帮助功能也使我们上手更容易。

- **命令提示及补全**

  当我们要输入一个关键字，但你却又忘记了这个关键字的全称而只是记得开头的几个字符，这时可以在键入头几个字符后按“?”号：

```
R1(config)#ho?
hostname
```

上面的例子中，我想给设备改个名字，但是只记得关键字的开头是 ho，于是我输入“ho?”，系统会自动弹出 ho 开头的所有关键字。另外，如果输入一个关键字后，后续的关键字或者参数我们不知道要再键入什么信息了，亦可在键入该关键字后，输入空格，再输入“?”号：

```
R1(config)#hostname ?
  WORD  This system's network name
```

上面的例子中，我们输入了 hostname，但是 hostname 后面该输入啥，我忘了，于是我使用问号，即可得到提示。

另外，IOS 还有命令补齐功能，也就是在你输入一个关键字的开头几个字符后，使用“[tab]”键（大小写切换键上面的那个键），系统会自动将命令补齐，例如下面的例子：

```
R1(config)#ho[tab]
R1(config)#hostname
```

我输入 ho 后接着按 Tab 键，系统自动将关键字补齐为 hostname。当然这里要求当前配置模式下 ho 开头的就这一个关键字。补充一点，其实 IOS 所有的命令，都可以采用简化的书写方式，只要不会引发歧义，例如“hostname hello”这条命令，等同于“ho hello”，也就是说关

键字只输入了“ho”这两个字符，这大大提升了我们的配置效率，当然采用这种输入方式的前提是，输入的关键字不能有歧义，“ho”开头的不能存在 2 个或以上的关键字，如果存在，那就多输入几个字符，例如“host hello”。

- **命令语法检查**

当输入的命令有错误时，系统会弹出相应的报错：

示例 1：

```
R1(config)#router ospf
% Incomplete command.                    !! 命令不完整
```

命令没输完就按了回车，router ospf 后面还有其他参数。

示例 2：

```
R1(config)#router ospd 1
                          ^
% Invalid input detected at '^' marker.    !! 箭头所指字符无法识别，可能输入有误
```

示例 3：

```
R1(config)#s
% Ambiguous command:  "s"                 !! 未知的输入
```

s 开头的关键字太多，只输入了一个字符，系统表示不知道你想输入啥。

- **热键和快捷方式**

常用的快捷键：

- Tab	填写命令或关键字的剩下部分。
- Ctrl-U	删除一整行
- Ctrl-Z	退出配置模式并返回到执行模式
- 向下箭头	用于在前面用过的命令的列表中向前滚动
- 向上箭头	用于在前面用过的命令的列表中向后滚动
- Ctrl-Shift-6	用于中断诸如 ping 或 traceroute 之类的 IOS 进程
- Ctrl-C	放弃当前命令并退出配置模式

1.3 GNS 模拟器

几乎所有知识或者技能的学习都不应该脱离实践或者应用，这条规则在网络工程领域显得更为突出，学了那么多协议或者技术，最终都应该与业务及项目相结合。但是对于大多数数据通信的爱好者、从业者而言，拥有一套由真实设备组建的实验机架是非常奢侈的事情。GNS 模拟器 —— 这款经典的网络模拟软件绝对是不可多得的学习及工作的好伴侣。它能够从底层模拟包括 Cisco 在内的多个厂商的设备，支持的设备类型丰富多样，其中包括路由器、交换机、防火墙、终端等等，整合了抓包工具使得报文分析更加简单，支持与真实设备桥接，支持 IOS 导入使得对设备的仿真上升到一个新的高度，几乎是 100%还原真实设备，而且操作非常简单快捷，上手容易，几乎是网络工程行业的技术工程师，或者技术爱好者必备的软件之一。

这款软件可以直接从其官方网站 http://www.gns3.com/ 免费下载。本文档以 GNS3-1.2.3 为例进行讲解。从官网上下载到 GNS3-1.2.3-all-in-one.exe 这个安装包，双击即可开始安装，安装过程比较简单，一路按 Next 即可。

1.3.1 实验准备

我们需要使用到的软件有：SecureCRT、GNS、Wireshark，以及 Cisco IOS。

其中 SecureCRT 用于终端管理，Wireshark 是经典的抓包工具，它被捆绑在 GNS 的安装包之中，可以随后者一并安装。关于 Cisco IOS，请自行上网搜索，本手册涉及到的相关实验，可使用 c3640-ik9o3s-mz.124-25.bin 这款 IOS 来完成。

1. 环境调试

GNS 软件安装完成后，需配置一些基本参数。打开 GNS，关掉弹出的“Getting Started”，就能看到 GNS 的主界面，如图 1-13 所示：

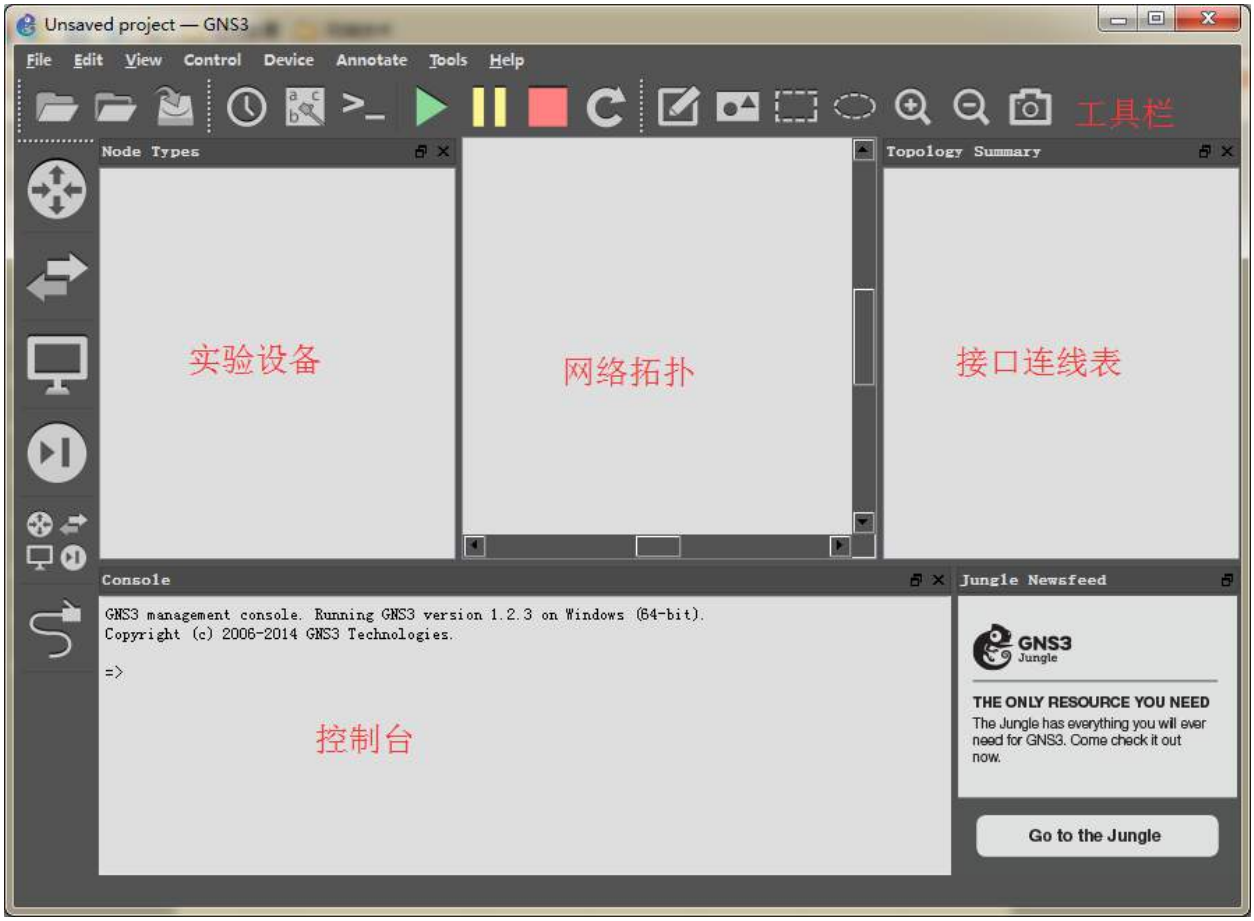

图 1-13 GNS 主界面

现在我们要对 GNS 做一些基本的设置。首先要创建路由器或交换机的实例，点击 Edit / Preferences：

图 1-14 配置首选项

选择 IOS Routers（如图 1-14 所示），点击 New 添加一个实例，在弹出的对话框中，找到我们事先准备好的 Cisco IOS，这里我以 c3640-ik9o3s-mz.124-25.bin 为例。完成后点击 Next（如图 1-15 所示）。

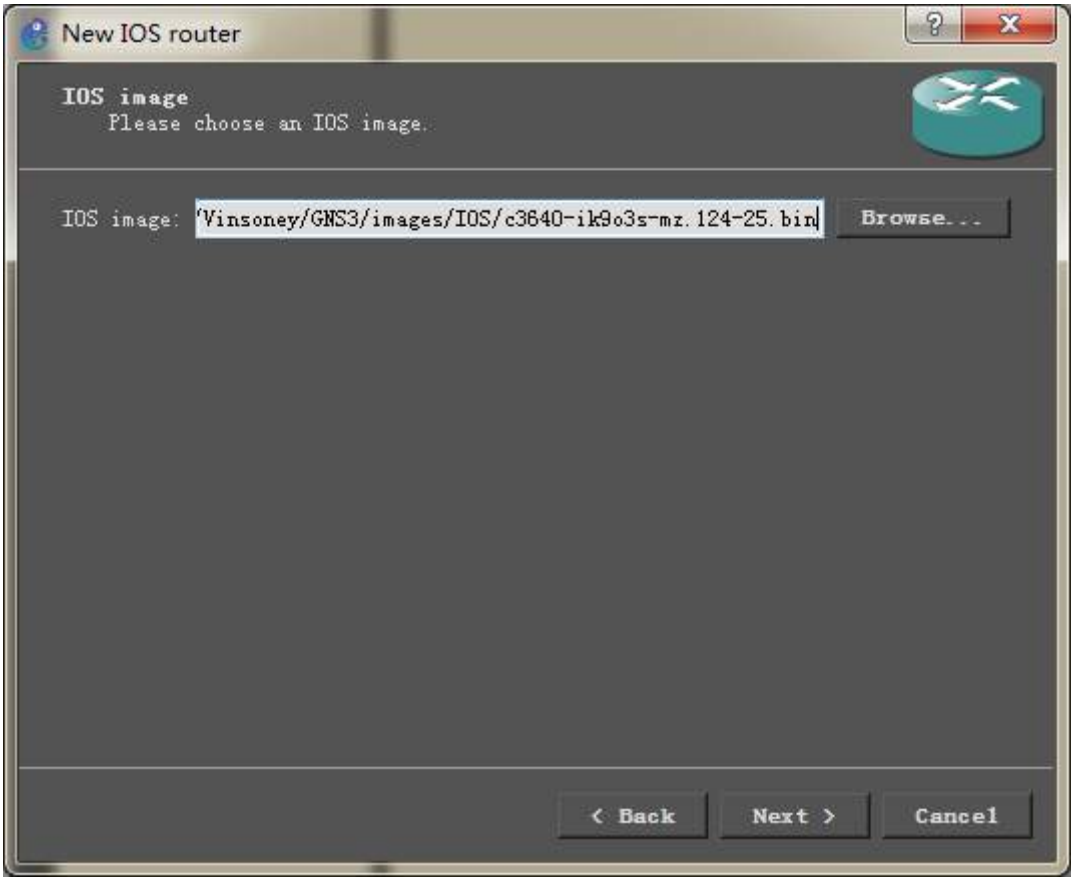

图 1-15 选择 IOS

现在选择实例名称及设备平台。实例名称可以随便设置，我这里设置的是 C3640-1，平台的选择则要根据 IOS 而定，这里当然应该选 C3600 平台，Chassis 可选择 3640（如图 1-16 所示），点击 Next。

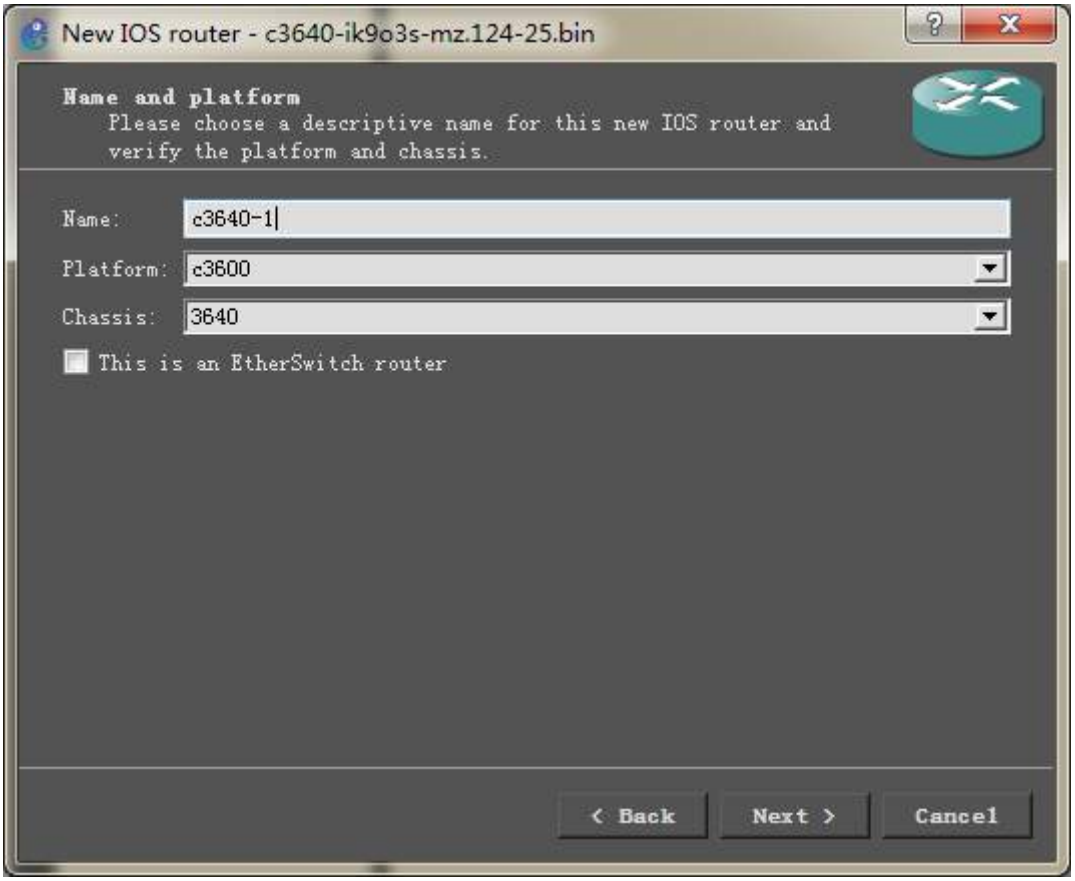

图 1-16 配置设备名称、平台类型，以及型号

接下来是 RAM 的分配，使用默认值即可，然后继续点 Next。

下面是为设备配置硬件模块，这个可以根据自己的需要进行配置，Slot 是模块的插槽，每个槽位都可以选择相应的模块，例如 NM-1FE-TX 为 1 个快速以太网接口的网络模块，NM-4T 为 4 个 Serial 接口的网络模块，NM-16ESW 为 16 个快速以太网接口的交换模块。在这个例子中，我为 0 号槽位配置了 NM-1FE-TX 模块，1 槽位配置了 NM-4T（如图 1-17 所示）。如此一来，我这台路由器就拥有了一个快速以太网接口和 4 个 Serial 接口。以这台路由器为例，快速以太网接口的编号是 FastEthernet0/0，而 4 个 Serial 接口分别是 Serial1/0、Serial1/1、Serial1/2、Serial1/3。

点击 Next，到了 Idle-PC 的配置界面，可以先不填写，直接点 Finish。

图 1-17 为设备配置模块

这就完成了一个路由器的配置。你可以根据需要再配置几个实例，可以为路由器配置不同的模块以便应对不同的实验，另外也可以增加交换机的实例，说到交换机的实例，c3640-ik9o3s-mz.124-25.bin 这个 IOS 可以在 GNS 环境中用于三层交换机，实例的创建与上述过程类似，只不过在 Name and Platform 界面需勾选 This is an EthernetSwitch 选项，同时在为设备加载模块时，选择 NM-16ESW。

回到主界面，点击左侧设备列的第一个图标（Browse Routers），就可以看到我们刚刚创建的路由器 C3640-1（如图 1-18 所示）：

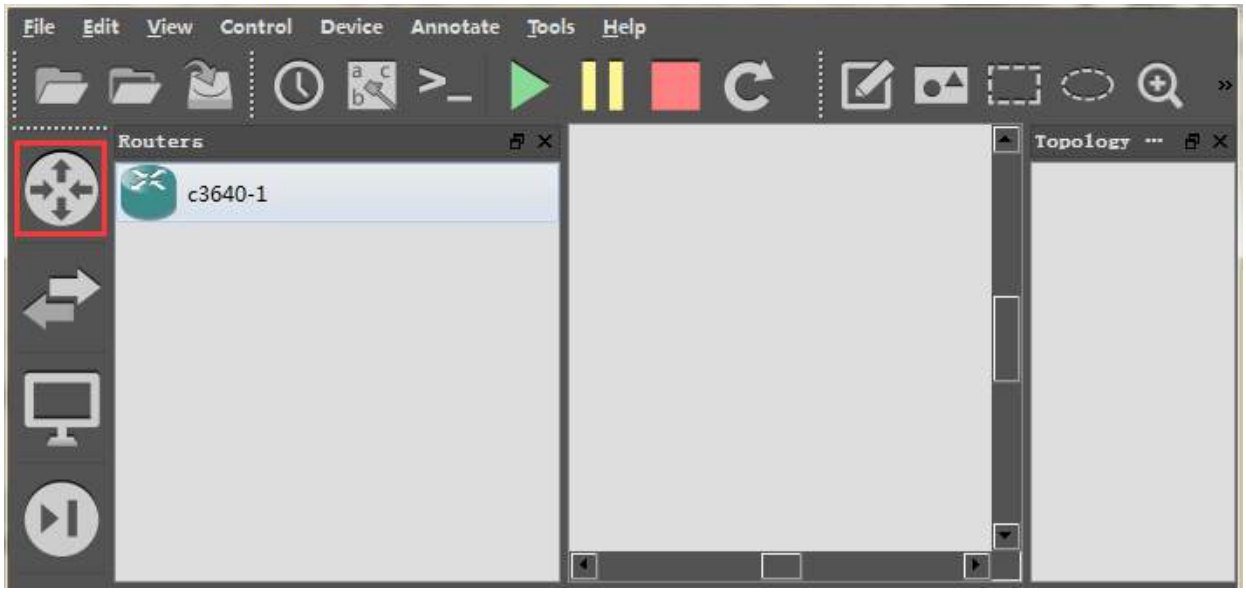

图 1-18 查看设备列表

现在选择 C3640-1 ,然后拖动鼠标到拓扑画布上放开 ,即在拓扑中添加了一台路由器(如图 1-19 所示)。

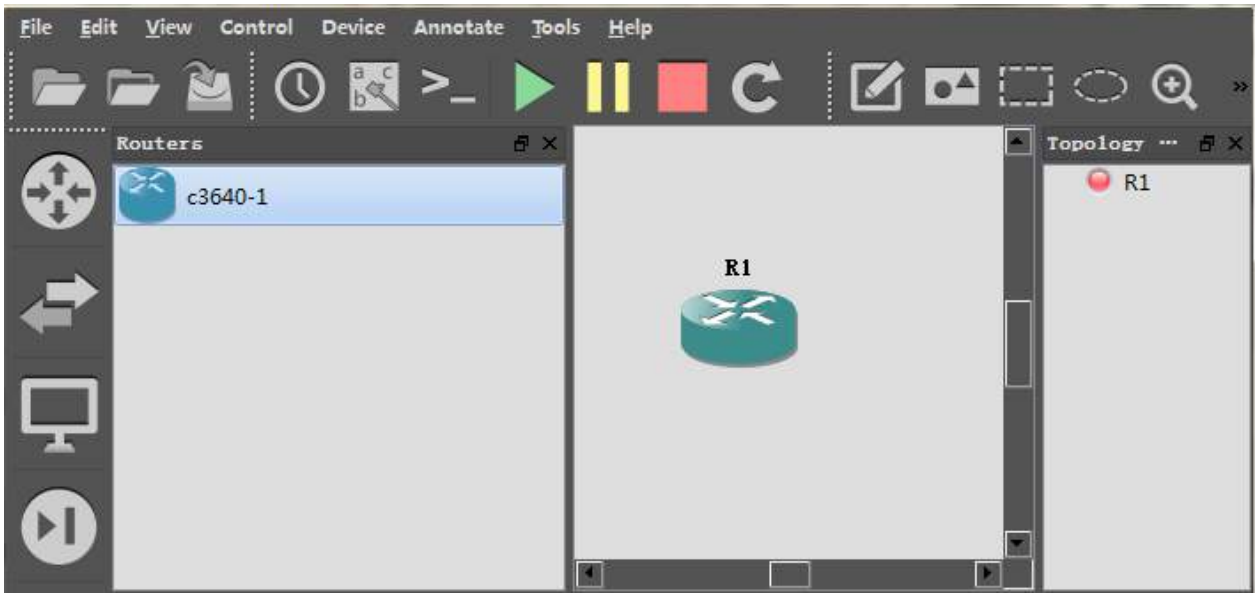

图 1-19 添加实验设备到拓扑中

对设备点击右键，选择 Start，该路由器就开始启动了。待启动之后，双击设备图标，即可弹出命令行窗口对设备进行配置。

2. 计算设备 IDLE 值

GNS 的 IDLE 值是模拟设备的一个全局参数，这个参数根据 IOS 的不同需计算不同的值，它会直接关系到模拟器对电脑 CPU 的占用，该值如果设置不妥，运行少数几台路由器就可能把电脑的 CPU 跑到 100%，因此在开始实验之前，需要对 IDLE 值进行调试，在拓扑中添加路由器，并把路由器开启后，双击图标进入配置界面，随便敲两下回车。

然后对路由器图标点右键，选择 Idle-PC，软件会自动计算 Idle 值，计算完成后会出现一个下拉单 (如

图 1-20)，我们选择带星号的值即可。Idle 值的计算只需一次，后续就无需在重复计算了。

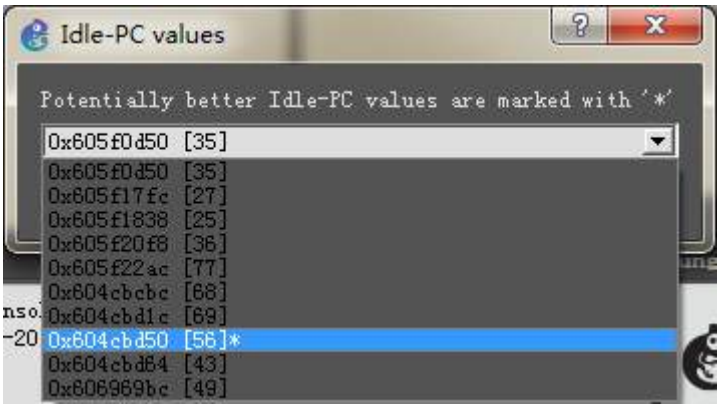

图 1-20 IDLE 值的计算

1.3.2 使用 GNS 搭建一个简单的拓扑

现在，我们要使用 GNS 来搭建一个简单的实验拓扑，这个拓扑非常简单，只有两台路由器，使用快速以太网接口直连。点击主界面左侧的 Browse Router 图标，选择 C3640-1 然后拖到拓扑画布上放开，重复这个动作，这样我们就由了两台路由器（如图 1-21 所示）：

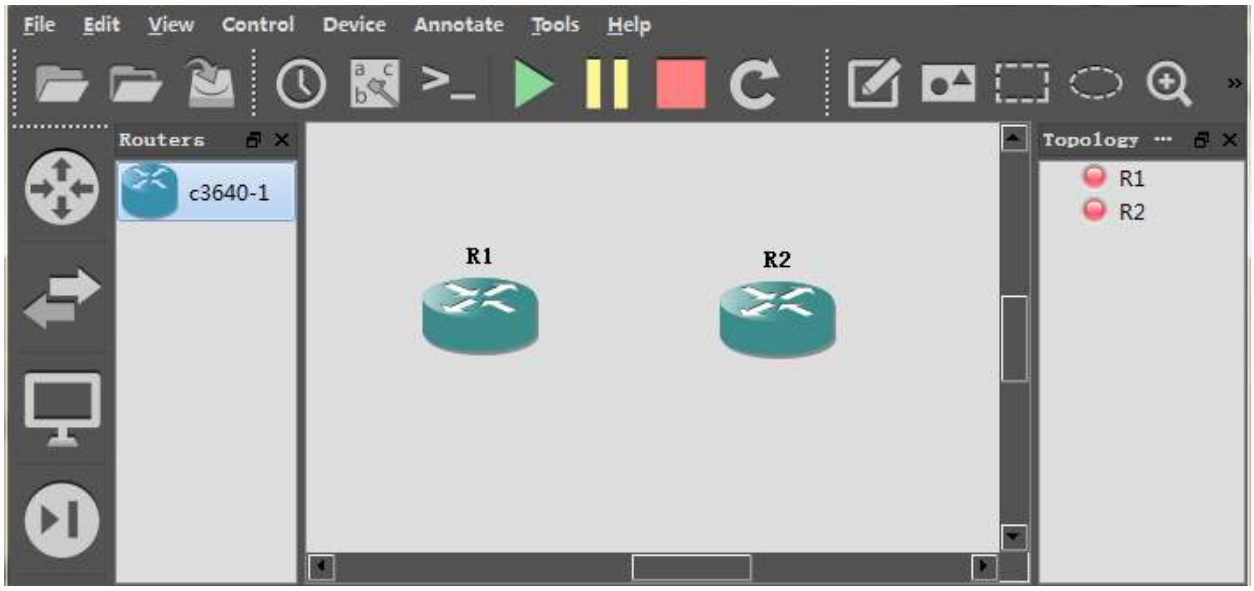

图 1-21 在拓扑中放置两台路由器

由于这种各类型的路由器之前就已经安装了相应的模块：1 个快速以太网接口及 4 个 Serial 接口，我们现在有两台这样的路由器。接下来要在路由器之间连线。在主界面左侧选择 Add a Link 图标，鼠标指针会变成一个十字，对着 R1 点击鼠标，会弹出 R1 的可选接口列表（如图 1-22 所示）：

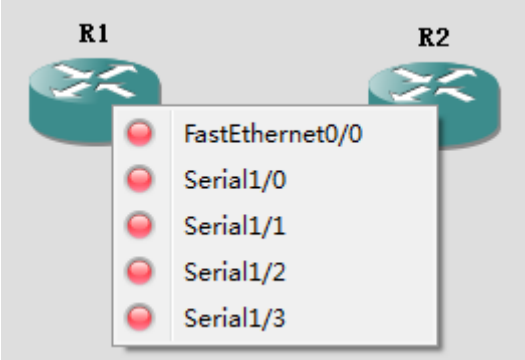

图 1-22 R1 的接口列表

选择 FE0/0 接口，然后再到 R2 上点击一下鼠标，弹出 R2 的接口列表，也是选择 FE0/0 接口，如此一来，我们就在 R1 的 FE0/0 接口及 R2 的 FE0/0 接口之间拉了一条网线（如图 1-23 所示）：

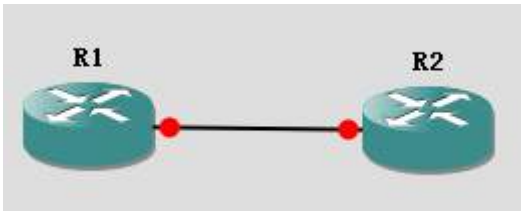

图 1-23 完成设备连线

现在，点击主界面工具栏中的绿色播放键（Start/Resume all Devices），将所有设备开启，我们会看到设备的接口“指示灯”变成了绿色。待设备开启完成后，双击图标，即可开始实验。GNS 缺省使用 putty 这个控制台管理软件进行设备配置，也就是双击图标后弹出的那个配置工具，这个工具的使用体验可能不是最佳的，可以将缺省的配置工具修改为 SecureCRT，点击菜单栏的 Edit/Preferences，点击左侧的 General，选择 Console applications 选项卡，在图 1-24 红色框内 Preconfigured commands 处选择 SecureCRT，然后点击 set 按钮，并修改 SecureCRT 的软件目录（根据实际的软件目录修改）：

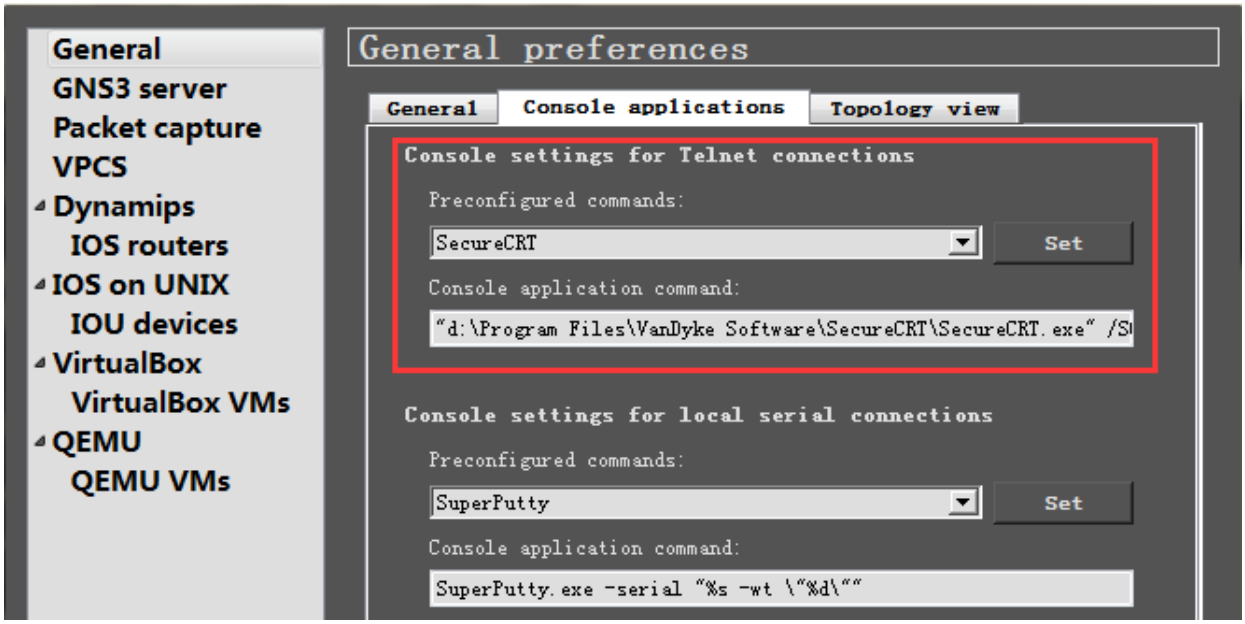

图 1-24 修改缺省的控制台软件

2 路由篇

2.1 路由器基础配置

实验目的

1. 掌握通过 Console 登录 CISCO 路由器的方法；
2. 掌握常用的 CISCO IOS 命令。

拓扑及需求

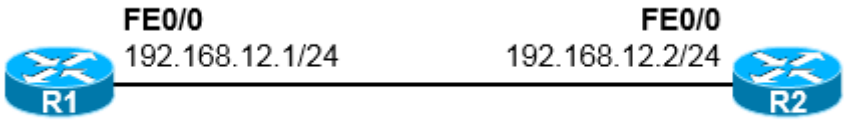

图 2-1 基础实验网络拓扑

配置及实现

如果你是采用真实的设备进行这个实验，那么请按照图示的要求在 R1 及 R2 这两台路由器的以太网接口之间连接一条网线（图 2-1 中的 FE0/0 表示的是路由器的一个快速以太网接口，FE 是 FastEthernet 的缩写，“0/0”的第一个 0 表示槽位 Slot，第二个 0 表示接口编号）。当然，可能实际互连的接口并非两者的 FE0/0 接口，如果不是 FE0/0，则请在配置接口时按照实际情况进行相应的配置。

1. 配置路由器 hostname

R1 的配置如下：

```
Router>                          !! Router>表示当前处于用户模式
Router> enable                   !! 使用 enable 命令进入特权模式
Router#                          !! Router#表示当前处于特权模式
Router# configure terminal       !! 使用 configure terminal 进入全局配置模式
Router(config)#                  !! Router(config)#表示当前处于全局配置模式
```

```
Router(config)# hostname R1                    !! 修改路由器名称为 R1
R1(config)# exit                               !! 从当前的全局配置模式退回到特权模式
Router#
```

R2 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R2
R2(config)# exit
R2#
```

2. 配置路由器的 MOTD

设置登录路由器时控制台显示的欢迎信息，欢迎信息使用一对起始字符和相同的结束字符区隔，例如下面的例子中，两个#号中间的内容就是我们所配置的欢迎信息：

```
R1(config)# banner motd #Hello , This is a Test of ccietea.com#
```

命令配置上去后，重新登录 R1 就会看到这条提示。

3. 在实验环境下，经常会用到如下三条命令来提高实验的效率

```
R2(config)# line console 0                     !! 进入 console 口
R2(config-line)# no exec-time
R2(config-line)# logging synchronous
R2(config-line)# exit
R2(config)# no ip domain lookup
```

line console 0 用于进入设备的管理接口，no exec-time 等同于 exec-time 0 0，用于将控制台设置为永远不超时，如此可以避免在实验过程中，设备因为一段时间未有操作导致超时退出，提高实验的效率。logging synchronous 命令用于配置 console 的日志同步功能，使得命令行界面弹出的日志信息不会打断用户正在输入中的配置命令。no ip domain lookup 命令用于关闭设备的域名解析功能，如果 ip domain lookup 这条命令没有 no 掉（缺省是打开的），则如果在特权模式下键入非关键字字符（例如误操作），则设备会认为用户输入的是一个主机名，于是开始在网络上解析这个主机名对应的 IP 地址，而解析过程中我们无法对设备进行进一步的配置，影响实验效率。

4. 设置和取消密码：Console 密码、特权模式密码、VTY 线路密码

R1 配置 Console 密码：

```
R1(config)# line console 0                     !! 进入 Console 口
```

```
R1(config-line)# password ccietea.com        !! 配置 Console 口密码
R1(config-line)# login                       !! 登录 Console 时，需提供密码进行身份验证
R1(config-line)# exit
```

配置完上述命令后，后续若再使用 console 口对设备进行管理就需输入相应的密码。

如果要实现通过 Console 口登陆时无需输入密码进行身份验证，则使用 no login 命令。删除密码使用：no password。

R1 配置特权密码：

```
R1(config)# enable password ccietea
```

enable password 命令用于设置特权明文密码，该密码在用户使用 enable 命令试图从用户模式进入特权模式时被要求输入。使用 show run 命令查看设备配置时，会看到这个密码的明文，因此强烈不建议使用这种方式指定特权密码。

```
R1(config)# enable secret ccietea.com
```

enable secret 命令用于设置 enable 密文密码，show run 只能查看到被加密的密码，安全级别高于上面的明文密码，与明文密码同时设置时，密文密码生效（也就是忽略 enable password）。取消 enable 密文密码可使用 no enable secret。在实际部署中，强烈建议配置密文密码而不是明文的。

配置 Telnet 密码：

```
R1(config)# line vty 0 4                     !! 进入 VTY 0-4 线路
R1(config-line)# password ccietea            !! 设密码，即用户通过 telnet 登入设备时输入的密码
R1(config-line)# login                       !! 使得 Telnet 登录到本设备需用密码验证
R1(config-line)# exit
```

5. 配置路由器的接口

R1 的配置如下：

```
R1(config)# interface fastethernet 0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
```

注意，“interface fastethernet 0/0”命令中 interface 是关键字，后面跟的 fastethernet 是接口的类型，0/0 为接口编号，第一个数字是接口板的编号，后一个数字是该接口在这个接口板上的编号，请根据自己的实验环境自行更改，例如如果实验使用的接口为串行口，那么命令就是 interface serial。同时，注意接口配置完 IP 地址后，一定要 no shutdown 取消关闭。

R2 的配置如下：

```
R2(config)# interface fastethernet 0/0
```

```
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
```

6. 查看接口状态

在特权模式下使用 show 关键字搭配各种命令可以查看路由器运行的各项参数和信息。使用 show ip interface brief 可以查看路由器所有接口的 IP 配置信息，以及接口状态。

R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.12.1	YES	manual	up	up

R2#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.12.2	YES	manual	up	up

从输出的结果可以看出，R1 及 R2 上的 FE0/0 接口的状态都为物理-Up、协议-Up。路由器的接口要能够正常工作，前提是接口的物理及协议状态都是 UP 的。物理状态指的就是接口的电气化特性，例如接口如果没有接网线，或者网线对端的设备没有加点，那么这个接口的物理状态就可能是 Down 的。如果一个接口的物理状态是 Up 的，而协议状态是 Down 的话，该接口可能无法正常工作。协议状态与该接口的封装协议有关，例如 Serial 接口如果采用 PPP 封装，而 PPP 要求认证用户名及密码，结果验证失败，那么此时接口的协议状态就是 Down 的。

7. 连通性测试

完成上述配置后，R1 与 R2 即可互相通信。我们首先做一个简单的 IP 连通性测试，让 R1 及 R2 互相 ping，所谓的 ping 其实是一个基于 ICMP 的小工具，这个小工具几乎在所有的操作系统上都被携带着，可以探测从本地到远端 IP 节点的可达性。

R1#ping 192.168.12.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/62/120 ms
```

注意 ping 命令要在用户模式或者特权模式下输入。

R2#ping 192.168.12.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.1, timeout is 2 seconds:
!!!!!
```

```
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/41/100 ms
```

让 R1 及 R2 进行互 ping，出现感叹号 !!!!! 代表网络是通的，这应该是网络工程师最喜欢看到的符号了。

8. telnet 远程登录到设备上

现在我们尝试从 R2 远程登陆（Telnet）到 R1。由于在前面的配置步奏中，我们已经在 R1 上配置了 vty 密码，而且现在 R1 及 R2 之间的网络又是联通的，因此在 R2 上可以直接 telnet 到 R1：

```
R2#telnet 192.168.12.1
Trying 192.168.12.1 ... Open

Hello , This is a Test of ccietea.com                    !!看到 banner 提示了
User Access Verification
Password:                 !! 密码输入时是不会显示字符的，这里输入的是 vty 的密码
R1>                       !! 看到 R1 的设备名了，已经登录到 R1 上来了
R1>enable                 !! 输入 enable 试图进入特权模式
Password:                 !! 密码输入时是不会显示字符的，这里输入的是 enable 密码
R1#                       !! 密码验证通过，进入 R1 的特权模式了
R1#exit                   !! 从 R1 登出

[Connection to 192.168.12.1 closed by foreign host]
R2#
```

当我们 telnet 到一台 Cisco 路由器时，使用 exit 命令可以从远端路由器登出，返回到本地设备。而如果我们仅仅是希望挂起 telnet，也就是不希望 telnet 会话断掉，只是想返回到本地设备做些操作，则可使用 ctrl+shift+6，然后按 x。举个例子，假设我们从 R2 telnet 到了 R1 上，使用 ctrl+shift+6，然后按 x 可以返回 R2：

```
R2#show session
Conn    Host                  Address               Byte    Idle Conn Name
*   1   192.168.12.1          192.168.12.1             0       0   192.168.12.1
```

在 R2 上 show session 可以看到本地挂起的会话，此时如果要返回 R1 的 telnet 会话，则只需键入会话的 ID，也就是 1 即可：

```
R2#1
[Resuming connection 1 to 192.168.12.1 ... ]

R1>
R1>
```

9. 查看设备配置

在特权模式或用户模式下使用 show running-config 命令可以查看当前设备的配置，这些配置信息是保存在 RAM 中的，也就是保存在动态存储器里的，这意味着如果设备重启或者掉电，这些配置就会丢失。我们每为设备增加或者修改的一条命令，都会记录在 running-config 中，如果要想让设备的配置不丢失，则要把配置保存到 startup-config，使用 write 命令，或者 copy running-config start-config 即可将当前配置(running-config)保存到启动配置(startup-config) 中，养成良好习惯，设备配置完成后注意保存。show startup-config 显示设备启动配置 (配置信息保存于 NVRAM 或 Flash 中)。

2.2 静态路由

实验目的

1. 巩固 Cisco IOS 的基本配置；
2. 掌握静态路由原理和配置命令；
3. 掌握默认路由原理和配置命令。

拓扑及需求

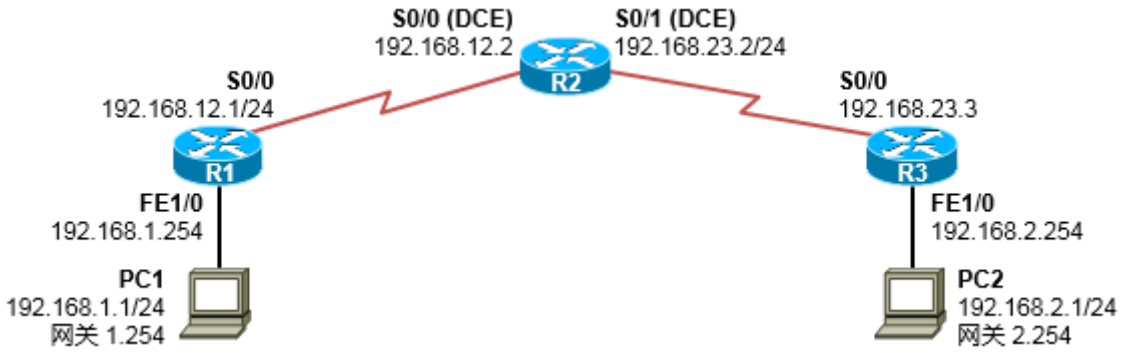

图 2-2 静态路由实验 网络拓扑

- 网络拓扑及 IP 编址如图所示；PC1 及 PC2 使用路由器模拟；
- 在 R1、R2、R3 上配置静态路由，保证全网可达；
- 在 R1、R3 上删掉上一步配置的静态路由，改用默认路由，仍然要求全网可达。

关键知识点

1. 基础知识

本实验主要考察静态路由的概念及实现。对于 PC1 来说，如需访问 192.168.1.0/24 以外的网络，则要将数据先发向网关 R1（因为 PC1 将网关地址设置为 192.168.1.254，也就是 R1 的 FE1/0 接口地址），因此 R1 要有到达远端网络的路由。在完成基本的 IP 配置后，R1 仅知晓 192.168.1.0/24 及 192.168.12.0/24 网络（直连路由），而对于 192.168.23.0/24 及 192.168.2.0/24 网络却并不知晓，因此需为其配置静态路由。同理，R2、R3 也是一样，这里务必要考虑数据的双向性，数据包有去得有回。

2. 关键命令

静态路由的配置命令如下：

```
router(config)# ip route 目的网络 网络掩码 下一跳地址/出接口
```

默认路由的配置命令如下：

```
router(config)# ip route 0.0.0.0 0.0.0.0 下一跳地址/出接口
```

配置及实现

1. 完成基本配置、接口 IP 配置

搭建如图所是的拓扑，完成各设备预配、接口 IP 地址的配置并进行相关测试。

R1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface serial 0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface fastethernet 1/0
R1(config-if)# ip address 192.168.1.254 255.255.255.0
R1(config-if)# no shutdown
```

R2 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R2
R2(config)# interface serial 0/0
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/1
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
```

注意，R2 的 S0/0 及 S0/1 接口都是 DCE 端，因此要配置时钟频率（clock rate）。一条串行链路两端的接口谁是 DCE 谁是 DTE ,在实验室环境中取决于线缆 ,线缆的 DCE 头接着哪端，它就是 DCE 端。当然在实际的项目中，DCE 端往往是运营商那头。

如果使用 GNS 模拟器做实验，那么可以不用配置 clock rate，但是真机环境下，必须配置。

R3 的配置如下

```
Router> enable
Router# configure terminal
Router(config)# hostname R3
R3(config)# interface serial 0/0
R3(config-if)# ip address 192.168.23.3    255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface fastethernet 1/0
R3(config-if)# ip address 192.168.2.254 255.255.255.0
R3(config-if)# no shutdown
```

对于 PC，我们是采用路由器来模拟，需要对其做以下配置

PC1（使用路由器来模拟）：

```
Router> enable
Router# configure terminal
Router(config)# hostname PC1
PC1(config)# no ip routing                          !! 将路由器模拟成 PC 机，关闭路由功能
PC1(config)# ip default-gateway 192.168.1.254       !! 为 PC 机指定网关
PC1(config)#interface fastethernet 0/0
PC1(config-if)# ip address 192.168.1.1 255.255.255.0    !! 为 PC 配置 IP 地址
PC1(config-if)# no shutdown
```

注意：一旦配置 no ip routing 后，路由器就失去了路由功能了，因此必须使用 ip default-gateway 的方式为其设置默认网关，而不能使用默认路由（ip route 0.0.0.0 0.0.0.0）的方式。

PC2 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname PC2
PC2(config)# no ip routing
PC2(config)# ip default-gateway 192.168.2.254
PC2(config)# interface fastethernet 0/0
PC2(config-if)# ip address 192.168.2.1 255.255.255.0
PC2(config-if)# no shutdown
```

在完成上述配置后，PC 与各自的网关、路由器与各直连接口都能够 ping 通。

测试 1：查看各设备直连接口状态。

PC1# show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.1	YES	manual	up	up

PC2#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.2.1	YES	manual	up	up

R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
serial0/0	192.168.12.1	YES	manual	up	up
FastEthernet1/0	192.168.1.254	YES	manual	up	up

R2#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
serial0/0	192.168.12.2	YES	manual	up	up
serial0/1	192.168.23.2	YES	manual	up	up

R3#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
serial0/0	192.168.23.3	YES	manual	up	up
FastEthernet1/0	192.168.2.254	YES	manual	up	up

测试 1 结论：各设备通过命令 show ip intereface brief 检测到各接口状态和协议双 up。

思考：若某接口的 Status up，Protocol down 是为什么？

测试 2：查看路由器直连网段是否能 ping 通。

R1# ping 192.168.1.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/24/72 ms
```

R1#ping 192.168.12.2

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.12.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 8/29/56 ms
```

在 Cisco IOS 设备上使用 ping 命令来探测远端节点的可达性时，如果看到一坨“!”，则表示

目的地可达。

R2 及 R3 的测试同理：

```
R2#ping 192.168.12.1
R2#ping 192.168.23.3
R3#ping 192.168.23.2
R3#ping 192.168.2.1
```

测试 2 结论：直连网段都能 ping 通。

测试 3：测试 PC1 和 PC2 能否正常通信。

```
PC1# ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
U.U.U                                   !! ping 返回 U 表示目的主机不可达
Success rate is 0 percent (0/5)
```

思考：为什么 PC1 与 PC2 无法正常通信？查看相关路由表，首先从 PC1 开始查起：

PC1#show ip route

```
Default gateway is 192.168.1.254
Host                Gateway             Last Use      Total Uses    Interface
ICMP redirect cache is empty
```

PC1 已经设置了缺省网关 192.168.1.254，因此当它发送数据到本地网段外的节点时，数据包会被送到网关，所以我们接着去它的网关也就是 R1 上看看。

R1# show ip route

```
     192.168.1.0/24 is subnetted, 1 subnets
C       192.168.1.0 is directly connected, FastEthernet1/0
     192.168.12.0/24 is subnetted, 1 subnets
C       192.168.12.0 is directly connected, serial 0/0
```

PC1 机可以 ping 通网关，但无法与 PC2 通信，因为沿途路由器没有相应的路由条目，R1 的路由表里并没有到达 192.168.2.0/24 网络的路由。

2. 配置静态路由，使 PC 之间可以互相通信

R1 配置去往 192.168.23.0/24 及 192.168.2.0/24 网段的路由

```
R1(config)# ip route 192.168.2.0 255.255.255.0 192.168.12.2
R1(config)# ip route 192.168.23.0 255.255.255.0 192.168.12.2
```

完成配置后查看路由表，可以看到我们刚才配置的静态路由条目。

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
S       192.168.23.0/24 [1/0] via 192.168.12.2
C       192.168.1.0/24 is directly connected, FastEthernet1/0
S       192.168.2.0/24 [1/0] via 192.168.12.2
```

以上输出的就是 R1 的路由表了：

- 从路由表中可以看到，一共有四个条目，也就是四条路由，其中两条路由标记为“C”，也就是 Connected，意思是直连网段的路由。另外还有两条标记为“S”，也就是 Static 静态路由，正是我们为 R1 配置的两条静态路由。
- 路由条目中的 [1/0] 意思是路由的 [管理距离/度量值]。
- 路由条目中的 via 192.168.12.2，是下一跳 IP 地址。

接着为 R2 配置去往 192.168.1.0/24 及 192.168.2.0/24 网段的路由：

```
R2(config)# ip route 192.168.2.0 255.255.255.0 192.168.23.3
!! R2 配置去往 PC2 所在网段的静态路由
R2(config)# ip route 192.168.1.0 255.255.255.0 192.168.12.1
!! R2 配置去往 PC1 所在网段的静态路由
```

R3 配置去往 192.168.1.0/24 及 192.168.12.0/24 网段的路由：

```
R3(config)# ip route 192.168.1.0 255.255.255.0 192.168.23.2
R3(config)# ip route 192.168.12.0 255.255.255.0 192.168.23.2
```

测试 4：查看两台 PC1 是否能够 ping 通 PC2。

PC1#ping 192.168.2.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/91/124 ms
```

测试 5：查看 PC1 能否 ping 通 192.168.23.0/24 网段

```
PC1#ping 192.168.23.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.23.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 56/91/124 ms
```

3. 将 R1、R3 的静态路由 no 掉，改为默认路由

R1 的配置如下：

```
R1(config)# no ip route 192.168.2.0 255.255.255.0 192.168.12.2
R1(config)# no ip route 192.168.23.0 255.255.255.0 192.168.12.2
R1(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2
```

去掉之前静态路由的配置，配置默认路由。默认路由可以匹配任何目的地，通常用在网络的出口设备上，完成配置后查看路由表：

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
C       192.168.1.0/24 is directly connected, FastEthernet1/0
S*      0.0.0.0/0 [1/0] via 192.168.12.2
```

R3 的配置如下：

```
R3(config)# no ip route 192.168.1.0 255.255.255.0 192.168.23.2
R3(config)# no ip route 192.168.12.0 255.255.255.0 192.168.23.2
R3(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

测试 6：查看两台 PC 机能否通信

```
PC1#ping 192.168.2.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 36/100/276 ms
```

提示 2：注意默认路由使用的场合以及在什么样的情况下才会选择默认路由转发数据。

2.3 RIPv2

实验目的

1. 了解动态路由协议的概念；
2. 了解 RIP 的工作机制；
3. 了解路由汇总的概念及 RIP 自动汇总、手工汇总机制。

拓扑及需求

图 2-3 RIPv2 基础实验 网络拓扑

1. 网络拓扑如上图所示。 PC1 使用模拟器模拟，R3 下联的网段使用 Loopback 接口来模拟，一共开设三个 Loopback 接口，用于模拟 R3 下联的三个网段；
2. 要求 R1、R2、R3 运行 RIPv2，并实现全网可达；
3. 在 R3 上部署 RIP 手工路由汇总，使得 R1、R2 学习到 R3 下联 Loopback 的汇总路由。

关键知识点

Loopback 接口，也叫回环口，是一种虚拟的接口。在网络实验或者项目中我们有可能在除了物理的接口或者物理网卡之外，针对某种特殊的需求还会用到一种虚拟的、稳定的、操作起来类似真实接口或者网卡的这么一种接口，这就是 Loopback 接口。

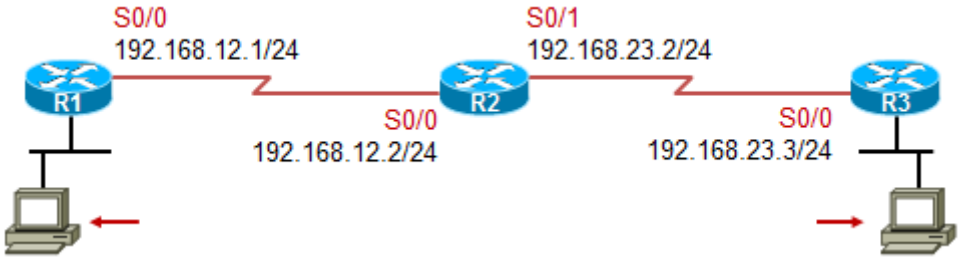

图 2-4 在 R1 及 R2 下各增加一个网段

以图 2-4 所示的网络为例，我们为了实验，需要在 R1 及 R3 上增加一个直连网段，这时候你可能会在他俩的物理接口上各连接一台 PC 来创建两个直连的物理环境。但是首先这需要 R1 及 R3 各出一个物理接口，另外还需增加两台 PC，总而言之比较麻烦。同样的需求，我们可以在 R1 及 R3 上增加一个 Loopback 接口搞定：

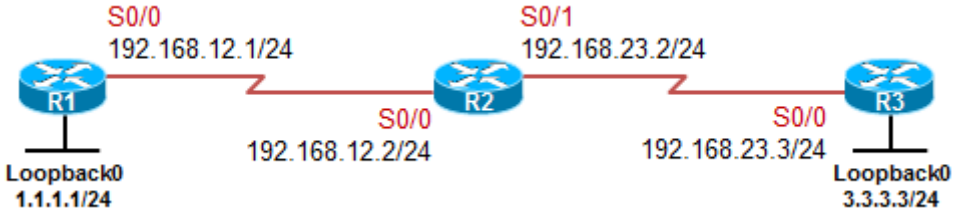

图 2-5 Loopback 接口的使用场景之一

在上图中，我们在 R1 及 R3 上各创建了一个 Loopback0 接口，这个接口是软件的、虚拟接口，使用全局配置命令 interface loopback 加上接口编号创建，创建完成后即可为接口配置 IP 地址。Loopback 接口在手工创建后，除非人为 shutdown，否则不会 Down 掉，因此非常稳定。上图中，R1 的关键性配置如下：

```
R1(config)# interface loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.0
```

这就创建了一个环回口，而且接口的 IP 地址为 1.1.1.1/24，可以将它视为 R1 的一个直连网段，网段上连着一台主机 1.1.1.1。实际上，Loopback 接口的应用场景还有许多，不仅仅局限在上面展示的例子中，还有例如：

- 模拟路由器的直连网段，可用于测试（如上文所述）；
- 可用于设备管理（Loopback 接口比较稳定）；
- 供其他协议使用，如 OSPF、BGP、MPLS 等使用 Loopback 接口地址作为协议的 Router-ID；
- SNMP Traps 消息的源地址；
- 其他用途。

配置及实现

1. 所有的设备完成基本配置（hostname、接口 IP 等）

R1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface serial0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface fastEthernet1/0
R1(config-if)# ip address 172.16.1.254 255.255.255.0
R1(config-if)# no shutdown
```

R2 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R2
R2(config)# interface serial0/0
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/1
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
```

R3 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R3
R3(config)# interface serial0/0
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# exit

R3(config)# interface loopback1
R3(config-if)# ip address 172.16.32.1 255.255.255.0
R3(config)# interface loopback2
R3(config-if)# ip address 172.16.33.1 255.255.255.0
R3(config)# interface loopback3
R3(config-if)# ip address 172.16.34.1 255.255.255.0
```

PC1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname PC1
PC1(config)# no ip routing
PC1(config)# ip default-gateway 172.16.1.254
PC1(config)# interface fastethernet 0/0
```

```
PC1(config-if)# ip address 172.16.1.1 255.255.255.0
PC1(config-if)# no shutdown
```

2. R1、R2、R3 运行 RIPv2

R1 的配置如下：

```
R1(config)# router rip                          #创建 RIP 进程
R1(config-router)# version 2                    #将 RIP 设置为版本 2
R1(config-router)# network 192.168.12.0         #在 S0/0 口上激活 RIPv2
R1(config-router)# network 172.16.0.0           #在 F1/0 口上激活 RIPv2
```

R2 的配置如下：

```
R2(config)# router rip
R2(config-router)# version 2
R2(config-router)# network 192.168.12.0
R2(config-router)# network 192.168.23.0
```

R3 的配置如下：

```
R3(config)# router rip
R3(config-router)# version 2
R3(config-router)# network 192.168.23.0
R3(config-router)# network 172.16.0.0
```

值得注意的是 RIP 在使用 network 命令指定网段时，只支持通告主类网络。R3 有三个 Loopback 接口，将这三个接口激活 RIP 时，只需 network 172.16.0.0 即可，实际上即使你输入诸如 network 172.16.32.0 这样的命令 ,系统也会按 network 172.16.0.0 生效。配置完成后，可做检查一下路由协议的运行情况：

R1#show ip protocols

```
Routing Protocol is "rip"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Sending updates every 30 seconds, next due in 12 seconds
  Invalid after 180 seconds, hold down 180, flushed after 240
  Redistributing: rip
  Default version control: send version 2, receive version 2
    Interface              Send  Recv  Triggered RIP  Key-chain
    Serial0/0              2     2                   !! 两个接口激活 RIPv2
    FastEthernet1/0        2     2
  Automatic network summarization is in effect       !! 自动汇总功能默认开启
  Maximum path: 4
```

```
  Routing for Networks:
    172.16.0.0
    192.168.12.0
  Routing Information Sources:
    Gateway         Distance      Last Update
    192.168.12.2         120      00:00:26
  Distance: (default is 120)                    !! 管理距离 120
```

现在 ,我们尝试让 PC1 ping 一下 R3 的 loopback 接口 ,发现 ping 不通。那么为什么不通呢？此刻我们需要做的就是保证沿途的每一跳路由器上 ,都有到达目的地的路由 ,并且回程的数据也要能顺利的返回到 PC1。因此我们在 R1 上查看路由表：

R1# show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/24 is subnetted, 1 subnets
C          172.16.1.0 is directly connected, FastEthernet1/0
R       192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:23, Serial0/0
```

R1 已经学习到 R2-R3 互联网段的路由 192.168.23.0/24，奇怪的是并没有看到 R3 Loopback 接口的路由，我们再去 R2 上看一下：

R2# show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
R       172.16.0.0/16    [120/1] via 192.168.23.3, 00:00:15, Serial0/1
                         [120/1] via 192.168.12.1, 00:00:01, Serial0/0
C       192.168.23.0/24 is directly connected, Serial0/1
```

R2 出现了一个奇怪的现象，路由表显示它从 R1 及 R3 都学习到了 172.16.0.0/16 路由，而且很显然，R2 将这两条路由都装载进了路由表里，进行“等价负载均衡”。这个现象将造成什么问题呢？R2 会认为，到达 172.16.0.0 网络即可从 R1 走又可以从 R3 走，那么问题来了，R2 要去往 172.16.1.0/24，也就是 R1 下联的网段，如果从 R3 走就肯定无法走通了，这就是问题所在了。为什么会造成这样的问题呢？其实这是 RIPv2 的自动汇总机制使然。由于 R1、R3 都处于“主类网络边界”，拿 R1 来说，有直连网络 192.168.12.0/24 及 172.16.1.0/24，因此 R1 就是两个主类网络的边界 ,RIPv2 在将子网路由 172.16.1.0/24 通告给 192.168.12.0/24 网络时，会将其进行自动汇总，汇总成 172.16.0.0/16，R3 也是类似的动作，就造成了 R2 路由表的诡异现象。

其实自动汇总的本意是为了减小路由条目，优化路由表，减小设备资源的损耗，但是在这个拓扑中却带来了这个问题，因为 172.16.0.0/16 网络的子网分处 R1、R3 两个地方。那么怎么解决呢？办法很简单，就是关闭自动汇总。

R1 的配置如下：

```
R1(config)# router rip
```

```
R1(config-router)# no auto-summary
```

R2 的配置如下：

```
R2(config)# router rip
R2(config-router)# no auto-summary
```

R3 的配置如下：

```
R3(config)# router rip
R3(config-router)# no auto-summary
```

如此一来，明细路由就都被放出来了：

R1 的路由表如下：

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/24 is subnetted, 4 subnets
R          172.16.32.0 [120/2] via 192.168.12.2, 00:00:03, Serial0/0
R          172.16.33.0 [120/2] via 192.168.12.2, 00:00:03, Serial0/0
R          172.16.34.0 [120/2] via 192.168.12.2, 00:00:03, Serial0/0
C          172.16.1.0 is directly connected, FastEthernet1/0
R       192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:03, Serial0/0
```

我们看到，R1 学习到了全网的明细路由，R2、R3 的路由表大家也可以自行查看。这样一来 PC1 就能 ping 通 R3 的所有 Loopback 接口了。

3. 优化路由（手工汇总）

我们在 R1、R2、R3 上关闭自动汇总后，每台路由器就能学习到全网明细路由，满足前面提到的数据访问的需求。但是这也带来了一个新的问题，在关闭自动汇总后，R3 将下联的所有 Loopback 明细路由都放出来了，这对网络中的其他设备而言是一种负担，这个动作丢失了路由汇总原有的利好。怎么办？我们可以用手工汇总，将 172.16.32.0/24、172.16.33.0/24、172.16.34.0/24 这三个子网进行精确汇总，根据我们的计算，得到汇总路由 :172.16.32.0/22，这个汇总地址刚好将三个明细子网囊括。在 R3 上增加的配置如下（注意 RIP 的手工汇总，是配置在接口上的）:

```
R3(config)# interface Serial0/0
R3(config-if)# ip summary-address rip 172.16.32.0 255.255.252.0
```

需要注意的是，RIP 的手工汇总不支持 CIDR。现在，我们再来看一下 R1 的路由表：

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
R          172.16.32.0/22 [120/2] via 192.168.12.2, 00:00:01, Serial0/0
C          172.16.1.0/24 is directly connected, FastEthernet1/0
R       192.168.23.0/24 [120/1] via 192.168.12.2, 00:00:01, Serial0/0
```

我们看到，R1 从 R2 学习到了 192.168.23.0/24 路由，也学到了 R3 发布的手工汇总的路由。再强调一下，RIP 的手工汇总，需要 RIP 版本 2 的支持，另外需要先在 RIP 进程中 no auto-summary 关闭自动汇总功能，然后再在接口上配置手工汇总命令，注意，该命令配置在汇总路由欲对外发布的那个接口上。

2.4 EIGRP

实验目的

1. 了解 EIGRP 基本配置；
2. 了解 EIGRP 自动汇总机制；
3. 了解 EIGRP 手工汇总方式；

拓扑及需求

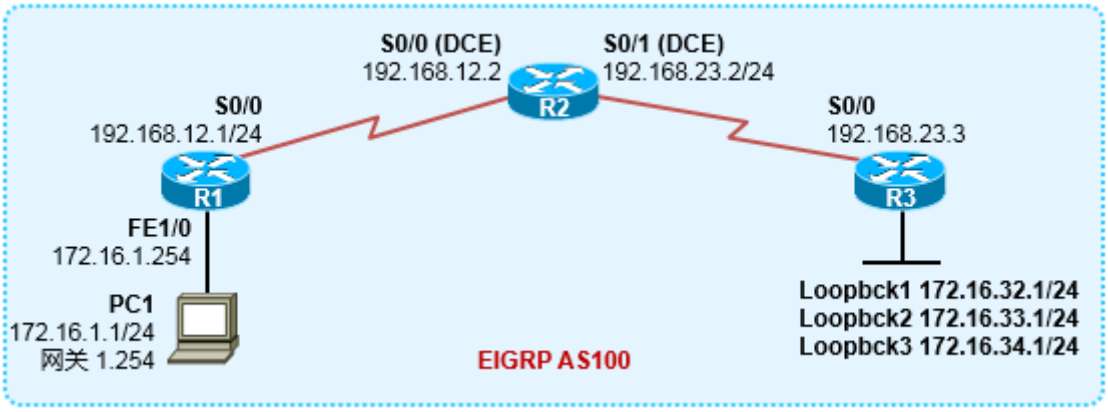

图 2-6 EIGRP 基础实验 网络拓扑

1. 网络拓扑如上图所示。 PC1 使用模拟器模拟，R3 下联的网段使用 Loopback 接口来模拟，一共开设三个 Loopback 接口，用于模拟 R3 下联的三个网段；
2. R1、R2 及 R3 运行 EIGRP 协议，保证全网可达；
3. 在 R3 上对其下属的 Loopback 直连路由进行手工汇总，使得 R1、R2 学习到一条汇总路由。

配置及实现

1. **所有的设备完成基本配置（hostname、接口 IP 等）**

R1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface serial0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface FastEthernet1/0
R1(config-if)# ip address 172.16.1.254 255.255.255.0
R1(config-if)# no shutdown
```

R2 的配置如下：

```
Router>configure terminal
Router(config)# hostname R2
R2(config)# interface serial0/0
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/1
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
```

R3 的配置如下：

```
Router>configure terminal
Router(config)# hostname R3
R3(config)# interface serial0/0
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config)# interface loopback1
R3(config-if)# ip address 172.16.32.1 255.255.255.0
R3(config-if)# interface loopback2
R3(config-if)# ip address 172.16.33.1 255.255.255.0
R3(config-if)# interface loopback3
R3(config-if)# ip address 172.16.34.1 255.255.255.0
```

PC1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname PC1
PC1(config)# no ip routing
PC1(config)# ip default-gateway 172.16.1.254
PC1(config)# interface fastethernet 0/0
PC1(config-if)# ip address 172.16.1.1 255.255.255.0
PC1(config-if)# no shutdown
```

2. 在 R1、R2、R3 上运行 EIGRP

R1 的配置如下：

```
R1(config)# router eigrp 100
R1(config-router)# network 192.168.12.0    0.0.0.255
R1(config-router)# network 172.16.1.0    0.0.0.255
```

R2 的配置如下：

```
R2(config)# router eigrp 100
R2(config-router)# network 192.168.12.0    0.0.0.255
R2(config-router)# network 192.168.23.0    0.0.0.255
```

R3 的配置如下：

```
R3(config)# router eigrp 100
R3(config-router)# network 192.168.23.0    0.0.0.255
R3(config-router)# network 172.16.0.0    0.0.255.255
!!方便起见，用一条命令在所有环回接口上激活 EIGRP
```

完成配置后先检查一下协议的运行情况：

R2#show ip eigrp neighbors

IP-EIGRP neighbors for process 100

H	Address	Interface	Hold (sec)	Uptime	SRTT (ms)	RTO	Q Cnt	Seq Num
1	192.168.23.3	Se0/1	14	00:02:47	86	516	0	5
0	192.168.12.1	Se0/0	11	00:03:00	54	324	0	3

上面就是 R2 的 EIGRP 邻居表，可以看到 R2 已经与 R1 及 R3 建立了 EIGRP 邻居关系。

- H　　邻居的顺序号。
- Address　　邻居路由器的接口地址。
- Interface　　本地连接该邻居的接口。

- Hold 保持时间，在收到邻居任何分组时，认为邻居 down 前等待的最长时间计时器在收到邻居发送的任何报文后复位。
- Uptime 邻居关系已经建立了多长时间。
- SRTT 向邻居路由器发送一个数据包后本路由器收到确认包的时间。
- RTO 当发了 update 包之后 ,对方在该时间内没有回应(确认)则需要进行重传。
- Q Cnt 队列中等待发送的报文数量，正常情况下应该为 0。
- Seq Num 从邻居收到的最后一个更新、查询或应答分组的序列号。

完成上述配置后，我们测试一下 PC1 去 ping 172.16.32.1，结果不通，问题和我们在 RIP 实验中是一样的，EIGRP 也有自动汇总机制，因此 R1 及 R3 都会将本地的 172.16 的子网汇总成 B 类的 172.16.0.0/16 的汇总路由公告出去。我们看一下 R1 的路由表：

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
D          172.16.0.0/16 is a summary, 00:00:37, Null0
C          172.16.1.0/24 is directly connected, FastEthernet1/0
D       192.168.23.0/24 [90/2681856] via 192.168.12.2, 00:00:30, Serial0/0
```

R1 已经学习到了 192.168.23.0，但是有一条奇怪的路由 172.16.0.0/16 指向的是 null0。这条路由是 R1 自动产生的，EIGRP 在路由汇总后，会在本地路由表里自动产生一条指向 Null0 的汇总路由，这是为了防环。

R2#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
D       172.16.0.0/16 [90/2172416] via 192.168.12.1, 00:11:49, Serial0/0
C       192.168.23.0/24 is directly connected, Serial0/1
```

看到 R2 的路由表中，仅有一条汇总路由，学习自 R1。但是为什么没有从 R3 学习到汇总路由呢？实际上 R2 已经学习到了 ,但是由于从 R3 那学到的 172.16.0.0/16 的路由度量值更大，因此没有被优选，从 R2 的 EIGRP 拓扑数据库里能够看到从 R3 学习到的汇总路由：

R2#show ip eigrp topology

```
IP-EIGRP Topology Table for AS(100)/ID(192.168.23.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
           r - reply Status, s - sia Status
P   192.168.12.0/24, 1 successors, FD is 2169856
             via Connected, Serial0/0
P   192.168.23.0/24, 1 successors, FD is 2169856
             via Connected, Serial0/1
P   172.16.0.0/16, 1 successors, FD is 2172416
```

```
        via 192.168.12.1 (2172416/28160), Serial0/0
        via 192.168.23.3 (2297856/128256), Serial0/1        !! 通告距离更大 ,Metric 更大
```

由于 R3 下挂的 172.16 的那些子网是来自 Loopback 接口的，而 R1 下挂的子网是来自快速以太网接口的，因此 R3 产生的汇总路由度量值要大于 R1 产生的那条，故 R2 优选来自 R1 的路由。而 R3 的那条就成了可行后继。现在我们在三台路由器上都关闭 EIGRP 自动汇总。

R1 的配置如下：

```
R1(config)# router eigrp 100
R1(config-router)# no auto-summary
```

R2 的配置如下：

```
R2(config)#router eigrp 100
R2(config-router)# no auto-summary
```

R3 的配置如下：

```
R3(config)#router eigrp 100
R3(config-router)# no auto-summary
```

在关闭了自动汇总后 ,R1 下联的 172.16.1.0/24 子网及 R3 底下的 Loopback 明细路由就能过来了，网络通信也就正常了。

R2#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/24 is subnetted, 4 subnets
D          172.16.32.0 [90/2297856] via 192.168.23.3, 00:00:22, Serial0/1
D          172.16.33.0 [90/2297856] via 192.168.23.3, 00:00:22, Serial0/1
D          172.16.34.0 [90/2297856] via 192.168.23.3, 00:00:22, Serial0/1
D          172.16.1.0 [90/2172416] via 192.168.12.1, 00:00:32, Serial0/0
C       192.168.23.0/24 is directly connected, Serial0/1
```

3. 配置 EIGRP 手工汇总

现在 ,我们在 R3 上 ,为这条三 Loopback 路由进行手工汇总 ,经过计算 ,网络号 172.16.32.0/22 正好可以将这三条 Loopback 明细路由囊括住。现在，R3 增加如下配置：

```
R3(config)#interface Serial0/0
R3(config-if)# ip summary-address eigrp 100 172.16.32.0 255.255.252.0
```

EIGRP 的手工汇总，是配置在接口上的，注意要跟上 eigrp 的 AS 号 100，然后我们上 R2 看看去：

R2#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
```

```
D       172.16.32.0/22 [90/2297856] via 192.168.23.3, 00:00:45, Serial0/1
D       172.16.1.0/24 [90/2172416] via 192.168.12.1, 00:01:02, Serial0/0
C    192.168.23.0/24 is directly connected, Serial0/1
```

这样一来 PC1 访问 R3 及其下的 Loopback 就没有问题了，同时，网络中的路由条目也做到了优化。

2.5 OSPF 单区域

实验目的

1. 了解 OSPF 基本配置；
2. 学会识别 OSPF 路由。

拓扑及需求

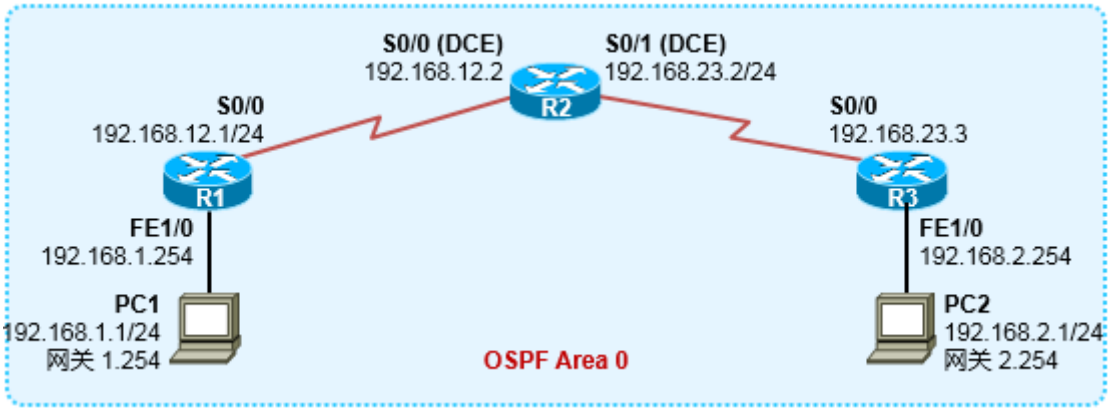

图 2-7 OSPF 单区域实验 网络拓扑

1) 网络拓扑如上图所示，PC1、PC2 使用路由器模拟；
2) 在 R1、R2、R3 上开设 Loopback0 接口，地址分别为 1.1.1.1/32、2.2.2.2/32 及 3.3.3.3/32；
3) 在三台路由器上部署 OSPF，OSPF 的 Router-ID 需设置为各自 Loopback0 接口地址，Loopback0 接口无需激活 OSPF；
4) 要求实现全网可达。

配置及实现

1. 所有的设备完成基本配置（hostname、接口 IP 等）

R1 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface serial 0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface fastethernet 1/0
R1(config-if)# ip address 192.168.1.254 255.255.255.0
R1(config-if)# no shutdown
```

R2 的配置如下：

```
Router> enable
Router# configure terminal
Router(config)# hostname R2
R2(config)# interface serial 0/0
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial 0/1
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
```

R3 的配置如下

```
Router> enable
Router# configure terminal
Router(config)# hostname R3
R3(config)# interface serial 0/0
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface fastethernet 1/0
R3(config-if)# ip address 192.168.2.254 255.255.255.0
R3(config-if)# no shutdown
```

PC1、PC2 的配置不再赘述。

2. R1、R2、R3 运行 OSPF

R1 的配置如下：

```
R1(config)# Interface loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255

R1(config)# router ospf 1                              !!创建 OSPF 进程，使用进程号 1
R1(config-router)# router-id 1.1.1.1                   !!手工指定 Router-ID 为 1.1.1.1
R1(config-router)# network 192.168.1.0 0.0.0.255 area 0      !!在接口 F1/0 上激活 OSPF
R1(config-router)# network 192.168.12.0 0.0.0.255 area 0     !!在接口 S0/0 上激活 OSPF
```

R2 的配置如下：

```
R2(config)# Interface loopback0
R2(config-if)# ip address 2.2.2.2 255.255.255.255

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 192.168.12.0 0.0.0.255 area 0
R2(config-router)# network 192.168.23.0 0.0.0.255 area 0
```

R3 的配置如下：

```
R3(config)# interface loopback0
R3(config-if)# ip address 3.3.3.3 255.255.255.255

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 192.168.23.0    0.0.0.255 area 0
R3(config-router)# network 192.168.2.0    0.0.0.255 area 0
```

OSPF 的 Router-ID 是一个非常重要的东西，是每一台 OSPF 路由器在整个 OSPF 域的标识符，在规划 OSPF 的时候切记不能存在 Router-ID 冲突的现象。Router-ID 在 OSPF 激活后默认会选择本地 Loopback 接口中的最大 IP 地址，如果没有配置 loopback，则选择活跃的物理接口中的最大 IP，因此为了保障 Router-ID 的稳定和可控，建议大家在每台 OSPF 上开设 loopback 接口，在 OSPF 配置模式中，使用 router-id 命令手工指定一下将 OSPF Router-ID 设置为 loopback 接口的 IP 地址。

另外，作为 Router-ID 使用的 Loopback，可以不通告(network)进 OSPF，当然也可以通告，

这要看具体的实现需求，在本实验中，我们在三台路由器上并没有宣告这些 loopback 接口。

3. 查看 OSPF 邻居

在 R1 上 show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
2.2.2.2	0	**FULL**/ -	00:00:32	192.168.12.2	Serial0/0

看到 state 为 FULL，这是 OSPF 邻居关系建立的最终稳定状态，如果我们发现两个 OSPF 之间的邻接关系不是 FULL，那么就要进行错误排查了。因此往往在做 OSPF Trouble shooting 的时候，第一步先看看邻居关系正不正常。接下去再看看 R1 的路由表：

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
O       192.168.23.0/24 [110/128] via 192.168.12.2, 00:03:45, Serial0/0
C       192.168.1.0/24 is directly connected, FastEthernet1/0
O       192.168.2.0/24 [110/129] via 192.168.12.2, 00:03:45, Serial0/0
```

我们看到 R1 学习到了全网的路由，路由的标记为 O，表示该条路由是 OSPF 区域内部路由。

R1#show ip ospf interface　　!!查看 R1 上激活 OSPF 的接口

```
Serial0/0 is up, line protocol is up                    !!第一个接口
  Internet Address 192.168.12.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:07
  Supports Link-local Signaling (LLS)
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 1
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2
  Suppress hello for 0 neighbor(s)
FastEthernet1/0 is up, line protocol is up              !!第二个接口
```

```
Internet Address 192.168.1.254/24, Area 0
Process ID 1, Router ID 1.1.1.1, Network Type BROADCAST, Cost: 1
Transmit Delay is 1 sec, State DR, Priority 1
Designated Router (ID) 1.1.1.1, Interface address 192.168.1.254
No backup designated router on this network
Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
  oob-resync timeout 40
  Hello due in 00:00:09
Supports Link-local Signaling (LLS)
Index 1/1, flood queue length 0
Next 0x0(0)/0x0(0)
Last flood scan length is 0, maximum is 0
Last flood scan time is 0 msec, maximum is 0 msec
Neighbor Count is 0, Adjacent neighbor count is 0
Suppress hello for 0 neighbor(s)
```

现在 PC1 与 PC2 已经可以通信了。

2.6 OSPF 多区域

实验目的

1. 了解 OSPF 基本配置；
2. 了解 OSPF 多区域的概念及配置；
3. 学会识别 OSPF 路由；
4. 了解 OSPF 区域内路由汇总的方法。

拓扑及需求

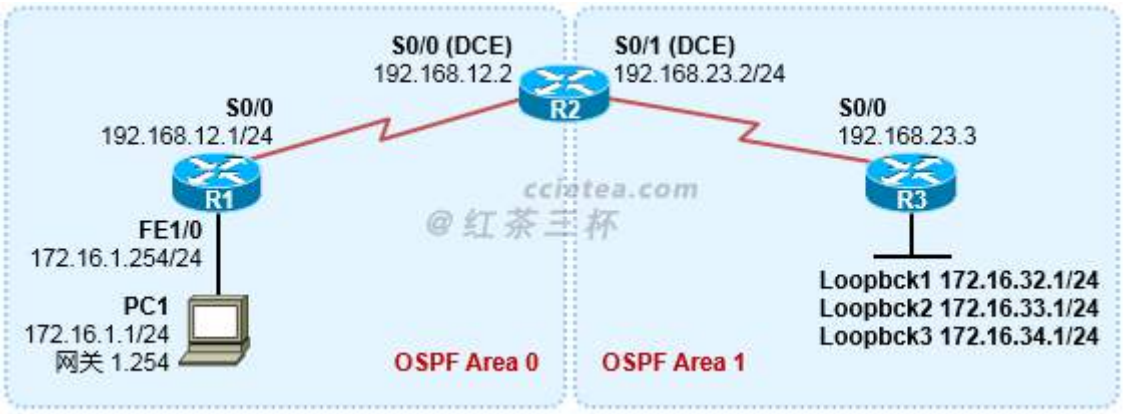

图 2-8 OSPF 多区域实验 网络拓扑

1. 网络拓扑如上图所示；在 R1、R2 及 R3 上开设 Loopback0 接口，地址分别为 1.1.1.1/32、2.2.2.2/32 及 3.3.3.3/32，确保这三台路由器的 OSPF Router-ID 都是各自的 Loopback0 接口地址。Loopback0 接口无需激活 OSPF；
2. R3 创建 Loopback1、Loopback2 及 Loopback3 接口，用于模拟下挂的三个网段；
3. R1、R2、R3 运行 OSPF，R2 为 ABR；保证全网可达（包括 R3 的 Loopback1-3）；
4. 默认情况下 R1、R2 学习到的 R3 下联的 Loopback1、2、3 接口所在网段的路由都是/32 位的主机路由，做适当的修改，使得 R1、R2 学习到的这些路由掩码恢复为这些 Loopback 的实际掩码长度；
5. 部署 OSPF 路由汇总，将 R3 下联的三个 Loopback 网段进行汇总。

配置及实现

1. 所有的设备完成基本配置（hostname、接口 IP 等）

各设备的基本配置不再赘述，大家自行完成。

2. R1、R2 及 R3 运行 OSPF

R1 的配置如下：

```
R1(config)# Interface loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255

R1(config)# router ospf 1                                  !!创建 OSPF 进程，使用进程号 1
R1(config-router)# router-id 1.1.1.1                       !!手工指定 RouterID 为 1.1.1.1
R1(config-router)# network 172.16.1.0 0.0.0.255 area 0     !!在接口 F1/0 上激活 OSPF
R1(config-router)# network 192.168.12.0 0.0.0.255 area 0   !!在接口 S0/0 上激活 OSPF
```

R2 的配置如下：

```
R2(config)# Interface loopback0
R2(config-if)# ip address 2.2.2.2 255.255.255.255

R2(config)# router ospf 1
R2(config-router)# router-id 2.2.2.2
R2(config-router)# network 192.168.12.0 0.0.0.255 area 0
R2(config-router)# network 192.168.23.0 0.0.0.255 area 1
```

R2 的配置是需要格外留意的，它是 ABR，一个接口再 area 0，另一个接口再 area 1。

R3 的配置如下：

```
R3(config)# Interface loopback0
R3(config-if)# ip address 3.3.3.3 255.255.255.255
R3(config)# Interface loopback1
R3(config-if)# ip address 172.16.32.1 255.255.255.0
R3(config)# Interface loopback2
R3(config-if)# ip address 172.16.33.1 255.255.255.0
R3(config)# Interface loopback3
R3(config-if)# ip address 172.16.34.1 255.255.255.0

R3(config)# router ospf 1
R3(config-router)# router-id 3.3.3.3
R3(config-router)# network 192.168.23.0 0.0.0.255 area 1
R3(config-router)# network 172.16.32.1 0.0.0.0 area 1
R3(config-router)# network 172.16.33.1 0.0.0.0 area 1
R3(config-router)# network 172.16.34.1 0.0.0.0 area 1
```

3. 调整 Loopback 路由

完成上述配置后，先看看 R1 的路由表：

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
        172.16.0.0/32 is subnetted, 3 subnets
O IA       172.16.33.1 [110/129] via 192.168.12.2, 00:00:13, Serial0/0
O IA       172.16.32.1 [110/129] via 192.168.12.2, 00:02:11, Serial0/0
O IA       172.16.34.1 [110/129] via 192.168.12.2, 00:00:02, Serial0/0
O IA 192.168.23.0/24 [110/128] via 192.168.12.2, 00:04:11, Serial0/0
```

```
C       192.168.1.0/24 is directly connected, FastEthernet1/0
```

我们看到 R1 学习到了全网的路由，标记为 O IA 的路由指的是区域间的路由，IA=inter area，是从其他区域过来的路由。另一点需要我们留意的是 R1 学习到的 R3 的三个 LOOPBACK 接口路由，在路由表中都是/32 的主机路由，而我们给 R3 这三个 LOOPBACK 接口实际分配的是/24 的掩码，为什么到了 R1 的路由表里就变成了/32？这其实是 OSPF 的一个特性，OSPF 认为“Loopback interface is treated as a stub Host”，也就是将 Loopback 当做一个直连的主机，因此无论你给 LOOPBACK 接口分配什么掩码，OSPF 在对外通告 LOOPBACK 接口网段路由的时候（通过 LSA）都以./32 位的形式。那么如何还原 LOOPBACK 接口的“本来面貌”呢？很简单，我们只需要在 R3 的这三个 Loopback 接口下：

```
R3(config)# Interface loopback 1
R3(config-if)# Ip ospf network point-to-point   !!将 Loopback 接口的 OSPF 类型修改为点到点
R3(config)# Interface loopback 2
R3(config-if)# Ip ospf network point-to-point
R3(config)# Interface loopback 3
R3(config-if)# Ip ospf network point-to-point
```

如此一来，给 LOOPBACK 在 LSA 中被通告的时候，掩码就是该接口所配置的/24。我们再去 R1 上看看。

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
        172.16.0.0/24 is subnetted, 3 subnets
O IA       172.16.32.0 [110/129] via 192.168.12.2, 00:00:11, Serial0/0
O IA       172.16.33.0 [110/129] via 192.168.12.2, 00:00:01, Serial0/0
O IA       172.16.34.0 [110/129] via 192.168.12.2, 00:00:01, Serial0/0
O IA 192.168.23.0/24 [110/128] via 192.168.12.2, 00:05:01, Serial0/0
C       192.168.1.0/24 is directly connected, FastEthernet1/0
```

现在，这些路由的掩码长度在 R1 的路由表中已经变成/24 的了。

4. OSPF 区域内路由汇总

现在我们要对网络中的路由进行优化，R1 其实没有必要知道 R3 下联所有 Loopback 的明细，因此我们做手工汇总（OSPF 与 RIP 及 EIGRP 不同，不支持自动汇总），具体的操作方法是在 ABR 也就是 R2 上来部署对区域内路由的汇总，R2 配置增加如下：

```
R2(config)# router ospf 1
R2(config-router)# area 1 range 172.16.32.0 255.255.252.0
```

上面这条命令的意思是，如果 area1 内存在 172.16.32.0/22 的子网路由，则将这些路由汇总

成 172.16.32.0/22，而不再向 area0 通告子网明细路由。

OSPF 支持两种手工路由汇总的方式，一种是部署在 ABR 上，另一种则是部署在 ASBR 上的，本例中我们使用到的是前者，至于后者，已经超出了本手册的内容范围。我们再去 R1 上看看：

R1#show ip route

```
C       192.168.12.0/24 is directly connected, Serial0/0
        1.0.0.0/32 is subnetted, 1 subnets
C          1.1.1.1 is directly connected, Loopback0
        172.16.0.0/22 is subnetted, 1 subnets
O IA       172.16.32.0 [110/129] via 192.168.12.2, 00:00:10, Serial0/0
O IA 192.168.23.0/24 [110/128] via 192.168.12.2, 00:05:41, Serial0/0
C       192.168.1.0/24 is directly connected, FastEthernet1/0
```

完成上述配置后，R1 不再学习到 R3 的 Loopback 明细路由，而是学习到一条汇总的路由，它的路由表规模自然就减小了。而此时 PC 是能够 ping 通 R3 的三个 Loopback 接口的。

3 交换篇

3.1 二层交换基础

实验目的

1. 了解二层交换机工作原理；
2. 理解 VLAN 的概念，掌握 VLAN 的配置；
3. 理解 trunk 的概念，掌握 trunk 的配置。

拓扑及需求

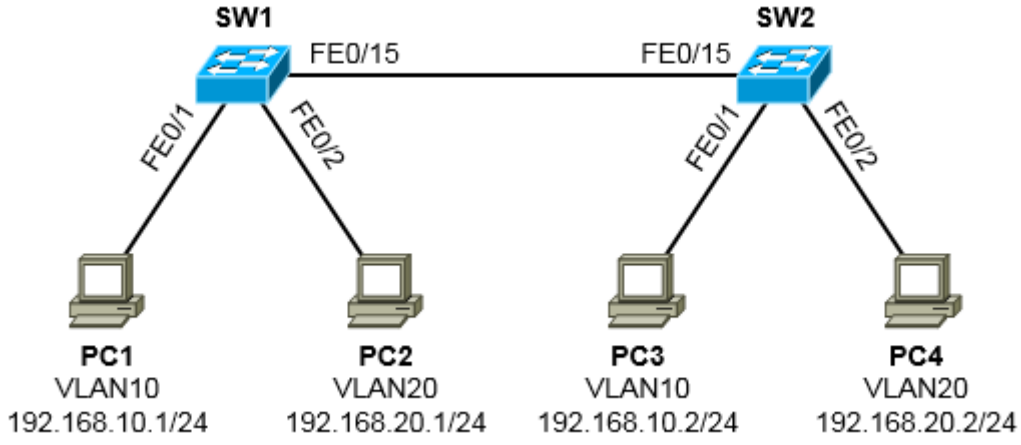

图 3-1 二层交换基础实验 网络拓扑

1) 网络拓扑如上图所示；
2) SW1 及 SW2 都创建 VLAN10 及 VLAN20，按照上图所示，将接口添加到特定的 VLAN；
3) 测试 PC 之间的连通性，要求相同 VLAN 内的 PC 能够相互通信。

配置及实现

1. **完成所有 PC 的配置**

   PC 的配置不再赘述。

2. 完成交换机的配置

SW1 的配置如下：

```
SW1# config terminal
SW1(config)# vlan 10                                    !!创建 VLAN10
SW1(config-vlan)# exit
SW1(config)# vlan 20                                    !!创建 VLAN20
SW1(config-vlan)# exit

SW1(config)# interface fastethernet0/1
SW1(config-if)# switchport mode access                  !!将接口类型修改为 access 模式
SW1(config-if)# swtichport access vlan 10               !!将接口添加到特定的 VLAN
SW1(config-if)# interface fastethernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# swtichport access vlan 20
SW1(config-if)# interface fastethernet0/15
SW1(config-if)# switchport mode trunk                   !!将接口模式定义为 trunk
SW1(config-if)# switchport trunk encapsulation dot1q    !!指定 trunk 封装协议为 dot1q
```

SW2 的配置如下：

```
SW2# config terminal
SW2(config)# vlan 10
SW2(config-vlan)# exit
SW2(config)# vlan 20
SW2(config-vlan)# exit

SW2(config)# interface fastethernet0/1
SW2(config-if)# switchport mode access
SW2(config-if)# swtichport access vlan 10
SW2(config-if)# interface fastethernet0/2
SW2(config-if)# switchport mode access
SW2(config-if)# swtichport access vlan 20
SW2(config-if)# interface fastethernet0/15
SW2(config-if)# switchport mode trunk
SW2(config-if)# switchport trunk encapsulation dot1q
```

如果是使用 GNS 模拟器做这个实验，有两个地方需要注意：

1) 给**模拟交换机的设备（可以使用 C3600 平台模拟）**安装的模块选择如下，选择交换模块 NM-16ESW：

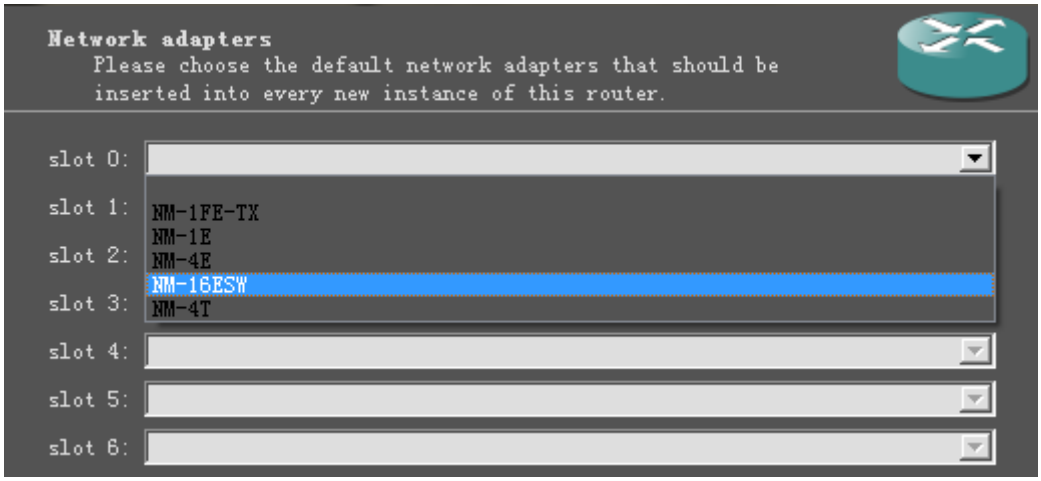

图 3-2 在 GNS 中为设备添加 NM-16ESW 交换模块

2) **在 GNS 模拟器上做交换实验，如需创建 VLAN，配置在 vlan database 中进行，拿 SW1 举例（仅限于在模拟器上）：**

```
SW1# vlan database
SW1(vlan)# vlan 10              !!创建 vlan10
VLAN 10 added:
     Name: VLAN0010
SW1(vlan)# vlan 20              !!创建 vlan10
VLAN 20 added:
     Name: VLAN0020
SW1 (vlan)# exit                !!注意必须使用 exit 退出，否则创建的 VLAN 不会生效
APPLY completed.
Exiting....
```

3. **查看及验证**

SW1# show vlan　　!!在 GNS 模拟器上这条命令为 show vlan-switch

```
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Fa0/0, Fa0/3, Fa0/4, Fa0/5
                                                 Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                                 Fa0/10, Fa0/11, Fa0/12, Fa0/13
                                                 Fa0/14
10   VLAN0010                          active    Fa0/1
```

```
20   VLAN0020                         active    Fa0/2
1002 fddi-default                     active
1003 token-ring-default               active
1004 fddinet-default                  active
1005 trnet-default                    active
```

从上面的输出可以看到我们已经成功创建了两个 VLAN：10 和 20，并且 FE0/1 及 FE0/2 口都划分到了相应的 VLAN。如果在模拟器上进行交换实验，使用 show vlan-switch 来查看 vlan 信息。

SW1# show interfaces trunk !!查看 trunk 接口

```
Port        Mode         Encapsulation  Status        Native vlan
Fa0/15      on           802.1q         trunking      1

Port        Vlans allowed on trunk
Fa0/15      1-1005

Port        Vlans allowed and active in management domain
Fa0/15      1,10,20

Port        Vlans in spanning tree forwarding state and not pruned
Fa0/15      1,10,20
```

可以看到，F0/15 已经工作在 trunk 模式，使用的封装协议是 802.1q。

接下去 PC1 去 ping PC3，或者 PC2 去 ping PC4，同 VLAN 的都能 ping 通，但是不同 VLAN 的 PC 到目前为止无法互访。

3.2 使用以太网子接口实现 VLAN 之间的互访

实验目的

1. 深入理解 VLAN 及 trunk 的概念；
2. 掌握通过以太网子接口实现 VLAN 间互访的方法。

拓扑及需求

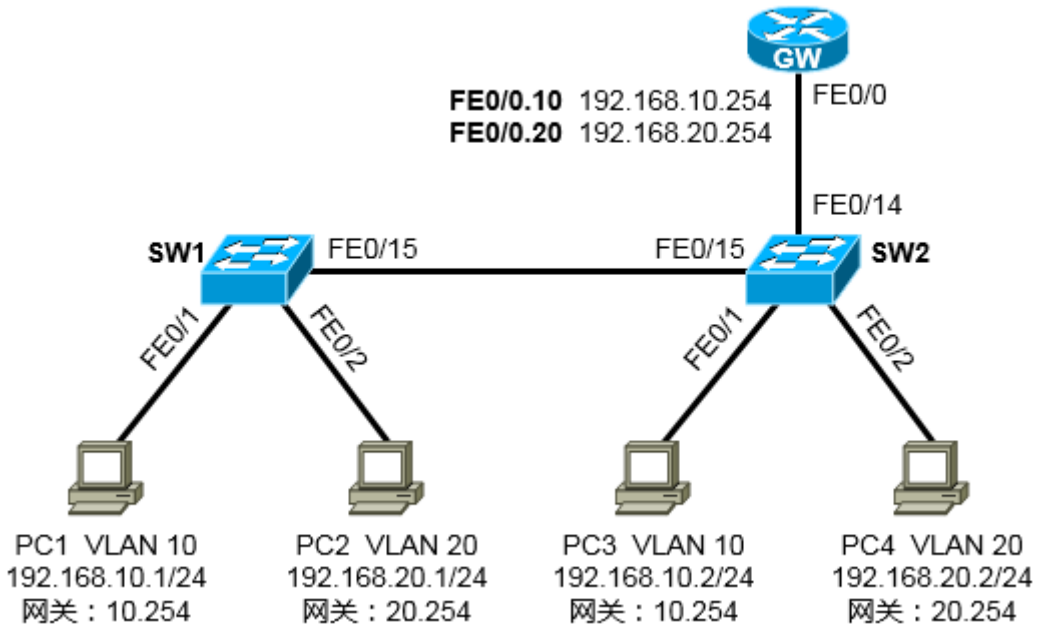

图 3-3 单播路由实验 网络拓扑

1. 网络拓扑如上图所示；PC 的网关均在路由器上；
2. SW1 及 SW2 都创建 VLAN10 及 VLAN20，并且根据拓扑所示将接口加入特定的 VLAN；
3. 将交换机的适当端口配置为 trunk 并设置封装协议为 Dot1q；
4. 在路由器上通过创建子接口的方式使得所有的 PC 都能够互相通信。

配置及实现

1. **完成所有 PC 的配置**

   PC 的配置不再赘述。

2. **完成交换机的配置**

   SW1 的配置如下：

```
SW1# config terminal
SW1(config)# vlan 10
SW1(config-vlan)# exit
SW1(config)# vlan 20
SW1(config-vlan)# exit
```

```
SW1(config)# interface fastethernet0/1
SW1(config-if)# switchport mode access
SW1(config-if)# swtichport access vlan 10
SW1(config)# interface fastethernet0/2
SW1(config-if)# switchport mode access
SW1(config-if)# swtichport access vlan 20
SW1(config)# interface fastethernet0/15
SW1(config-if)# switchport trunk encapsulation dot1q
SW1(config-if)# switchport mode trunk
```

SW2 的配置如下：

```
SW2# config terminal
SW2(config)# vlan 10
SW2(config-vlan)# exit
SW2(config)# vlan 20
SW2(config-vlan)# exit

SW2(config)# interface fastethernet0/1
SW2(config-if)# switchport mode access
SW2(config-if)# swtichport access vlan 10
SW2(config-if)# interface fastethernet0/2
SW2(config-if)# switchport mode access
SW2(config-if)# swtichport access vlan 20
SW2(config-if)# interface fastethernet0/14          !! 这是与路由器跑单臂的接口必须是trunk
SW2(config-if)# switchport trunk encapsulation dot1q
SW2(config-if)# switchport mode trunk
SW2(config-if)# interface fastethernet0/15
SW2(config-if)# switchport trunk encapsulation dot1q
SW2(config-if)# switchport mode trunk
```

这里一定要注意，SW2 的 FE0/14 及 FE0/15 口都需要同时承载多 VLAN 的数据，因此必须配置成 trunk，尤其要注意 FE0/14 接口，由于对端是路由器的子接口，因此必须配置为 Trunk 类型。当然，如果在其他场景中，路由器使用物理接口（不适用子接口）与交换机对接，则交换机这一段的接口通常配置为 access 类型。

3. 完成路由器 GW 的配置

```
Router(config)# hostname GW
```

```
GW(config)# interface FastEthernet 0/0
GW(config-if)# no shutdown                    !!注意一定要将物理接口 no shutdown
GW(config)# interface FastEthernet 0/0.10     !!这个子接口作为 VLAN10 的网关
GW(config-if)# encapsulation dot1Q 10         !!设定封装协议，10 表示的是 vlan tag ID
GW(config-if)# ip address 192.168.10.254 255.255.255.0
GW(config)# interface FastEthernet 0/0.20     !!这个子接口作为 VLAN20 的网关
GW(config-if)# encapsulation dot1Q 20
GW(config-if)# ip address 192.168.20.254 255.255.255.0
```

4. 连通性测试

完成上述配置后，VLAN10 与 VLAN20 之间的用户就能够互访了。这个实验的拓扑形象点理解如下：

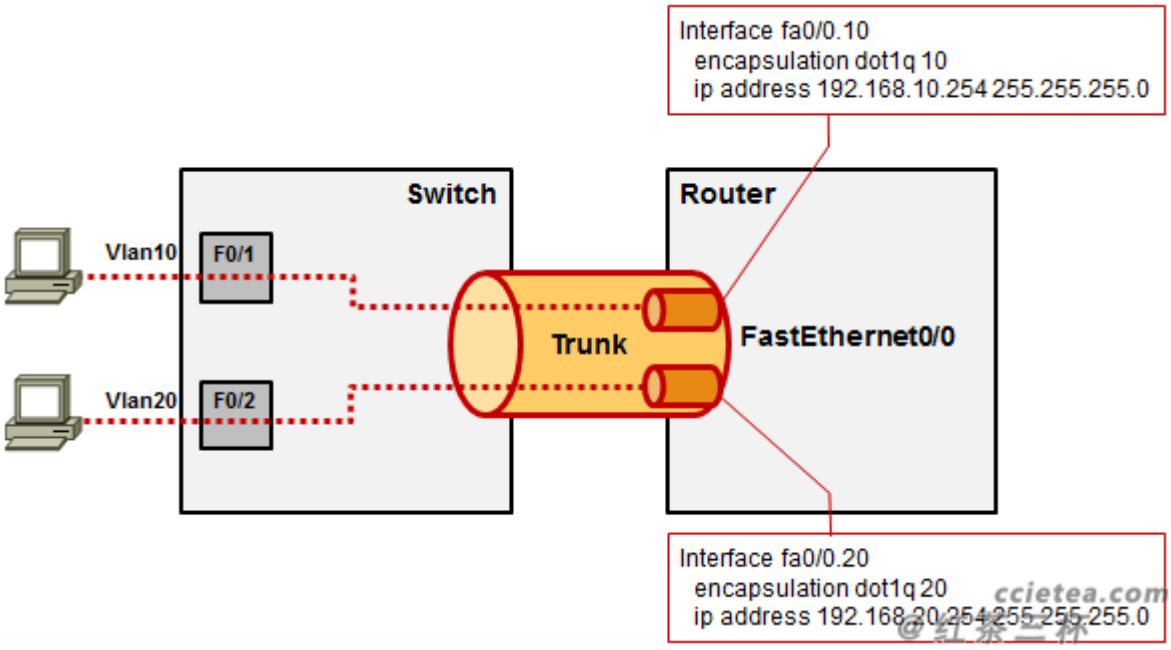

图 3-4 形象地理解单播路由

3.3 二层交换机的管理 VLAN

实验目的

1. 了解二层交换机管理 VLAN 的概念；

2. 了解二层交换机设备管理的部署方法。

拓扑及需求

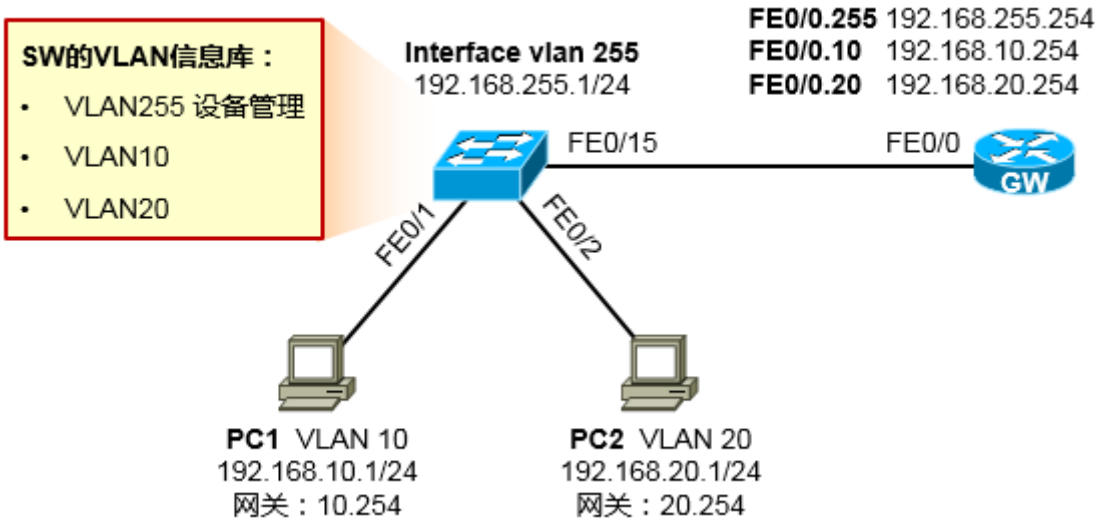

图 3-5 二层交换机的管理实验 网络拓扑

1. 这个环境虽然简单，但是初学者理解起来还是有一定的困难的。交换机下联 2 台 PC，又连接一台路由器。路由器作为内网 VLAN10、VLAN20 用户的网关。交换机增加一个 VLAN255 用于其自身的设备管理，交换机的管理 IP 为 192.168.255.1，网关在路由器上；
2. 交换机创建 VLAN10、VLAN20，这是用户 VLAN，是 PC 用户的 VLAN；
3. 路由器 FE0/0 口划分子接口做单臂，作为 VLAN10、VLAN20 及 VLAN255 的网关；
4. 要求 VLAN10、VLAN20 之间的用户能够互访，同时只允许 VLAN10 的用户 Telnet 交换机进行设备的管理。

关键知识点

这个实验中，涉及到的主要知识点有：VLAN 的配置、单臂路由的概念及配置，同时还有一个，交换机的管理 VLAN 及设备管理。

我们知道，在一个园区网中数量最多的设备一般而言应该是交换机（二层及三层交换机），其中又以二层交换机的数量居多，二层交换机在典型的三层结构中处于“接入层”，主要的任务是为终端的 PC 和服务器提供接入，同时划分 VLAN 隔离广播域，再者运行 STP 来实现二层的防环机制。一个中小型的园区网中，二层交换机的数量往往是上百台，这些设备在刚刚在客户现场被拆箱后，一般是由工程师现场用 Console 线缆一台台的调试的（不要惊讶，笔者的纪录是一个下午的时间，个人完成近 100 台交换机的调试任务）。这些交换机在调试好之后，就会被安装到客户现场的各个

机房或者弱电间去，一切妥当后就正式上线运行了。

那么这就有一个问题，在设备上线后，如果我们要变更设备的配置、要管理这些交换机怎么办？难道要拿着笔记本电脑带着 console 线到机房去现场调试么？这种屌丝级的方法完全不可理喻嘛，对了，可以通过 telnet 或者 SSH 来远程管理。路由器上的 telnet 我们已经很熟悉了，只要是三层可达，就能 telnet 到路由器，那么接下去我们来看看，如何管理交换机。

这里我们讲的是二层交换机，大家都知道，二层交换机是无法读取报文的三层头部的，它压根不会去看三层的 IP 头，但是这不影响二层交换机自己拥有一个 IP 地址。在路由器上，我们是给路由器的物理接口配置 IP 地址，而对于二层交换机，我们是在 VLAN 接口上配置 IP 地址，VLAN 接口也称为 SVI（交换式虚拟接口），是跟 VLAN 对应的一个逻辑的、虚拟的接口。一台二层交换机，只能够给一个 VLAN 接口分配 IP 地址。考虑一个最简单的模型：

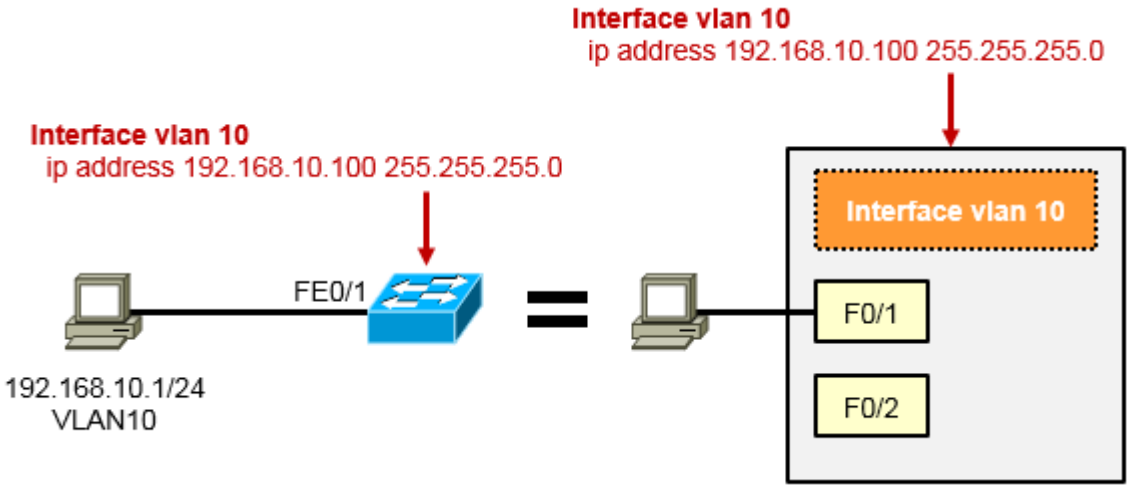

图 3-6 交换机管理 VLAN 的概念

如上图，PC 要想 Telnet 交换机，首先 PC 要能 ping 通交换机，其次交换机上要激活 VTY 线路并配置密码。我们在二层交换机上创建一个 VLAN10，将连接 PC 的接口 FE0/1 口划入 VLAN10，同时给二层交换机的 VLAN10 的三层逻辑接口（VLAN 接口）配置一个与 PC 在同一个网段的 IP 地址。如此一来，PC 就能访问到交换机了。这时 PC 的 IP 地址与交换机自身的 IP 是同网段的，而且两者在同一个广播域（VLAN），因此可以进行二层直接通信。

但是 PC 与交换机的管理 VLAN 重叠，万一下面有一台 PC 配置的 IP 地址与交换机存在冲突那可就麻烦了，因此我们考虑给交换机划分一个单独的 VLAN 用于自身被管理，这个 VLAN 适用于整个交换网络 ,统一的 VLAN 统一的 IP 规划 ,它就是管理 VLAN ,因此管理 VLAN 并不一定是 VLAN1，许多人在这里存在误解。

一般情况下，关于管理 VLAN 我们会使用一个较为“生僻的”VLAN ID 和 IP 编制，例如本实验中的 VLAN255，以及网段 192.168.255.0/24。

问题又来了，给交换机弄一个单独的设备管理 VLAN 固然可以起到与用户 VLAN 隔离的作用，但是这样一来用户不就无法访问到交换机了么？这就需要借助三层设备—例如路由器了。与此同时，

由于二层交换机没有路由功能，无法像路由器哪样拥有一个 IP 路由表，因此，你还需给交换机配置一个默认网关，强调一下，这里的网关是交换机自身的网关，而不是 PC 的网关。

配置及实现

先不忙着做实验，我们来看看这个实验的拓扑：

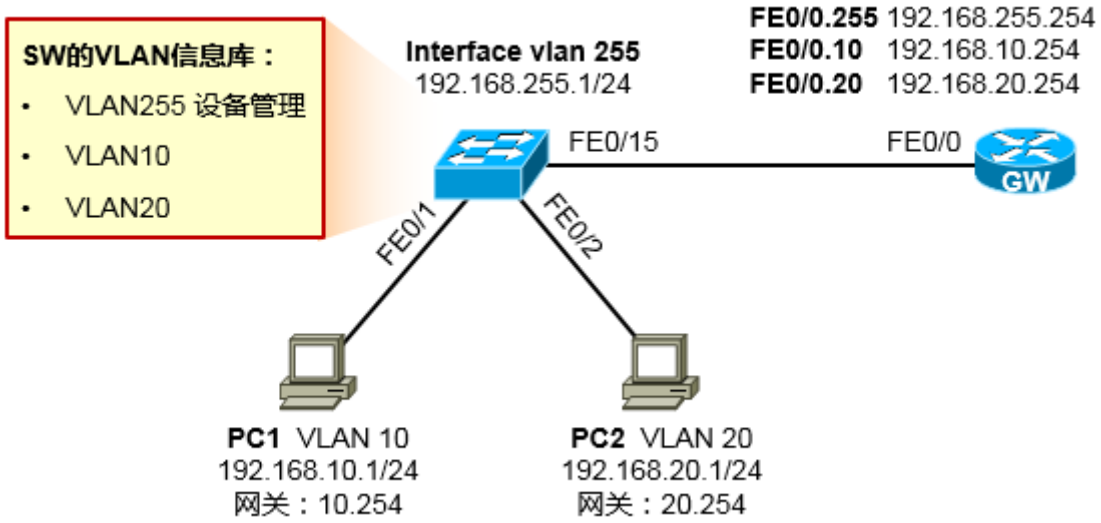

图 3-7 回顾一下实验拓扑

这个拓扑其实可以用一个更形象的方式来理解：

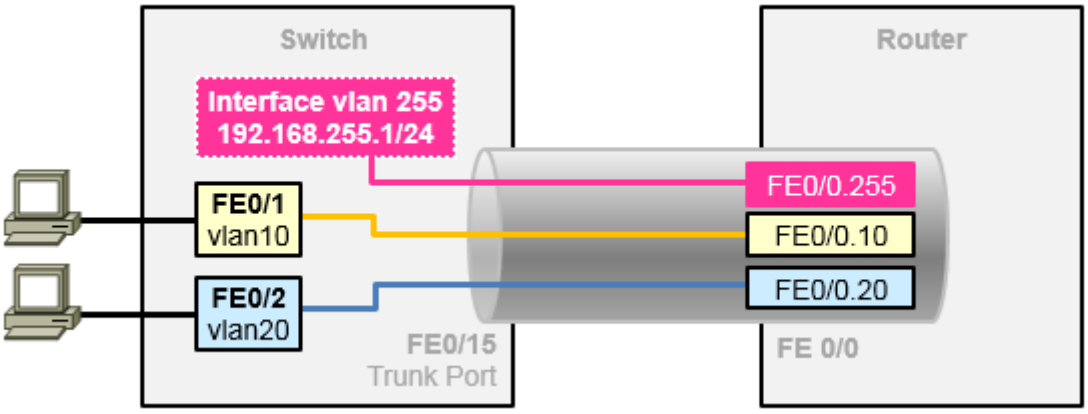

图 3-8 形象地理解本实验的网络拓扑

这样总能看懂了吧？于是当 PC 要管理交换机时（例如要 Telnet 交换机），直接 telnet 192.168.255.1，数据包会被发送给 PC 自己的网关，该网关其实在 Router 的子接口上的，再经由 Router 路由到子接口 FE0/0.255，然后走二层到交换机。有一点值得提醒的是，为了让回程的数据能够顺利返回 PC，我们还要给这台二层交换机配个默认网关，否则 PC 无法正常管理交换机。下面来看具体的配置：

1. **完成 PC1 及 PC2 的配置**

   这个就不再赘述了。

2. **完成交换机的配置**

```
switch# config terminal
switch(config)# vlan 10
switch(config-vlan)# exit
switch(config)# vlan 20
switch(config-vlan)# exit
switch(config)# vlan 255                                    !!这是管理 VLAN
switch(config-vlan)# exit

switch(config)# interface fastethernet0/1
switch(config-if)# swtichport mode access
switch(config-if)# swtichport access vlan 10
switch(config-if)# interface fastethernet0/2
switch(config-if)# swtichport mode access
switch(config-if)# swtichport access vlan 20
switch(config-if)# interface fastethernet0/15
switch(config-if)# switchport trunk encapsulation dot1q
switch(config-if)# switchport mode trunk
switch(config-if)# exit

switch(config)# interface vlan 255
switch(config-if)# ip address 192.168.255.1 255.255.255.0
!!给 VLAN255 的 SVI 口配置 IP 地址，这个 IP 地址就是交换机的管理 IP
switch(config-if)# exit

switch(config)# ip default-gateway 192.168.255.254          !!为交换机指定网关

switch(config)# line vty 0 4                                !!配置 VTY
switch(config-line)# password ccietea
switch(config-line)# login
switch(config-line)# exit
switch(config)#
```

注意，本实验中的交换机为二层交换机。如果使用三层交换机（例如 C3640）做本实验，则需先在全局配置模式下 no ip routing 关闭设备的路由功能，让其模拟一台二层交换机，如果

不去 no ip routing，则 ip default-gateway 指定的缺省网关是无效的，此时可以使用 ip route 0.0.0.0 0.0.0.0 的方式来替代 ip default-gateway 命令。注意，ip defaut-gateway 这条命令，是给交换机自己用的，而不是为下联的 PC 配置网关，关于这点，许多初学者经常搞混。

3. 完成路由器的配置

```
Router(config)# hostname GW
GW(router)# interface fastethernet 0/0
GW(router-if)# no shutdown                    !!注意一定要将物理接口打开
GW(router)# interface fastethernet 0/0.10     !!这个子接口作为 VLAN10 的网关
GW(router-if)# encapsulation dot1Q 10
GW(router-if)# ip address 192.168.10.254 255.255.255.0
GW(router-if)# interface fastethernet 0/0.20  !!这个子接口作为 VLAN20 的网关
GW(router-if)# encapsulation dot1Q 20
GW(router-if)# ip address 192.168.20.254 255.255.255.0
GW(router-if)# interface fastethernet 0/0.255 !!这个子接口作为 VLAN255 的网关
GW(router-if)# encapsulation dot1Q 255
GW(router-if)# ip address 192.168.255.254 255.255.255.0
```

GW#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	up	up
FastEthernet0/0.10	192.168.10.254	YES	manual	up	up
FastEthernet0/0.20	192.168.20.254	YES	manual	up	up
FastEthernet0/0.255	192.168.255.254	YES	manual	up	up

4. 验证及扩展

完成上述配置后，PC1 及 PC2 就都能 telnet 上交换机进行管理了。这就是二层交换机的管理概念。

PC1#telnet 192.168.255.1

```
Trying 192.168.255.1 ... Open


User Access Verification
Password:
SW>
```

在实际的部署中，如果直接将交换机暴露在网络中，内网所有的 PC 都能随意登录，那是存在风险的，我们还可以在交换机上增加如下配置，来限制管理交换机的网段，例如我们只让

192.168.10.0/24 这个网段的用户管理交换机：

```
Switch(router)# access-list 1 permit 192.168.10.0 0.0.0.255
Switch(router)# line vty 0 4
Switch(router-line)# access-class 1 in
```

如此一来，只有 192.168.10.0/24 网段内的 PC 才能够 telnet 到交换机。

3.4 使用 SVI 实现 VLAN 间的互访

实验目的

1. 理解 SVI（交换式虚拟接口，也称为 VLAN 接口）的概念；
2. 掌握 SVI 实现 VLAN 间互访的方法。

拓扑及需求

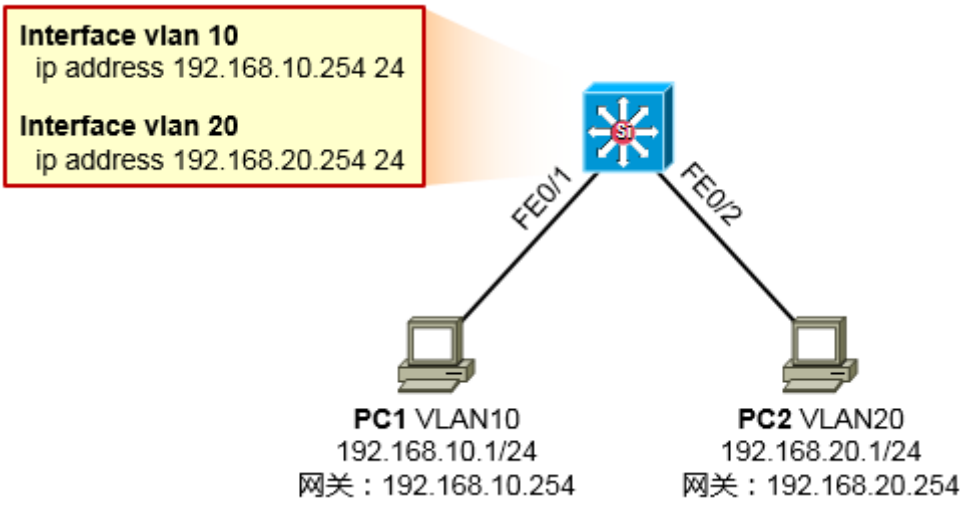

图 3-9 三层交换机基础实验 网络投票

1. PC1 及 PC2 分别属于 vlan10 及 vlan20；
2. 交换机为三层交换机，支持三层功能；
3. 两台 PC 的网关均在交换机上，要求 PC1 与 PC2 能够互访。

关键知识点

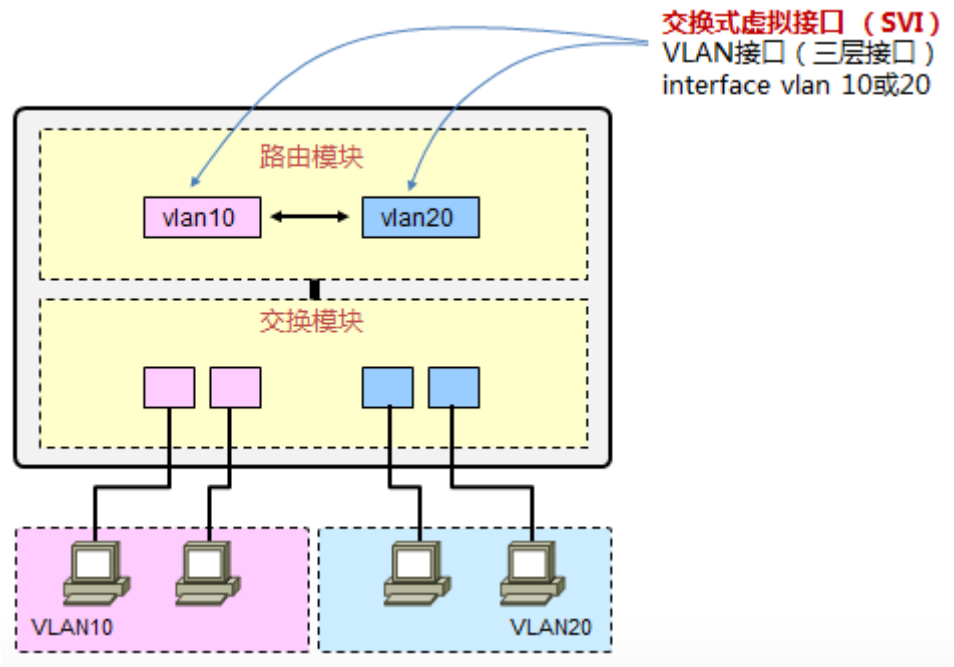

图 3-10 SVI 的概念

在理解了单臂路由之后，我们再来看看三层交换机是如何实现 VLAN 间的数据互访的，我们从这里为切入点，开始理解并部署三层交换。我们知道二层交换机是可以实现二层交换的，它看的是数据帧，对帧头的二层信息进行读取并且根据自己的 MAC 表进行转发。而三层交换机相当于在二层交换机的基础上，多了个路由模块，于是乎它就能支持路由功能了，当然，也能够支持路由协议、支持三层数据的转发、支持 IP 路由查找、支持三层接口等等。

先来认识一下第一种三层接口：SVI 交换式虚接口，SVI 是一个逻辑接口，也就是说不是一个物理接口，当我们在交换机上创建了一个 VLAN 之后，紧接着就可以创建一个与这个 VLAN 对应的 SVI 接口，例如我们创建了 VLAN10，那么 VLAN10 对应的 SVI 接口就是 interface vlan10 或者叫 SVI10，这个 SVI10 是一个三层接口，你可以为这个 SVI 口配置 IP 地址，与 VLAN10 内的 PC 用户的 IP 地址同一网段，那么这样一来，VLAN10 内的用户就能够将网关指向这个 SVI 接口，当 VLAN10 的 PC 需要访问本网段以外的网络时他们将数据交给网关，也就是 SVI10，再由三层交换机去做路由查找及数据转发。实际上，在这个理解过程中，我们可以拿单臂路由那个模型对类比。

所以看上面这图，在三层交换机上创建了两个 VLAN：10 和 20，同时为两个 VLAN 的 SVI 分配了地址作为各自 VLAN 的用户网关，这样一来，这台交换机的路由表里就有了两个 VLAN 网段的路由。那么当两 VLAN 之间要互访时，VLAN10 的用户将数据丢给自己的网关，也就是 VLAN10 的 SVI，数据到了 SVI10 之后，三层交换机查表，发现目的地是 VLAN20 的所在网段，因此将数据从 VLAN20 扔出去，最终抵达目的地的 VLAN20 的 PC。

配置及实现

1. 完成两台 PC 的配置

PC 的配置不再赘述。

2. 完成交换机的配置

注意，用 GNS 模拟本实例中的三层交换机时，必须使用三层交换机的 IOS，例如 C3640 平台的 IOS。并且为设备添加的模块为 NM-16ESW（具体的添加方法请见本手册 GNS 模拟器章节）。

SW 的配置如下：

```
SW(config)# ip routing                                    !!开启三层交换机的路由功能

SW(config)# vlan 10
SW(config-vlan)# exit
SW(config)# vlan 20
SW(config-vlan)# exit

SW(config)# interface fastethernet 0/1
SW(config-if)# switchport mode access
SW(config-if)# switchport access vlan 10
SW(config-if)# interface fastethernet 0/2
SW(config-if)# switchport mode access
SW(config-if)# switchport access vlan 20
SW(config-if)# exit

SW(config)# interface vlan 10                             !!配置 vlan10 的 SVI
SW(config-if)# ip address 192.168.10.254 255.255.255.0
SW(config-if)# interface vlan 20                          !!配置 vlan20 的 SVI
SW(config-if)# ip address 192.168.20.254 255.255.255.0
```

SW#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/1	unassigned	YES	unset	up	up
....					
Vlan1	unassigned	YES	unset	up	down
Vlan10	192.168.10.254	YES	manual	up	up
Vlan20	192.168.20.254	YES	manual	up	up

现在 PC1 与 PC2 即可互访了。

4 安全篇

4.1 标准 ACL

实验目的

1. 掌握标准 ACL 的配置；
2. 理解标准 ACL 在接入控制中的运用。

拓扑及需求

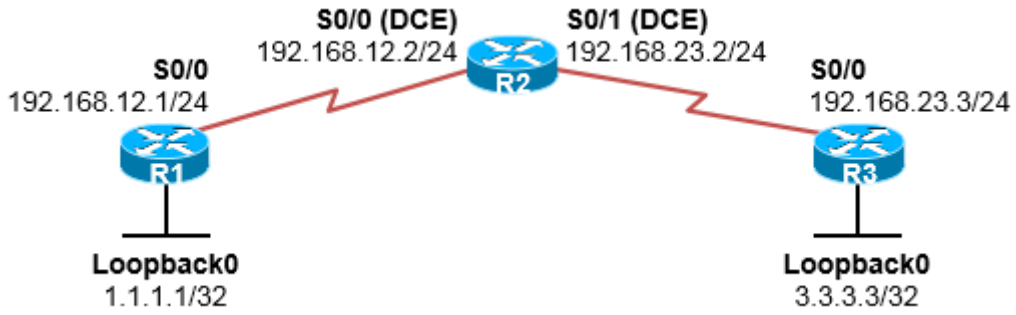

图 4-1 标准 ACL 实验 网络拓扑

1. 完成各设备配置使得全网互通；
2. 在 R2 上部署标准访问控制列表，只允许 192.168.12.0/24 网段的用户穿越 R2 访问 3.3.3.3，其他进入 R2 S0/0 接口的流量全部丢弃。

配置及实现

1. **完成设备的基本配置**

R1 的配置如下：

```
R1# configure terminal
R1(config)# Interface serial0/0
R1(config-if)# Ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# exit

R1(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2
```

R2 的配置如下：

```
R2# configure terminal
R2(config)# interface serial0/0
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# interface serial0/1
R2(config-if)# clock rate 64000
R2(config-if)# ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# exit

R2(config)# ip route 1.1.1.1 255.255.255.255 192.168.12.1
R2(config)# ip route 3.3.3.3 255.255.255.255 192.168.23.3
```

R3 的配置如下：

```
R3# configure terminal
R3(config)# interface serial0/0
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface loopback0
R3(config-if)# ip address 3.3.3.3 255.255.255.255
R3(config-if)# exit

R3(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

完成上述配置后，全网是能够互通的。

2. 在 R2 上部署标准 ACL

在完成上述配置后我们测试一下，首先在 R1 上直接 ping 3.3.3.3，这时候由于 ICMP 包是从 R1 始发，因此这个 ICMP 报文的源 IP 地址就是其出接口的 IP 地址，由于报文从 R1 的 S0/0 接口发出，因此源 IP 地址为 192.168.12.1，当然数据包的目的地址是 3.3.3.3。那么如果我们希望在 R1 上以 1.1.1.1 为源去 ping 3.3.3.3 呢？很简单，在 R1 上使用“ping 3.3.3.3 source 1.1.1.1”命令即可，source 关键字后面的 IP 地址就是我们所设定的源地址，如果不指定 source 关键字，则缺省情况下源地址为报文出接口的地址。由于前面为各设备配置了路由，所以此时 1.1.1.1 与 3.3.3.3 是能够互访的。接下去聚焦我们的需求：“在 R2 上部署标准访问控制列表，只允许 192.168.12.0/24 网段的用户访问 3.3.3.3”。

R2 增加配置如下：

```
R2(config)# access-list 1 permit 192.168.12.0 0.0.0.255
R2(config)# interface serial 0/0
R2(config-if)# ip access-group 1 in          !!将 ACL 应用在 R2 的 S0/0 口 in 方向
```

与此一来从 R1 上直接 ping 3.3.3.3 是通的，因为数据的源地址是 192.168.12.1，而使用 1.1.1.1 为源去 ping 3.3.3.3，却无法 ping 通，因为此刻的源地址是 1.1.1.1，所以被 ACL 干掉了：

```
R1#ping 3.3.3.3

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 48/74/120 ms
R1#ping 3.3.3.3 source 1.1.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 3.3.3.3, timeout is 2 seconds:
Packet sent with a source address of 1.1.1.1
U.U.U
Success rate is 0 percent (0/5)
```

4.2 扩展 ACL

实验目的

1. 掌握扩展 ACL 的配置；
2. 理解扩展 ACL 在接入控制中的运用。

拓扑及需求

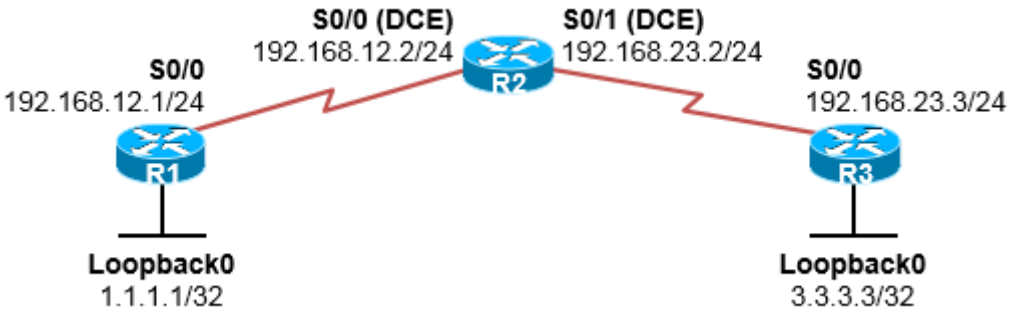

图 4-2 扩展 ACL 实验 网络拓扑

1. 完成各设备的配置，保证全网可达；
2. 在 R2 上部署 ACL，只允许从 1.1.1.1 到 3.3.3.3 的 ICMP 流量以及 R1 到 3.3.3.3 的 telnet 流经过 R2，其他从 R2 的 S0/0 接口进入的流量过滤掉。

配置及实现

1. 完成设备的基本配置。

R1 的配置如下：

```
R1# configure terminal
R1(config)# interface serial0/0
R1(config-if)# ip address 192.168.12.1 255.255.255.0
R1(config-if)# no shutdown
R1(config-if)# interface loopback0
R1(config-if)# ip address 1.1.1.1 255.255.255.255
R1(config-if)# exit

R1(config)# ip route 0.0.0.0 0.0.0.0 192.168.12.2
```

R2 的配置如下：

```
R2# configure terminal
R2(config)# Interface serial0/0
```

```
R2(config-if)# clock rate 64000
R2(config-if)# Ip address 192.168.12.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# Interface serial0/1
R2(config-if)# clock rate 64000
R2(config-if)# Ip address 192.168.23.2 255.255.255.0
R2(config-if)# no shutdown
R2(config-if)# exit

R2(config)# ip route 1.1.1.1 255.255.255.255 192.168.12.1
R2(config)# ip route 3.3.3.3 255.255.255.255 192.168.23.3
```

R3 的配置如下：

```
R3# configure terminal
R3(config)# Interface serial0/0
R3(config-if)# ip address 192.168.23.3 255.255.255.0
R3(config-if)# no shutdown
R3(config-if)# interface loopback0
R3(config-if)# ip address 3.3.3.3 255.255.255.255
R3(config-if)# exit

R3(config)# line vty 0 4
R3(config-line)# password ccietea
R3(config-line)# login
R3(config-line)# exit

R3(config)# ip route 0.0.0.0 0.0.0.0 192.168.23.2
```

完成上述配置后，全网的数据是能够互通的。

2. 在 R2 上部署扩展 ACL。

R2 增加的配置如下：

```
R2(config)# access-list 100 permit icmp host 1.1.1.1 host 3.3.3.3 echo
R2(config)# access-list 100 permit tcp host 192.168.12.1 host 3.3.3.3 eq telnet
R2(config)# interface serial 0/0
R2(config-if)# ip access-group 100 in
```

测试 1　R1 直接 ping 3.3.3.3，不通，因为数据包的源地址是 192.168.12.1。

测试 2　R1 ping 3.3.3.3 source 1.1.1.1 ，通了，因为匹配 ACL100 第一条语句。

测试 3　R1 telnet 3.3.3.3，连上了，因为匹配住了第二条语句。

其他流量全被干掉了。

4.3 NAT

实验目的

1. 理解 NAT 地址转换的机制；
2. 掌握静态 NAT（及 NAT 端口映射）、动态 NAT（地址池）、PAT 的配置。

拓扑及需求

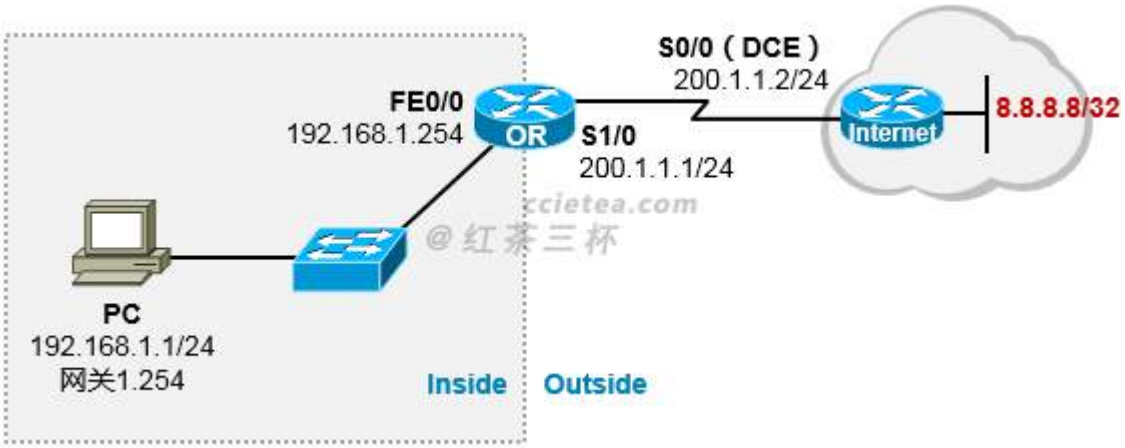

图 4-3 NAT 基础实验 网络拓扑

1. 内网 PC 使用私有 IPv4 地址空间，网关在出口路由器上。
2. 完成 Internet 路由器的配置 ,该设备模拟互联网 ,配置 Loopback 接口并设置 IP 地址 8.8.8.8/32，该地址用于模拟公网上的一个节点。
3. 配置 OR 路由器，采用 NAT 静态一对一 IP 映射，将 192.168.1.1 映射到 200.1.1.10，使得 PC 能够访问 8.8.8.8，同时 Internet 路由器也能够使用 200.1.1.10 来访问 PC。
4. 上一步需求实现后，删除 NAT 的配置，OR 改用静态端口映射，仅将 PC 的 WEB 端口映射到外网 200.1.1.10：8080，使得 Internet 用户能够使用该地址和端口来访问 PC 的 WEB 服务。
5. 上一步需求实现后，删除 NAT 的配置，OR 改用动态 NAT 地址池一对一 IP 的方式，使得 PC

能够访问外网。

6. 上一步需求实现后，删除 NAT 的配置，OR 改用动态 NAT 地址池 Overload 的方式，使得 PC 能够访问外网。

7. 上一步需求实现后，删除 NAT 的配置，OR 改用接口 Overload（使用外网接口）的方式，使得 PC 能够访问外网。

配置及实现

1. 完成所有设备的基本配置

PC 的配置不再赘述。

OR 的基本配置如下：

```
OR# configure terminal
OR(config)# interface fastEthernet 0/0
OR(config-if)# Ip address 192.168.1.254 255.255.255.0
OR(config-if)# no shutdown
OR(config-if)# interface serial 1/0
OR(config-if)# ip address 200.1.1.1 255.255.255.0
OR(config-if)# no shutdown

OR(config)# Ip route 0.0.0.0 0.0.0.0 200.1.1.2
```

为了让内网用户能够访问 Internet 的资源，要在 OR 出口路由器上指一条默认路由出去，下一跳是运营商的设备，也就是图中的 200.1.1.2

Internet 路由器的基本配置如下：

```
Internet# configure terminal
Internet(config)# interface serial 0/0
Internet(config-if)# clock rate 64000
Internet(config-if)# ip address 200.1.1.2 255.255.255.0
Internet(config-if)# no shutdown
Internet(config-if)# interface loopback0
Internet(config-if)# ip address 8.8.8.8 255.255.255.255
```

要注意，我们在 OR 上配置了一条默认路由，这是为了让内网用户能够到达 Internet。但是在 Internet 路由器上，不能配置任何回指的路由，运营商是不可能给你指一条到你内网私有地址段的路由回来的。这是许多同学经常犯的一个错误。

经过上面的配置后我们从 PC 已经能 ping 通网关 192.168.1.254 ,但是肯定是无法出外网的。

2. 配置静态一对一 IP 映射。

在 OR 上，增补的命令如下：

```
OR(config)# ip nat inside source static 192.168.1.1 200.1.1.10

OR(config)# interface fastEthernet 0/0
OR(config-if)# ip nat inside                #设置为 NAT 内部接口
OR(config)# Interface serial 1/0
OR(config-if)# ip nat outside               #设置为 NAT 外部接口
```

ip nat inside source static 192.168.1.1 200.1.1.10 这条命令是在 OR 上创建一条静态的 NAT IP 一对一映射条目，这样做的结果是，内网地址 192.168.1.1 与公网地址 200.1.1.10 对应了起来。一方面 192.168.1.1 这个私有地址在访问外网时，源地址被替换成 200.1.1.10，使得它能够在 Internet 上畅游 (当然 200.1.1.10 这个地址是需要向运营商购买或者申请的) , 另一个结果是，外网用户能直接访问 200.1.1.10 这个公网 IP 从而访问 192.168.1.1 这台内网 PC，换而言之，这台 PC 将直接暴露在公网，这将带来一定的安全隐患。

OR#show ip nat translations !!查看 NAT 映射

Pro	Inside global	Inside local	Outside local	Outside global
---	200.1.1.10	192.168.1.1	---	---

可以看到，OR 的 NAT 映射表里创建了这一条静态的 NAT 映射条目。现在我们再去 PC1 上 8.8.8.8，就能够 ping 通了。ping 完之后可以在 OR 上查看到临时创建的 NAT 映射表项：

OR#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.10:1	192.168.1.1:1	8.8.8.8:1	8.8.8.8:1
---	200.1.1.10	192.168.1.1	---	---

当然现在 Internet 路由器也是能够主动发起访问到 PC1 的：

Internet#ping 200.1.1.10

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.1.1.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/53/104 ms
```

3. 配置 NAT 静态端口映射。

上面我们说过了，静态的 IP 一对一映射实质是将内网的一个 IP 完全暴露在了公网，折是存在

风险的。在实际的环境中，我们可能并不需要把整个 IP（的所有端口）映射到公网，而只需映射特定的端口，例如内网假设了 WEB 服务器，需要让外网来访问，那么我们可能只需要将内网服务器的 TCP 80 端口映射到公网 IP 即可，这就需要用到端口映射了。同样是上面的拓扑，我们假设现在 PC 是一台 WEB 服务器，我们通过在 PC 上配置如下命令来开启路由器的 Http 服务：

```
PC(config)# ip http server
```

然后我们要修改 OR 的配置，先将上一步配置的 IP 一对一映射去掉：

```
OR(config)# no ip nat inside source static 192.168.1.1 200.1.1.10
```

再配置 NAT 端口映射，OR 的配置就变成了：

```
OR(config)# Interface FastEthernet0/0
OR(config-if)# Ip nat inside
OR(config)# Interface serial 1/0
OR(config-if)# Ip nat outside

OR(config)# ip nat inside source static tcp 192.168.1.1 80 200.1.1.10 8080
```

这里要注意，FastEthernet0/0 接口的 ip nat inside 及 serial1/0 的 ip nat outside 也要配上，千万别忘了。ip nat inside source static tcp **192.168.1.1 80 200.1.1.10 8080** 这条命令的意思是，将内网主机 192.168.1.1 的 TCP 80 端口，映射到外网的 IP 200.1.1.10 的 8080 端口上，那么这样一来公网用户通过访问 200.1.1.10:8080 就相当于访问这台内网服务器的 80 端口的 web 服务了。接下来我们测试一下，在 Internet 路由器上，去 **telnet 200.1.1.10 8080。**

注意上面这条命令的写法，一定要跟上后面的 8080，否则你默认访问的是 TCP 23 端口。这条命令敲完之后，你会发现再敲回车虽然屏幕显示一直在换行，但是没有任何输出，这是因为 Console 界面不支持 HTML 脚本的可视化显示而已，而且我们用的是模拟器，PC 上也没有任何 web 文件。其实我们已经登录到 PC 的 80 端口上了，这个时候只要在 PC 上：

PC#show tcp brief

```
TCB         Local Address          Foreign Address        (state)
64D3F430    192.168.1.1.80         200.1.1.2.59728          ESTAB
```

就能看到外网用户 200.1.1.2 已经登录进来了。这里顺便提一句 ,使用 telnet 加端口号的方法，常用于我们在网络中检测目标节点的某个 TCP 端口是否开放。

4. 配置基于地址池的一对一 NAT

静态的 IP 映射毕竟需要手工维护，如果 IP 资源有限，为了让不同的用户能上网，管理员还得频繁更换配置，太麻烦了，我们可以使用动态 NAT 地址池的方式来实现。这种方式，为路由器分配一个 NAT 地址池，这个地址池内包含用户向运营商申请到的一部分公网 IP，将这些 IP 放入 NAT 池中，当有用户需要上网的时候，从池里找一个公网 IP 出来分配给这个用户，如果

该用户不上了，那么又将 IP 放回池中，供其他人使用。当然，这种方式虽然是“动态的”，但是实质上仍然是一对一的 IP 映射关系，也就是一个内网 IP 映射到一个外网 IP，如果申请了 10 个公网地址，也就只能满足同时 10 人在线，没有从根本上解决 IPv4 地址稀缺的问题。

同样的，我们还是在 OR 上进行配置，先把上一步的命令 no 掉。

```
OR(config)# no ip nat inside source static tcp 192.168.1.1 80 200.1.1.10 8080
```

接着如下配置：

```
OR(config)# access-list 1 permit 192.168.1.0 0.0.0.255
!! 创建一个 ACL 用于匹配允许进行 NAT 动态映射的内网地址段
OR(config)# ip nat pool ccietea 200.1.1.11 200.1.1.20 netmask 255.255.255.0
!! 定义 NAT 地址池 ccietea，该地址池里包含从 200.1.1.11 到 200.1.1.20 共计 10 个公网 IP
OR(config)# ip nat inside source list 1 pool ccietea
!! 将 NAT 地址池与所定义的 ACL 进行关联，如此一来，被 ACL 所匹配的内网用户，当访问外网时，就可以使用 NAT 池中的地址进行地址转换。
```

同样的，别忘记了 FE0/0 接口下的 ip nat inside 及 serial1/0 口下的 ip nat outside 命令。现在我们再做测试，让 PC 去 ping 8.8.8.8，发现是能够 ping 通的。到 OR 上看一下：

OR#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.11:2	192.168.1.1:2	8.8.8.8:2	8.8.8.8:2
---	200.1.1.11	192.168.1.1	---	---

可以看到动态创建的 NAT 映射表项。当 PC 访问外网时，由于 PC 的地址能够被 ACL1 所匹配，因此是允许被地址转换的，路由器从地址池中取出一个公网地址 200.1.1.11，然后与 192.168.1.1 建立一对一的映射，注意此刻这个公网地址就是完完全全给 PC1 使用了。换句话说，这个地址池中一共 10 个公网地址，那么同时就只允许 10 个内网用户访问公网。

接下去我们考虑另一个特性：overload，使用这个特性，我们可以实现一对多的映射，也就是说，多个内网的私有 IP，共用一个公网的 IP 地址上外网，正是这种技术的出现，真正缓解了 IPv4 地址稀缺的问题。配置的方法非常简单，只要在我们本例中定义的命令后，加上一个 overload 关键字即可：

```
OR(config)# ip nat inside source list 1 pool ccietea overload
```

这样一来，私有 IP 访问公网时，端口号也会被转换，这就可以实现多个私有 IP 地址共用一个公网 IP 地址，通过公网 IP 地址的端口号来区分不同的会话。

5. 配置接口 Overload 方式的动态 NAT

端口过载的特性其实很好理解了。如果用户经济条件有限，没法买多个公网 IP 放进 NAT 地址池里进行 overload，那么完全可以借用 OR 的公网接口 IP 来进行 overload 嘛，毕竟这个出口 IP 也是一个公网地址，而且闲着也是闲着是吧？

OR 路由器的配置如下：

```
OR(config)# interface FastEthernet0/0
OR(config-if)# ip nat inside
OR(config-if)# exit
OR(config)# interface Serial1/0
OR(config-if)# ip nat outside
OR(config-if)# exit
OR(config)# no ip nat inside source list 1 pool ccietea overload !!no 掉上一个需求的命令

OR(config)# access-list 1 permit 192.168.1.0 0.0.0.255
OR(config)# ip nat inside source list 1 interface Serial1/0 overload
```

配置完成后，内网用户访问外网时，将直接使用 serial1/0 口的地址进行转换，该地址会被 overload，你可能会看到如下的现象：

OR#sh ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
icmp	202.101.100.1:13	192.168.1.1:13	200.1.1.1:13	200.1.1.1:13
icmp	202.101.100.1:14	192.168.1.2:14	200.1.1.1:14	200.1.1.1:14
icmp	202.101.100.1:15	192.168.1.3:15	200.1.1.1:15	200.1.1.1:15

值得一提的是，PAT 和动态 NAT 映射，丢失了端到端的寻址能力，换而言之，一方面内，这两种机制确实将内网“保护了起来”，但是，如若有特定的业务需求，要让公网用户访问内网的资源，那么就还是得要用到静态的 NAT 映射。

4.4 DHCP

实验目的

1. 理解 DHCP 的工作原理；
2. 掌握 DHCP 服务端及客户端在 CISCO IOS 路由器上的配置。

拓扑及需求

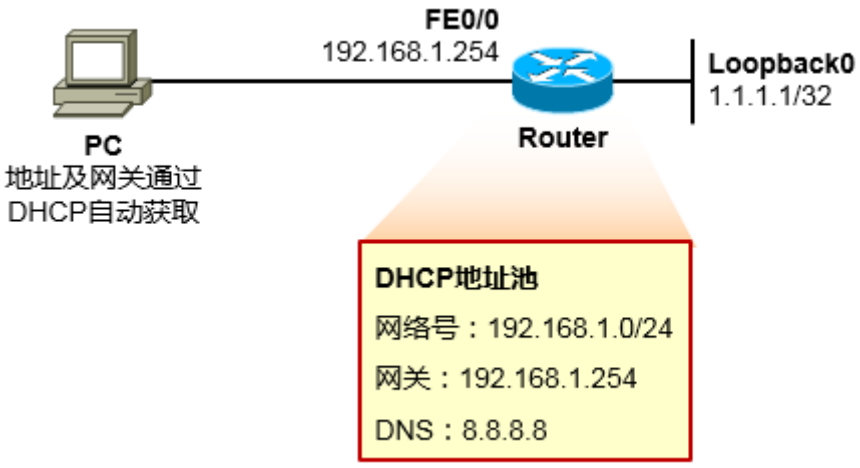

图 4-4 DHCP 基础实验 网络拓扑

1. 在 Router 上配置 DHCP 地址池使得 PC 能够通过 DHCP 自动获取地址、网关、DNS 等信息。
2. 完成所有设备的配置，使得 PC 能够访问 1.1.1.1。

配置及实现

Router 的配置如下：

```
Router# configure terminal
Router(config)# Interface fastethernet0/0
Router(config-if)# ip address 192.168.1.254 255.255.255.0
Router(config-if)# no shutdown
Router(config-if)# interface loopback0
Router(config-if)# ip address 1.1.1.1 255.255.255.255

Router(config)# service dhcp                    !!开启 DHCP 服务
```

```
Router(config)# ip dhcp excluded-address 192.168.1.254
!!由于 192.168.1.254 已经被网关使用，因此该地址必须排除在 DHCP 地址池之外
Router(config)# ip dhcp pool ccietea                    !!创建一个地址池，名字叫 ccietea
Router(dhcp-config)# network 192.168.1.0 /24            !!地址池使用的网络号 192.168.1.0/24
Router(dhcp-config)# default-router 192.168.1.254       !!分配给 PC 的网关地址
Router(dhcp-config)# dns-server 8.8.8.8                 !!分配给 PC 的 DNS 服务器地址
```

现在你可以直接连接一台真机在路由器的 FE0/0 口上，并将真机的网卡设置为自动获取 IP 地址，那么网卡就能够通过 DHCP 获取到地址等信息。当然，Cisco IOS 路由器也是支持作为 DHCP 客户端的，例如本实验拓扑中的 PC，如果使用路由器来模拟，那么配置如下：

```
PC(config)# no ip routing

PC(config)# interface fastethernet0/0
PC(config-if)# ip address dhcp                          !!接口地址采用 DHCP 自动获取
PC(config-if)# no shutdown
```

接口一旦激活，就会开始发送 DHCP Discovery 去发现链路上的 DHCP 服务器。不一会儿 PC1 就能够获取到地址。

```
*Mar  1 00:05:47.399: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned
DHCP address 192.168.1.1, mask 255.255.255.0, hostname PC

PC#show ip interface brief
Interface              IP-Address      OK? Method Status                Protocol
FastEthernet0/0        192.168.1.1     YES DHCP   up                    up
PC#
```

网关也获取到了：

PC1#show ip route

```
Default gateway is 192.168.1.254
Host              Gateway           Last Use    Total Uses  Interface
ICMP redirect cache is empty
```

在 DHCP 服务器 Router 上也能够查询到地址的分配信息

Router #show ip dhcp binding

```
Bindings from all pools not associated with VRF:
IP address      Client-ID/              Lease expiration        Type
                Hardware address/
                User name
192.168.1.1     0063.6973.636f.2d63.    Mar 02 2002 12:05 AM    Automatic
```

```
                    6330.382e.3161.3830.
                    2e30.3030.302d.4661.
                    302f.30
```

PC#ping 1.1.1.1

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 1.1.1.1, timeout is 2 seconds:
!!!!!
```

4.5 综合实验 1

实验目的

1. 全面理解交换原理并掌握交换机的配置；
2. 掌握 DHCP 的部署；
3. 通过本实验初步建立整体网络模型的概念。

拓扑及需求

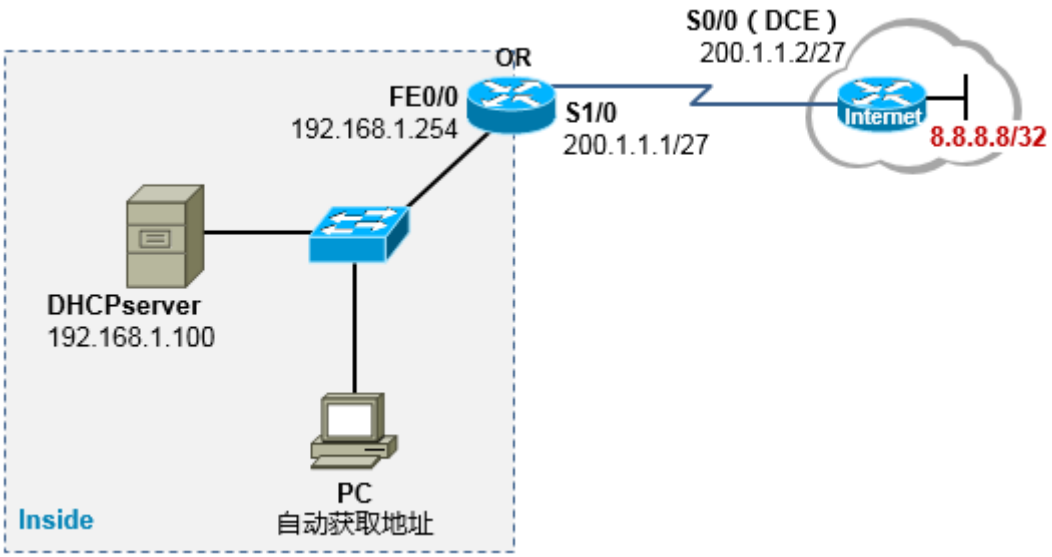

图 4-5 NAT 综合实验 1 网络拓扑

1. 内网 PC 通过 DHCP 服务器自动获取地址（DHCP 地址池中包含的网段是 192.168.1.0/24，

网关地址是 192.168.1.254，DNS 服务器是 8.8.8.8），DHCP 服务器采用路由器来部署；

2. 内网 PC 及服务器均能访问 Internet，采用接口 Overload 的方式；
3. 外网用户能够通过公网地址 200.1.1.10：8080（此地址是客户向运营商购买的）访问内网服务器 192.168.1.100 的 WEB 服务。

配置及实现

1. 完成 DHCPserver 的配置

DHCPserver 的配置如下：

```
Router# configure terminal
Router(config)# hostname DHCPServer
DHCPserver(config)# no ip routing                          !!关闭路由功能，模拟一台服务器

DHCPserver(config)# Interface fastethernet0/0
DHCPserver(config-if)# ip address 192.168.1.100 255.255.255.0
DHCPserver(config-if)# no shutdown

DHCPserver(config)# ip default-gateway 192.168.1.254       !!配置默认网关

DHCPserver(config)# service dhcp                           !!开启 DHCP 服务
DHCPserver(config)# ip dhcp excluded-address 192.168.1.100
DHCPserver(config)# ip dhcp excluded-address 192.168.1.254
DHCPserver(config)# ip dhcp pool ccietea                   !!创建一个 DHCP 地址池
DHCPserver(dhcp-config)# network 192.168.1.0 /24           !!网络号 192.168.1.0/24
DHCPserver(dhcp-config)# default-router 192.168.1.254      !!默认网关
DHCPserver(dhcp-config)# dns-server 8.8.8.8                !!DNS 服务器地址
```

2. 完成 PC 的配置

PC 的配置如下

```
PC(config)# no ip routing

PC(config)# Interface fastethernet0/0
PC(config-if)# ip address dhcp                 !!地址采用 DHCP 自动获取
PC(config-if)# no shutdown
```

接口一旦激活后，PC 即可获取到地址：

PC#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	192.168.1.1	YES	DHCP	up	up

3. 完成 Internet 路由器的配置

Internet 路由器的配置最为简单：

```
Internet(config)# Interface serial0/0
Internet(config-if)# clock rate 64000
Internet(config-if)# ip address 200.1.1.2    255.255.255.224
Internet(config-if)# no shutdown
Internet(config-if)# interface loopback0
Internet(config-if)# ip address 8.8.8.8 255.255.255.255
```

4. 完成 OR 出口路由器的配置

出口路由器 OR 的基本配置如下：

```
OR(config)# Interface fastethernet0/0
OR(config-if)# ip address 192.168.1.254    255.255.255.0
OR(config-if)# no shutdown
OR(config)# Interface serial1/0
OR(config-if)# ip address 200.1.1.1    255.255.255.224
OR(config-if)# no shutdown

OR(config)# ip route 0.0.0.0 0.0.0.0 200.1.1.2
```

OR 路由器作为网络的出口设备，对外连接着 Internet，因此为了保证内网用户能够访问浩瀚的 Internet，需要为其配置一条默认路由。

完成上述配置后 PC 及 DHCPserver 都能够 ping 通网关 192.168.1.254；OR 也能够 ping 通公网节点 8.8.8.8。但是 PC 无法访问 Internet（8.8.8.8），当然 Internet 路由器也无法访问 DHCPserver。现在我们为 OR 配置 NAT，这里要部署两种 NAT，一是使用 OR 的 S1/0 口做端口 Overload（用于保证内网用户能够访问 Internet），另外是配置 NAT 静态端口映射（用于保证 Internet 用户能够访问内网 Server）。

```
OR(config)# Interface fastethernet0/0
OR(config-if)# ip nat inside
OR(config)# Interface serial1/0
OR(config-if)# Ip nat outside
```

```
OR(config)# access-list 1 permit 192.168.1.0 0.0.0.255
OR(config)# ip nat inside source list 1 interface serial 1/0 overload

OR(config)# Ip nat inside source static tcp 192.168.1.100 80 200.1.1.10 8080
```

完成上述配置后，PC 即可 ping 通 8.8.8.8，而 Internet 路由器也能使用 200.1.1.10 8080 来访问内网 DHCPServer 的 WEB 服务，测试方法在上一个实验中有介绍，不再赘述。

4.6 综合实验 2

实验目的

1. 全面理解交换原理并掌握交换机的配置；
2. 掌握 DHCP（多地址池）的部署；
3. 通过本实验巩固对整体网络模型的理解。

拓扑及需求

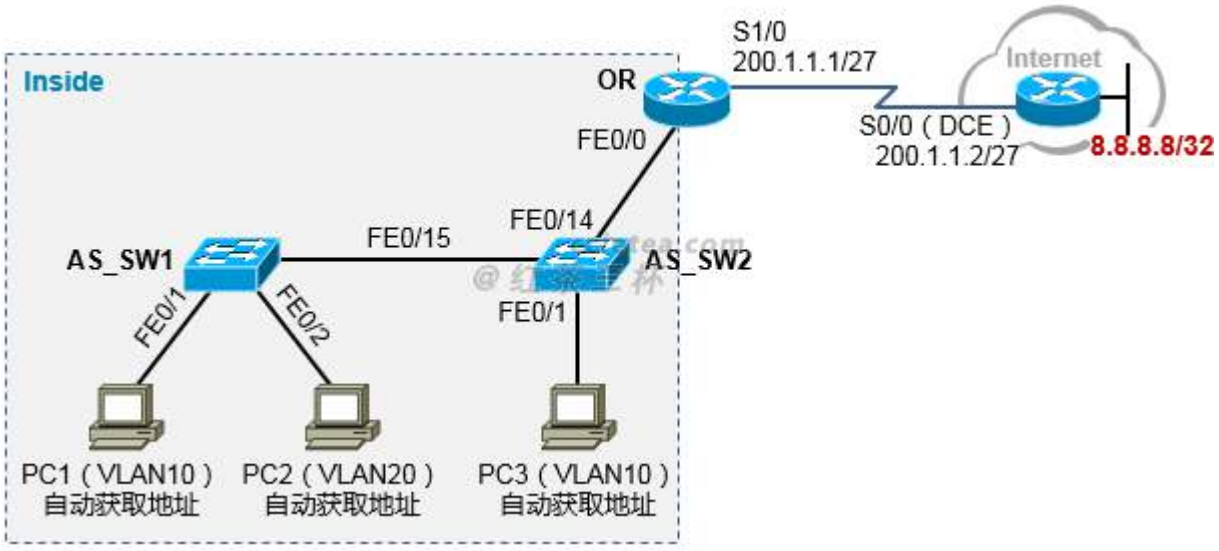

图 4-6 NAT 综合实验 2 网络拓扑

1. PC1、PC3 属于 VLAN10；PC2 属于 VLAN20；所有的 PC 均采用 DHCP 的方式获取地址，OR 路由器充当 DHCP 服务器。其中 VLAN10 用户的地址段为 192.168.10.0/24；VLAN20 用户的地址段为 192.168.20.0/24；

2. OR 路由器是内网 PC 的网关。VLAN10 用户使用网关地址 192.168.10.254；VLAN20 用户使用网关地址 192.168.20.254；
3. 内网 PC 能够访问 Internet 路由器，使用 S1/0 接口 Overload 的方式部署源地址转换。

配置及实现

1. 完成 AS_SW1 及 AS_SW2 的配置

AS_SW1 的配置如下：

```
switch# configure terminal
switch(config)# hostname AS_SW1
AS_SW1(config)# vlan 10
AS_SW1(config-vlan)# exit
AS_SW1(config)# vlan 20
AS_SW1(config-vlan)# exit

AS_SW1(config)# interface fastethernet0/1
AS_SW1(config-if)# switchport mode access
AS_SW1(config-if)# switchport access vlan 10
AS_SW1(config)# interface fastethernet0/2
AS_SW1(config-if)# switchport mode access
AS_SW1(config-if)# switchport access vlan 20
AS_SW1(config)# interface fastethernet0/15
AS_SW1(config-if)# switchport encapsulation dot1q
AS_SW1(config-if)# switchport mode trunk
```

AS_SW2 的配置如下：

```
switch# configure terminal
switch(config)# hostname AS_SW2
AS_SW2(config)# vlan 10
AS_SW2(config-vlan)# exit
AS_SW2(config)# vlan 20
AS_SW2(config-vlan)# exit

AS_SW2(config)# interface fastEthernet 0/1
AS_SW2(config-if)# switchport mode access
```

```
AS_SW2(config-if)# switchport access vlan 10
AS_SW2(config)# interface fastEthernet 0/15
AS_SW2(config-if)# switchport encapsulation dot1q
AS_SW2(config-if)# switchport mode trunk
AS_SW2(config)# interface fastEthernet 0/14
AS_SW2(config-if)# switchport encapsulation dot1q
AS_SW2(config-if)# switchport mode trunk
```

AS_SW2 有一个小细节要注意，虽然 AS_SW2 上实际只连接着 VLAN10 的 PC，因此许多童鞋做实验的时候可能会只在 AS_SW2 上创建 VLAN10 而并未创建 VLAN20，这将导致 AS_SW1 的 VLAN20 用户无法穿越 AS_SW2 去访问路由器，因为 AS_SW2 没有 VLAN20 的 VLAN 信息。所以务必注意在 AS_SW2 上要创建 VLAN10 及 VLAN20。

2. 完成 Internet 路由器的配置。

Internet 路由器的配置最为简单：

```
Router# configure terminal
Router(config)# hostname Internet
Internet(config)# Interface serial0/0
Internet(config-if)# clock rate 64000
Internet(config-if)# ip address 200.1.1.2    255.255.255.224
Internet(config-if)# no shutdown
Internet(config-if)# interface loopback0
Internet(config-if)# ip address 8.8.8.8 255.255.255.255
```

3. 完成 OR 出口路由器的配置

出口路由器 OR 的基本配置如下：

```
router# configure terminal
router(config)# hostname OR
OR(config)# Interface fastethernet0/0
OR(config-if)# no shutdown                          !!将物理接口 no shutdown

OR(config-if)# Interface fastethernet0/0.10
OR(config-if)# encapsulation dot1q 10
OR(config-if)# ip address 192.168.10.254    255.255.255.0
OR(config-if)# Interface fastethernet0/0.20
OR(config-if)# encapsulation dot1q 20
OR(config-if)# ip address 192.168.20.254    255.255.255.0
```

```
OR(config)# Interface serial1/0
OR(config-if)# ip address 200.1.1.1    255.255.255.224
OR(config-if)# no shutdown

OR(config)# ip route 0.0.0.0 0.0.0.0 200.1.1.2
```

接下去是 DHCP 服务的配置：

```
OR(config)# service dhcp

OR(config)# ip dhcp excluded-address 192.168.10.254
OR(config)# ip dhcp excluded-address 192.168.20.254

OR(config)# ip dhcp pool vlan10
OR(dhcp-config)# network 192.168.10.0 /24
OR(dhcp-config)# default-router 192.168.10.254
OR(dhcp-config)# exit

OR(config)# ip dhcp pool vlan20
OR(dhcp-config)# network 192.168.20.0 /24
OR(dhcp-config)# default-router 192.168.20.254
OR(dhcp-config)# exit
```

接下去是 OR 路由器的 NAT 配置：

```
OR(config)# access-list 1 permit 192.168.10.0 0.0.0.255
OR(config)# access-list 1 permit 192.168.20.0 0.0.0.255
OR(config)# ip nat inside source list 1 interface serial1/0 overload
OR(config)# interface fastethernet0/0.10
OR(config-if)# ip nat inside
OR(config-if)# interface fastethernet0/0.20
OR(config-if)# ip nat inside
OR(config)# interface serial1/0
OR(config-if)# ip nat outside
```

注意，由于 OR 的 F0/0 接口上创建了两个子接口，因此 ip nat inside 命令是配置在子接口上的，而不是物理接口上。

4. 完成所有 PC 的配置

PC1 的配置如下：

```
Router(config)# hostname PC1
PC1(config)# no ip routing
PC1(config)# interface fastEthernet 0/0
PC1(config-if)# ip address dhcp
PC1(config-if)# no shutdown
```

PC2 及 PC3 的配置类似，不再赘述。

完成上述配置后 PC1、PC2 及 PC3 都能够获取到地址，并且都能 ping 通 8.8.8.8。

5 广域网篇

5.1 PPP（PAP 认证）

实验目的

1. 掌握 PPP 封装的配置；
2. 掌握 PAP 认证的原理。

拓扑及需求

图 5-1 PPP PAP 认证实验 网络拓扑

1. Remote 路由器及 Core 路由器的直连接口采用 PPP 封装；
2. 要求 PPP 链路采用 PAP 认证，Remote 路由器为客户端，Core 路由器为认证服务端，Core 路由器为 Remote 路由器开设的用户名是 remote，密码是 ccietea。

配置及实现

1. 配置 Remote 及 Core Router

Remote Router（认证客户端）的配置如下：

```
Router# configure terminal
Router(config)# hostname Remote
Remote(config)# interface serial0/0
Remote(config-if)# encapsulation ppp                    !!接口封装 PPP
Remote(config-if)# ip address 12.1.1.1 255.255.255.252
Remote(config-if)# ppp pap sent-username remote password ccietea
                                                        !!用于认证的用户名及密码
Remote(config-if)# no shutdown
```

“remote”是 Remote 路由器用来验证时使用的用户名，ccietea 是密码。

Core（认证服务端）的配置如下：

```
Router# configure terminal
Router(config)# hostname Core
Core(config)# username remote password ccietea          !! 定义本地数据库，用于认证的时候进行查询检验
Core(config)# interface serial0/0
Core(config-if)# encapsulation ppp
Core(config-if)# clock rate 64000
Core(config-if)# ip address 12.1.1.2    255.255.255.252
Core(config-if)# ppp authentication pap                 !! 开启 pap 认证，这条命令仅在认证服务端配置
Core(config-if)# no shutdown
```

2. 查看及验证

注意，PAP 使用明文传输密码，可通过抓包的方式将密码截获：

```
Point-to-Point Protocol
PPP Password Authentication Protocol
  Code: Authenticate-Request (0x01)
  Identifier: 0x10
  Length: 19
⊟ Data (15 bytes)
  ⊟ Peer ID length: 6 bytes
      Peer-ID (6 bytes)
  ⊟ Password length: 7 bytes
      Password (7 bytes)

00  ff 03 c0 23 01 10 00 13  06 72 65 6d 6f 74 65 07  ...#.... .remote.
10  63 63 69 65 74 65 61                             ccietea
```

图 5-2 使用抓包工具进行抓包，能看到 PAP 报文中所包含的铭文密码

R1#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
Serial0/0	12.1.1.1	YES	manual	**up**	**up**

完成上述配置后，R1-R2 之间的 PPP 链路即可协商并建立成功，两者可互访。

5.2 PPP（CHAP 单向认证）

实验目的

1. 理解 PPP CHAP 认证过程；
2. 掌握 CHAP 认证的配置。

拓扑及需求

图 5-3 PPP CHAP 单向认证实验 网络拓扑

1. Remote 路由器及 Core 路由器的直连接口采用 PPP 封装；
2. 要求 PPP 链路采用 CHAP 认证，Remote 路由器为客户端，Core 路由器为认证服务端，Core

路由器为 Remote 路由器开设的用户名是 remote，密码是 ccietea。

配置及实现

1. 配置 Remote 及 Core

Remote 的配置如下：

```
Router# configure terminal
Router(config)# hostname Remote
Remote(config)# interface serial0/0
Remote(config-if)# encapsulation ppp
Remote(config-if)# ip address 12.1.1.1 255.255.255.252
Remote(config-if)# ppp chap hostname remote          !! 用于 CHAP 认证的名称，如果不配置这条，则默认发送 hostname
Remote(config-if)# ppp chap password ccietea         !! 用于 CHAP 认证的密码
Remote(config-if)# no shutdown
```

Core 的配置如下：

```
Router# configure terminal
Router(config)# hostname Core
Core(config)# username remote password ccietea

Core(config)# interface serial0/0
Core(config-if)# encapsulation ppp
Core(config-if)# clock rate 64000
Core(config-if)# ip address 12.1.1.2 255.255.255.252
Core(config-if)# ppp authentication chap             !!开启 CHAP 认证
Core(config-if)# no shutdown
```

注意，ppp authentication chap 这条命令是配置在认证端。完成上述配置后 R1、R2 之间的 PPP 链路即可协商成功。

2. CHAP 小结

注意,CHAP 认证过程中 ,密码是不会被发送的 ,CHAP 使用一个 challenge 来完成验证动作：

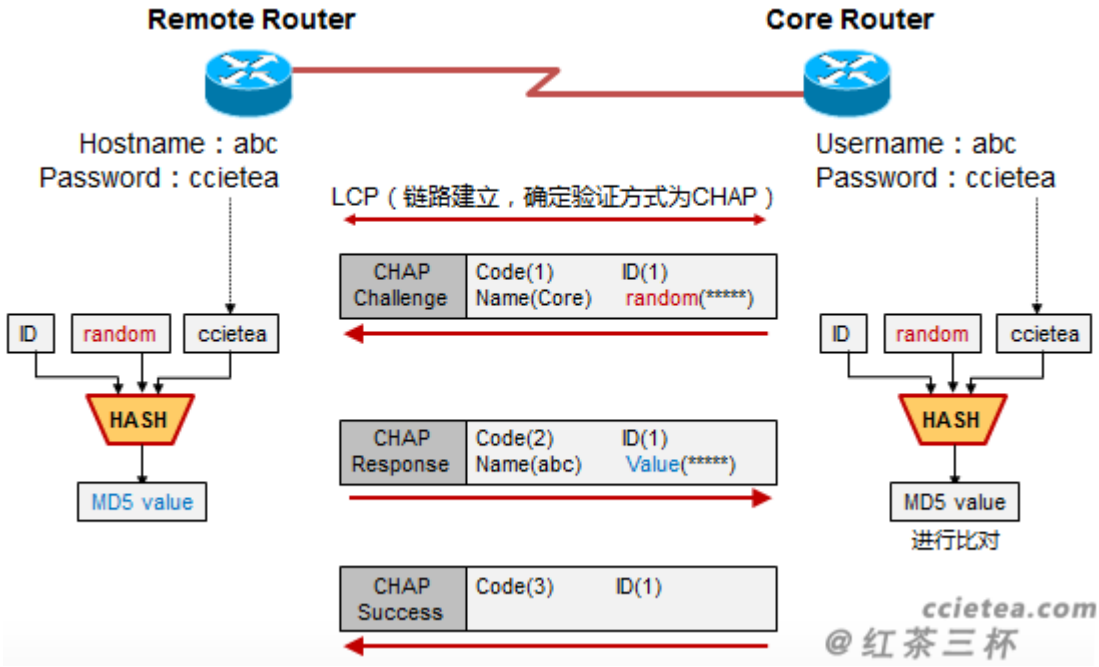

图 5-4 CHAP 认证的过程

5.3 PPP（CHAP 双向认证）

实验目的

1. 理解 CHAP 认证过程；
2. 掌握 CHAP 认证的配置。

拓扑及需求

图 5-5 PPP CHAP 双向认证实验 网络拓扑

1. R1 及 R2 的直连接口采用 PPP 封装；
2. 要求 PPP 链路采用 CHAP 双向认证。

配置及实现

R1 的配置如下：

```
Router# configure terminal
Router(config)# hostname R1
R1(config)# username R2 password ccietea        !! 注意这里 username 配置的是对端的 hostname，密码必须一致
R1(config)# interface serial0/0
R1(config-if)# encapsulation ppp
R1(config-if)# ip address 12.1.1.1 255.255.255.252
R1(config-if)# ppp authentication chap          !!开启 CHAP 验证
R1(config-if)# no shutdown
```

R2 的配置如下：

```
Router# configure terminal
Router(config)# hostname R2
R2(config)# username R1 password ccietea
R2(config)# interface serial0/0
R2(config-if)# encapsulation ppp
R2(config-if)# clock rate 64000
R2(config-if)# ip address 12.1.1.2 255.255.255.252
R2(config-if)# ppp authentication chap
R2(config-if)# no shutdown
```

R1 及 R2 即是认证端又是被认证端。完成上述配置后，PPP 链路即可协商并建立成功。

5.4 帧中继基础实验

实验目的

1. 了解帧中继基本原理；
2. 了解帧中继交换机、终端设备的配置；
3. 了解静态帧中继映射的配置；

拓扑及需求

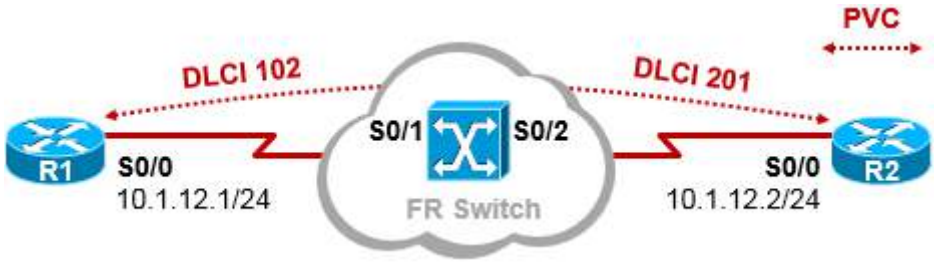

图 5-6 帧中继基础实验 网络拓扑

1. 网络拓扑如上图所示，FRSwitch 是帧中继交换机，其 S0/1 及 S0/2 接口分别连着 R1 和 R2；
2. 完成帧中继交换机的配置，DLCI 号码如图所示；
3. 完成 R1、R2 的配置，两者采用 inverse-arp 的方式建立帧中继映射，要求 R1 与 R2 能互通；
4. 取消 R1、R2 为实现上一个需求所做的配置，改为静态帧中继映射，要求 R1 与 R2 能互通。

配置及实现

1. 完成帧中继交换机的配置

帧中继交换机可使用路由器来模拟。FRSwitch 的配置如下：

```
Router# configure terminal
Router(config)# hostname FRSwitch
FRswitch(config)# frame-relay switching          !! 在模拟帧中继交换机的路由器上配置

FRswitch(config)# interface Serial0/1
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay route 102 interface Serial0/2 201
!! 配置 PVC，上面这条命令可以形象的理解为，S0/1 接口的 DLCI 102 对应到 S0/2 接口的 DLCI 201
FRswitch(config-if)# no shutdown
FRswitch(config-if)# exit

FRswitch(config)# interface Serial0/2
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
```

```
FRswitch(config-if)# frame-relay route 201 interface Serial0/1 102
FRswitch(config-if)# no shutdown
```

完成上述配置后，这条 PVC 就建立好了，在帧中继交换机上可以做一些基本的查看。

FRSwitch#show frame-relay route

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0/1	102	Serial0/2	201	inactive
Serial0/2	201	Serial0/1	102	inactive

2. 完成 R1、R2 的配置，两者采用 Inverse-arp 的方式建立帧中继映射。

R1 的配置如下：

```
R1(config)#interface serial 0/0
R1(config-if)#encapsulation frame-relay
R1(config-if)#ip address 10.1.12.1 255.255.255.0
R1(config-if)#no shutdown
```

R2 的配置如下：

```
R2(config)#interface serial 0/0
R2(config-if)#encapsulation frame-relay
R2(config-if)#ip address 10.1.12.2 255.255.255.0
R2(config-if)#no shutdown
```

由于缺省情况下，Inverse-arp 已经是开启的，因此无需再手动激活。稍等片刻之后，R1、R2 就会自动建立帧中继映射表项：

R1#show frame-relay map

```
Serial0/0 (up): ip 10.1.12.2 dlci 102(0x66,0x1860), dynamic,
                broadcast,
                CISCO, status defined, active
```

我们看到，R1 已经出现帧中继的映射表项了，从表项可以读出，10.1.12.2 这个 IP 地址映射到了本地的 DLCI102，而且这个映射条目是动态获取的（dynamic），另外这条 PVC 支持广播（实际上是伪广播），PVC 当前是 active 的。

R2#show frame-relay map

```
Serial0/0 (up): ip 10.1.12.1 dlci 201(0xC9,0x3090), dynamic,
                broadcast,
                CISCO, status defined, active
```

R2 的现象类似。现在 R1 与 R2 已经可以相互通信了。

3. **完成 R1、R2 的配置，两者采用静态的方式创建帧中继映射。**

现在，我们的需求是 R1、R2 不使用 inverse-arp 的方式建立帧中继映射表，而是采用手工配置的方式来创建。R1 的配置修改如下：

```
R1(config)#interface serial 0/0
R1(config-if)#no frame-relay inverse-arp                    !!关闭 inverse-arp
R1(config-if)#frame-relay map ip 10.1.12.2 102 broadcast    !!创建静态帧中继映射条目
R1(config-if)#ip address 10.1.12.1 255.255.255.0
```

frame-relay map ip 10.1.12.2 102 这条命令用于将远端 IP 地址 10.1.12.2 映射到本地的 DLCI102，需要强调的是，DLCI 只具有本地意义。这条命令还增加了 broadcast 关键字，加上这个关键字后，这条 PVC 就可以支持广播，否则不能支持广播。

R2 的配置修改如下：

```
R2(config)#interface serial 0/0
R2(config-if)#no frame-relay inverse-arp
R2(config-if)#frame-relay map ip 10.1.12.1 201 broadcast
R2(config-if)#ip address 10.1.12.2 255.255.255.0
```

下面来观察一下 R1 的帧中继映射表：

R1#show frame-relay map

```
Serial0/0 (up): ip 10.1.12.2 dlci 102(0x66,0x1860), static,
                broadcast,
                CISCO, status defined, active
```

R2 的现象类似，此时，R1 与 R2 即可相同通信。

实际上这里还有一个小细节被我们遗漏了，大家可以尝试着在 R1 上 ping 一下它自己的接口 IP 10.1.12.1，或者在 R2 上 ping 一下 10.1.12.2，会发现无法 ping 通，这是为什么呢？

5.5 帧中继 Hub&Spoke 模型基础实验

实验目的

1. 了解帧中继基本原理；
2. 了解帧中继交换机、终端设备的配置；

3. 了解静态帧中继映射的配置；
4. 了解 Hub-and-Spoke 结构的帧中继部署。

拓扑及需求

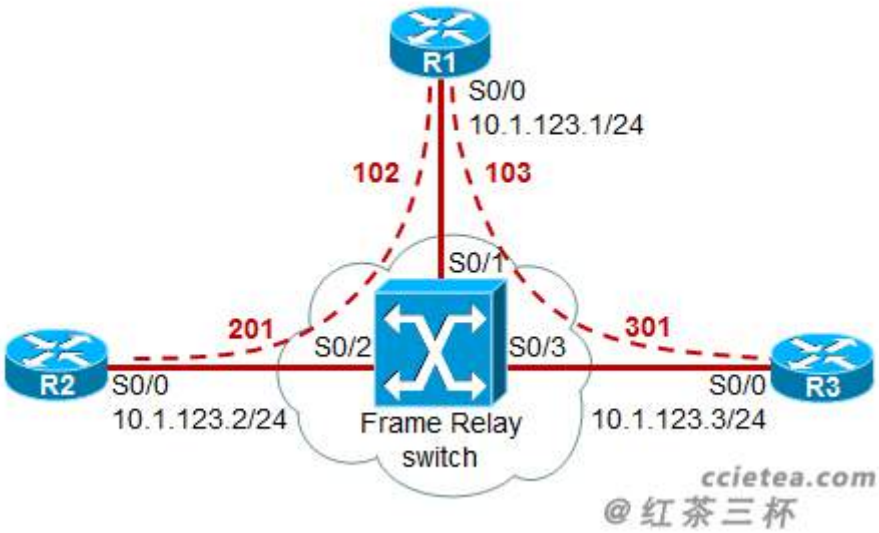

图 5-7 帧中继 Hub&Spole 实验 网络拓扑

1. R1、R2、R3 为终端路由器，帧中继交换机使用 R4（路由器）模拟，端口、连线及 IP 编址请见上图。这是一个典型的 hub&spoke 拓扑；
2. 根据图示完成帧中继交换机的配置；
3. 要求 R1、R2 及 R3 均关闭 inverse-arp，使用静态 map 的方式创建帧中继映射；
4. 要求 R1 能够 ping 通 R2、R3，R2、R3 之间也能够相互 ping 通；
5. 要求 R1 、R2 及 R3 都能够 ping 通自己。

配置及实现

1. 配置帧中继交换机

帧中继交换机的配置如下：

```
Router# configure terminal
Router(config)# hostname FRswitch
FRswitch(config)# frame-relay switching

FRswitch(config)# interface Serial0/1
```

```
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 102 interface Serial0/2 201
FRswitch(config-if)# frame-relay route 103 interface Serial0/3 301
FRswitch(config-if)# no shutdown
FRswitch(config-if)# exit

FRswitch(config)# interface Serial0/2
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 201 interface Serial0/1 102
FRswitch(config-if)# no shutdown
FRswitch(config-if)# exit

FRswitch(config)# interface Serial0/3
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 301 interface Serial0/1 103
FRswitch(config-if)# no shutdown
FRswitch(config-if)# exit
```

完成后，可使用 show frame-relay route 查看配置好的 PVC

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0/1	102	Serial0/2	201	inactive
Serial0/1	103	Serial0/3	301	inactive
Serial0/2	201	Serial0/1	102	inactive
Serial0/3	301	Serial0/1	103	inactive

2. 配置终端设备 R1、R2 及 R3

R1 的配置如下：

```
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface Serial0/0
```

```
R1(config-if)# ip address 10.1.123.1 255.255.255.0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no frame-relay inverse-arp                    !!关闭 inverse-arp
R1(config-if)# frame-relay map ip 10.1.123.2 102 broadcast   !!手工配置帧中继映射
R1(config-if)# frame-relay map ip 10.1.123.3 103 broadcast
R1(config-if)# no shutdown
```

Broadcast 关键字为可选，加上此关键字，则该条 PVC 将具有“广播”的支持能力，当然，所谓的帧中继环境下的广播，指的是向所有的 PVC 都发送一份数据的拷贝，实现类似广播的操作。

R2 的配置如下：

```
Router# configure terminal
Router(config)# hostname R2
R2(config)# interface Serial0/0
R2(config-if)# ip address 10.1.123.2 255.255.255.0
R2(config-if)# encapsulation frame-relay
R2(config-if)# no frame-relay inverse-arp
R2(config-if)# frame-relay map ip 10.1.123.1 201 broadcast
R2(config-if)# no shutdown
```

R3 的配置如下：

```
Router# configure terminal
Router(config)# hostname R3
R3(config)#interface Serial0/0
R3(config-if)# ip address 10.1.123.3 255.255.255.0
R3(config-if)# encapsulation frame-relay
R3(config-if)# no frame-relay inverse-arp
R3(config-if)# frame-relay map ip 10.1.123.3 301 broadcast
R3(config-if)# no shutdown
```

如此一来，R1、R2、R3 的帧中继映射就建立好了。我们可以查看一下：

R1#show frame-relay map

```
Serial0/0 (up): ip 10.1.123.2 dlci 102(0x66,0x1860), static,
                broadcast,
                CISCO, status defined, active
Serial0/0 (up): ip 10.1.123.3 dlci 103(0x67,0x1870), static,
                broadcast,
```

```
                CISCO, status defined, active
```

两条 PVC 现在都是 active 的。

3. 连通性测试

基于上面的配置，R1 已经能够 ping 通 R2，R1 也能 ping 通 R3。但是有两个问题，一是 R2 无法 ping 通 R3，这是为什么？这是因为 R1、R2、R3 三台路由器虽然共处一个 NBMA 网络，使用的地址段也是同一个 IP 地址段，但是帧中继环境下，正常通信同时双方需具备彼此的二层、三层地址，现在三层地址是有了，但是二层地址呢？R2、R3 彼此并没有对方的 IP 地址对应的本地 DLCI。简单的说，R2 ping R3，在构造数据帧的时候，并没有 10.1.123.3 的对应的本地 DLCI，因此数据帧构造失败。那么在 R2 上增加配置：

```
R2(config)# Interface serial0/0
R2(config-if)# frame-relay map ip 10.1.123.3 201 broadcast
```

在 R3 上也要增加配置：

```
R3(config)# Interface serial0/0
R3(config-if)# frame-relay map ip 10.1.123.2 301 broadcast
```

如此一来，R2、R3 之间就能正常通信了。其实 R2、R3 之间的通信流量是需要从 R1 中转的，因为 R2、R3 之间并没有直接的建立 PVC（也许是为了省钱）。还有一个问题，R1、R2、R3 无法 ping 通自己？问题跟上面描述的类似，我们还需在三台路由器上，将各自的 IP 地址映射到本地的 DLCI，相关的配置这里就不再罗列了。

5.6 帧中继 P2P 子接口实验

实验目的

掌握帧中继点对点子接口的配置。

拓扑及需求

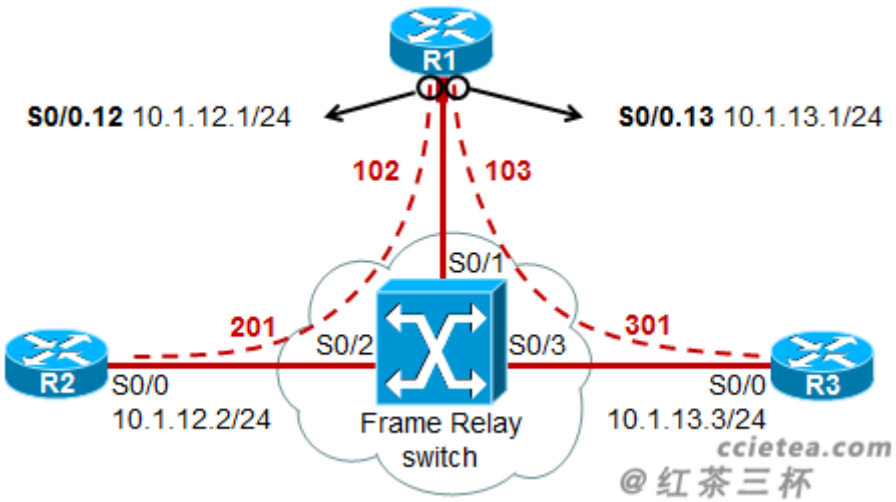

图 5-8 帧中继 P2P 子接口实验 网络拓扑

R1 使用物理接口 S0/0 与帧中继交换机直连。在物理接口上，划分两个子接口，S0/0.12 与 R2 建立 PVC，S0/0.13 与 R3 建立 PVC。注意，一旦拓扑变更为如上的情况后，R1、R2 之间使用一个 IP 地址段，R1、R3 之间使用另外一个地址段，相当于这是两段链路。

配置及实现

1. 完成帧中继交换机的配置：

帧中继交换机的配置如下：

```
Router# configure terminal
Router(config)# hostname FRswitch
FRswitch(config)# frame-relay switching

FRswitch(config)# interface Serial0/1
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 102 interface Serial0/2 201
FRswitch(config-if)# frame-relay route 103 interface Serial0/3 301
FRswitch(config-if)# no shutdown

FRswitch(config)# interface Serial0/2
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
```

```
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 201 interface Serial0/1 102
FRswitch(config-if)# no shutdown

FRswitch(config)# interface Serial0/3
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 301 interface Serial0/1 103
FRswitch(config-if)# no shutdown
```

完成后，可使用 show frame-relay route 查看配置好的 PVC

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0/1	102	Serial0/2	201	inactive
Serial0/1	103	Serial0/3	301	inactive
Serial0/2	201	Serial0/1	102	inactive
Serial0/3	301	Serial0/1	103	inactive

2. 完成 R2、R3 的配置

R2 的配置如下：

```
Router# configure terminal
Router(config)# hostname R2
R2(config)# interface Serial0/0
R2(config-if)# ip address 10.1.12.2 255.255.255.0
R2(config-if)# encapsulation frame-relay
R2(config-if)# no frame-relay inverse-arp
R2(config-if)# frame-relay map ip 10.1.12.1 201 broadcast
R2(config-if)# no shutdown
```

R3 的配置如下：

```
Router# configure terminal
Router(config)# hostname R3
R3(config)#interface Serial0/0
R3(config-if)# ip address 10.1.13.3 255.255.255.0
R3(config-if)# encapsulation frame-relay
R3(config-if)# no frame-relay inverse-arp
```

```
R3(config-if)# frame-relay map ip 10.1.13.1 301 broadcast
R3(config-if)# no shutdown
```

这个实验关键点在于 R1 的配置。帧中继交换机的配置这里不再赘述。R1 的配置如下：

```
Router# configure terminal
Router(config)# hostname R1
R1(config)# interface Serial0/0
R1(config-if)# encapsulation frame-relay
R1(config-if)# no shutdown
!!注意，物理接口一定要封装帧中继并且 no shutdown

R1(config)# interface Serial0/0.12 point-to-point          !!创建点对点子接口
R1(config-if)# ip address 10.1.12.1 255.255.255.0
R1(config-if)# frame-relay interface-dlci 102              !!配置该点对点帧中继子接口对应的 DLCI 号，注意该命令只能配置在点对点的帧中继子接口上。
R1(config-fr-dlci)# exit

R1(config-if)# interface Serial0/0.13 point-to-point
R1(config-if)# ip address 10.1.13.1 255.255.255.0
R1(config-if)# frame-relay interface-dlci 103
R1(config-fr-dlci)# exit
```

完成上述配置后，R1 与 R2，以及 R1 与 R3 都可以相互通信了。而 R2 与 R3 之间要想相互通信，就需要在 R2 及 R3 上配置静态路由。

6 综合实验

6.1 综合实验 1

实验目的

1. 学会分析网络、解构网络；
2. 掌握从无到有配置一个网络的思路；
3. 巩固 L2 switching、DHCP、单臂路由、FrameRelay、PPP、路由选择原理等知识；
4. 掌握基本的故障排除（Trouble shooting）方法。

拓扑及需求

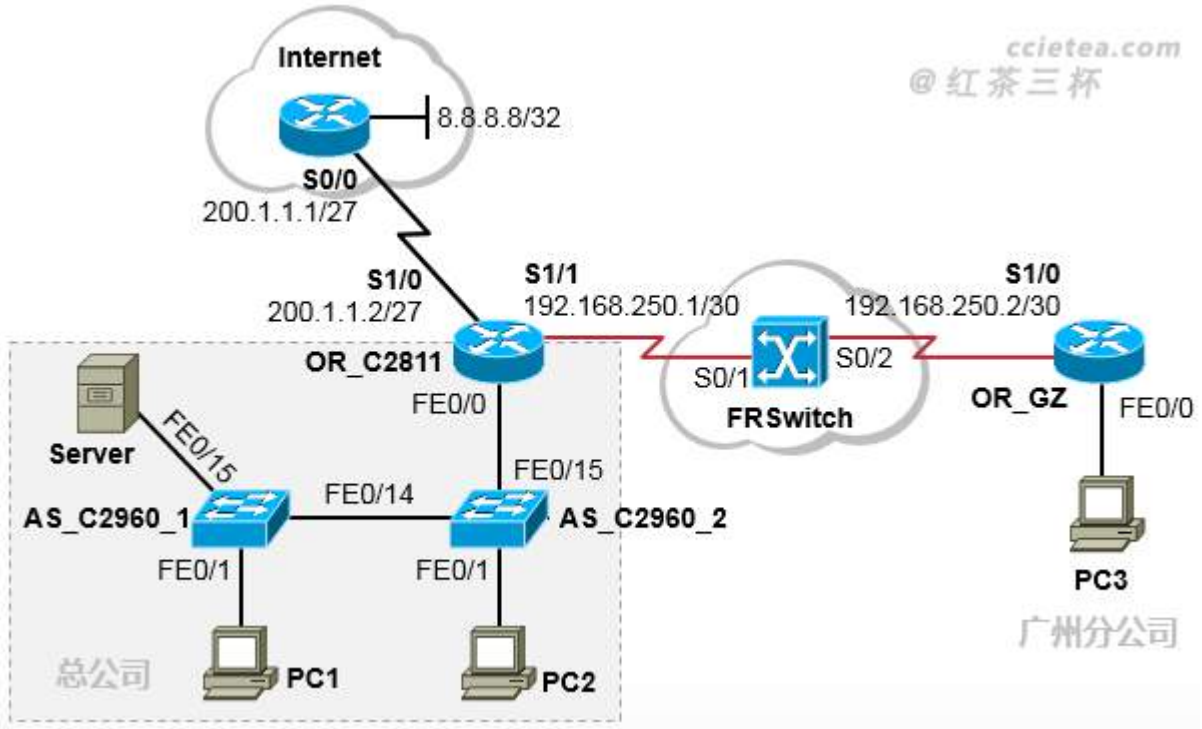

图 6-1 综合实验 1 网络拓扑

1. PC1 属于 VLAN10；PC2 属于 VLAN20；网关均在 OR_C2811。VLAN10、20 对应网段分别为 192.168.10.0/24、192.168.20.0/24 ,这两个网段的 PC 均采用 DHCP 动态获取地址 ,DHCP 服务器为内网的 Server；
2. Server 属于 VLAN200，IP 地址为 192.168.200.1/24，网关也在 OR_C2811 上；
3. 总公司两台交换机的管理 VLAN 为 255，IP 地址段为 192.168.255.0/24，要求只允许总公司内网的 VLAN10 及 VLAN20 用户 telnet 到交换机上进行管理；
4. PC3 的地址为 192.168.30.1/24（静态配置），通过总公司出口访问 Internet，PC3 的网关在 OR_GZ 路由器上；
5. VLAN10、20 用户均能访问 Internet 及广州站点 PC3；
6. 广州分公司的 PC3 能访问总公司内网资源；

7. Internet 用户能访问总公司 Server 的 WEB 服务，通过 8080 端口访问；
8. 总公司路由器与运营商路由器之间的链路采用 PPP 封装，使用的是 PAP 认证，用户名为 C39001，密码为 CO5566，OR_C2811 路由器为被认证端；
9. 总公司与广州分公司之间的广域网链路为帧中继链路，OR_C2811 这一侧的 DLCI 为 117，OR_GZ 的 DLCI 为 133。

配置及实现

1. 完成所有 PC 的配置

PC1（使用路由器模拟）的配置如下：

```
Router(config)# hostname PC1
PC1(config)# no ip routing
PC1(config)# interface fastEthernet 0/0
PC1(config-if)# ip address dhcp                    !! 通过 DHCP 获取地址及网关
PC1(config-if)# no shutdown
```

PC2（使用路由器模拟）的配置如下：

```
Router(config)# hostname PC2
PC2(config)# no ip routing
PC2(config)# interface fastEthernet 0/0
PC2(config-if)# ip address dhcp
PC2(config-if)# no shutdown
```

PC3（使用路由器模拟）的配置如下：

```
Router(config)# hostname PC3
PC3(config)# no ip routing
PC3(config)# interface fastEthernet 0/0
PC3(config-if)# ip address 192.168.30.1 255.255.255.0
PC3(config-if)# no shutdown
PC3(config-if)# exit
PC3(config)# ip default-gateway 192.168.30.254
```

2. 配置服务器 Server

Server 的配置如下：

```
Server(config)# no ip routing
Server(config)# ip http server                     !!激活 http 服务，也就是侦听 80 端口，因为需
```

```
求中提到，外网需访问这台服务器的 web 服务

Server(config)# service dhcp                    !!激活 DHCP 服务
Server(config)# ip dhcp excluded-address 192.168.10.254
Server(config)# ip dhcp excluded-address 192.168.20.254
Server(config)# ip dhcp pool vlan10                    !!该地址池对应 VLAN10
Server(dhcp-config)# network 192.168.10.0 /24
Server(dhcp-config)# default-router 192.168.10.254
Server(dhcp-config)# exit

Server(config)# ip dhcp pool vlan20                    !!该地址池对应 VLAN20
Server(dhcp-config)# network 192.168.20.0 /24
Server(dhcp-config)# default-router 192.168.20.254
Server(dhcp-config)# exit

Server(config)# Interface fastEthernet 0/0
Server(config-if)# ip address 192.168.200.1 255.255.255.0
Server(config-if)# no shutdown
Server(config-if)# exit

Server(config)# ip default-gateway 192.168.200.254      !! 为 server 配置默认网关
```

3. 配置接入层交换机

AS_C2960_1 的配置如下：

```
Switch(config)# hostname AS_C2960_1
AS_C2960_1(config)# vlan 10                    !!创建 vlan10、20、200 以及 255
AS_C2960_1(config-vlan)# exit
AS_C2960_1(config)# vlan 20
AS_C2960_1(config-vlan)# exit
AS_C2960_1(config)# vlan 200
AS_C2960_1(config-vlan)# exit
AS_C2960_1(config)# vlan 255
AS_C2960_1(config-vlan)# exit

AS_C2960_1(config)# Interface fastEthernet 0/1
AS_C2960_1(config-if)# switchport mode access
AS_C2960_1(config-if)# switchport access vlan 10
```

```
AS_C2960_1(config)# Interface fastEthernet 0/15
AS_C2960_1(config-if)# switchport mode access
AS_C2960_1(config-if)# switchport access vlan 200
AS_C2960_1(config)# Interface fastEthernet 0/14
AS_C2960_1(config-if)# switchport trunk encapsulation dot1q
AS_C2960_1(config-if)# switchport mode trunk

AS_C2960_1(config)# interface vlan 255
AS_C2960_1(config-if)# ip address 192.168.255.1 255.255.255.0
AS_C2960_1(config-if)# exit
!! 配置交换机的管理 IP（配置在管理 VLAN 接口上）

AS_C2960_1(config)# ip default-gateway 192.168.255.254
!! 为交换机配置默认网关

AS_C2960_1(config)# enable secret ccietea.com        !! 配置特权密码

AS_C2960_1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
AS_C2960_1(config)# access-list 1 permit 192.168.20.0 0.0.0.255
!!用于匹配允许 Telnet 本设备的的源
AS_C2960_1(config)# line vty 0 4
AS_C2960_1(config-line)# password ccietea
AS_C2960_1(config-line)# login
AS_C2960_1(config-line)# access-class 1 in
```

注意，如果采用模拟器做这个实验，则创建 VLAN 是在特权模式下先进入 vlan database 再创建。另外如果使用三层交换机模拟这两台接入层交换机，那么上面设置默认网关的命令输入进去设备是不识别的，除非使用 no ip routing 关闭三层交换机的路由功能。或者如果不想关闭 ip routing，则可以使用 ip route 0.0.0.0 0.0.0.0 192.168.255.254 来替代上面的 ip default-gateway 命令。

AS_C2960_2 的配置如下：

```
Switch(config)# hostname AS_C2960_2
AS_C2960_2(config)# vlan 10
AS_C2960_2(config-vlan)# exit
AS_C2960_2(config)# vlan 20
AS_C2960_2(config-vlan)# exit
AS_C2960_2(config)# vlan 200
AS_C2960_2(config-vlan)# exit
```

```
AS_C2960_2(config)# vlan 255
AS_C2960_2(config-vlan)# exit

AS_C2960_2(config)# Interface fastEthernet 0/1
AS_C2960_2(config-if)# switchport access vlan 20
AS_C2960_2(config)# Interface fastEthernet 0/14
AS_C2960_2(config-if)# switchport trunk encapsulation dot1q
AS_C2960_2(config-if)# switchport mode trunk
AS_C2960_2(config)# Interface fastEthernet 0/15
AS_C2960_2(config-if)# switchport trunk encapsulation dot1q
AS_C2960_2(config-if)# switchport mode trunk
AS_C2960_2(config-if)# exit

AS_C2960_2(config)# interface vlan 255
AS_C2960_2(config-if)# ip address 192.168.255.2 255.255.255.0
AS_C2960_2(config-if)# exit

AS_C2960_2(config)# ip default-gateway 192.168.255.254

AS_C2960_2(config)# enable secret ccietea.com

AS_C2960_2(config)# access-list 1 permit 192.168.10.0 0.0.0.255
AS_C2960_2(config)# access-list 1 permit 192.168.20.0 0.0.0.255
AS_C2960_2(config)# line vty 0 4
AS_C2960_2(config-line)# password ccietea
AS_C2960_2(config-line)# login
AS_C2960_2(config-line)# access-class 1 in
```

4. 配置总公司出口路由器使得总公司内网的不同 vlan 之间能够互访

OR_C2811 的配置如下

```
Router(config)# hostname OR_C2811
OR_C2811(config)# Interface fastEthernet 0/0
OR_C2811(config-if)# no shutdown
OR_C2811(config-if)# exit
OR_C2811(config)# Interface fastEthernet0/0.10
OR_C2811(config-if)# encapsulation dot1q 10
OR_C2811(config-if)# ip address 192.168.10.254 255.255.255.0
```

```
OR_C2811(config-if)# Interface fastEthernet 0/0.20
OR_C2811(config-if)# encapsulation dot1q 20
OR_C2811(config-if)# ip address 192.168.20.254 255.255.255.0
OR_C2811(config-if)# Interface fastEthernet0/0.200
OR_C2811(config-if)# encapsulation dot1q 200
OR_C2811(config-if)# ip address 192.168.200.254 255.255.255.0
OR_C2811(config-if)# Interface fastEthernet0/0.255
OR_C2811(config-if)# encapsulation dotq 255
OR_C2811(config-if)# ip address 192.168.255.254 255.255.255.0
OR_C2811(config-if)# exit
```

5. 验证 DHCP

在上述配置完成后，在 PC1、PC2 上使用 show ip interface brief 验证一下看看是否获取到 IP 地址。然而等了老久，PC1 及 PC2 就是获取不到地址。碰到这样的故障该如何定位呢？一般的操作是，手工为 PC1 配置一个 IP 地址以及网关地址，例如配置一个 192.168.10.1/24 的地址，网关为 192.168.10.254。然后从 PC1 去 ping 网关看看是否能通，再去 ping DHCP 服务器看看是否能通，如果发现都能通，则表明从 PC 到 DHCPserver 之间的三层通路没有问题，如果不通，那么就要检查一下各个设备的配置。

现在按照我们上面所做的配置，PC 跟 DHCPserver 之间的三层通路是没有问题的，那么为什么两台 PC 都无法通过 DHCP 获取到地址呢？仔细思考一下，DHCP 的数据交互过程，一般是由于 PC 也就是 DHCP 客户端发送广播的 DHCP Discovery 消息，试图发现网络中的 DHCP 服务器，为什么要用广播的方式呢，这是因为 PC 压根不知道 DHCP 服务器在哪里。但是再一想，广播数据包是无法跨三层设备的，但是这个拓扑中，PC1、PC2 和 DHCPserver 是属于不同网段的，中间隔着 OR_C2811 路由器(的子接口)，因此 PC1 及 PC2 发送出来的 DHCP 广播消息其实是被透传到了 OC_C2811 的子接口 FE0/0.10 及 F0/0.20 上，然后被阻挡，那么自然 DHCPserver 就无法收到这些报文了。

因此为了让 PC 能够通过 DHCP 获取到地址，我们需要在 OR_C2811 上增加如下配置：

```
OR_C2811(config)# Interface fastethernet 0/0.10
OR_C2811(config-if)# ip helper-address 192.168.200.1
OR_C2811(config-if)# Interface fastethernet 0/0.20
OR_C2811(config-if)# ip helper-address 192.168.200.1
```

ip helper-address 192.168.200.1 这条命令的意思是当该接口收到一个广播的 DHCP 消息，则将该广播消息“变成”单播包，然后发送给目的地 192.168.200.1，这叫做 DHCP 中继。

完成上述配置后，可以在 PC1 及 PC2 上 shutdown 再 no shutdown 一下接口，应该马上就能拿到地址：

```
*Mar  1 00:11:26.787: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned
DHCP address 192.168.10.1, mask 255.255.255.0, hostname PC1

PC1#show ip interface breif
Interface              IP-Address      OK?  Method  Status         Protocol
FastEthernet0/0        192.168.10.1    YES  DHCP    up             up
```

6. 验证内网连通性

我们要做的测试包含如下几项：

- PC1 及 PC2 是否能获取到 IP 地址。
- PC1 及 PC2 互相 ping 是否能 ping 通。
- PC1 及 PC2 telnet 接入交换机 AS_C3640_1（192.168.255.1）及 AS_C3640_1（192.168.255.2）。

上述几项需全部成功，才算满足需求。有一点要提醒的是，如果 PC1 或 PC2 迟迟无法获取到地址，可通过将其接口 shutdown 再 no shutdown 的方式进行“重激活”，再看看是否能获取到 IP 地址，如果仍然无法获取到，那么就要进行错误的排查。

7. 配置 Internet 路由器

Internet 路由器的配置是最简单的：

```
Internet(config)# username C39001 password CO5566
Internet(config)# Interface serial0/0
Internet(config-if)# Ip address 200.1.1.1 255.255.255.224   !!与总公司路由器对接的接口
Internet(config-if)# clock rate 64000
Internet(config-if)# no shutdown
Internet(config-if)# encapsulation ppp                      !!接口封装 PPP
Internet(config-if)# ppp authentication pap                 !!开启 PAP 验证
Internet(config-if)# exit

Internet(config)# interface loopback0
Internet(config-if)# ip address 8.8.8.8 255.255.255.255    !!模拟 Internet 上的一个节点
```

8. 配置 PAT 使得总公司用户能够访问外网 ;配置静态 NAT 端口映射 ,使得外网能够访问 Server

增加配置到 OR_C2811 路由器：

```
OR_C2811(config)# Interface fastethernet 0/0.10
```

```
OR_C2811(config-if)# ip nat inside
!! 注意在这个实验中，ip nat inside 是配置在子接口上，而不是在物理接口上的
OR_C2811(config)# Interface fastethernet 0/0.20
OR_C2811(config-if)# ip nat inside
OR_C2811(config)# Interface fastethernet 0/0.200
OR_C2811(config-if)# ip nat inside

OR_C2811(config)# Interface serial1/0
OR_C2811(config-if)# ip address 200.1.1.2 255.255.255.224
OR_C2811(config-if)# no shutdown
OR_C2811(config-if)# ip nat outside
OR_C2811(config-if)# encapsulation ppp                    !!接口封装 PPP
OR_C2811(config-if)# ppp pap sent-username C39001 password CO5566          !! 配 置 用
于 pap 的用户名及密码

OR_C2811(config)# Ip route 0.0.0.0 0.0.0.0 200.1.1.1

OR_C2811(config)# access-list 1 permit 192.168.10.0 0.0.255.255
OR_C2811(config)# access-list 1 permit 192.168.20.0 0.0.255.255  !! 这个 ACL 用于 PAT

OR_C2811(config)# Ip nat inside source list 1 interface serial1/0 overload
OR_C2811(config)# ip nat inside source static tcp 192.168.200.1 80 200.1.1.10 8080
!!将内网服务器 192.168.200.1 的 80 端口映射到 200.1.1.10 的 8080 端口
```

首先查看一下 OR_C2811 的公网出口：

OR_C2811#show ip interface brief

Interface	IP-Address	OK?	Method	Status	Protocol
FastEthernet0/0	unassigned	YES	unset	up	up
FastEthernet0/0.10	192.168.10.254	YES	manual	up	up
FastEthernet0/0.20	192.168.20.254	YES	manual	up	up
FastEthernet0/0.200	192.168.200.254	YES	manual	up	up
FastEthernet0/0.255	192.168.255.254	YES	manual	up	up
Serial1/0	200.1.1.2	YES	manual	up	up

S1/0 接口的物理和协议均已 UP 了。

测试方法：

PC1#ping 8.8.8.8

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
```

在 OR_C2811 上查看 NAT 表项：

OR_C2811#show ip nat translations

Pro	Inside global	Inside local	Outside local	Outside global
icmp	200.1.1.2:1	192.168.10.1:1	8.8.8.8:1	8.8.8.8:1
icmp	200.1.1.2:2	192.168.10.1:2	8.8.8.8:2	8.8.8.8:2
tcp	200.1.1.10:8080	192.168.200.1:80	---	

再尝试从 Internet 路由器 telnet 200.1.1.10 8080，看到的正确现象应该是能登陆上去，敲回车一直在换行，没有回显，但是从 Server 上 show tcp brief 能看到现象：

Server#show tcp brief

TCB	Local Address	Foreign Address	(state)
63FAD828	192.168.200.1.80	200.1.1.1.55262	ESTAB

9. 配置 FR switch

FRswitch 的配置相对比较简单：

```
FRswitch(config)# frame-relay switching
FRswitch(config)# Interface serial 0/1
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# frame-relay route 117 interface s0/2 133
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# no shutdown

FRswitch(config-if)# Interface serial 0/2
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# frame-relay route 133 interface s0/1 117
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# no shutdown
```

FRswitch#show frame-relay route

Input Intf	Input Dlci	Output Intf	Output Dlci	Status

Serial0/1	117	Serial0/2	133	inactive
Serial0/2	133	Serial0/1	117	inactive

10. 配置广州分公司出口路由器及总公司出口路由器使得广州分公司能访问总公司以及外网

OR_GZ 的配置如下：

```
OR_GZ(config)# Interface serial 1/0
OR_GZ(config-if)# ip address 192.168.250.2 255.255.255.252
OR_GZ(config-if)# encapsulation frame-relay
OR_GZ(config-if)# no frame-relay inverse-arp
OR_GZ(config-if)# frame-relay map ip 192.168.250.1 133 broadcast
OR_GZ(config-if)# no shutdown

OR_GZ(config-if)# Interface fastEthernet 0/0
OR_GZ(config-if)# ip address 192.168.30.254 255.255.255.0
OR_GZ(config-if)# no shutdown

OR_GZ(config)# Ip route 0.0.0.0 0.0.0.0 192.168.250.1
```

OR_C2811 的配置增加如下：

```
OR_C2811(config)# Interface serial 1/1
OR_C2811(config-if)# ip address 192.168.250.1 255.255.255.252
OR_C2811(config-if)# encapsulation frame-relay
OR_C2811(config-if)# no frame-relay inverse-arp
OR_C2811(config-if)# frame-relay map ip 192.168.250.2 117 broadcast
OR_C2811(config-if)# no shutdown
OR_C2811(config-if)# ip nat inside        !!分公司要通过这个接口，进而 PAT 出去访问外网，所以这个接口必须配置 ip nat inside

OR_C2811(config)# Ip route 192.168.30.0 255.255.255.0 192.168.250.2      !!静态路由指向广州分公司
```

完成配置后，简单的验证一下。

OR_C2811#show frame-relay map

```
Serial1/1 (up): ip 192.168.250.2 dlci 117(0x75,0x1C50), static,
                broadcast,
                CISCO, status defined, active
```

帧中继的 PVC 已经 active 了。

```
PC1#ping 192.168.30.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
!!!!!
PC2#ping 192.168.30.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.30.1, timeout is 2 seconds:
!!!!!
PC3#ping 8.8.8.8

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 8.8.8.8, timeout is 2 seconds:
!!!!!
```

6.2 综合实验 2

实验目的

1. 系统性地了解数据网络层次结构；
2. 掌握从无到有配置一个网络的思路；
3. 巩固 L3 switching、FrameRelay、PPP、路由选择原理等知识；
4. 掌握基本的故障排除（Trouble shooting）方法。

拓扑及需求

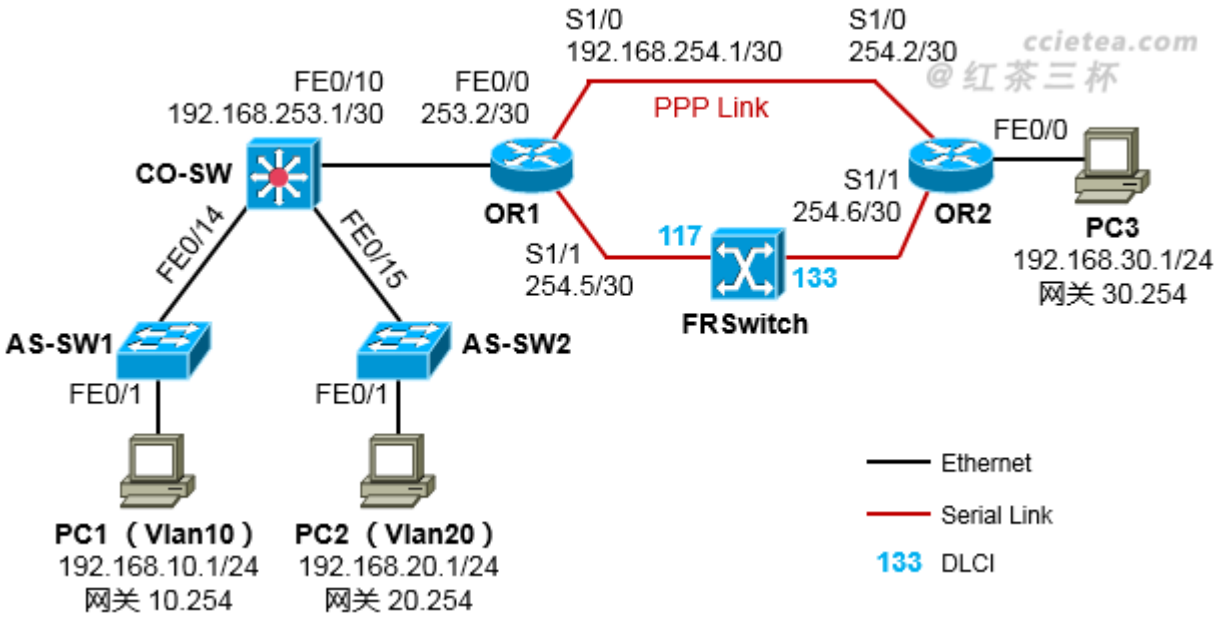

图 6-2 综合实验 2 网络拓扑

1. 网络拓扑如图所示，全网的 IP 地址统一采用 192.168 开头的地址段；
2. PC1、PC2 分别属于 Vlan10 及 vlan20，网关在核心交换机 CO-SW 上；
3. CO-SW、OR1、OR2 运行 OSPF，要求全网可达；OSPF Router-ID 采用 192.168.252.0/24 地址段，具体每台设备的 Router-ID 请自行规划；
4. OR1 的 S1/0 与 OR2 的 S1/0 通过 PPP 链路互连，要求使用 CHAP 验证，OR1 为被认证端，用户名为 ccie，密码为 ccietea.com；
5. OR1 的 S1/1 与 OR2 的 S1/1 通过帧中继链路互连，DLCI 号码如图所示；OR1、OR2 关闭 inverse-arp，使用静态 FR 映射；
6. 要求默认情况下，PC1、PC2 与 PC3 互访的流量始终走帧中继线路，当帧中继线路发生故障时则自动切换到 PPP 链路；
7. AS-SW1、AS-SW2 的管理地址分别为 192.168.255.1/24 及 192.168.255.2/24，管理 VLAN 是 vlan255，网关在核心交换机上，要求只允许 Vlan10 及 Vlan20 的用户对其进行 telnet 远程管理。

配置及实现

1. 完成所有 PC 的配置

PC1 的配置如下：

```
Router(config)# hostname PC1
PC1(config)# no ip routing
```

```
PC1(config)# interface fastEthernet 0/0
PC1(config-if)# ip address 192.168.10.1 255.255.255.0
PC1(config-if)# no shutdown
PC1(config-if)# exit
PC1(config)# ip default-gateway 192.168.10.254
```

PC2 的配置如下：

```
Router(config)# hostname PC2
PC2(config)# no ip routing
PC2(config)# interface fastEthernet 0/0
PC2(config-if)# ip address 192.168.20.1 255.255.255.0
PC2(config-if)# no shutdown
PC2(config-if)# exit
PC2(config)# ip default-gateway 192.168.20.254
```

PC3 的配置如下：

```
Router(config)# hostname PC3
PC3(config)# no ip routing
PC3(config)# interface fastEthernet 0/0
PC3(config-if)# ip address 192.168.30.1 255.255.255.0
PC3(config-if)# no shutdown
PC3(config-if)# exit
PC3(config)# ip default-gateway 192.168.30.254
```

2. 完成接入交换机及核心交换机的配置，使得 PC1、PC2 能够互相通信，并且能够管理到三台交换机

AS-SW1 的配置如下：

```
Switch(config)# hostname AS-SW1
AS-SW1(config)# vlan 10
AS-SW1(config-vlan)# exit
AS-SW1(config)# vlan 20
AS-SW1(config-vlan)# exit
AS-SW1(config)# vlan 255
AS-SW1(config-vlan)# exit

AS-SW1(config)# enable secret ccietea.com

AS-SW1(config)# Interface fastEthernet 0/1
```

```
AS-SW1(config-if)# switchport mode access
AS-SW1(config-if)# switchport access vlan 10
AS-SW1(config)# Interface fastEthernet 0/14
AS-SW1(config-if)# switchport trunk encapsulation dot1q
AS-SW1(config-if)# switchport mode trunk

AS-SW1(config)# interface vlan 255
AS-SW1(config-if)# ip address 192.168.255.1 255.255.255.0
!! 配置交换机的管理 IP（配置在管理 VLAN 的 SVI 上）
AS-SW1(config)# ip default-gateway 192.168.255.254
!! 为交换机配置默认网关，如果该交换机为三层交换机，则需先配置 no ip routing 命令关闭其路由功能再配置 ip default-gateway，因为此处模拟的是二层交换机

AS-SW1(config)# access-list 1 permit 192.168.10.0 0.0.0.255
AS-SW1(config)# access-list 1 permit 192.168.20.0 0.0.0.255
!!ACL1 用于匹配允许 Telnet 本设备的的源地址

AS-SW1(config)# line vty 0 4
AS-SW1(config-line)# password ccietea
AS-SW1(config-line)# login
AS-SW1(config-line)# access-class 1 in
```

AS-SW2 的配置如下：

```
Switch(config)# hostname AS-SW2
AS-SW2(config)# vlan 10
AS-SW2(config-vlan)# exit
AS-SW2(config)# vlan 20
AS-SW2(config-vlan)# exit
AS-SW2(config)# vlan 255
AS-SW2(config-vlan)# exit

AS-SW2(config)# enable secret ccietea.com

AS-SW2(config)# Interface fastEthernet 0/1
AS-SW2(config-if)# switchport mode access
AS-SW2(config-if)# switchport access vlan 20
AS-SW2(config)# Interface fastEthernet 0/15
```

```
AS-SW2(config-if)# switchport trunk encapsulation dot1q
AS-SW2(config-if)# switchport mode trunk

AS-SW2(config)# interface vlan 255
AS-SW2(config-if)# ip address 192.168.255.2 255.255.255.0
AS-SW2(config)# ip default-gateway 192.168.255.254

AS-SW2(config)# access-list 1 permit 192.168.10.0 0.0.0.255
AS-SW2(config)# access-list 1 permit 192.168.20.0 0.0.0.255

AS-SW2(config)# line vty 0 4
AS-SW2(config-line)# password ccietea
AS-SW2(config-line)# login
AS-SW2(config-line)# access-class 1 in
```

核心交换机 CO-SW 的配置如下：

```
Switch(config)# hostname CO-SW
CO-SW(config)# vlan 10
CO-SW(config-vlan)# exit
CO-SW(config)# vlan 20
CO-SW(config-vlan)# exit
CO-SW(config)# vlan 255
CO-SW(config-vlan)# exit

CO-SW(config)# enable secret ccietea.com

CO-SW(config)# Interface fastEthernet 0/14
CO-SW(config-if)# switchport trunk encapsulation dot1q
CO-SW(config-if)# switchport mode trunk
CO-SW(config)# Interface fastEthernet 0/15
CO-SW(config-if)# switchport trunk encapsulation dot1q
CO-SW(config-if)# switchport mode trunk

CO-SW(config)# ip routing                    !!激活 IP 单播路由功能
CO-SW(config)# interface vlan 10
CO-SW(config-if)# ip address 192.168.10.254 255.255.255.0
CO-SW(config-if)# interface vlan 20
```

```
CO-SW(config-if)# ip address 192.168.20.254 255.255.255.0
CO-SW(config-if)# interface vlan 255
CO-SW(config-if)# ip address 192.168.255.254 255.255.255.0
CO-SW(config-if)# exit

CO-SW(config)# access-list 1 permit 192.168.10.0 0.0.0.255
CO-SW(config)# access-list 1 permit 192.168.20.0 0.0.0.255
CO-SW(config)# line vty 0 4
CO-SW(config-line)# password ccietea
CO-SW(config-line)# login
CO-SW(config-line)# access-class 1 in
```

完成上述配置后，PC1、PC2 即可 ping 通对方，而且也都能够远程 telnet 到 AS-SW1、AS-SW2 及 CO-SW 上。

3. 完成核心交换机与 OR1 的三层对接

我们已经知道三层交换机是具备路由功能的，到目前为止也已经熟悉了三层交换机上的一种三层接口类型：VLAN 接口，或者叫做交换式虚拟接口（SVI）。每个 VLAN 都有一个对应的 VLAN 三层接口。

现在是时候了解三层交换机的另外一种三层接口了。思科的三层交换机支持将物理接口在二层与三层之间切换。也就是说三层交换机的物理接口既可以工作在二层，又可以工作在三层。如果交换机的物理接口工作在二层，那么它势必与 VLAN 相关，这个接口可以是 access 模式，也可以是 trunk 模式。而如果将某个物理接口设置为工作在三层，则该接口就成为类似路由器接口的一个路由口，这个接口可以直接配置 IP 地址，可以隔绝广播。在某个接口的配置模式下使用 no switchport 命令，使其工作在三层，使用 switchport 命令使其工作在二层（缺省就是）。因此在本实验中，CO-SW 的 FE0/10 接口被我们规划成三层接口，并分配 IP 地址 192.168.253.1/30，用于和 OR1 进行三层对接。

CO-SW 的配置如下：

```
CO-SW(config)# Interface fastEthernet 0/10
CO-SW(config-if)# no switchport
CO-SW(config-if)# ip address 192.168.253.1 255.255.255.252
CO-SW(config-if)# no shutdown
```

OR1 的配置如下：

```
OR1(config)# Interface fastEthernet 0/0
OR1(config-if)# ip address 192.168.253.2 255.255.255.252
OR1(config-if)# no shutdown
```

4. 完成 OR1、OR2 上关于 PPP 的配置

OR1 的配置如下：

```
OR1(config)# Interface serial 1/0
OR1(config-if)# encapsulation ppp
OR1(config-if)# ip address 192.168.254.1 255.255.255.252
OR1 (config-if)# ppp chap hostname ccie                    !! 用于 CHAP 认证的名称
OR1 (config-if)# ppp chap password ccietea.com             !! 用于 CHAP 认证的密码
OR1(config-if)# no shutdown
```

OR2 的配置如下：

```
OR2(config)# username ccie password ccietea.com
OR2(config)# Interface serial 1/0
OR2(config-if)# encapsulation ppp
OR2(config-if)# ip address 192.168.254.2 255.255.255.252
OR2(config-if)# ppp authentication chap
OR2(config-if)# no shutdown
```

完成配置后需查看一下 OR1 及 OR2 的 Serial1/0 接口的物理及协议状态是否都 UP。

5. 完成 OR1、OR2 上关于帧中继的配置

帧中继交换机可使用路由器来模拟。FRSwitch 的配置如下：

```
Router# configure terminal
Router(config)# hostname FRSwitch
FRswitch(config)# frame-relay switching                    !! 在模拟帧中继交换机的路由器上配置

FRswitch(config)# interface Serial0/1
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
FRswitch(config-if)# frame-relay route 117 interface Serial 0/2 133
FRswitch(config-if)# no shutdown
FRswitch(config-if)# exit

FRswitch(config)# interface Serial0/2
FRswitch(config-if)# encapsulation frame-relay
FRswitch(config-if)# clock rate 64000
FRswitch(config-if)# frame-relay intf-type dce
```

```
FRswitch(config-if)# frame-relay route 133 interface Serial0/1 117
FRswitch(config-if)# no shutdown
```

完成上述配置后，这条 PVC 就建立好了，在帧中继交换机上可以做一些基本的查看。目前由于 OR1 及 OR2 这两台终端还未配置，状态显示为 inactive。

FRSwitch#show frame-relay route

Input Intf	Input Dlci	Output Intf	Output Dlci	Status
Serial0/1	117	Serial0/2	133	inactive
Serial0/2	133	Serial0/1	117	inactive

现在开始配置 OR1：

```
OR1(config)#interface serial 1/1
OR1(config-if)# encapsulation frame-relay
OR1(config-if)# ip address 192.168.254.5 255.255.255.252
OR1(config-if)# no frame-relay inverse-arp
OR1(config-if)# frame-relay map ip 192.168.254.6 117 broadcast
OR1(config-if)# no shutdown
```

OR2 的配置如下：

```
OR2(config)#interface serial 1/1
OR2(config-if)# encapsulation frame-relay
OR2(config-if)# ip address 192.168.254.6 255.255.255.252
OR2(config-if)# no frame-relay inverse-arp
OR2(config-if)# frame-relay map ip 192.168.254.5 133 broadcast
OR2(config-if)# no shutdown
```

完成配置后，在 OR1 及 OR2 上查看帧中继映射：

OR1#show frame-relay map

```
Serial1/1 (up): ip 192.168.254.6 dlci 117(0x75,0x1C50), static,
                broadcast,
                CISCO, status defined, active
```

6. 完成 OSPF 的配置

CO-SW 的配置如下：

```
CO-SW(config)# interface loopback0
CO-SW(config-if)# ip address 192.168.252.11 255.255.255.255
```

```
CO-SW(config-if)# exit

CO-SW(config)# router ospf 1
CO-SW(config-router)# router-id 192.168.252.11
CO-SW(config-router)# network 192.168.10.0 0.0.0.255 area 0
CO-SW(config-router)# network 192.168.20.0 0.0.0.255 area 0
CO-SW(config-router)# network 192.168.253.0 0.0.0.3 area 0
CO-SW(config-router)# exit
```

OR1 的配置如下：

```
OR1(config)# interface loopback0
OR1(config-if)# ip address 192.168.252.1 255.255.255.255
OR1(config-if)# exit

OR1(config)# router ospf 1
OR1(config-router)# router-id 192.168.252.1
OR1(config-router)# network 192.168.253.0 0.0.0.3 area 0
OR1(config-router)# network 192.168.254.0 0.0.0.3 area 0
OR1(config-router)# network 192.168.254.4 0.0.0.3 area 0
OR1(config-router)# neighbor 192.168.254.6                    !!手工指定对端 IP 地址
OR1(config-router)# exit
```

OR2 的配置如下：

```
OR2(config)# interface loopback0
OR2(config-if)# ip address 192.168.252.2 255.255.255.255
OR2(config-if)# exit

OR2(config)# router ospf 1
OR2(config-router)# router-id 192.168.252.2
OR2(config-router)# network 192.168.30.0 0.0.0.255 area 0
OR2(config-router)# network 192.168.254.0 0.0.0.3 area 0
OR2(config-router)# network 192.168.254.4 0.0.0.3 area 0
OR2(config-router)# neighbor 192.168.254.5                    !!手工指定对端 IP 地址
OR2(config-router)# exit

OR2(config)# interface fastEthernet 0/0
OR2(config-if)# ip address 192.168.30.254 255.255.255.0
```

```
OR2(config-if)# no shutdown
OR2(config-if)# exit
```

注意，当路由器接口的二层封装为帧中继时，OSPF 如果在该接口被激活，则缺省情况下这个接口的 OSPF 网络类型为 NBMA（Non-Broadcast Multiple Access），也就是非广播多路访问，当接口的网络类型为 NBMA 时，OSPF 不会在这个接口上发送组播的 Hello 包，而这么一来，OR1 及 OR2 也就无法通过帧中继链路建立邻接关系。解决的办法主要有两个，其中之一是手工修改接口的 OSPF 网络类型，另一个方法是在 OSPF 配置模式下，使用 neighbor 命令来指定对端的 IP 地址，从而使 OSPF 发送单播的 Hello 报文到该地址，这样，OSPF 邻接关系就能够使用单播的报文来建立。OR1 的邻居表如下：

OR1#show ip ospf neighbor

Neighbor ID	Pri	State	Dead Time	Address	Interface
192.168.252.2	**1**	**FULL/BDR**	**00:01:59**	**192.168.254.6**	**Serial1/1**
192.168.252.2	0	FULL/ -	00:00:33	192.168.254.2	Serial1/0
192.168.252.11	1	FULL/DR	00:00:35	192.168.253.1	FastEthernet0/0

7. 测试及验证

现在我们来测试最后一个需求：要求默认情况下，PC1、PC2 与 PC3 互访的流量始终走帧中继线路，当帧中继线路发生故障时则自动切换到 PPP 链路。完成上述配置后，OR1 的路由表如下：

OR1#show ip route

```
O     192.168.30.0/24 [110/65] via 192.168.254.6, 00:04:43, Serial1/1
                              [110/65] via 192.168.254.2, 00:04:43, Serial1/0
O     192.168.10.0/24 [110/2] via 192.168.253.1, 00:04:43, FastEthernet0/0
O     192.168.20.0/24 [110/2] via 192.168.253.1, 00:04:43, FastEthernet0/0
      192.168.254.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.254.4/30 is directly connected, Serial1/1
C        192.168.254.2/32 is directly connected, Serial1/0
C        192.168.254.0/30 is directly connected, Serial1/0
      192.168.253.0/30 is subnetted, 1 subnets
C        192.168.253.0 is directly connected, FastEthernet0/0
      192.168.252.0/32 is subnetted, 1 subnets
C        192.168.252.1 is directly connected, Loopback0
```

我们看到 192.168.30.0/24 路由出现了等价负载均衡。这意味着当 PC1 或者 PC2 访问 PC3 时，流量在到达 OR1 后，OR1 将会把这些流量在 S1/0 及 S1/1 接口上执行等价负载分担（具体的报文转发细节这里不做讨论），也就说流量有可能走 PPP 链路，也有可能走 FR 链路。再看看 OR2 的路由表：

OR2#show ip route

```
C       192.168.30.0/24 is directly connected, FastEthernet0/0
O       192.168.10.0/24 [110/66] via 192.168.254.5, 00:05:31, Serial1/1
                        [110/66] via 192.168.254.1, 00:05:31, Serial1/0
O       192.168.20.0/24 [110/66] via 192.168.254.5, 00:05:31, Serial1/1
                        [110/66] via 192.168.254.1, 00:05:31, Serial1/0
        192.168.254.0/24 is variably subnetted, 3 subnets, 2 masks
C           192.168.254.4/30 is directly connected, Serial1/1
C           192.168.254.0/30 is directly connected, Serial1/0
C           192.168.254.1/32 is directly connected, Serial1/0
        192.168.253.0/30 is subnetted, 1 subnets
O           192.168.253.0 [110/65] via 192.168.254.5, 00:05:33, Serial1/1
                          [110/65] via 192.168.254.1, 00:05:33, Serial1/0
        192.168.252.0/32 is subnetted, 1 subnets
C           192.168.252.2 is directly connected, Loopback0
```

OR2 关于 PC1 及 PC2 所在网段的路由也出现了等价负载均衡。现在，我们为了让流量始终走帧中继链路，就要采取一定的策略，方法非常简单，就是将 PPP 链路的 Cost 调高。具体的操作需要在 OR1 及 OR2 上来完成。

OR1 的配置增加如下：

```
OR1(config)# interface serial 1/0
OR1(config-if)# ip ospf cost 8888
```

OR2 的配置增加如下：

```
OR2(config)# interface serial 1/0
OR2(config-if)# ip ospf cost 8888
```

OR1#show ip route

```
O       192.168.30.0/24 [110/65] via 192.168.254.6, 00:00:01, Serial1/1
O       192.168.10.0/24 [110/2] via 192.168.253.1, 00:00:01, FastEthernet0/0
O       192.168.20.0/24 [110/2] via 192.168.253.1, 00:00:01, FastEthernet0/0
        192.168.254.0/24 is variably subnetted, 3 subnets, 2 masks
C           192.168.254.4/30 is directly connected, Serial1/1
C           192.168.254.2/32 is directly connected, Serial1/0
C           192.168.254.0/30 is directly connected, Serial1/0
        192.168.253.0/30 is subnetted, 1 subnets
C           192.168.253.0 is directly connected, FastEthernet0/0
        192.168.252.0/32 is subnetted, 1 subnets
```

```
C        192.168.252.1 is directly connected, Loopback0
```

OR2#show ip route

```
C     192.168.30.0/24 is directly connected, FastEthernet0/0
O     192.168.10.0/24 [110/66] via 192.168.254.5, 00:00:25, Serial1/1
O     192.168.20.0/24 [110/66] via 192.168.254.5, 00:00:25, Serial1/1
      192.168.254.0/24 is variably subnetted, 3 subnets, 2 masks
C        192.168.254.4/30 is directly connected, Serial1/1
C        192.168.254.0/30 is directly connected, Serial1/0
C        192.168.254.1/32 is directly connected, Serial1/0
      192.168.253.0/30 is subnetted, 1 subnets
O        192.168.253.0 [110/65] via 192.168.254.5, 00:00:27, Serial1/1
      192.168.252.0/32 is subnetted, 1 subnets
C        192.168.252.2 is directly connected, Loopback0
```

现在 OR1 的路由表中，192.168.30.0/24 路由优选的是 FR 链路，而 OR2 的路由表中关于 192.168.10.0/24 和 192.168.20.0/24 的路由也是优选 FR 链路，满足我们的需求。在 PC1 上测试一下：

PC1#traceroute 192.168.30.1

```
Type escape sequence to abort.
Tracing the route to 192.168.30.1
  1 192.168.10.254 108 msec 60 msec 64 msec
  2 192.168.253.2 76 msec 104 msec 88 msec
  3 192.168.254.6 144 msec 116 msec 88 msec
  4 192.168.30.1 144 msec 196 msec 116 msec
```

PC1 访问 PC3 的流量，走的是 CO-SW > OR1 –FR 链路-- OR2 这条路径。然后，我们再测试一下，当 FR 链路发生故障时间，流量能否自动切换到 PPP 链路，OR1 关闭 S1/1 接口：

```
OR1(config)#interface serial 1/1
OR1(config-if)#shutdown
```

PC1#traceroute 192.168.30.1

```
Type escape sequence to abort.
Tracing the route to 192.168.30.1
  1 192.168.10.254 64 msec 88 msec 60 msec
  2 192.168.253.2 100 msec 88 msec 60 msec
  3 192.168.254.2 160 msec 92 msec 88 msec
  4 192.168.30.1 148 msec 100 msec 120 msec
```

我们看到，PC1 访问 PC3 的流量，已经切换到了 CO-SW > OR1 –PPP 链路-- OR2 这条路径。实验到此就完成了。