

## Table of Contents

60 天通过 CCNA 考试	1.1
第1天, 网络、线缆、OSI及TCP模型	1.2
第2天, CSMA/CD, 交换和虚拟局域网	1.3
第3天, 中继、DTP 及 VLAN 间路由	1.4
第4天, 路由器和交换机安全	1.5
第5天, IP 地址分配	1.6
第6天, 网络地址转换	1.7
第7天, 互联网协议版本6	1.8
第8天, IPv4与IPv6共存的网络环境	1.9
第9天, 访问控制清单	1.10
第10天, 路由的一些概念	1.11
第11天, 静态路由	1.12
第12天, OSPF基础知识	1.13
第13天, OSPF版本3	1.14
第14天, DHCP及DNS	1.15
第15天, 一二层排错	1.16
第31天, 生成树协议	1.17
第32天, 快速生成树协议	1.18
第33天, 以太网通道及链路聚合协议	1.19
第34天, 第一跳冗余协议	1.20
第35天, 启动与IOS	1.21
第36天, EIGRP	1.22
第37天, EIGRP故障排除	1.23
第38天, IPv6下的EIGRP	1.24
第39天, 开放最短路径优先协议	1.25
第40天, 系统日志、简单网络管理协议与NetFlow软件	1.26
第41天, 广域组网	1.27
第42天, 帧中继与点对点协议	1.28
附录一：华为交换机端口镜像	1.29
附录二：IPv6地址空间	1.30
附录三：GNS3简介	1.31

# 60 天通过 CCNA 考试

## Cisco CCNA in 60 Days

- Github: [github.com/gnu4cn/ccna60d](https://github.com/gnu4cn/ccna60d)
  - Gitlab: [gitlab.com/unisko/ccna60d](https://gitlab.com/unisko/ccna60d)
  - Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)
  - PDF 版本: [ccna60d.pdf](https://ccna60d.pdf)
- 

## 推荐模拟器



[GNS3 下载\(Linux, Windows, MacOS\)](#), [GNS3入门教程](#)

---

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以捐赠译者：



图 0-1 - 赞助译者 - 支付宝 付款码



图 0-2 - 赞助译者 - 微信 付款码



图 0-3 - 赞助译者 - **Bitcoin** 付款码

## 捐赠记录

2017-08-03

- “十円”通过支付宝进行了捐赠，并留言“谢谢译者的辛勤付出！”

2017-05-21

- “远”通过支付宝进行了捐赠，并留言“60天通过ccna对我帮助很大期待更新”

## 更新记录

2020-10-27

- 生成PDF版本

2019-10-30

- 完成全部章节翻译
- 重新通过 Gitbook 进行发布

2017-07-16

- 完成第37天--EIGRP故障排除章节
- 完成第38天--IPv6下的EIGRP章节

2017-07-14

- 完成第36天--EIGRP 章节的修订， EIGRP已无问题

# 第1天 网络、线缆、OSI 以及 TCP 模型

## Networks, Cables, OSI, and TCP Models

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第一天的任务

- 阅读今天的课程内容
- 阅读 ICND1 cram 指南

今天你将学到以下内容：

- 各种网络设备及网络图表
- OSI 和 TCP 模型
- 各种线缆和传输介质
- 连接到一台路由器

此模块覆盖了 ICND1 大纲要求的以下内容：

- 了解各种不同网络设备，如路由器、交换机、网桥及集线器等的用途和功能
- 选出需要的部件来满足指定规格网络的需求
- 区分常见的不同应用，以及它们对网络的影响
- 描述 OSI 和 TCP 模型中众多协议的目的及其运作过程
- 预测网络上两台主机之间数据流的走向
- 找出合适的传输媒介、线缆、接口以及连接器以将思科网络设备连接到局域网上的其它网络设备和主机

## 网络设备

作为一名网络工程师，你将用到很多网络线缆及其它传输媒介。你需要知道哪些线缆能够与 WAN、LAN 上的设备和接口，以及管理端口工作起来。如果你之前曾学过 CompTIA Network+，那么这些知识将会是对其的复习。

### 常见的网络设备

#### 网络交换机

就在几年前，网络仍是相当小型的。这就意味着你只需简单地将所有设备连到一台或几台集线器上就够了。集线器的作用是在需要对信号放大时对其进行放大，然后传至所有其它插上集线器的设备。问题在于，一条只希望传给某台特定主机的消息，被传给了成百上千台网络上的主机。

网络交换机是更为智能版的集线器。交换机使用内容可寻址存储器(Content Addressable Memory, CAM)，因此可以记住设备所插入的端口。思科公司生产的交换机被设计可用于从小型机构到有数千台设备的大型企业网络。

交换机的基本功能是利用设备的 MAC 地址（第二层）和 IP 地址（第三层）来运行，它们也能完成更为复杂的一些工作，比如包括基于 permit/deny、协议及端口号（第四层），还有综合各层及其它方面的策略和规则的处理列表（processing lists）。

早期版本的交换机又被叫做网桥。网桥查看数据帧的源端口和 MAC 地址，以建立一个表并做出转发决定。在网桥上，有相应软件来访问这个表，在交换机上，则是由硬件（专用集成电路， Application Specific Integrated Chips, ASIC）去访问 CAM 表。因此交换机可以看成是一台多端口网桥。

采用交换机可以将你的网络划分成更小的、更可管理的部分（就是网段， segments）。进而允许单位内部的不同部门，比如人力资源、财务、法务等等，得以同时在各自的网段上工作，这是十分有用的，因为同一部门的设备大部分时间都是用于各自之间的通信。



图 1.1 -- 思科 2960 交换机

每台设备都连接到交换机的一个接口上，这样的接口被称为“端口(port)”。常见的接口速度为 100Mbps 或 1000Mbps(又叫做 1Gbps)。通常有用于连接到另一交换机的光纤端口。每台交换机又有各自的管理端口，用于连接计算机，完成初始配置并获得通过网络进行维护的访问能力。

图 1.2 展示的是一台思科 2960 交换机的近景。2960 系列有多个型号，以满足从小型到中型企业的需求。

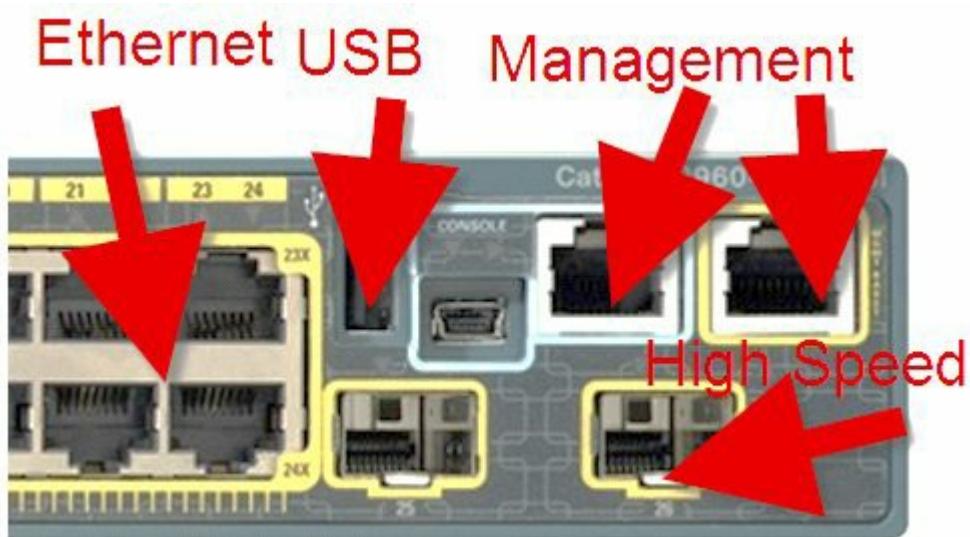


图 1.2 -- 交换机接口类型

通过交换机，你还可以更好地使用 IP 电话，因为交换机端口能够通过端口为其供电（使用 PoE 接口）。基本的网络交换机用于：

- 连接诸如打印机和 PC 这样的网络设备

- 赋予网络服务器和路由器网络访问
- 使用 VLAN 对网络进行划分

VLAN 就是虚拟局域网。

## 路由器

作为一名思科工程师，你将耗费大量时间来对路由器进行安装、配置以及故障排除。为此，CCNA 大纲超过半数的内容都是用于学习路由器配置的。

路由器（如图 1.3 所示）是用于建立网络的设备。与负责同一网络上的设备相互通信的交换机不同，路由器实现不同网络上的设备之间的通信。老旧型号的路由器上只有端口，这些端口都是物理内建与其中，固定在主板上的。这样的路由器仍然时不时的可以见到，但现代网络需要路由器具备 IP 电话、交换及安全以及能够连接到不同类型电讯公司的功能。因此，路由器是模块化的了，这就是说你有路由器机架和一些空着的插槽，能够连接大量的路由或交换模块。

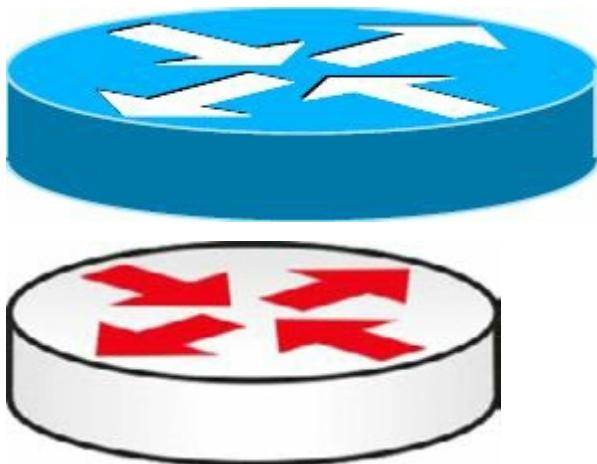


图 1.3 -- 一台模块化路由器，右边有一个空着的插槽

## 怎样在图表中表示网络

所有网络工程师都需要一种通用的方法来进行沟通，尽管在不同企业和电讯公司会用到不同的方法。如果我必须要就我的网络拓扑向你征询设计或安全方面的建议，比起我随手画出的来，如有某种一致认可的格式，肯定会来得更好。CCDA (Cisco Certified Design Associate, 思科认证的设计助理) 考试中有更多关于网络拓扑方面的知识。

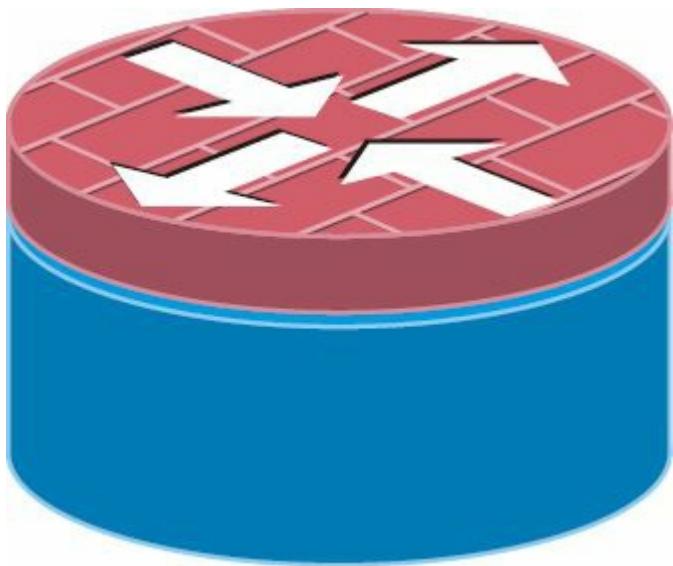
下面是一些在你作为网络工程师将会遇到的那些网络设备的符号。



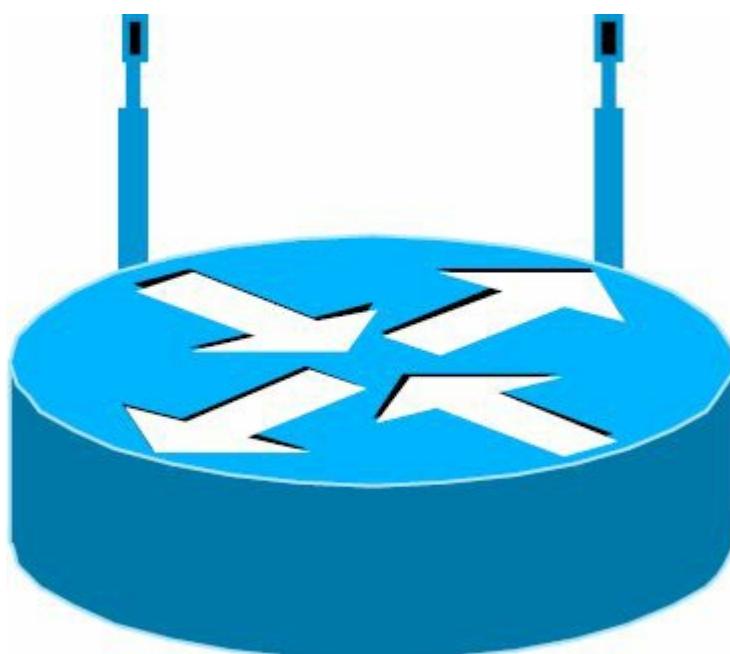
路由器, routers



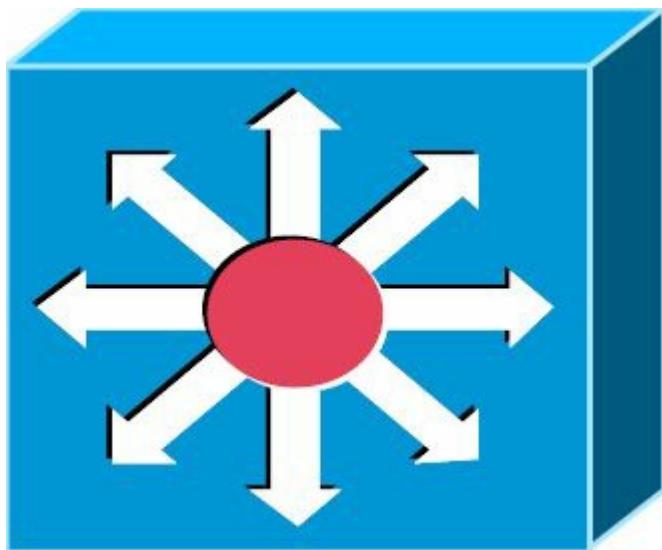
交换机, switch



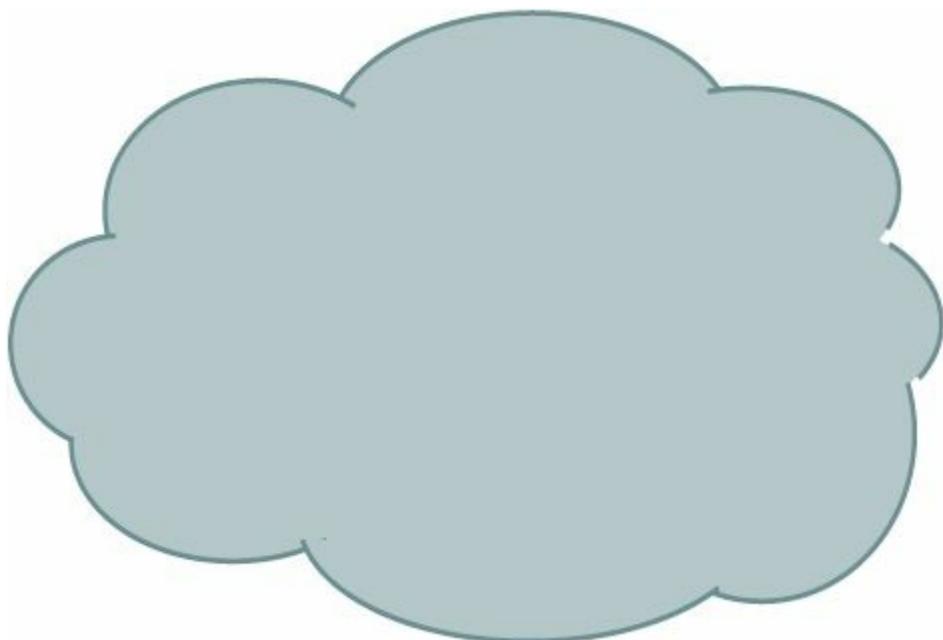
有防火墙的路由器



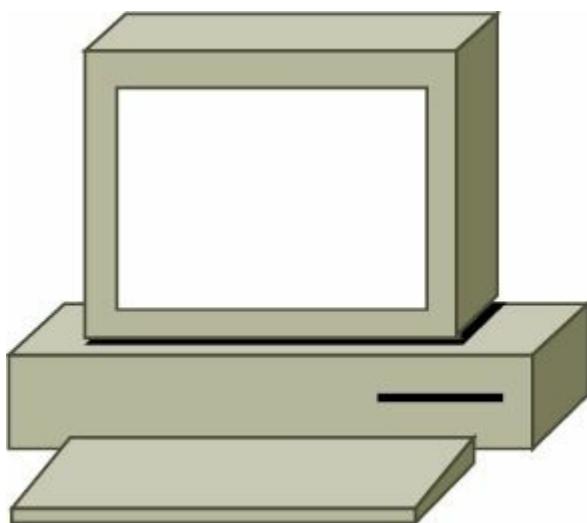
无线路由器



多层交换机



云--电讯端设备



终端设备 -- 一台 PC/串行线/以太网连接



IP 电话



防火墙

## 局域网和广域网拓扑

拓扑是指设备以何种方式进行安排，以实现通信。会因设备使用的通信协议、造价、地理分布及诸如考虑主要线路的失效所需的冗余需求等其它因素，而确定拓扑。

你也会注意到，物理拓扑和逻辑拓扑通常会有不同。物理拓扑是你所看到的网络的样子，逻辑拓扑是网络自身的样子。下面是常见的拓扑类型。

### 点对点 (point-to-point)

此种拓扑主要用在广域网中。一条点对点链路即是简单的一台设备到另一设备的连接。你可以在两台设备之间再增加一条连接，但如果设备本身失效的话，你仍将失去连通性。



图 1.4 -- 点对点拓扑

### 总线拓扑

伴随初代以太网的建立，诞生了此种拓扑，所有设备都必须连接到一条粗同轴线（a thick cable），这条粗同轴线被称为主干(the backbone)。如主干失效，则网络就会失效。如一条连接设备主干的同轴线失效，则只有该设备将失去连接。

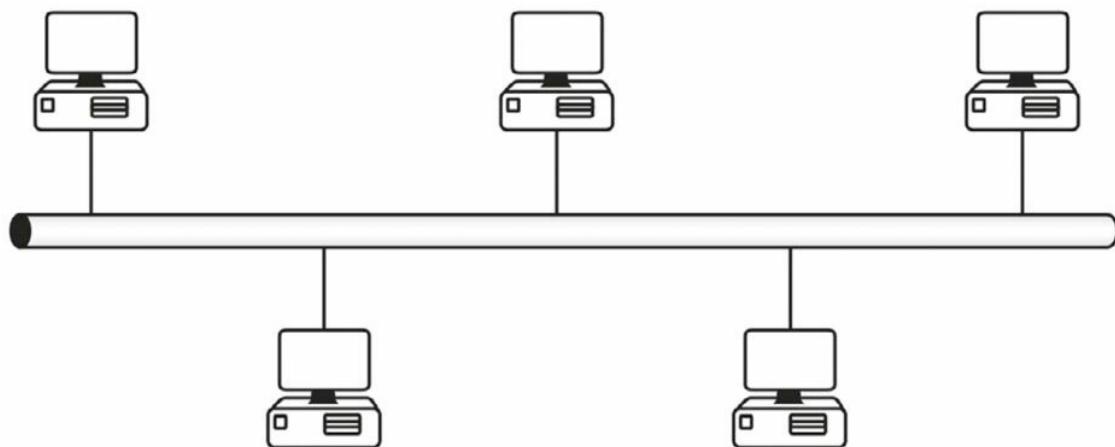


图 1.5 -- 总线拓扑

### 星形拓扑

这或许是你将遇到的最为常见的拓扑。每台网络设备都被连接到一台中心集线器或交换机。如果其中一台设备的线缆失效，则只有该设备会失去连接。

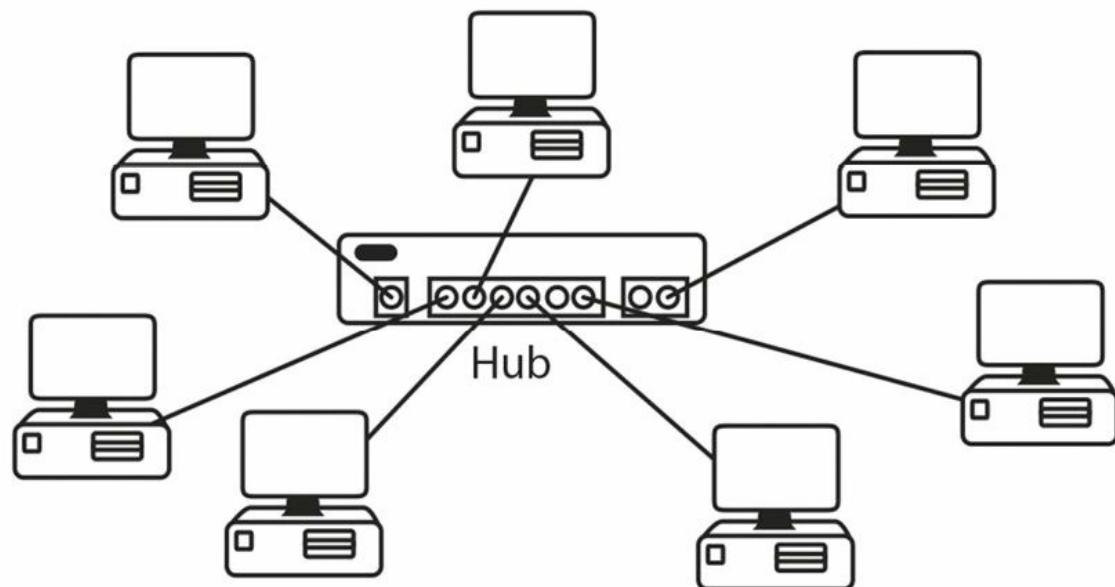


图 1.6 -- 星形拓扑

### 环形拓扑

令牌环网络 (token ring networks) 和光纤分布式数据接口网络 (Fiber Distributed Data Interface, FDDI) 网络使用此种拓扑，而两种网络在多年前就已被弃用了。

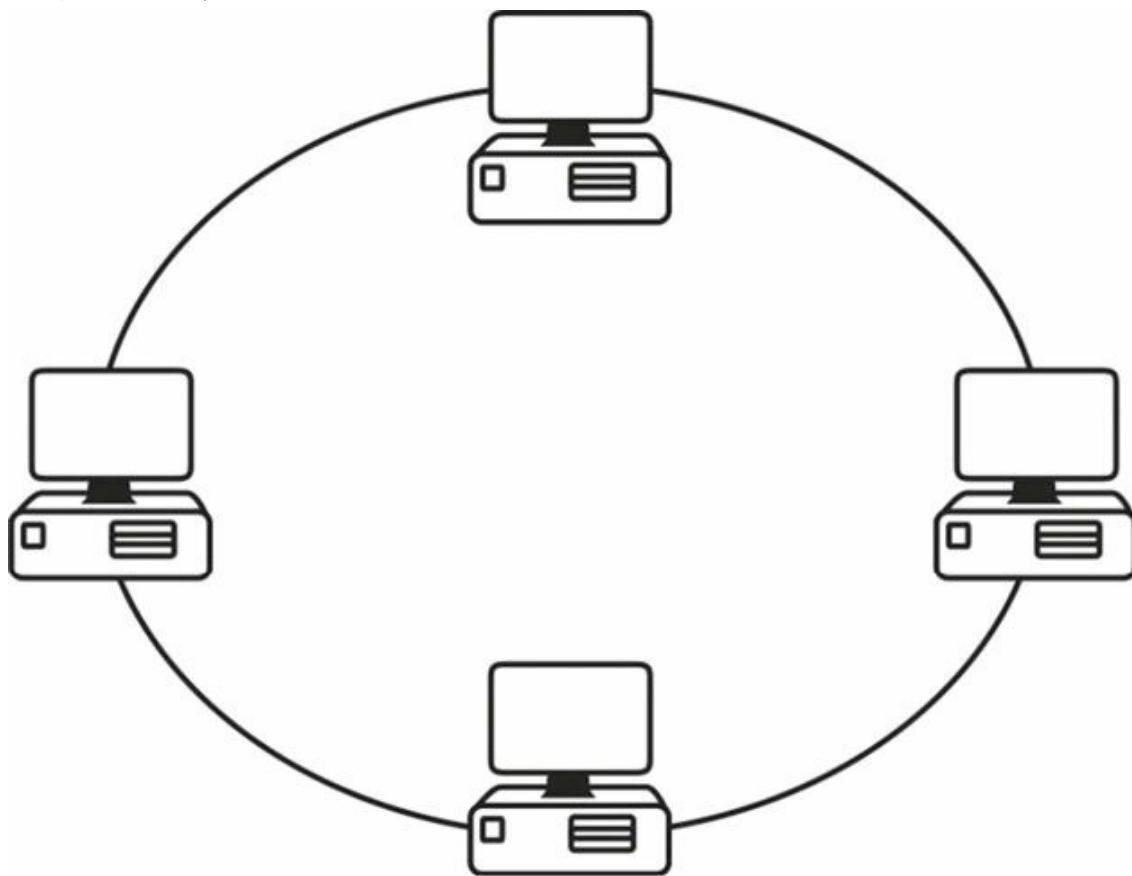


图 1.7 -- 令牌环拓扑

FDDI 网络中会用到双环连接的环形拓扑，以提供在一个环失效时的冗余。

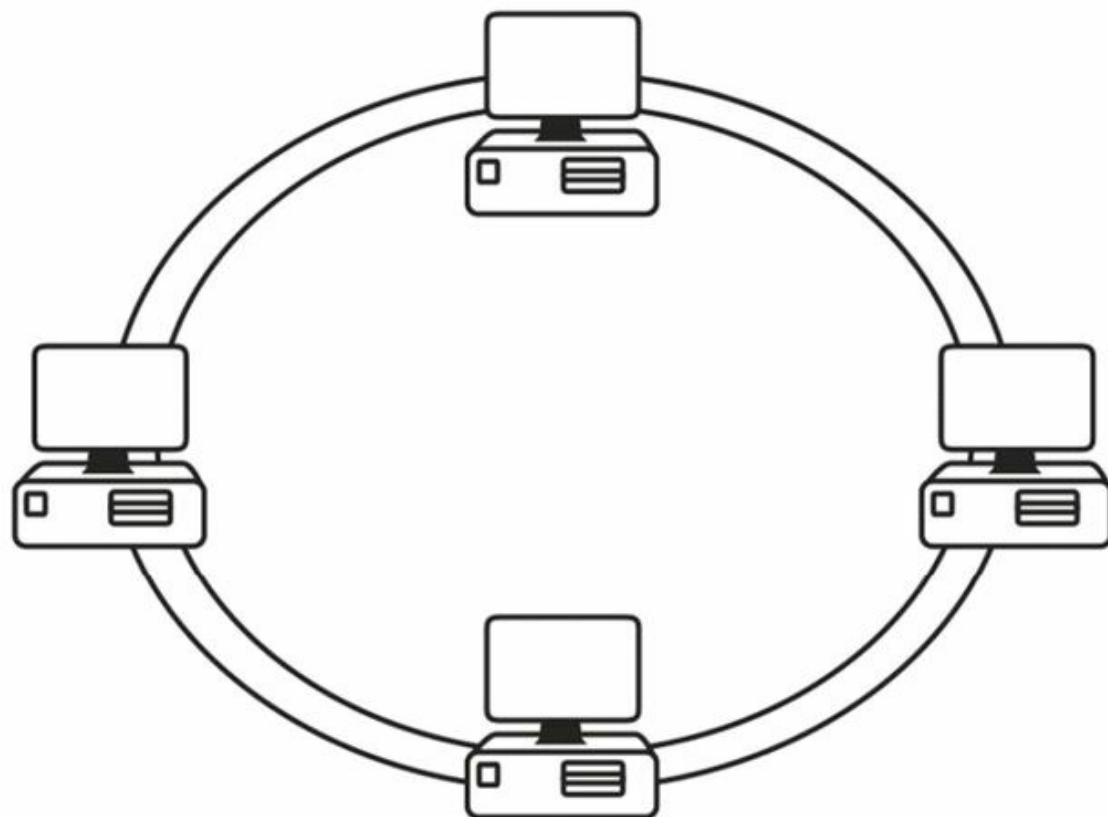


图 1.8 -- 双环拓扑

#### 网状拓扑

在不容许出现故障时间 (downtime) 时，就要考虑使用此种拓扑。完全的网状网络中每台设备都有一条到其它设备的连接。这种方案一般用在广域连接上。

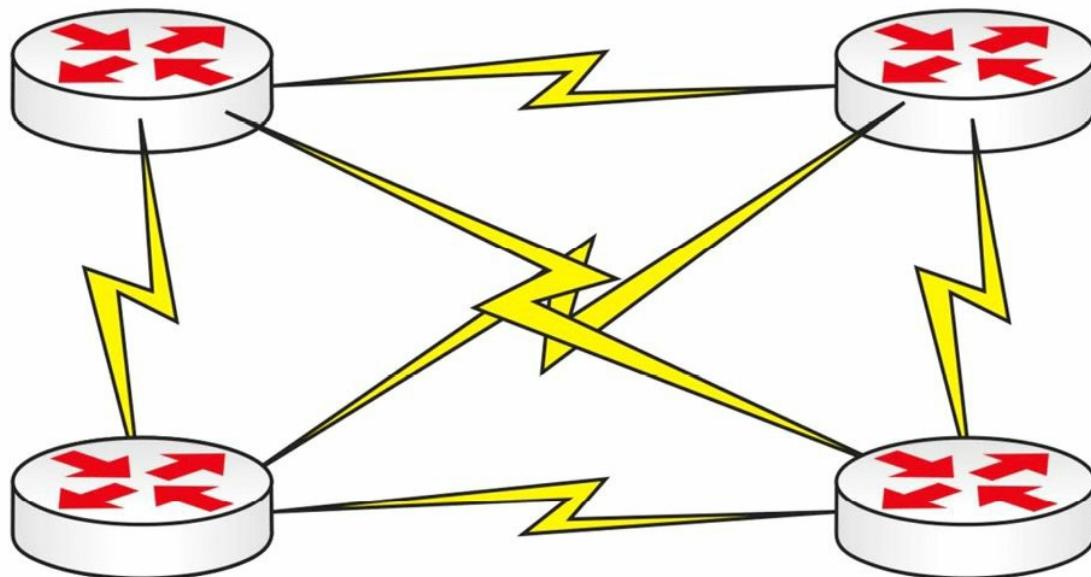


图 1.9 -- 完全的网状拓

通常这样的方案是非常费钱的。为此，会考虑采用部分网状拓扑。此时，一台设备到其它设备之间将会有  
一跳 (hops) 或几跳，它们之间会有一台以上的路由器。

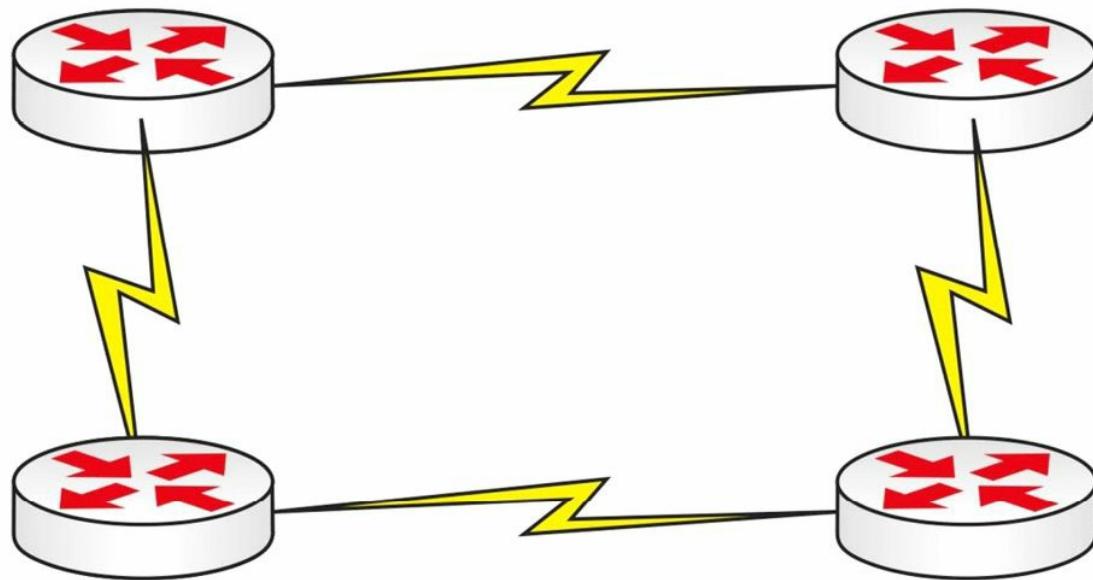


图 1.10 -- 部分网状拓扑

#### 轴辐式拓扑 (Hub-and-Spoke)

考虑到设备造价、广域网连接和带宽成本，企业往往会采用轴辐式拓扑。一台高性能路由器被放置于拓扑的轴心位置 (hub)，其往往位于企业的总部。而辐条 (spokes) 节点代表公司的各分支机构，只需不那么强大的路由器。这种拓扑有一些明显的问题，但其仍然被广泛使用。由于轴辐式拓扑所导致的路由故障占据了 CCNA 考试的较大篇幅，我们将在帧中继 (Frame Relay) 章节中回顾它。

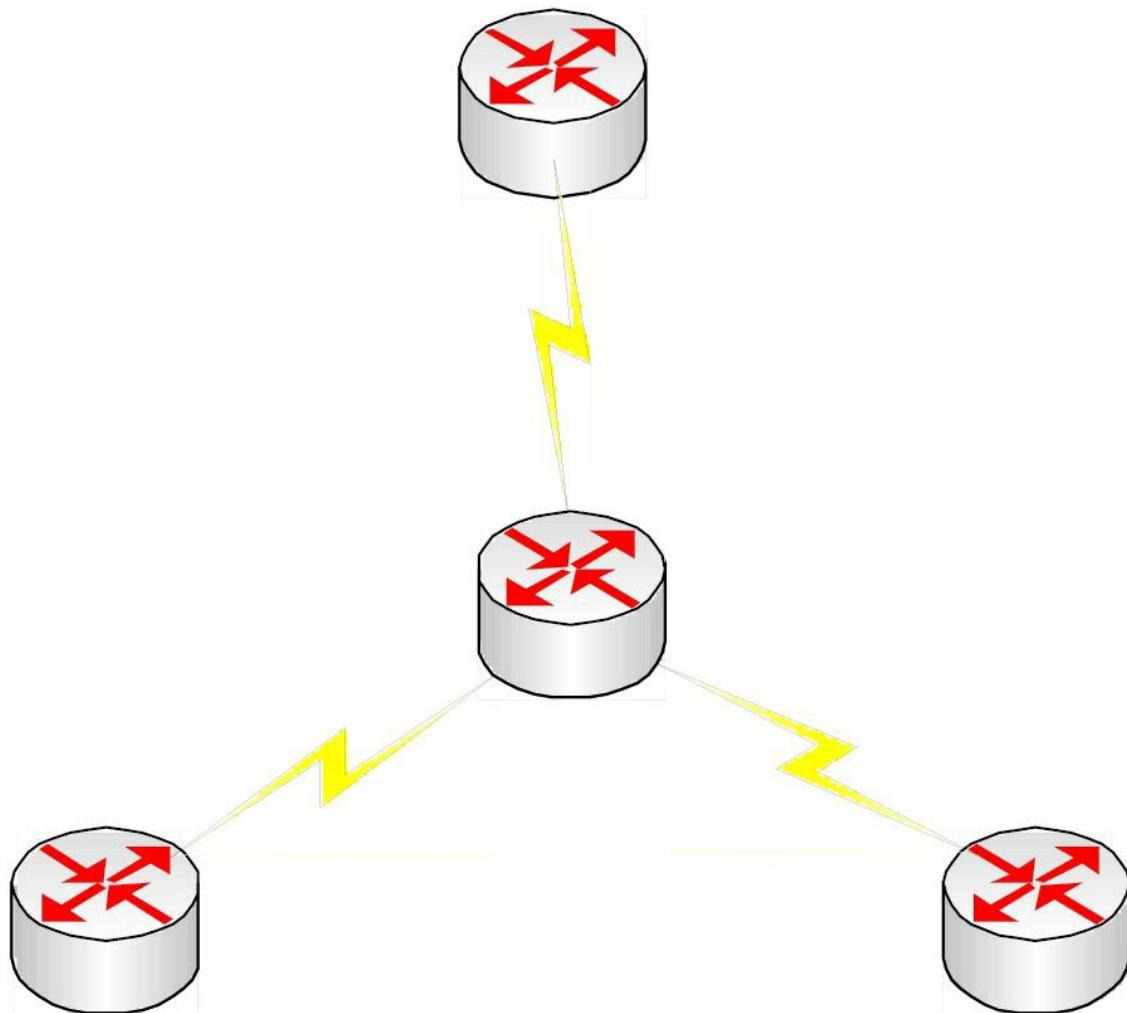


图 1.11 -- 轴辐式拓扑 (Hub-and-Spoke Topology)

#### 物理和逻辑拓扑的关系

当你能看到网络设备时，你所见的就是物理拓扑。这可能会产生误导，比如明明看起来网络是以星形拓扑布线的，实际上却是以环状逻辑运行。关于此的一个经典例子是环网。尽管流量沿环循环传输，所有设备却插入在一台集线器上。这个环其实位于令牌环集线器中，只是你无法从外界察觉到它，正如下图 1.12 所示。

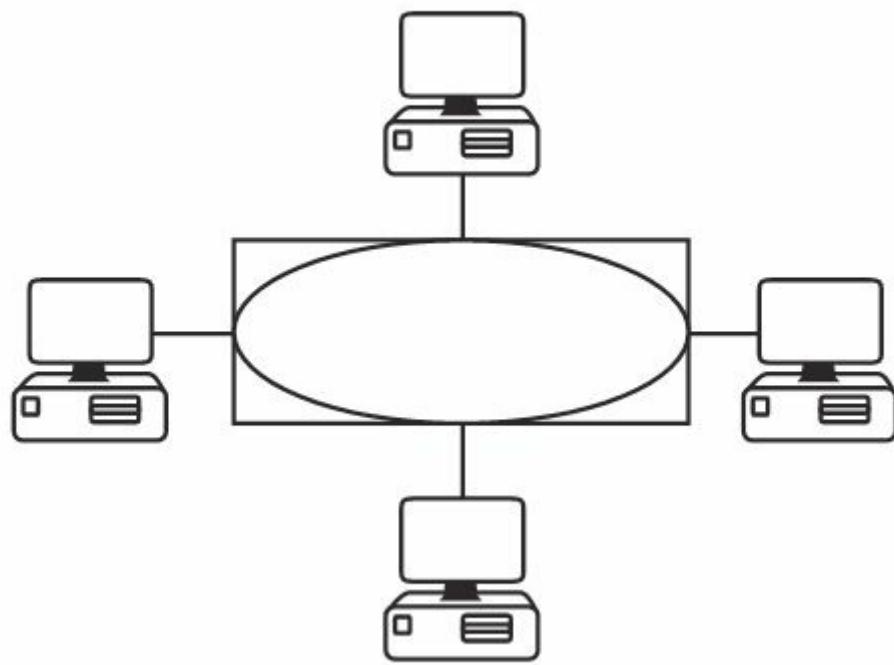


图 1.12 -- 环是在集线器内部

考试中你可能会要求从物理上和逻辑上去区分不同的网络。记住物理拓扑是你看到的网络，而逻辑拓扑是网络本身所见到的网络。表 1.1 对其进行了总结。

表 1.1 -- 物理和逻辑拓扑

拓扑	物理	逻辑
总线	总线	总线
星形	星形	总线
令牌环	星形	环形
点对点	总线	总线
FDDI	环形	环形

## OSI 和 TCP 模型

开放标准互连（Open Standards Interconnection, OSI）是有国际标准化组织创建的。伴随技术喷发，网络设备和网络软件行业兴起了几家巨头，包括思科、微软、Novell、IBM、惠普、苹果以及其它几家公司。每家都有自己的线缆和端口类型，允许各自的商业性协议。此时，如你从一家买路由器、另一家买交换机，又从别家买服务器，就会出现兼容性问题。

有一些处理这些问题的通容办法，比如在网络上部署网关来转换不同的协议，这会导致性能上的瓶颈（比如网络慢速部分）并会令到故障排除十分困难和费时。最终，厂商们不得不达成一个在各自产品上都能工作的通用标准，一套叫做 TCP/IP 的免费协议包。最后，那些未能采行 TCP/IP 的厂商失去市场份额，走向破产。

ISO 创建出 OSI 模型，以助力于各厂商就通用标准达成一致，实现厂商之间的兼容。此模型包括了将总多网络功能分解为一套逻辑分层，或通俗地称为层的东西。各层只需完成其特定的一些功能，比如说你的公司专注于防火墙，那么这些防火墙将自然地与其它厂商的设备一起工作。

此模型的优势在于每件设备设计用来出色完成一个角色，而非不充分地完成多个角色。客户可以根据其解决方案选出最好的设备，而不用死栓在一家厂商那里。同时故障排除也变得更为容易，因为确定的出错可被追踪到具体的某层。

OSI 模型将所有网络功能划分为七个不同的层。该层次化模型从第七层一路去往第一层。那些离用户更近、更为复杂的功能，在顶部，一直到处于底层的网络线缆规格，如同表 1.2 所示。

表 1.2 OSI 模型

层 #	层名
7	应用层, Application
6	表示层, Presentation
5	会话层, Session
4	传输层, Transport
3	网络层, Network
2	数据链路层, Data Link
1	物理层, Physical

#### "All People Seem To Need Data Processing"

在数据为通过物理网络介质传输而自顶层传至底层时，数据被放入不同的逻辑数据套盒子。尽管我们常把这些数据盒子称作“包（packets）”，实际上根据其处于 OSI 不同的层而有不同的名称（如图 1.13 所示）。从 OSI 模型往下的数据处理，叫做封装（见图 1.13）。而往上的处理中从盒子里取出数据的过程，叫做解封装。

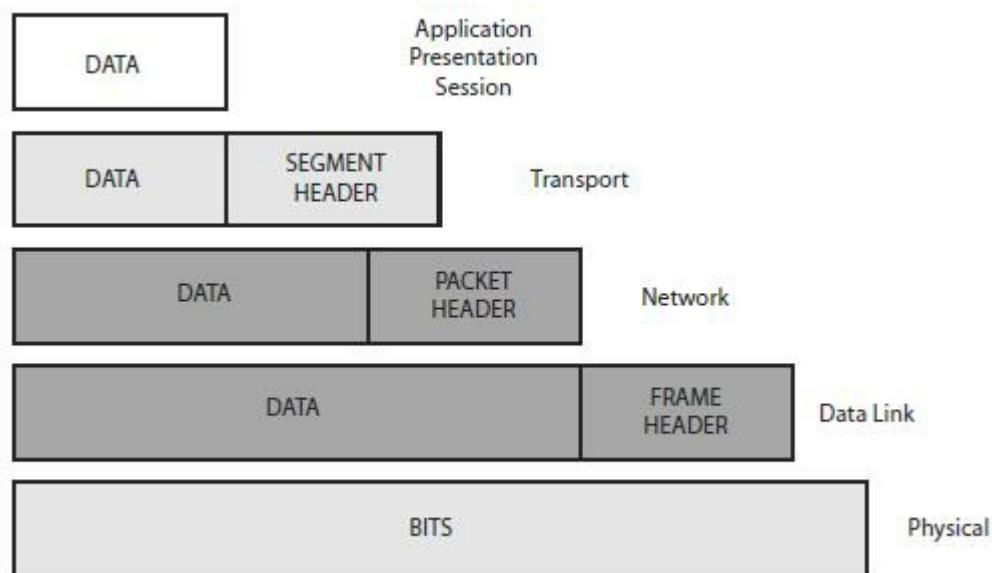


图 1.13 -- 封装

CCNA 考试要求你理解 OSI 模型，以及各层有哪些应用和协议。同时也要求你能够运用 OSI 层次化方法到故障排除中。

### 第七层 -- 应用层

这是到你我这些终端用户最近的层。应用层并非操作系统，但它通常提供了诸如 e-mail(SMTP 以及 POP3)、web 浏览（使用 HTTP） 、以及文件传输服务（使用 FTP）。应用层决定了资源的可用性。

### 表示层

表示层将数据提供给应用层。多媒体技术工作在这一层，你可以想到 MP4、JPEG、GIF 等等。而加密、解密以及数据压缩都发生在这一层。

### 会话层

会话层的角色是建立、管理及中断设备之间的会话。这些动作发生在逻辑链路上，而真正干的事情是将两个软件应用程序连接起来。SQL、RPC 以及 NFS 都工作于会话层。

### 传输层

传输层的角色是将来自更高层的数据分拆成被称为数据段（segments）的更小片。虚电路（virtual circuits）在这里建立，在设备之间能够通信之间有赖于虚电路的建立。

在数据得以跨网络传输前，传输层需要确认多少数据能发往远端设备。这取决于端到端链路的速率和可靠性。如你有一条高速链路，而终端用户只有一条低速链路，数据仍然需要以较小数据块进行发送。

以下是三种控制数据流的方法：

- 流控 flow control
- 窗口机制 windowing
- 通告机制 acknowledgements

### 流控

如发往接收系统的信息多于它所能处理的量时，它将请求发送系统暂停一段时间。这一般发生在一段使用宽带而另一端使用拨号上网的时候。这个用于通知其它设备停止的包叫做源抑制消息（a source quench message）。

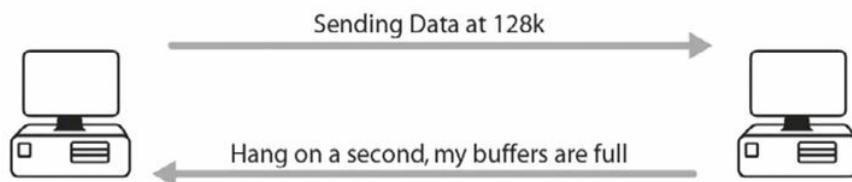


图 1.14 -- 流控

### 窗口机制

窗口机制下，每个系统就能在收到应答（acknowledgement）前发送多少数据达成一致。“窗口”随着数据的传输时开时合，以维持一个持续的数据流。

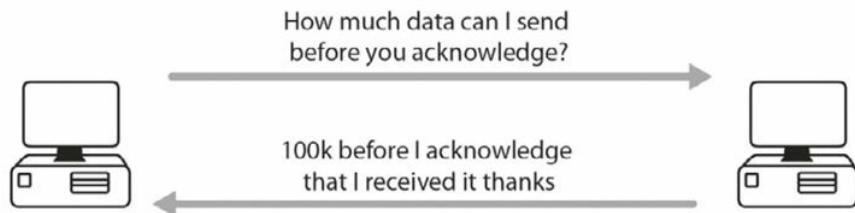


图 1.15 -- 窗口机制

### 通告机制

在收到一定数量的数据段后，接收端需要就这些数据段的安全抵达和顺序正确，通告发送端。

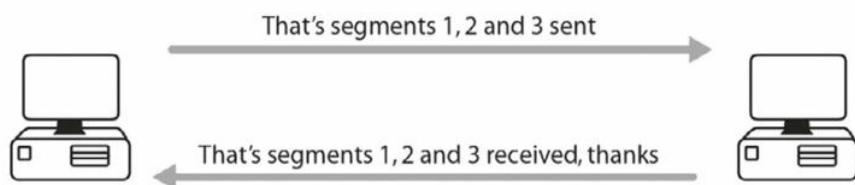


图 1.16 -- 通告机制

这些都是在一个叫做三次握手（a three-way handshake）的过程中达成一致（见图 1.17）。你要发出一个包来建立会话。第一个包叫做同步(synchronise, SYN)包。远端设备以同步应答（a synchronise acknowledgement, SYN-ACK）包予以回应。第三步的应答包（acknowledgement, ACK）的发出标志着会话的建立。这都是通过 TCP 业务完成的。

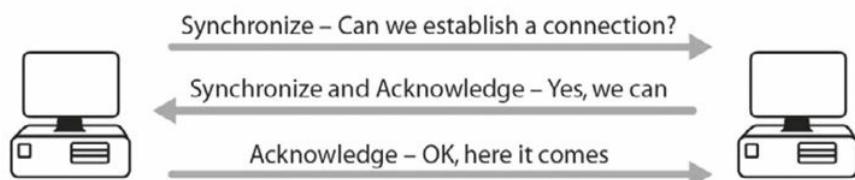


图 1.17 -- 三次握手

传输层包含了好几个协议，其中最熟知的是传输控制协议（Transmission Control Protocol, TCP）和用户数据报协议（User Datagram Protocol, UDP），它们都是 TCP/IP 协议包的组成部分。这个协议包因为是互联网所使用的标准而相当出名。**TCP 是面向连接的协议**。它使用了三次握手、窗口机制以及其它技巧来保证数据安全地到达其目的地。有许多协议都使用了 TCP，比如 Telnet、HTTPS 以及 FTP（尽管 FTP 位于应用层，它确实使用了 TCP）。

**UDP 是一个无连接协议（a connectionless protocol）**。它在对数据包进行编号后就发往目的地了。它绝不会管这些数据包是否安全抵达，也绝不会在发送数据包前建立一条连接。在数据不是那么重要，应用开发者觉得信息总是可以在未能到达目的地时重新发送时，往往采用 UDP。

那么为什么要用到 UDP 呢？TCP 协议本身会消耗许多网络的带宽，甚至在数据还没发送时，为建立其连接，也要往复发送很多流量。这都会耗去一些宝贵的时间和网络资源。UDP 数据包比起 TCP 包要小很多，在无需真正可靠连接时，它是是否有用的。使用到 UDP 的协议有 TFTP 等。

### 第三层 -- 网络层

网络层接手来自传输层的数据段，将其拆分为叫做数据包的更小单位。许多网络工程师不管在 OSI 的哪层，都把数据叫做包，这也是可以的。但是，请记住，技术上说只有在网络层，才可以叫包。

**网络层必须确定从一个网络到另一网络的最优路径；**为此，路由器工作在该层。路由器在此采用逻辑寻址，而 TCP/IP 的寻址方式是 IP 寻址，稍后会讲到。

### 数据链路层

数据链路层将数据包拆分成帧这种更小的单位。二层交换机工作在这层，使用硬件地址，或是 MAC 地址，它们之所以能够更快地交换流量，是因为无需查看 IP 地址和路由表。广域网协议工作在第二层，包括 HDLC、ISDN 以及 PPP。以太网也是第二层的。

为了向其上下两层提供接口，数据链路层又分了两个子层，分别是逻辑链路控制子层（the Logical Link Control, LLC），以及介质访问控制子层（the Media Access Control, MAC）。逻辑链路控制子层与网络层交互，介质访问控制子层与物理层交互。

### 物理层

在这层上，帧被转换为位，以将其放到线路上。这些位是由电脉冲构成，读作“开”“关”位，或是二进制的 1 和 0。集线器工作于此层，在这里你会找到线缆规格，比如 RJ45。

## OSI 故障排除

在对你的网络进行故障排除时，采用层次化方法是十分有效的。至此，你唯一要明确的是从哪个方向上运用 OSI 协议栈，是自顶向下呢，还是自底往上，抑或采用分而治之、各个击破方法，这又涉及到聚焦网络的那些部分。

我建议在初期采用自底向上的方法，对于通常处在较低的层上的问题，比如因为线缆松掉或破损、或者配置了不正确的 IP 地址时，你无需浪费时间在检查应用上。在你有更多经验后，在处理某些故障现象时采用各个击破方法会更为快速。在从底层往上处理问题时，你大概会做下面这些事情：

第一层 -- 所有线缆都恰当地插入到端口了吗？还是有的松掉了？网线头已经弯掉或是磨损了吗？如果网线有问题，设备上的指示灯会呈黄色，而非正常的绿色。是有人没有往接口上配置正确的速率吗？以太网端口速率有被设置正确吗？接口有开放给网络管理员以使用吗？

第二层 -- 接口有采用正确的协议，比如 Ethernet/PPP/HDLC，以便能够与另一端保持一致吗？

第三层 -- 接口有使用正确的 IP 地址以及子网掩码？

第四层 -- 有使用正确的路由协议吗？从路由器通告的网络是正确的吗？

在完成本书的实验过程中，你会见识到如何采行这些步骤。专家们会认为一些第四层问题是在第三层，而第二层问题实际上是在第一层，如此等等。我个人偏好于聚焦于使用层次化故障排除这一方法本身，而不是就问题所在的层去探讨。

## TCP/IP、DoD 模型

TCP/IP 模型是另一个框架，作为 OSI 模型的替代。它是由 高级防务研究项目署（the Defense Advanced Research Projects Agency, DARPA）创建的四层或五层模型。它就是为人熟知的国防部模型。自顶向下的四层分别是：

- 4 - 应用, Application [Telnet/FTP/DNS/RIP]
- 3 - 传输/主机到主机, Transport/Host-to-Host [UDP/TCP/ICMP]
- 2 - 互联网/网际网络, Internet or Internetwork [IPSec/IP]
- 1 - 链路/网络接口, Link/Network Interface [Frame Relay/Ethernet/ATM]

TCP/IP 模型已由四层更新为五层，所以你会在考试中被问到有关五层 TCP 模型（a five-layered TCP model）的问题。较高的层离用户较近，而较低的层描述了其它系统交互时所采用的技术或协议。五层 TCP 模型如下所示：

- 5 - 应用, Application [Telnet/FTP/DNS/RIP]
- 4 - 传输/主机到主机, Transport/Host-to-Host [UDP/TCP/ICMP]
- 3 - 网络层, Network [IPSec/IP]
- 2 - 数据链路层, Data Link [Ethernet/Frame Relay/PPP]
- 1 - 链路/网络接口/物理, Link/Network Interface/Physical [Bits on the wire]

五层的 TCP 模型具有更细的粒度，能更精确地表示数据放在线路之前所发生的事情。比如在第二层处，进行了数据封装以及寻址（如数据链路寻址）。考试中思科偏向选择五层模型。

数据会如同上述的 OSI 模型那样在自应用层往物理层的途中进行封装，如表 1.3 所示：

表 1.3 五层的 TCP 模型

<b>应用, Application</b>	仍未封装的数据，	
<b>传输, Transport</b>	将 TCP 头部添加到数据上, TCP header added to the data	段, Segment
<b>网络, Network</b>	IP 头部被添加上去（包括 IP 地址）, IP header added(including IP address)	包, Packet
<b>数据链路, Data Link</b>	添加数据链路头部（数据链路地址）, Data Link header added(Data Link address)	帧, Frame
<b>物理, Physical</b>	转变成电信号, Turned into electrical signals	线路上的位, Bits on the wire

你可能会被问及 TCP/IP 模型与 OSI 模型的对应关系。如表 1.4 所示：

表 1.4 -- TCP/IP 模型到 OSI 模型的对应关系

层号, Layer #	OSI 模型	TCP 模型
7	应用, Application	应用, Application
6	表示, Presentation	
5	会话, Session	
4	传输, Transport	主机到主机, Host to Host
3	网络, Network	网际网络, Internetwork
2	数据链路, Data Link	网络接口, Network Interface
1	物理, Physical	

思科选择了新的 TCP 模型而不再是 OSI 模型作为网络框架，但仍要求你理解 OSI 模型，所以在大纲中保留了 OSI 模型。

表 1.5 新旧 TCP 模型对比

旧 TCP 模型, Old TCP Model	层, Layer	新 TCP 模型, New TCP Model
应用, Application	5	应用, Application
传输, Transport	4	传输, Transport
互联网, Internet	3	网络, Network
链路/网络接口, Link/Network Interface	2	数据链路, Data Link
	1	物理, Phycial

## TCP/IP

TCP/IP 是一套完整的，可以实现通过网络通信的协议和服务套件。诸如 IPX/SPX 这样的 TCP/IP 早期竞争者也是完整的，却由于它们的应用量极少且缺乏后续演进，而消亡了。

TCP/IP 是由互联网工程任务组（the Internet Engineering Task Force, IETF）所维护的一套可自由获取和免费使用的标准，用于端端设备连通性的建立（it is used for end-to-end device connectivity）。通过请求评议（Request fo Comments, RFCs）的提交方式，它得以开发和改进。请求评议是众多工程师提交的，用于将一些新点子传送给其他成员审核的一系列文档。2663 号请求评议（RFC 2663）就是一个关于网络地转换（Network Address Translation, NAT）的实例。IETF 采纳了这些请求评议作为互联网的标准。你可以在这里了解更多的 IETF 和 RFCs 的知识：

[www.ietf.org/rfc.html](http://www.ietf.org/rfc.html)

TCP/IP 提供了很多业务，那些不包含在 CCNA 大纲中的在本书不会涉及。也会忽略那些其它部分如 DNS 和 DHCP 中的内容。以下部分是 TCP/IP 中的基础部分。因为 CCNA 并是一个基础的网络考试，所以它要求你已经对 CompTIA 的 Network+ 考试内容有很好的掌握。

## 传输控制协议，Transmission Control Protocol, TCP

TCP 运行于 OSI 模型的传输层。提供了一种用于网络设备间可靠数据传输的面向连接服务。TCP 提供流控、队列（sequencing）、窗口机制以及错误侦测。它将一个 32 位的头部附加到应用层数据，接着就封装到 IP 头部。RFC 793 描述了 TCP。常见的 TCP 端口如下所示：

- FTP 数据 -- 20
- FTP 控制 -- 21
- SSH -- 22
- Telnet -- 23
- SMTP -- 25
- DNS -- 53(也使用 UDP)
- HTTP -- 80
- POP3 -- 110
- NNTP -- 119
- NTP -- 123
- TLS/SSL -- 443

## 互联网协议，Internet Protocol, IP

IP 协议工作于 OSI 模型的网络层。它是**无连接的**，负责将数据进行跨网络传输。IP 寻址是互联网协议的一项功能。IP 检查每个数据包的网络层地址，以此确定该数据包到达目的地的最优路径。RFC 791 对 IP 进行了讨论。

## 用户数据报协议，User Datagram Protocol, UDP

UDP 也是工作于 OSI 模型的网络层。它不像 TCP 那样事先建立起连接，在网络设备之间传输信息。UDP 是**无连接的**，只是尽力投送，不保证数据抵达目的地。UDP 像是发出一封没有退回地址的信件。你只知道数据发送出去了，而永远不知道是否送到。

比起 TCP，UDP 消耗更少的带宽，适合用于相比可靠性和有保证来说，低延迟更为重要的应用。TCP 和 UDP 都是由 IP 承载的。RFC 768 对 UDP 进行了叙述。常见的 UDP 端口号有以下这些：

- DNS -- 53
- TFTP -- 69
- SNMP -- 161/162

## 文件传输协议，File Transfer Protocol, FTP

文件传输协议工作于应用层，负责透过一条远程链路**可靠地**传数据。因为它是可靠的，所以使了 TCP 来传输数据。

你可以使用 `debug ip ftp` 命令来对 FTP 流量进行调试。

FTP 使用了 `20` 和 `21` 号端口。通常，自客户端发起的到 FTP 服务器的第一次连接是在 `21` 号端口上。随后的数据连接可以是从 FTP 服务器的 `20` 号端口上离开，或者从客户端的随机端口到 FTP 服务器的 `20` 端口的连接建立。关于主动（active）和被动（passive）FTP 的内容，CCNA 考试不要求。

## 简单的文件传输协议，Trivial File Transfer Protocol, TFTP

如需不那么可靠的数据传输时，TFTP 提供了一种好的替代。TFTP 使用 UDP 端口 69，提供了一种**无连接**的数据传输方法。TFTP 可能会因为需要指定文件的位置而难于使用。

你需要有一个客户端（这里的路由器）以及 TFTP 服务器，这可以是一台路由器或者 PC，或者是网络上的服务器（最好是在同一子网上），来使用 TFTP。在服务器上有 TFTP 软件，这样文件才能拉出来并转发给客户端。

**真实世界：**将启动配置（startup configuration）以及 IOS 备份到网络上的一台服务器上，是一个非常好的主意。

TFTP 在思科路由器上用到很多，用来备份配置以及升级路由器。下面的命令执行这些功能：

```
Router#copy tftp flash:
```

会提示你输入新的 flash 文件所在的其它主机的 IP 地址：

```
Address or name of remote host []? 10.10.10.1
```

然后你必须输入其它路由器上的 flash 镜像的文件名：

```
Source filename []? / c2500-js-1.121-17.bin
Destination filename [c2500-js-1.121-17.bin]?
```

如你有一个旧版本的 IOS，你会收到是否要在拷贝前擦除路由器 flash 的提示，之后文件将被传输。当路由器再次启动时，你的新 flash 镜像就可使用了。

其它可选命令有在保存备份时用到的 `copy flash tftp` 或者在备份当前配置文件时用的 `copy running-config tftp`。

你可以使用 `debug tftp` 命令来调试 TFTP 流量。

## 简单邮件传输协议，Simple Mail Transfer Protocol, SMTP

SMTP 定义了邮件怎样从客户端发往邮件服务器。使用 TCP 来确保一条可靠的连接。SMTP 邮件以三种不同方式从 SMTP 服务器上拉出，多数网络都将 SMTP 作为一种邮件投递服务。POP3 是另一种流行的方式。POP3 是一个将邮件从服务器传至客户端的协议。SMTP 使用 TCP 端口 25。

## 超文本传输协议，Hyper Text Transfer Protocol, HTTP

HTTP 使用 TCP（80 端口）来将文本、图形以及其它多媒体文件从网页服务器发往客户端。此协议让你可以查看网页，位于 OSI 模型的应用层。HTTPS 是 HTTP 的安全版本，使用了安全的套接字层技术（Secure Socket Layer, SSL），或者传输层安全技术（Transport Layer Security, TLS）来在发送前加密数据。

你可以用 `debug ip http` 命令对 HTTP 流量进行调试。

## 远程登陆 Telnet

Telnet 使用 TCP（端口 23）来允许建立一条到某台网络设备的远程连接。在后续实验中你将对其了解更多。因为 Telnet 是不安全的，所以现今很多管理员都使用 SSH，SSH 使用 TCP 端口 22，作为一个确保安全连接的替代。Telnet 是唯一一个能够对 OSI 模型全部七层进行检查的工具，如你能够 Telnet 到某个地址，那么所有七层都是正确工作的。如你不能 Telnet 到另一设备，则并不能说明存在网络问题。那有可能存在一台防火墙，或是有访问控制列表（an access control list）特意阻止了连接，或是设备的 Telnet 没有打开。

需要事先设置好 VTY 线路的认证方式，方能远程连接到这台交换机或路由器上。如你不能连接上某台设备，你可以输入 `Ctrl+Shift+6` 然后输入 `x` 来退出。要退出一个活动的 Telnet 会话，你可以输入 `exit` 或者 `disconnect`。

用 `debug telnet` 命令来调试 Telnet。

## 互联网控制消息协议，Internet Control Message Protocol, ICMP

ICMP 是一个在某网络上用 IP 数据包（或数据报）来报告问题或故障的协议。ICMP 是那些任何要在其网络上采用 IP 技术的企业所需要的。在一个 IP 数据包发生问题时，此数据包将被销毁，同时生成一条 ICMP 消息并发送给发出该数据包的主机。

如同 RFC 792 所定义的那样，ICMP 在 IP 数据包内部投递消息。ICMP 最流行的使用是发出一个 `ping` 数据包来测试远端主机的连通性。在一台网络设备上运行 `ping` 命令时，便生成一个请求回应的数据包（a echo request packet），发往目的设备。目的设备收到该请求回应后，生成一条回应应答。

因为这些 ping 包有一个生存时间的字段（a Time to Live, TTL），它们提供了一个很好的网络延迟数据。下面的 ping 输出来自一台桌面 PC：

```
C:\ping cisco.com

Pinging cisco.com [198.133.219.25] with 32 bytes of data:

Reply from 198.133.219.25: bytes=32 time=460ms TTL=237
Reply from 198.133.219.25: bytes=32 time=160ms TTL=237
Reply from 198.133.219.25: bytes=32 time=160ms TTL=237
Reply from 198.133.219.25: bytes=32 time=160ms TTL=237

Ping statistics for 192.133.219.25:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 160ms, Maximum = 460ms, Average = 240ms
```

在上述输出中，ping 数据包是 32 字节长，时间字段报告响应耗时的毫秒数，TTL 是存活时间字段（数据包在多少毫秒后过期）。

思科路由器的 ping 命令有着复杂的参数，提供了更细的粒度，你可以指定 ping 发出的源地址，发出多少次 ping，ping 数据包的大小，以及其它参数。此特性在测试中是很有用的，在后面的实验部分用到很多次，如下面的输出所示：

```
Router#ping <- press Enter here
Protocol [ip]:
Target IP address: 172.16.1.5
Repeat count [5]:
Datagram size [100]: 1200
Timeout in Seconds [2]:
Extended commands [n]: yes
Source address: <- you can specify a source address or interface here
Type of service [0]:
Set DF bit in IP header? [no]: yes
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose [none]:
Type escape sequence to abort.
Sending 5, 1000-byte, ICMP Echos to 131.108.2.27, timeout is 2 seconds:
U U U U U
Success rate is 0% percent, round-trip min/avg/max = 4/6/12 ms
```

下面是其它几个表示回应 ping 数据包的符号所表示的意义：

- ! -- 每个回应有一个感叹号
- . -- 一次超时一个句点
- U -- 目的主机不可达
- N -- 网络不可达消息
- P -- 协议不可达消息
- Q -- 源抑制消息
- M -- 无法分片
- ? -- 未知数据包类型

通过 `Ctrl+Shift+6` 并输入 `x` 来终止一个 ping 会话。

RFC 1700 中定义了 ICMP 数据包。CCNA 大纲不包括所有代码和名称的内容。

在故障排除时，许多初级网络工程师会误用 ping 工具。ping 失败可以说明网络有问题，也可能是由于 ICMP 流量在网络上被阻止了。应为 ping 常会成为一种网络攻击的方法，ICMP 通常会被阻止。

## 追踪路由，Traceroute

Traceroute 可以用来测试网络的连通性，是一个广泛使用的工具，它又被用来做测量和管理。Traceroute 通过发出一些带有小的 TTL 字段 UDP 数据包，然后等待 ICMP 超时回应，以此来跟随目的 IP 的数据包。在 Traceroute 数据包的进行过程中，记录就会一跳接一跳地显示出来。每跳会测试 3 次。一个星号 (\*) 表明该跳超出了时间限制。

思科路由器的命令是 traceroute，Windows 计算机是 tracert。如下所示：

```
C:\Documents and Settings\pc>tracert hello.com
Tracing route to hello.com [63.146.123.17]
over a maximum of 30 hops:
 1  81 ms 70 ms 80 ms imsnet-cl10-hg2-berks.ba.net [213.140.212.45]
 2  70 ms 80 ms 70 ms 192.168.254.61
 3  70 ms 70 ms 80 ms 172.16.93.29
 4  60 ms 81 ms 70 ms 213.120.62.177
 5  70 ms 70 ms 80 ms core1-pos4-2.berks.ukore.ba.net [65.6.197.133]
 6  70 ms 80 ms 80 ms core1-pos13-0.ealng.core.ba.net [65.6.196.245]
 7  70 ms 70 ms 80 ms transit2-pos3-0.eang.ore.ba.net [194.72.17.82]
 8  70 ms 80 ms 70 ms t2c2-p8-0.uk-eal.eu.ba.net [165.49.168.33]
 9  151 ms 150 ms 150 ms t2c2-p5-0.us-ash.ba.net [165.49.164.22]
10  151 ms 150 ms 150 ms dcp-brdr-01.inet.qwest.net [205.171.1.37]
11  140 ms 140 ms 150 ms 205.171.251.25
12  150 ms 160 ms 150 ms dca-core-02.inet.qwest.net [205.171.8.221]
13  190 ms 191 ms 190 ms atl-core-02.inet.qwest.net [205.171.8.153]
14  191 ms 180 ms 200 ms atl-core-01.inet.net [205.171.21.149]
15  220 ms 230 ms 231 ms iah-core-03.inet.net [205.171.8.145]
16  210 ms 211 ms 210 ms iah-core-02.inet.net [205.171.31.41]
17  261 ms 250 ms 261 ms bur-core-01.inet.net [205.171.205.25]
18  230 ms 231 ms 230 ms bur-core-02.inet.net [205.171.13.2]
19  211 ms 220 ms 220 ms buc-cntr-01.inet.net [205.171.13.158]
20  220 ms 221 ms 220 ms msfc-24.buc.qwest.net [66.77.125.66]
21  221 ms 230 ms 220 ms www.hello.com [63.146.123.17]
Trace complete.
```

Traceroute 的输出字段有如下定义：

- ... -- 超时
- U -- 端口不可达消息
- H -- 主机不可达消息
- P -- 协议不可达消息
- N -- 网络不可达消息
- ? -- 未知包类型
- Q -- 收到源抑制 (source quench received)

在你想要对网络连通性进行故障排除时，Traceroute 是一个非常有用的命令。尽管有超出 CCNA 大纲，下面还是对此有更多的说明。

Traceroute 以逐步增加 UDP 数据包的 TTL 字段数值方式工作（仅在思科和 Linux 是这样的；微软 Windows 的 tracert 命令使用 ICMP 请求回应数据报，而不是 UDP 数据报来探测），这些 UDP 数据包将某台主机作为其目的地，并记录下收到的二者之间的那些路由器的回应。

每个数据包都有一个 TTL 值，当数据包到达一台路由器时，其 TTL 减 1。第一个数据包的 TTL 值为 1，当该数据包到第一台路由器时，TTL 降为 0，此时该路由器将发出一条错误消息（TTL 超时）。此时发出第二个数据包，TTL 设置为 2。当该数据包到达第二台路由器时，就会发出第一台路由器那的错误消息。这个过程持续下去，直到到达目的主机。

除了最后一跳外，所有的跳数都将返回一条“TTL 超时”的消息，最后一跳发回的消息将是“目的不可达/端口不可达”，表明它不能处理收到的流量（UDP Traceroute 数据包会将地址设置为一个不存在的端口号，终端主机一般不会去理会）。

## 地址解析协议，Address Resolution Protocol, ARP

有两种寻址方式来鉴别网络主机 -- IP （或三层）地址以及本地（或数据链路层）地址。数据链路层地址又叫做 MAC 地址。RFC 826 中定义的地址解析，是指 IOS 从 网络层（或 IP）地址得到数据链路层地址的过程。

ARP 将一个已知的 IP 地址解析为 MAC 地址。当主机需要在其网络上传输数据时，它需要知道另一主机的 MAC 地址。主机会检查它的 ARP 缓存，如果没有需要的 MAC 地址，你就发出一条 ARP 广播消息来找到该主机，如图 1.18 所示。

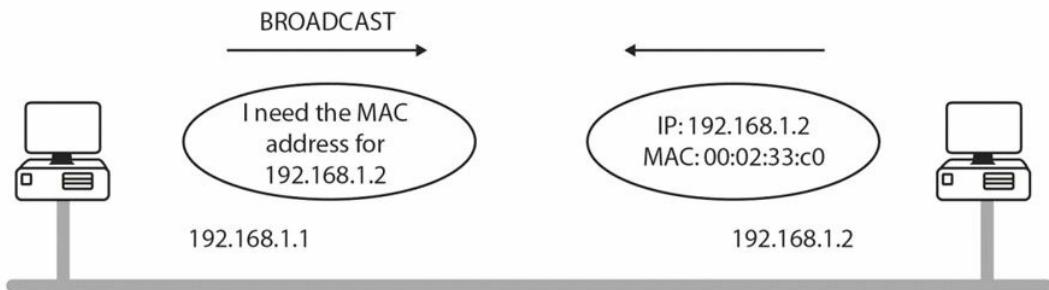


图 1.18 -- 一台主机为找到另一主机 MAC 地址而发出的广播

你可以用 `debug arp` 命令来调试 ARP。

为在网络上通信，一条 ARP 条目是需要的。你会看到，在没有需要的 ARP 条目时，就会产生一条广播。理解到路由器和交换机上的 ARP 表在一段时间后（默认 4 小时）就会刷新，是重要的，这是为了节约资源以及防止过时条目的留存。

在下面的路由器中，只有一条它自己的快以太网接口 ARP 条目，知道它邻居对其进行了 ping 操作后，因此，头 5 个 ping 数据包 (ICMP) 将会失败，就像下面的句点后有 4 个感叹号：

```

Router#show arp
Protocol      Address      Age (min)    Hardware Addr    Type      Interface
Internet      192.168.1.1 -          0002.4A4C.6801  ARPA     FastEthernet0/0
-
Router#ping 192.168.1.2
Type escape sequence to abort.
Hardware Addr Type Interface
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.!!!! - first packet fails due to ARP request
Success rate is 80 percent(4/5),round-trip min/avg/max = 31/31/31 ms
Router#show arp
Protocol      Address      Age (min)    Hardware Addr    Type      Interface
Internet      192.168.1.1          0002.4A4C.6801  ARPA     FastEthernet0/0
Internet      192.168.1.2 0        0001.97BC.1601  ARPA     FastEthernet0/0
Router#

```

## 代理 ARP

代理 ARP（见图 1.19）是在 RFC 1027 中定义的。代理 ARP 令到位于一个以太网络上的主机，在无需知道路由的情况下，能够与其它子网或网络的主机进行通信。

如有一条ARP广播到达某台路由器，路由器不会转发该ARP广播（在默认下）。路由器不转发广播，但如果它知道怎样去找到该主机（比如它们有一条到该主机的路由）的话，它们将会把自己的 MAC 地址发给广播主机。这个过程就叫做代理 ARP，此技术令到像是直接到达远端主那样发送数据。路由器将MAC替换后，将数据包转发给恰当的下一跳。

`ip proxy-arp` 命令在思科路由器上是默认开启的。

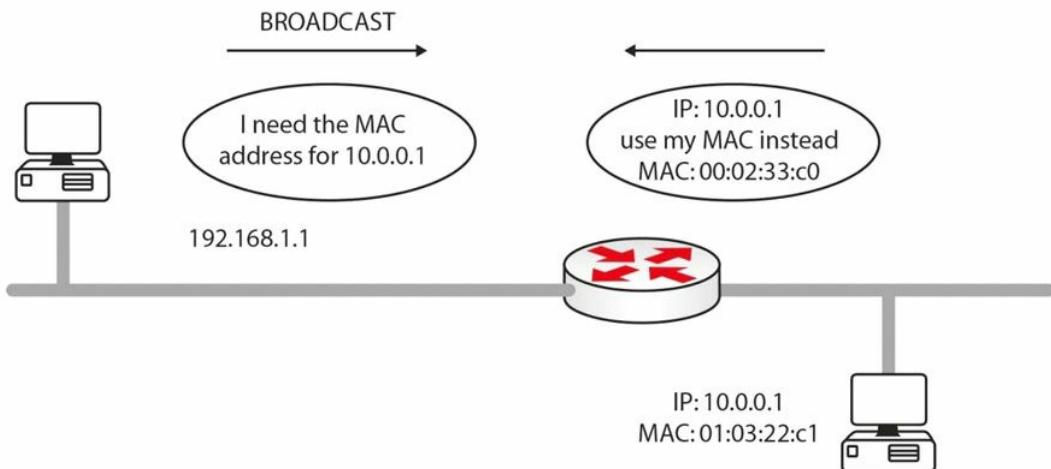


图 1.19 -- 路由器采用代理 ARP 技术以允许主机能够连接

对上述知识点进行拓展，考试要求理解数据包来回过程中寻址的改变。在数据包在网络上来回穿越的时候，两台终端设备都需要有某种方法来进行通信，其间的那些设备也要能交换数据包的下一跳地址才行。代理 ARP 当然给出了解决办法。数据包的 IP 地址始终保持不变，但为了让数据包能够传到下一跳，帧的 MAC 地址在设备之间发生了改变。

在下面的图 1.20 中，数据帧将离开 HOST A，它的 IP 地址是 192.168.1.1，目的 IP 地址是 172.16.1.2，源 MAC 地址为 AAAA:AAAA:AAAA，目的 MAC 地址是 AAAA:AAAA:BBBB。路由器 R1 将保留 IP 地址，而将源地址修改为 AAAA:AAAA:CCCC。而在数据包离开路由器 R2 前往 HOST B 之前，IP 地址仍然不会改变，源地址将是 AAAA:AAAA:DDDD，同时目的地址为 AAAA:AAAA:EEEE。

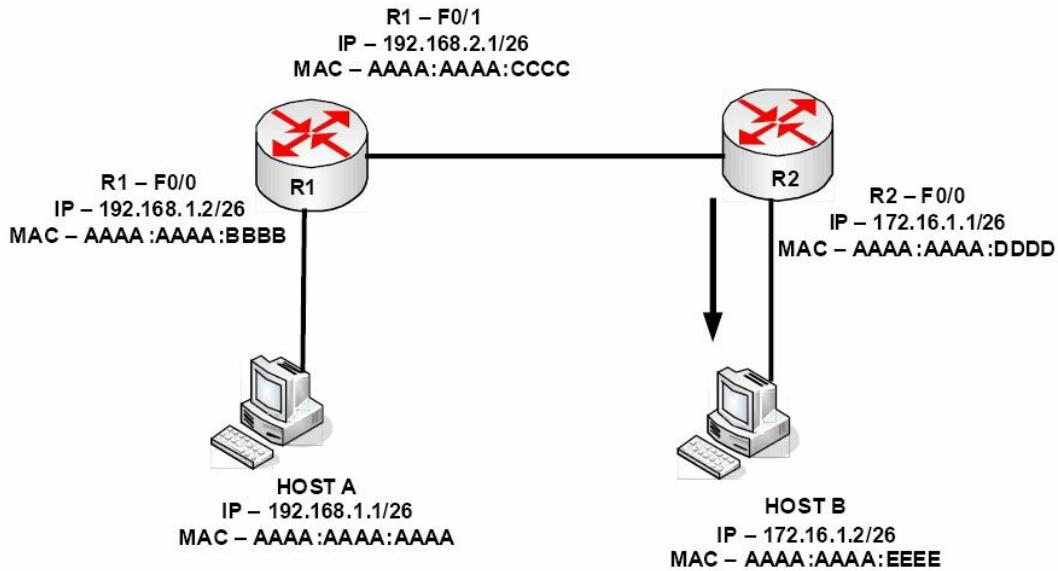


图 1.20 -- 数据包在设备间往复时，MAC 地址的改变

## 反向地址解析协议，Reverse Address Resolution Protocol, RARP

RARP 将一个已知的 MAC 地址映射到一个 IP 地址。像是无盘工作站（又叫做瘦客户机）那样的主机，在它们启动时知道自己的 MAC 地址。它们就会使用 RARP 来网络上的服务器那里发现自己的 IP 地址。

## 无故 ARP，Gratuitous Address Resolution Protocol, GARP

GARP 是一类特殊的 ARP 数据包。普通主机通常会在链路建立起来时或网络接口开启时发出一个 GARP 请求。这里的“无故（Gratuitous）”是指一种无需 ARP 的 RFC 通过，但仍可在某些场合使用的请求/回应。一次无故 ARP 请求就是一个将其源 MAC 地址、源 IP 以及目的 IP 地址都设置为发出该数据包的机器的 IP 地址的 ARP 数据包。目的 MAC 地址为广播地址 FFFF:FFFF:FFFF。通常，不会回应数据包产生。

而一个 GARP 回应是一个没有请求的回应（如你看到一个 GARP 回应，那就意味着网络上的另一计算机和你的计算机用了相同的 IP 地址）。当 FHRP 协议（比如 HSRP）中出现状态改变时，会用到 GARP，为达到更新第二层 CAM 表的目的。IPv6 章节也会讨论到 GARP。

## 简单网络管理协议，Simple Network Management Protocol, SNMP

SNMP 为众多网络管理业务所使用。一套 SNMP 管理系统中，网络设备将名为陷阱（traps）的消息发送给管理工作站。这会想网络管理员报告任何的网络故障（比如接口故障），或是服务器上 CPU 使用等情况。

使用 `debug snmp` 命令对 SNMP 流量进行调试。SNMP 使用 UDP 端口 161 及 162。

## 安全版超文本传输协议，Hyper Text Transfer Protocol Secure, HTTPS

TLS，以及旧版的 SSL，被用到加固互联网上的通信，是通过采用各种加密方法实的。在电子邮件以及 VoIP，以及访问那些以 `http://` 开头的站点时，你会发现这些加密方法。带有 TLS/SSL 的 HTTP（HTTPS）使用 443 端口。

### IP 配置命令，IP Configuration Command

这不是一件属于思科的工具，但它你的故障排除工具包中的一部分。命令 `ipconfig` 是在Windows命令提示符下运行的命令，你可以用到数个命令开关，但可能用得最多的就是 `ipconfig /all` 命令，如下面的屏幕截图那样。

图 1.21 -- ipconfig /all 命令的输出

## 线缆和介质，Cables and Media

作为网络工程师的你，布线及线缆相关的事情将成为日常工作的一部分。你需要知道哪些线应该插入哪些设备，诸多工业限制，以及怎样将设备配置起来使用这些线缆。

## 局域网的线缆， LAN Cables

以太网线

因为局域网上有着为数众多的线缆和连接头，同时又存在因设备迁移及测试带来的线缆频繁插拔，大多数线缆有关的网络问题都是发生在局域网上，而不会是广域网。

以太网线用于将工作站连接至交换机，交换机之间以及交换机与路由器的连接。其规格和速率在近年来有多次修订和提升，这就是说你可以很快用到将今天的标准速率甩得老远的速率，到你的桌面的高速链路也会很快到来。目前的标准以太网线仍然使用 8 条、4 对缠绕的电线，以消除电干扰（electromagnetic interference, EMI），也就是串扰（crosstalk）这种会蔓延到相邻线路上的信号。

ANSI/TIA/EIA-568-A 标准中对以太网线的类别进行了定义，有 3 类、5 类、5e 类以及 6 类共 4 个类别。每个类别都有其相应标准、规格以及在限定距离范围内能够达到的数据吞吐速率。3 类以太网线布线可以最高 10Mbps 速率传输数据。5 类布线主要用于快速以太网络，100BASE-TX 以及 1000BASE-T 都是 5 类网线。5e 类布线使用了增强的 100-MHz (100-Mhz-enhanced) 双绞线来组建千兆以太网 (GigabitEthernet)，

就是 1000Base-T。最后的 6 类布线，每对电线以 250MHz 运作，以提供出改进了的 1000Base-T 的性能。（“1000”表示数据传输速度有多少 Mbps，“Base”代表基带传输--baseband，而“T”则是指双绞线 --twisted pair）。表 1.6 给出了你所熟悉的一些常见的以太网标准。

表 1.6 常见以太网标准

速率	名称	IEEE 名称	IEEE 标准	线缆类型/长度
10Mbps	以太网， Ethernet	10BASE-T	802.3	铜线/100米
100Mbps	快速以太网， FastEthernet	100BASE-T	802.3u	铜线/100米， Copper/100m
1000Mbps	千兆以太网， GigabitEthernet	1000BASE-LX	802.3z	光纤/5000米， Fibre/5000m
1000Mbps	千兆以太网	1000BASE-T	802.3ab	铜线/100米， Copper/100m
10Gbps	万兆以太网， TenGigabitEthernet	10GBASE-T	802.3an	铜线/100米， Copper/100m

思科喜欢将线缆规格有关的问题偷偷摸摸地放到考试中去，所以务必要记住这个表格。

### 双工, Duplex

在以太网投入使用的早期阶段，同一时间数据只能在一个方向上传输。这是因为那个时候所使用线缆的限制造成的。发送设备在线缆上发送数据前必须等待直到线缆可用，否则将会发生冲突（collision）。因为后来有了不同组别的电线负责发送和接收信号，这就不成问题了。

半双工（half duplex）是指数据只能在一个方向上传输，全双工则是数据能够在两个方向上同时传输（见图 1.22）。这是通过使用以太网线内部的外加电线实现的。现在的所有设备都以全双工方式运行，除非是设置为半双工。

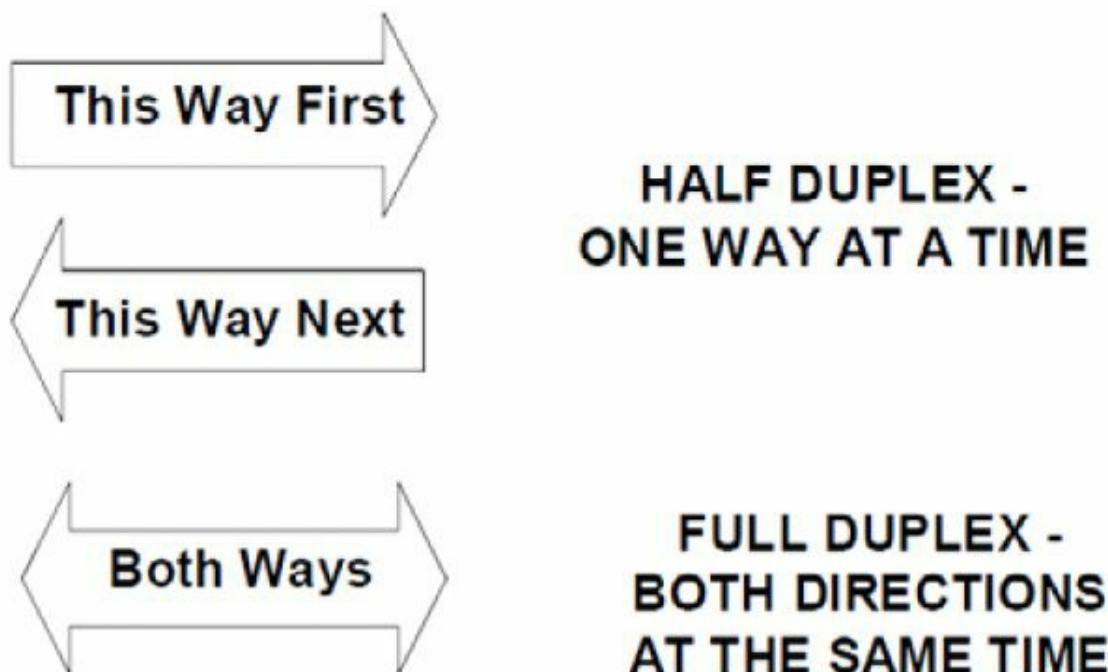


图 1.22 -- 双工拓扑, Duplex Topology

考试中仍然要求你能理解并排除全双工方面的故障；本书后面的第一层和第二层故障排除部分将详细介绍。使用 `show interface X` 命令就可以轻易地检查接口的双工设置。

```
Switch#show interface FastEthernet0/1
FastEthernet0/1 is down, line protocol is down (disabled)
    Hardware is Lance, address is 0030.a388.8401 (bia 0030.a388.8401)
    BW 100000 Kbit, DLY 1000 usec,
        reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA, Loopback not set
    Keepalive set (10 sec)
    Half-duplex, 100Mb/s
```

如果此接口与某台全双工设备连接起来，你将立即看到有错误发生，同时链路流量将极为慢速。你可以在一台真实交换机上执行 `show interfaces status` 命令，但考试中这条命令可能不会工作，因为像 Packet Tracer 这样的路由器模拟软件仅能运行有限的一些命令。在下面的输出中，你会发现接口 `FastEthernet 1/0/2` 存在一些问题。

```
Switch#show interfaces status
Port     Name      Status     Vlan      Duplex   Speed    Type
Fa1/0/1  notconnect  1         auto     auto    10/100BaseTX
Fa1/0/2  notconnect  1         half    10     10/100BaseTX
Fa1/0/3  notconnect  1         auto     auto    10/100BaseTX
Fa1/0/4  notconnect  1         auto     auto    10/100BaseTX
Fa1/0/5  notconnect  1         auto     auto    10/100BaseTX
```

当然要修复这个问题也是十分容易的，像下面这样：

```
Switch(config)#int f1/0/2
Switch(config-if)#duplex ?
    auto Enable AUTO duplex configuration
    full Force full duplex operation
    half Force half-duplex operation
Switch(config-if)#duplex full
```

请务必要在真实思科设备上，或 GNS3 中，或最新版的 Packet Tracer 中去试试这些命令，来记住它们！

### 速率，speed

你可将路由器或交换机的速率保留成自动协商（auto-negotiate），或者硬性设置为 `10Mbps`、`100Mbps` 或者 `1000Mbps`。

像下面这样就可以手动设置速率：

```
Router#config t
Router(config)#interface GigabitEthernet 0/0
Router(config-if)#speed ?
    10      Force 10 Mbps operation
    100     Force 100 Mbps operation
    1000    Force 1000 Mbps operation
    auto    Enable AUTO speed configuration
```

下面的命令是要查看以太网接口的设置：

```
Router#show interface FastEthernet0
FastEthernet0 is up, line protocol is up
  Hardware is DEC21140AD, address is 00e0.1e3e.c179 (bia 00e0.1e3e.c179)
  Internet address is 1.17.30.4/16
  MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec, rely 255/255, load 1/255
  Encapsulation ARPA, Loopback not set, keepalive set (10 sec)
  Half-duplex, 10Mb/s, 100BaseTX/FX
```

EIA/TIA 的以太网线规格要求网线的末端务必是 RJ45 公头（见图 1.23；图 1.24 展示了其母头），你可以将网插入路由器/交换机/PC 的端口上。

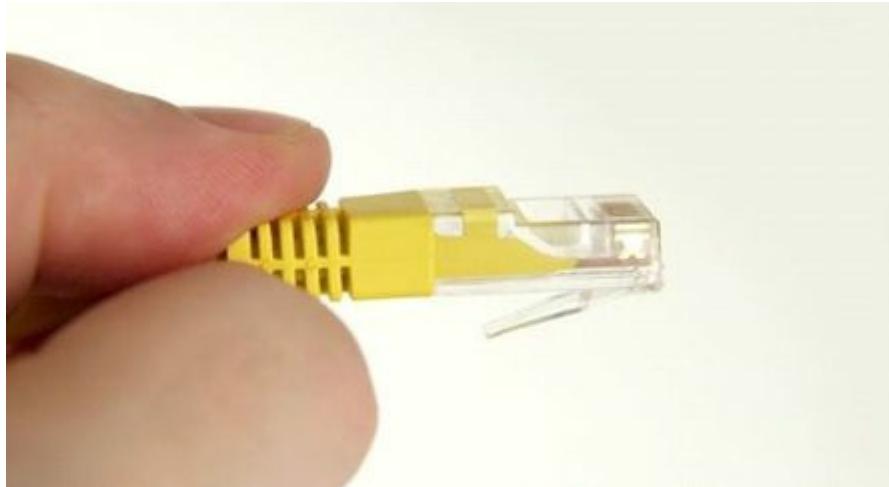


图 1.23 -- RJ45 公头



图 1.24 -- RJ45 母头

### 直通线

以太网线有 8 根，每根都与水晶头上一个针脚连接起来。而每条线与这些针脚位置的接法，确定做出来的网线的用途。如果网线两端的接法完全相同，则做出来的网线就叫直通线。这种线用于将**终端设备连接交换机，或者连接交换机和路由器**。将网线的两端放在一起对比一看，就知道它们是否是一样的接法。如图 1.25 和 1.26 所示。



图 1.25 -- 对比网线两端

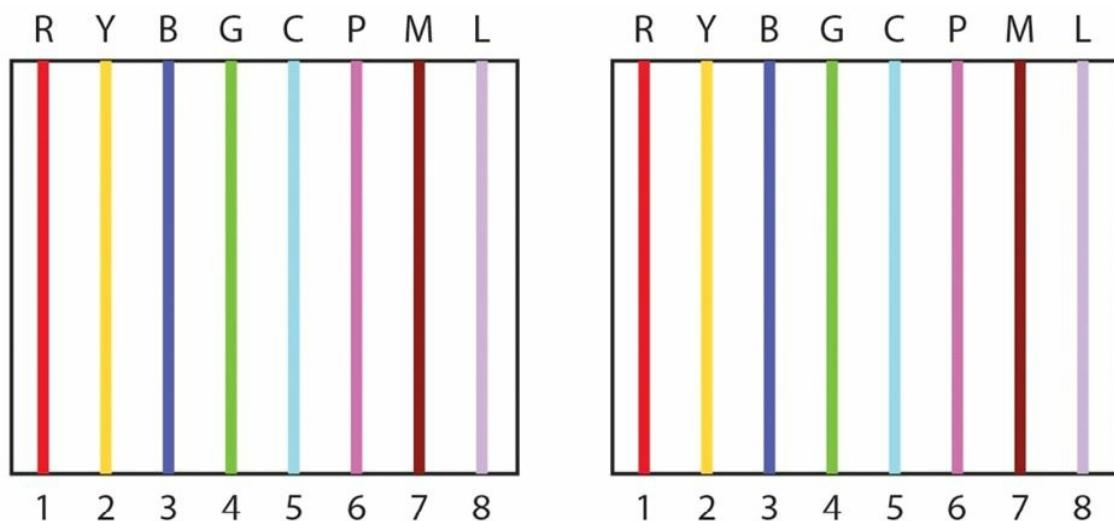


图 1.26 -- 网线两端是一样的

### 交叉线, crossover cables

通过将网线的两对电线的位置交换一下, 就可以用来在无需交换机、集线器的情况下, 连接两台 PC 或是两台交换机(较新的网卡的 Auto-MDIX 功能能够自动侦测连接是否需要交叉连接, 选择 MDI 或是 MDIX 配置来与链路的另一端恰当匹配)。一端的针脚 1 需要连接到另一端的针脚 3, 针脚 2 要连接到针脚 6 (见图 1.27)。

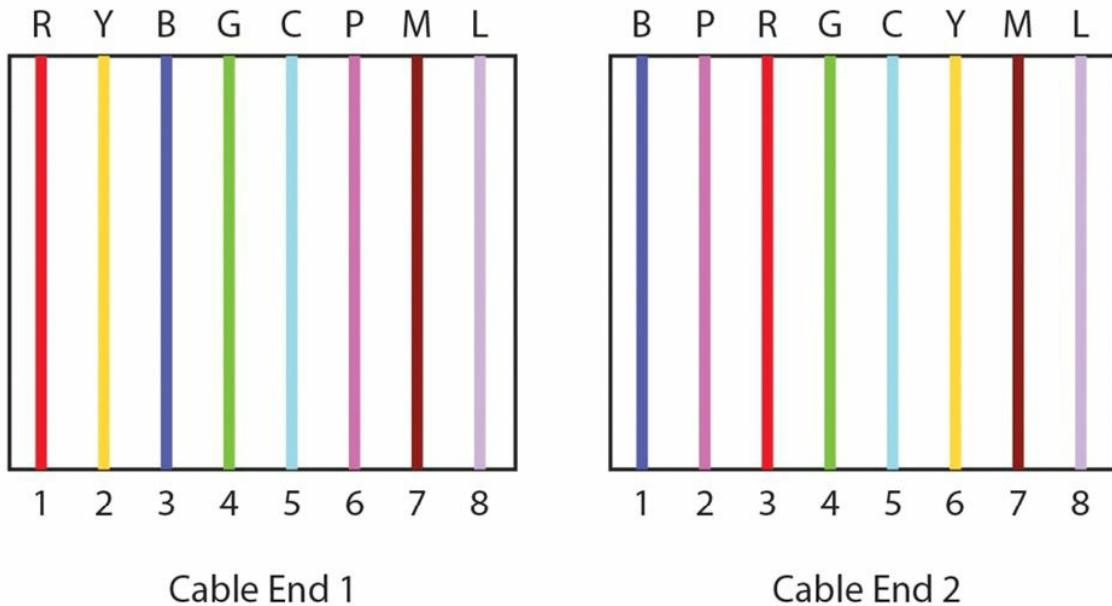


图 1.27 -- 针脚 1 到针脚 3, 针脚 2 到针脚 6

#### 翻转线/控制台线, Rollover/Console Cables

所有的思科路由器和交换机都有用于初始设置以及灾难恢复和访问的几个物理端口。这些端口叫做控制台端口，作为思科工程师，你肯定会用到这些端口。你需要一种叫做翻转线或者控制台线的特殊类型线缆来连接这个端口（见图 1.28）。有时又称其为扁线（a flat cable），因为它与一般圆形的网线不同，它是扁的。



图 1.28 -- 一条典型的翻转线

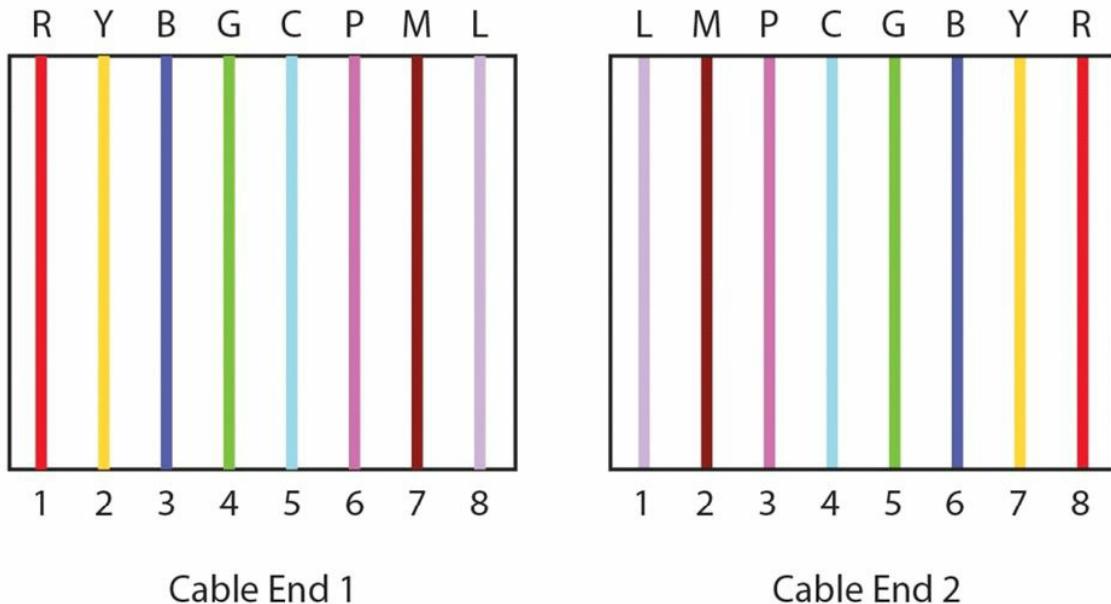


图 1.29 -- 所有针脚都交换了位置

翻转线通常一端有一个 RJ45 接头，另一端是一个 9 针 D 形连接器，设计用于连接 PC 或笔记本电脑的 COM 端口。问题是现今的设备通常有不再有 COM 端口了，因为 COM 端口用得很少很少。不过你可以从电子商店或网上买到 DB9-to-USB 转换器（如图 1.30）。它们带有驱动程序，允许你通过如 PuTTY 或 HyperTerminal 等终端程序，连接到 PC 的逻辑 COM 端口( a logical COM port)。

思科已经开始在他们的设备上放 mini-USB 端口，作为 RJ45 端口的补充，可以通过 USB A 型 (Type A) 至 5 针 B 型 (5-pin Type B) 插头线，获得对控制台的访问。如同时插入两种控制台线，那么 mini-USB 优先。图 1.31 及 1.32 是不同的连接类型。

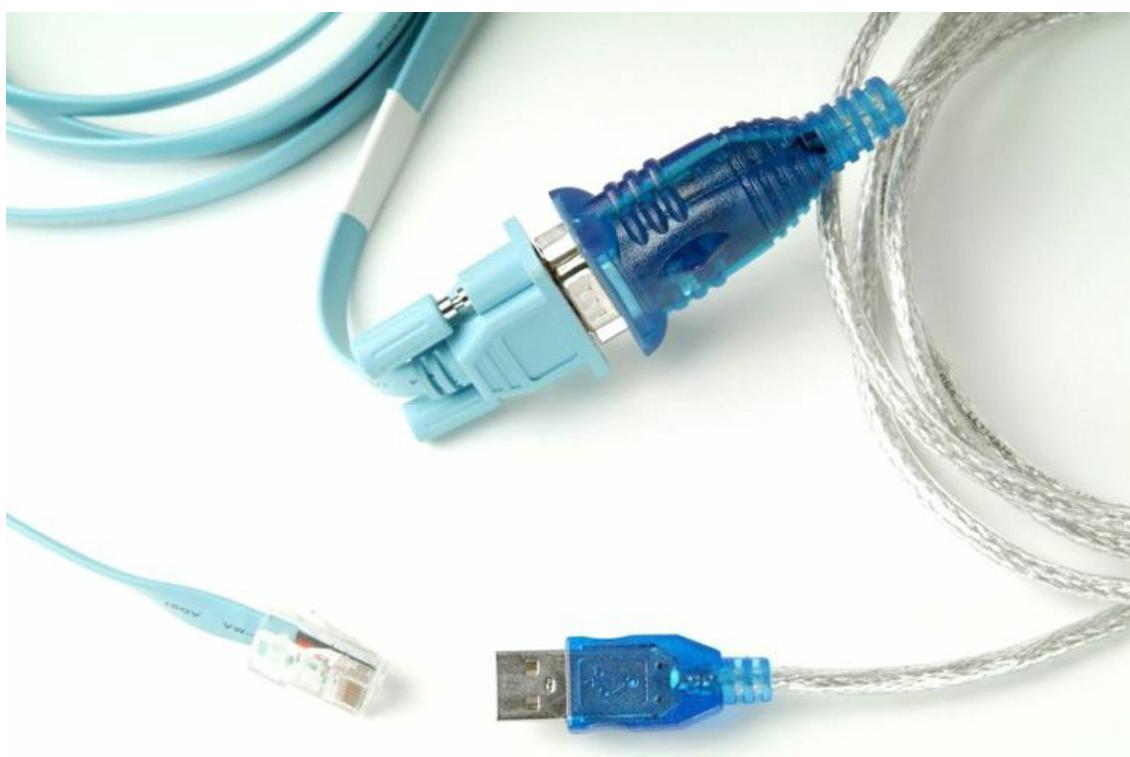


图 1.30 -- 一条 COM 到 USB 的转换线

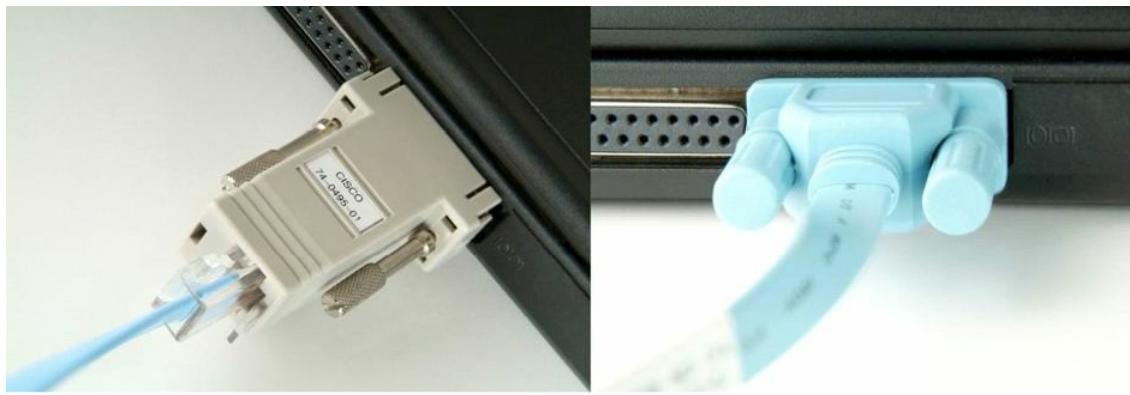


图 1.31 -- 将串行线连接到笔记本电脑的串行端口

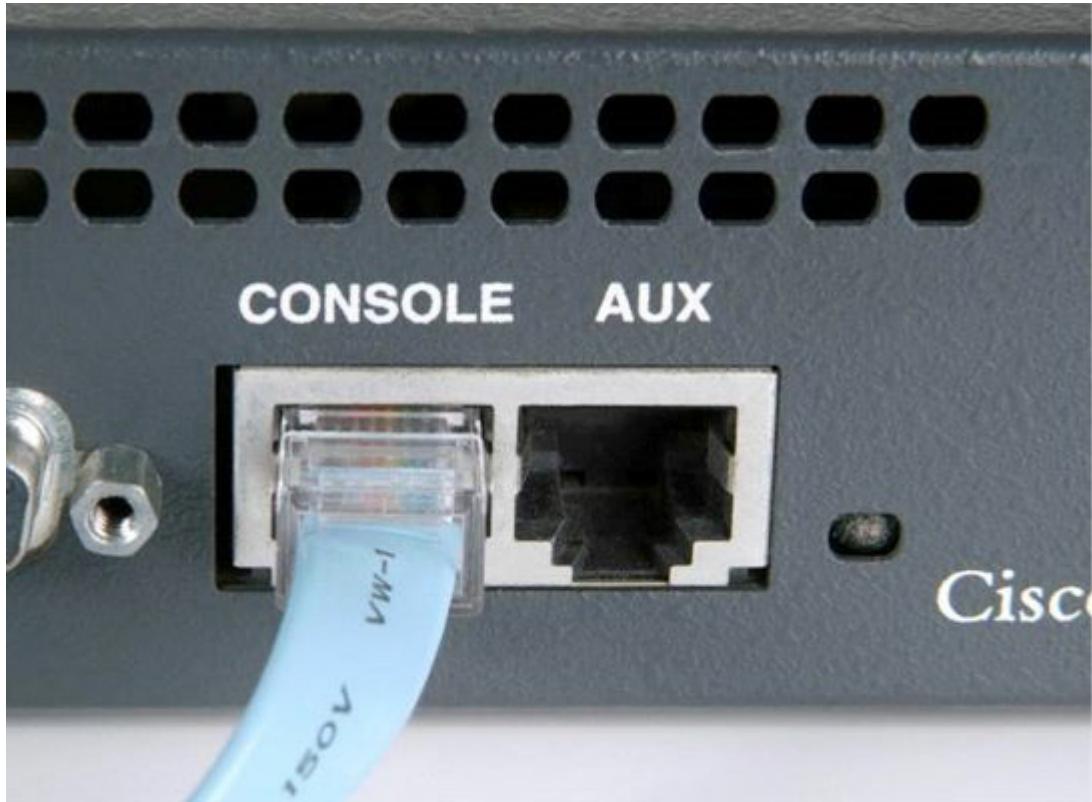


图 1.32 -- 将串行线连接到路由器或交换机的控制台端口

## 广域网线缆, WAN Cables

依路由器接口及连接类型的不同，广域网连接所使用到的 **串行通讯线(serial cables)** 在形状、大小以及规格上有好几种。比如 ISDN 就会使用到与帧中继（Frame Relay）或 ADSL 所不同的一些线缆。

尤其是在家庭网络的实践下，你会用到的一种常见的叫做 DB60（见图 1.33）的 WAN 线缆。此种线缆有一个数据终端设备（a data terminal equipment, DTE）端，这端是要插入到客户设备上，另一端是数据通信设备端，它决定来自 ISP 处的连接速率。图 1.34 是一个 WIC-1T 插卡上的 DB60 串行接口。



图 1.33 -- 一条 DB60 线缆



图 1.34 -- 一块 WIC-1T 插卡上的 DB60 串行接口

还有一种思科经常推介的、用于广域网接口卡（WAN Interface Cards, WICs）的，叫做小巧串行线（smart serial cable）的类型。如图 1.35 所示。



图 1.35 -- 小巧串行线, Smart Serial Cable

在使用这种类型线缆时, 你当然需要恰当的接口卡, 此种接口卡如图 1.36 所示。



图 1.36 -- 使用小巧串行线的 WIC-2T 卡

该 WIC 卡使用路由器的一个插槽, 能提供两条连接, 而 标准的 WIC-1T 卡仅有一条连接。每条连接都可以使用不同的封装类型, 比如一条使用 PPP, 另一条使用帧中继。

关于 DCE 和 DTE 线缆的最重要之处在于, 你需要在 DCE 端指定时钟(a clock rate)。通常情况下, 你的 ISP 会干这件事, 因为他们持有 DCE 端, 但在家中或是用真实机架 (live rack) 做实验时, 是你持有 DCE 端, 在一台路由器上你是客户, 另一台路由器上你又是 ISP 了。需要输入的命令是 `clock rate 64000` (或者任何可用的速率, 单位是 bits per second)。`clock rate ?` 命令可以调出那些速率选项。

在输入以下命令前, 你务必先要搞清楚它们。首先, 要确认哪台路由器接上了 DCE 线, 你需要命令 `show controllers` 之后接上接口编号。在真实考试的故障排除中 (现实工作中也一样), 这是一个有用的命令。命令 `show ip interface brief` 让你掌握路由器上有哪些接口。

实际上，你可以简化思科 IOS 命令的输入，就像下面的输出那样。但考试中简化输入的命令可能不会运行，因为考试使用的是路由器模拟器（而不是真实的路由器）。

```

Router#sh ip int bri
Interface      IP-Address  OK? Method   Status          Protocol
FastEthernet0/0 unassigned  YES unset   administratively down    down
FastEthernet0/1 unassigned  YES unset   administratively down    down
Serial0/1/0     unassigned  YES unset   administratively down    down
Vlan1          unassigned  YES unset   administratively down    down

Router#show controllers s0/1/0
Interface Serial0/1/0
Hardware is PowerQUICC MPC860
DCE V.35, no clock
Router(config-if)#clock rate ?
Speed (bits per second)
  1200
  2400
  4800
  9600
  19200
  38400
  56000
  64000
...
[Truncated Output]

```

## 连接到一台路由器， Connecting to a Router

这是你头一次连接到一台路由器或交换机，看起来有些艰巨吧。前面的内容已经讲到了控制台连接了，所以在连上串行线后，你的 PC 或笔电就需要一个终端模拟程序了。有了这些，你就可以查看路由器的输出并敲入那些配置命令了。

超级终端（HyperTerminal）作为默认程序已经用了很多年了，在完成灾难备份时，你可能仍需要这个程序；但是你可以选择 PuTTY 这个广泛使用的程序。从 [www.putty.org](http://www.putty.org) 可以下载到它。老式的 PC 上的 COM 端口连接总是会用到标为 COM1 或 COM2 的逻辑端口。PuTTY 中有一个有关逻辑端口的设置，我们实际上叫这个是一条串行连接（a serial connection）。如图 1.37 所示。

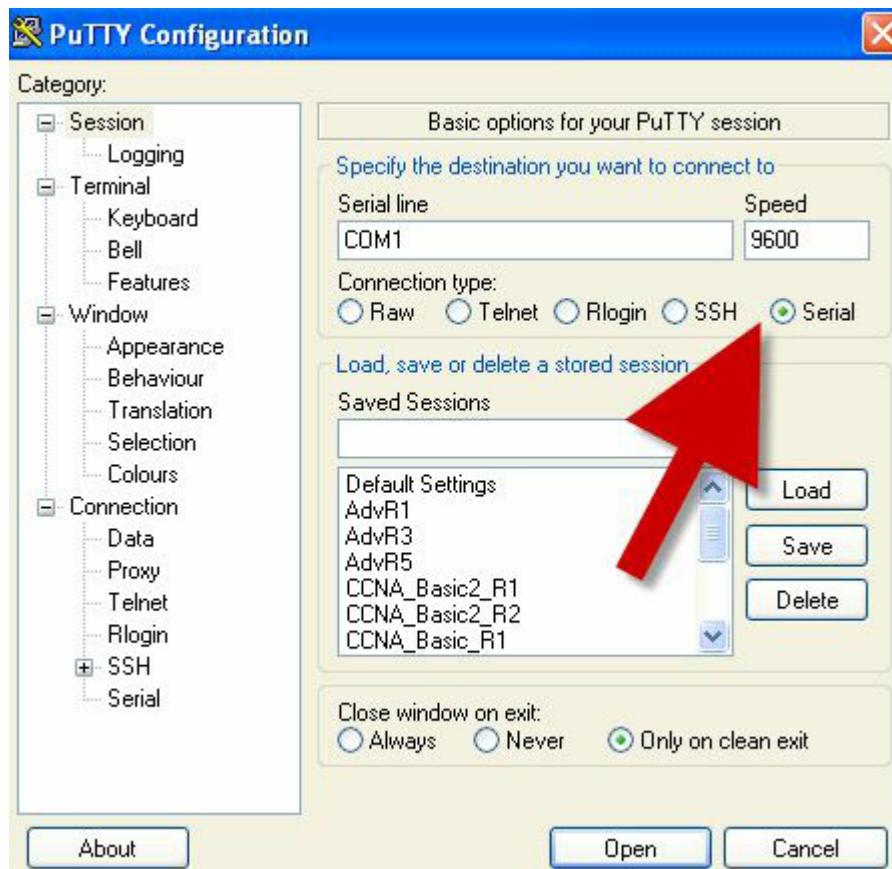


图 1.37 -- PuTTY 使用 COM 端口得到串行访问

如你使用的是 USB 到翻转线（USB-to-rollover）转换器，那么你会收到一张包含其驱动程序的安装光盘，在安装好驱动程序后，你将得到一个可以使用的 COM 端口。如你使用的是 Windows 系统，在设备管理器中你会发现这个端口。如图 1.38 所示。

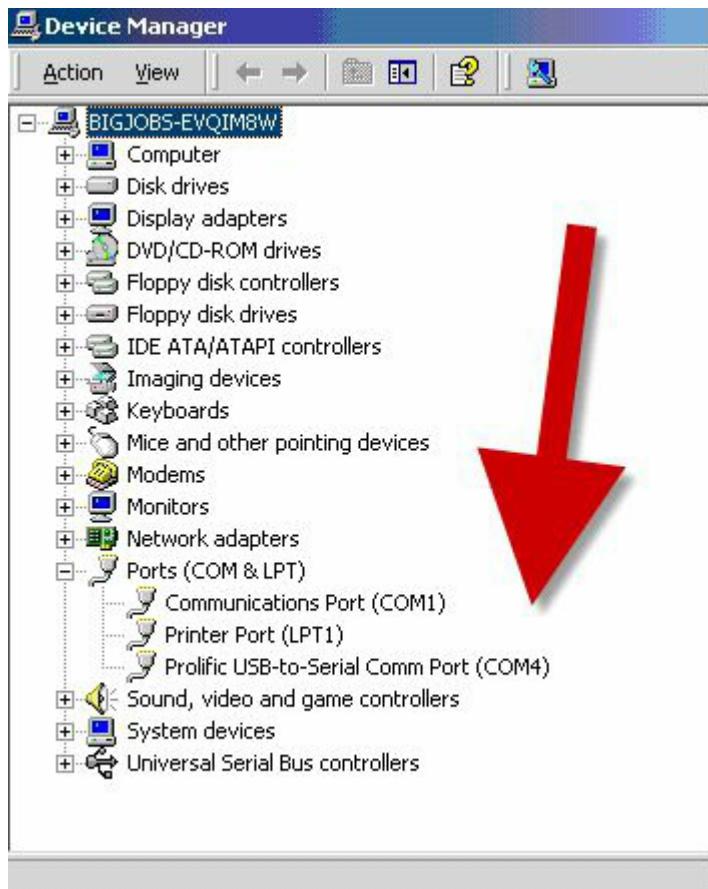


图 1.38 -- 驱动程序将 COM4 作为控制台连接的端口

在使用超级终端时，你还要选择一些连接参数，比如波特率等。你需要做如下选择，如图 1.39 所示。

- 每秒位数，Bits per second: 9600
- 数据位数，Data bits: 默认值 8
- 校验，Parity: 无/None
- 停止位，Stop bits: 默认值 1
- 流控，Flow control: 必须是 无/None

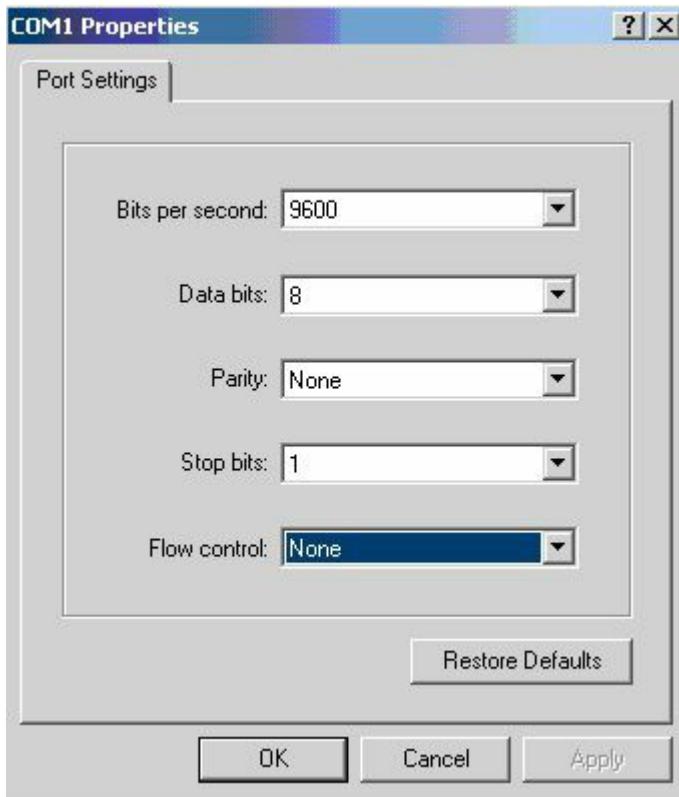


图 1.39 -- 超级终端连接参数设置

在开启路由器时，如你已经选择了正确的 COM 端口，并将翻转线插入到路由器的控制台端口，你将看到路由器的启动文字（见图 1.40）。如你不能看到任何文本，那么敲几下回车键并在此检查一下你的设置。

如路由器没有在它的 NVRAM 中找到启动配置文件（a startup configuration file）时，或者路由器的配置寄存器（the configuration register）被设置为 0x2142 而忽略启动配置文件时，路由器会询问你是否要进入**初始设置模式(Initial Configuration mode)**。请输入“n”或“no”，输入“yes”会进入配置模式（setup mode），你是不会想要进入到这个模式的。

```
Would you like to enter the initial configuration dialog?
[yes/no]:
% Please answer 'yes' or 'no'.
Would you like to enter the initial configuration dialog?
[yes/no]: no
Press RETURN to get started!
Router>
```

在另一个型号的路由器上，你会看到下面的输出：

```
Technical Support: www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
--- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]: no
Press RETURN to get started!
Router>
```

## 路由器的各种模式，Routers Modes

为通过 CCNA 考试，你需要理解在完成各种操作时，需要进入的不同路由器模式提示符。而不管你要执行何种功能，首先你要是在正确的模式下（有不同的提示符来区分）。新手在配置路由器的过程中遇到找不到正确命令来使用的问题时，他们犯的最大错误往往就在这里。请一定要确定你在正确的模式下！

### 用户模式，User Mode

在路由器启动后，第一个展现在你面前的叫用户模式（User Mode）或者用户执行模式（User Exec Mode）。用户模式下只有很小的一套命令可供使用，但在查找基本的路由器元素上是有用的。路由器默认名称是“Router”，后面你会看到该名称可以修改。

```
Router>
```

### 特权模式，Privileged Mode

在用户模式下输入 `enable` 命令，就带你进入了下一模式，叫做特权模式或特权执行模式（Privileged Exec mode）。输入 `disable` 命令退回到用户模式。而要退出整个会话，输入 `logout` 或者 `exit`。

```
Router>enable
Router#
Router#disable
Router>
```

在查看路由器的整个配置、路由器的运行统计数据，以致路由器插入了哪些模块时，特权模式是有用的。在此提示符下，你会输入 `show` 命令，和用于调试的 `debug` 命令。

### 全局配置模式，Global Configuration Mode

为了真正配置路由器，你需要进入全局配置模式。在特权运行模式下，输入 `configure terminal` 命令，或其简短版本 `config t` 来进入此模式。此外，仅输入 `config` 时，路由器会询问你要进入何种模式。`terminal` 是模式的（默认选项会被中括号括起来）。如你按下了回车键，就会接受中括号里的命令。

```
Router#config
Configuring from terminal, memory, or network[terminal]? - press Enter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#
```

### 接口配置模式，Interface Configuration Mode

接口模式下，可以输入路由器接口，如快速以太网、串行接口等，的命令。在一台全新的路由器上，默认所有接口都是关闭的，没有任何配置。

```
Router>enable
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial0
Router(config-if)#
```

`show ip interface brief` 命令可以查看到路由器有哪些接口。你的串行接口可能不是 `Serial0`。

### 线路配置模式，Line Configuration Mode

线路配置模式用来对控制台、Telnet 或者辅助端口（auxiliary ports）进行改变。你可以控制哪些人可以通过这些端口访问到路由器，以及在这些端口上部署口令或者“访问控制列表（access control lists）”这种安全特性。

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line console 0
Router(config-line)#

```

你还可以在此模式下设置波特率、执行级别（exec levels）等参数。

### 路由器配置模式，Router Configuration Mode

为了给路由器配置一种路由协议，以便它能够建立起网络图（build a picture of the network），你需要用到路由器配置模式。

```
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#router rip
Router(config-router)#

```

### 虚拟局域网配置模式，VLAN Configuration Mode

此种模式实际上是属于交换机的，但既然我们在此讨论不同模式，所以也有必要提一下。本书的交换机实验中，你会用到很多这种模式。

```
Switch#conf t
Enter configuration commands, one per line.
Switch(config)#vlan 10
Switch(config-vlan)#

```

在具备以太网交换机卡的路由器上，会使用虚拟局域网数据库配置模式（VLAN Database Configuration mode，该模式在交换机上已被废除），其与 VLAN 配置模式是相似的。

```
Router#vlan database
Router(vlan)#vlan 10
VLAN 10 added:
  Name: VLAN0010
Router(vlan)#exit
APPLY completed.
Exiting....
Router#

```

## 配置一台路由器，Configuring a Router

路由器是没有菜单的，你也不能用鼠标在不同模式之间切换，这些都是经由命令行界面(command line interface, CLI)完成的。有些上下文敏感（context-sensitive）的帮助信息以 [?] 关键字形式给出。在路由器提示符处输入问号，所有可用的命令都将显示出来。

```

Router#?
Exec commands:
access-enable      Create a temporary Access-List entry
access-profile     Apply user-profile to interface
access-template    Create a temporary Access-List entry
alps               ALPS exec commands
archive            manage archive files
bfe                For manual emergency modes setting
cd                 Change current directory
clear              Reset functions
clock              Manage the system clock
cns                CNS subsystem
configure          Enter configuration mode
connect            Open a terminal connection
copy               Copy from one file to another
debug              Debugging functions (see also 'undebbug')
delete             Delete a file
dir                List files on a directory
disable            Turn off privileged commands
disconnect         Disconnect an existing network
enable             Turn on privileged commands
erase              Erase a file
exit               Exit from the EXEC mode
help               Description of the interactive help system
-- More --

```

如果有多于屏幕能显示的信息，你将看到 `--More--` 栏。按空格键来查看下一页。按 `Ctrl+Z` 或者 `Q` 回到提示符。

此外，如你已经开始输入一个命令，却忘记了该命令的剩下部分，输入 `?` 系统就会给出一个可用的命令清单。`?` 在 CCNA 考试中是可用的，但如你用了问号，说明你就没有认真完成本书的那些实验:)

```

Router#cl?
clear clock

```

按 `Tab` 键有命令补全功能。

```

Router#copy ru
← press the Tab key here
Router#copy running-config

```

路由器有好几个可供选择的模式。这是为了避免对不打算修改的路由器配置部分造成不必要的改变而设置的。看一下提示符就知道你当前所处哪个模式。比如你打算对某个快速以太网接口做一些改变，你需要在接口配置模式下来完成。

首先，进入全局配置模式：

```

Router#config t
Router(config)#

```

接着，告诉路由器你要配置哪个接口：

```

Router(config)#interface FastEthernet0
Router(config-if)#exit
Router(config)#

```

如你不确定采用何种方式输入接口编号，就使用 [?] 关键字。无需担心你所看到的所有选项。大多数人都只会用到快速以太网、串行接口及环回接口（Loopback interfaces）。

```
Router(config)#interface ?
Async           Async interface
BRI            ISDN Basic Rate Interface
BVI            Bridge-Group Virtual Interface
CTunnel        CTunnel interface
Dialer         Dialer interface
FastEthernet   IEEE 802.3u
Group-Async    Async Group interface
Lex             Lex interface
Loopback       Loopback interface
Multilink      Multilink-group interface
Null           Null interface
Serial          Serial interface
Tunnel          Tunnel interface
Vif             PGM Multicast Host interface
Virtual-Template Virtual Template interface
Virtual-TokenRing Virtual TokenRing interface
range          interface range command

Router(config)#interface FastEthernet?
<0-0> FastEthernet interface number
Router(config)#interface FastEthernet0
```

最终，路由器进入到了接口配置模式。

```
Router(config-if)#
```

在这里，你可以为接口配置上 IP 地址，设置其带宽，部署一条访问控制清单，以及完成很多其它事项。你的路由器或交换机可能会与我（作者）的有不同的接口编号，所以请使用 `?` 或 `show ip interface brief` 命令去查看你的选项。

输入 `exit` 命令从某个配置模式中退出。这会将你带回到其第二高的级别(the next-highest level)。而要从任何的配置模式中退出，按下 `Ctrl+Z` 或输入 `end` 命令就可以了。

```
Router(config-if)#exit
Router(config)#
```

或是 `Ctrl+Z` 的办法。

```
Router(config-if)^Z
Router#
```

## 环回接口，Loopback Interfaces

CCNA 大纲通常不会涉及环回接口的知识点，但不管在工作中，还是在操作实验中，都是有用的。环回接口是你配置得来的虚拟或逻辑接口（a virtual or logical interface），而不是物理存在的（所以你不会在路由器的面板上见到环回接口）。你可以往这类接口上执行 ping 操作，而无需在实验用有设备连接到路由器的快速以太网接口上。

使用环回接口的一大好处在于随路由器的运行，它们总是保持开启的，因为它们是逻辑的，意味着它们绝不会宕下去（go down）。而又由于它们是虚拟的，所以你不可以将网线插到它们上面。

```

Router#config t
Router#(config)#interface Loopback0
Router#(config-if)#ip address 192.168.20.1 255.255.255.0
Router#(config-if)#^z ~ press Ctrl+Z
Router#
Router#show ip interface brief
Interface   IP-Address      OK?    Method Status Protocol
Loopback0   192.168.20.1    YES   manual   up       up

```

此命令的输出将显示出你的路由器的所有可用接口的信息。

**真实世界：**可以在接口配置模式下输入 shutdown 命令来关掉一个环回接口。

务必要给环回接口一个有效的 IP 地址。可以**用于那些路由协议 或者测试路由器是否允许某些流量通过**。本课程中会大量使用到环回接口。

### 编辑命令，Editting Commands

与其将已输入的整行命令全部删除，你可以对其进行编辑。下面这些键盘输入可以将光标移至该行命令的任意位置。

键盘输入，Keystroke	用途，Meaning
Ctrl+A	将光标移至命令行开头
Ctrl+E	将光标移至命令行末尾
Ctrl+B	将光标移往后移动一个字符
Ctrl+F	将光标移往前移动一个字符
Esc+B	将光标往前移动一个词
Esc+F	将光标往后移动一个词
Ctrl+P 或向上箭头	翻出上一条命令
Ctrl+N 或向下箭头	翻出下一条命令
Ctrl+U	删除这条命令
Ctrl+W	删除一个词
Tab	补全命令
show history	默认情况下，显示前 10 条命令
退格按键， Backspace	删除一个字符

考试中出一道有关这些编辑命令的题目是很常见的。

### 配置一个路由器接口，Configuring a Router Interface

基于其以下两个因素，路由器接口可以分为几种。

- 所采用的技术（比如，以太网）
- 接口之带宽

在现代企业网络中使用到的常见路由器及交换机接口带宽有：

- 100Mbps (通常叫做快速以太网, FastEthernet)
- 1Gbps (通常叫做千兆以太网, GigabitEthernet)
- 10Gbps (通常叫做万兆以太网, TenGigabitEthernet)

为定位到 (address) 一个指定的路由器接口并进入到接口配置模式以设置其特定参数, 你必须知道接口命名法。在不同路由器生产商之间, 其接口命名法会有不同, 但接口命名法通常由两部分组成:

- 接口类型 ( Ethernet, FastEthernet 等)
- 接口插槽/模块以及端口号

比如, 常见的接口命名法有以下这些:

- Ethernet1/0 (第 1 号插槽, 第 0 号端口)
- FastEthernet0/3 (第 0 号插槽, 第 3 号端口)
- GigabitEthernet0/1/1 (第 0 号模块, 第 1 号插槽, 第 1 好端口)

**注意:** 第 0 号插槽通常表示那些内建的端口, 而其它插槽则表示那些可以随时添加上去的拓展插槽。插槽和端口的编号通常是从 0 开始的。

在进行配置时, 为令到路由器具有基本的那些功能, 你务必要配置以下参数:

- 速率, Speed
- 双工, Duplex
- IP 地址, IP address

你可以将这三个参数作为一台路由器的典型配置, 因为它们常用在现代企业网络中。要查看所有可用的接口及其当前状态, 你可以执行以下命令。

```
Router#show ip interface brief
Interface      IP-Address  OK? Method   Status          Protocol
FastEthernet0/0 unassigned  YES unset   administratively down    down
FastEthernet0/1 unassigned  YES unset   administratively down    down
```

从以上输出可以看出, 该路由器在插槽 0 上有两个快速以太网接口 (FastEthernet, 100Mbps), 都没有配置过 (也就是, 没有 IP 地址) 且是管理性关闭的 (也就是, 状态为: administratively down)。

在开始配置接口参数前, 你必须要在思科设备上使用命令 `configure terminal` 进入路由器的配置模式, 在使用命令 `interface <interface name>` 进入到接口配置模式。接口配置过程的第一步是开启该接口。比如, 使用 `no shutdown` 命令可以开启接口 FastEthernet0/0 :

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#no shutdown
Router(config-if)#no shutdown
Router(config-if)#
*Mar 1 00:32:05.199: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:32:06.199: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to
```

接着的配置步骤涉及到配置速率以及双工设置, 前面我们已经看到了。

### 给接口配置一个 IP 地址, Configuring an IP Address on an Interface

**注意:** 为让路由器与其它设备实现通信, 它需要在连接的接口上有一个 IP 地址。配置一个 IP 地址是相当直接的, 你还是要记住, 在此之前需要进入接口配置模式。

先不要担心到哪里去找到 IP 地址，我们后面会解决这个问题。

```
Router>enable ← takes you from User mode to Privileged mode
Router#config t ← from Privileged mode to Configuration mode
Router(config)#interface Serial0 ← and then into Interface Configuration mode
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown ← the interface is opened for traffic
Router(config-if)#exit ← you could also hold down the Ctrl+Z keys together to exit
Router(config)#exit
Router#
```

如下面的输出那样，可以为该接口加入一些描述信息。

```
RouterA(config)#interface Serial0
RouterA(config-if)#description To_Headquarters
RouterA(config-if)#^Z ← press Ctrl+Z to exit
```

在完成路由器的接口配置后，于思科路由器上，你可以使用以下命令，通过检查完整的接配置参数来验证其设置：

```
RouterA#show interface Serial0
Serial0 is up, line protocol is up
Hardware is HD64570
Description: To_Headquarters
Internet address is 12.0.0.2/24
MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation HDLC, loopback not set
Keepalive set (10 sec)
Last input 00:00:02, output 00:00:03, output hang never
[Output restricted...]
```

### show 命令，Show commands

通过在特权模式（privileged mode）下使用 `show x` 命令，你可以十分简单地看到路由器的绝大部分设置项，其中的 `x` 是下一条命令，`x` 的选择有以下这些：

```

Router#show ?
access-expression  List access expression
access-lists       List access lists
accounting        Accounting data for active sessions
adjacency         Adjacent nodes
aliases           Display alias commands
alps              Alps information
apollo            Apollo network information
appletalk          AppleTalk information
arap              Show AppleTalk Remote Access statistics
arp               ARP table
async             Information on terminal lines used as router interfaces
backup            Backup status
bridge            Bridge Forwarding/Filtering Database [verbose]
bsc               BSC interface information
bstun             BSTUN interface information
buffers           Buffer pool statistics
cca               CCA information
cdapi             CDAPI information
cef               Cisco Express Forwarding
class-map         Show QoS Class Map
clns              CLNS network information
--More--

```

下面列出了一些常用的 `show` 命令及其意义，连同两个实例。

Show 命令	结果
<code>show running-configuration</code>	显示 DRAM 中的配置
<code>show startup-configuration</code>	显示 NVRAM 中的配置
<code>show flash</code>	显示闪存中的 IOS
<code>show ip interface brief</code>	显示所有接口的简要信息
<code>show interface Serial0</code>	显示串行接口的统计信息
<code>show history</code>	显示输入的前 10 条命令

```

Router#show ip interface brief
Interface    Address      OK? Method   Status          Protocol
Ethernet0    10.0.0.1    YES manual   up              up
Ethernet1    unassigned   YES unset   administratively down
Loopback0    172.16.1.1   YES manual   up              up
Serial0      192.168.1.1  YES manual   down            down
Serial1      unassigned   YES unset   administratively down

```

其中的 `method` 标签表明地址指定的方式。可以是 `unset`，`manual`，`NVRAM`，`IPCP` 或者 `DHCP`。

路由器能够检索（`recall`）出先前于路由器提示符处输入的一些命令 -- 默认 10 条，方法是使用向上箭头。使用这个特性能够让你无再次输入长命令行，从而节省大量时间和精力。`show history` 命令显示前 10 条命令的缓冲区。

```
Router#show history  
show ip interface brief  
show history  
show version  
show flash:  
conf t  
show access-lists  
show process cpu  
show buffers  
show logging  
show memory
```

通过命令 `terminal history size` 命令来增大历史命令缓冲区 (the history buffer) :

```
Router#terminal history ?  
size Set history buffer size  
<cr>  
Router#terminal history size ?  
<0-256> Size of history buffer  
Router#terminal history size 20
```

### 验证基础路由器配置及网络连通性，Verifying Basic Router Configuration and Network Connectivity

下面的内容解释了一些最为有用的验证基础路由器配置的命令。

#### 版本查看，Show Version

`show version` 命令提供了那些可以说是验证大多数路由器操作的起点的有用信息。包括：

- 路由器的种类 (`show inventory` 是另一个列出路由器硬件信息的有用命令)
- IOS 的版本
- 内存容量
- 内存使用情况
- CPU 类型
- 闪存容量
- 其它硬件参数
- 上次重启原因

这里有一个 `show version` 命令的缩短了的输出。请自己动手输入这个命令。

```
Router#show version  
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.  
Processor board ID FTX0947Z18E  
M860 processor: part number 0, mask 49  
2 FastEthernet/IEEE 802.3 interface(s)  
2 Low-speed Serial(sync/async) network interface(s)  
191K bytes of NVRAM.  
63488K bytes of ATA CompactFlash (Read/Write)  
  
Configuration register is 0x2102
```

#### Show Running-config

`show running-config` 命令提供了路由器的完整配置，用于验证设备已被配置了恰当特性。因为其输出太过宽泛，这里就不给出来了。

#### Show IP Interface Brief

在前一部分提到的 `show ip interface brief` 命令，列出了路由器的接口以及它们的状态，包括以下项目：

- 接口的名称及编号
- IP 地址
- 链路状态
- 协议状态

```
Router#show ip interface brief
Interface      IP-Address  OK? Method    Status        Protocol
FastEthernet0/0 unassigned  YES unset   administratively down    down
FastEthernet0/1 unassigned  YES unset   administratively down    down
Serial0/0/0     unassigned  YES unset   administratively down    down
Serial0/1/0     unassigned  YES unset   administratively down    down
Vlan1          unassigned  YES unset   administratively down    down
Router#
```

### Show IP Route

`show ip route` 命令提供了有关设备路由能力的更深层次信息。它列出路由器所能到达的所有网络及到达这些网络的路径的信息，包括这些项目：

- 网络
- 路由协议
- 下一跳
- 外出接口

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter
           area, N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external
           type 2, E1 - OSPF external type 1, E2 - OSPF external type 2,
           i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
           inter area, * - candidate default, U - per-user static route,
           o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
R      80.1.1.0/24 [120/1] via 10.1.1.2, 00:00:04, Ethernet0/0.1
D      80.0.0.0/8 [90/281600] via 10.1.1.2, 00:02:02, Ethernet0/0.1
O E2   80.1.0.0/16 [110/20] via 10.1.1.2, 00:00:14, Ethernet0/0.1
```

除了上面的这些 `show` 命令外，还有一些用于验证路由器连通性的命令，比如 `ping` 和 `traceroute` 命令。

### Ping 命令

`ping` 命令提供了一种到特定目标的基本连特性测试。这种方式用以测试路由器能否到达一个网络。Ping 使用 ICMP，通过往一台机器发送 echo 请求方式来验证这台机器是否在运行。如果那台机器是在运行，它就会发出一个 ICMP 的 echo 回应消息给源机器，以确认它的可用性。一个 `ping` 的样例如下所示。

```
Router#ping 10.10.10.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:
!!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 20/40/76 ms
```

标准的 `ping` 命令发出 5 个到目标的 ICMP 数据包。而 `ping` 输出中，点（.）表示失败，叹号（！）表示成功收到数据包。`ping` 命令的输出还给出了到目标网络的往返时间（the round-trip time），有最长时间、平均时间以及最大时间。

如你需要调整 `ping` 相关的参数，你可在思科路由器上执行扩展的 `ping` 命令。通过在控制台处输入 `ping` 并按下回车键来执行。路由器就会通过一个交互式菜单进行提示，你就可以指定包含以下的这些参数了。

- ICMP 数据包的个数
- 包的大小
- 超时量
- 源接口
- 服务类型

```
Router#ping
Protocol [ip]:
Target IP address: 10.10.10.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: FastEthernet0/0
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.2, timeout is 2 seconds:
Packet sent with a source address of 10.10.10.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 20/36/72 ms
```

### Traceroute 命令

`traceroute` 命令是另一个用于查看数据包在到达其目的地前所经过的跳数。下面的输出表示数据包在到达其目标前必须经过一跳。

```
R2#traceroute 192.168.1.1
Type escape sequence to abort.
Tracing the route to 192.168.1.1
 1 10.10.10.1 60 msec * 64 msec
```

跟 `ping` 一样，思科路由器也允许你执行扩展的 `traceroute` 命令，搭配一些相关参数，而这些参数大多与 `ping` 相关的参数一样。

```
Router#traceroute
Protocol [ip]:
Target IP address: 192.168.1.1
Source address: 10.10.10.2
Numeric display [n]:
Timeout in seconds [3]:
Probe count [3]:
Minimum Time to Live [1]:
Maximum Time to Live [30]:
Port Number [33434]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Type escape sequence to abort.
Tracing the route to 192.168.1.1
 1 10.10.10.1 76 msec * 56 msec
```

## 第一天的问题

### OSI/TCP 模型的问题 OSI/TCP Model Questions

1. Name each layer of the OSI model, from Layer 7 down to Layer 1.
2. The role of the Session Layer is to \_\_\_\_\_, \_\_\_\_\_, and \_\_\_\_\_ sessions or dialogues between devices.
3. What are the three methods used to control data flow at Layer 4?
4. The Transport Layer includes several protocols, and the most widely known are \_\_\_\_\_ and \_\_\_\_\_.
5. Why is UDP used at all if TCP/IP offers guaranteed delivery?
6. What is data referred to at each OSI layer?
7. In order to interface with the upper and lower levels, the Data Link Layer is further subdivided into which two Sublayers?
8. What are the five TCP/IP layers from the top down?
9. How does the TCP/IP model map to the OSI model?
10. Layer 2 addresses are also referred to as \_\_\_\_\_ addresses.
11. Using a switch will allow you to divide your network into smaller, more manageable sections known as \_\_\_\_\_.

### 线缆的问题 Cable Questions

1. The current standard Ethernet cable still uses eight wires twisted into pairs to prevent \_\_\_\_\_, \_\_\_\_\_ and \_\_\_\_\_.
2. \_\_\_\_\_ is when a signal from one Ethernet wire spills over into a neighbouring cable.
3. Which command would set the FastEthernet router interface speed to 10Mbps?
4. On a crossover cable, the wire on pin 1 on one end needs to connect to pin \_\_\_\_\_ on the other end and pin 2 needs to connect to pin \_\_\_\_\_.
5. Which cable would you use to connect a router Ethernet interface to a PC?
6. You can see a summary of which interfaces you have on your router with the show \_\_\_\_\_ command.
7. Line Configuration mode lets you configure which ports?
8. A Loopback interface is a \_\_\_\_\_ or \_\_\_\_\_ interface that you configure.
9. The keyboard shortcut Ctrl+A does what?
10. The \_\_\_\_\_ keyboard shortcut moves the cursor back one word.
11. By default, the \_\_\_\_\_ command shows the last 10 commands entered.

# 第一天的答案

## OSI/TCP 模型答案

1. Application, Presentation, Session, Transport, Network, Data Link, and Physical.
2. Set up, manage, and terminate.
3. Flow control, windowing, and acknowledgements.
4. TCP and UDP.
5. TCP uses a lot of bandwidth on the network and there is a lot of traffic sent back and forth to set up the connection, even before the data is sent. This all takes up valuable time and network resources. UDP packets are a lot smaller than TCP packets and they are very useful if a really reliable connection is not that necessary. Protocols that use UDP include DNS and TFTP.
6. Bits (Layer 1), Frames (Layer 2), Packets (Layer 3), Segments (Layer 4) and Data (Layers 5-7).
7. LLC and MAC.
8. Application, Transport, Network, Data Link, and Network.
- 9.

Layer #	OSI	Data
7	Application	Application
6	Presentation	
5	Session	
4	Transport	Host to Host
3	Network	Internetwork
2	Data Link	Network Interface
1	Physical	

1. MAC.
2. Collision domains.

## 线缆答案 Cable Answers

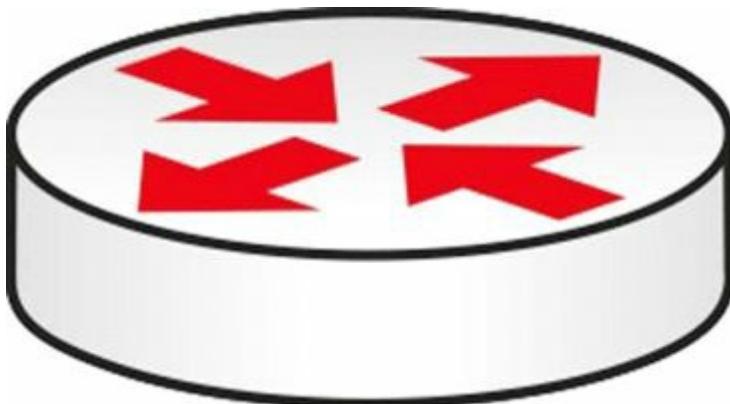
1. Electromagnetic interference (EMI) and crosstalk.
2. Crosstalk.

3. The `speed 10` command.
4. 3 and 6.
5. A crossover cable.
6. `ip interface brief`.
7. The console, Telnet, and auxiliary ports.
8. Virtual or logical.
9. Moves the cursor to the beginning of the command line.
10. Esc+B.
11. `show history`.

## 第一天的实验 Day 1 Lab

### IOS 命令导航实验 IOS Command Navigation Lab

#### 拓扑, Topology



#### 实验目的, Purpose

学习如何通过控制台接口连接到一台路由器，以及尝试一些命令。

#### 步骤, Walkthrough

1. 使用一条控制台线缆，和 PuTTY 程序（可免费在线获取，请搜索“PuTTY”），连接到一台路由器的控制台端口。
2. 在 `Router>` 提示符处，输入下面的这些命令，探寻不同的路由器模式和命令。如你遇到询问进入配置模式，输入 `no` 并按下回车键。

```

Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE
SOFTWARE (fc2)
Technical Support: www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
    --- System Configuration Dialog ---
Continue with configuration dialog? [yes/no]:no
Press RETURN to get started!
Router>enable
Router#show version
Cisco 1841 (revision 5.0) with 114688K/16384K bytes of memory.
Processor board ID FTX0947Z18E
M860 processor: part number 0, mask 49
2 FastEthernet/IEEE 802.3 interface(s)
2 Low-speed Serial(sync/async) network interface(s)
191K bytes of NVRAM.
63488K bytes of ATA CompactFlash (Read/Write)
Configuration register is 0x2102
Router#show ip interface brief
Interface      IP-Address  OK? Method  Status          Protocol
FastEthernet0/0 unassigned  YES unset   administratively down  down
FastEthernet0/1 unassigned  YES unset   administratively down  down
Serial0/0/0     unassigned  YES unset   administratively down  down
Serial0/1/0     unassigned  YES unset   administratively down  down
Vlan1          unassigned  YES unset   administratively down  down
Router#
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface Serial0/1/0 ~ put your serial # here
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#interface Loopback0
Router(config-if)#ip address 10.1.1.1 255.0.0.0
Router(config-if)#^Z ~ press Ctrl+Z keys together
Router#
Router#show ip interface brief
Interface      IP-Address  OK? Method  Status          Protocol
FastEthernet0/0 unassigned  YES unset   administratively down  down
FastEthernet0/1 unassigned  YES unset   administratively down  down
Serial0/0/0     unassigned  YES unset   administratively down  down
Serial0/1/0     192.168.1.1 YES manual  administratively down  down
Loopback0       10.1.1.1   YES manual  up            up
Vlan1          unassigned  YES unset   administratively down  down
Router#show history
Router(config)#hostname My_Router
My_Router(config)#line vty 0 ?
    <1-15>  Last Line number
    <cr>
My_Router(config)#line vty 0 15 ~ enter 0 ? to find out how many lines you have
My_Router(config-line)#
My_Router(config-line)#exit
My_Router(config)#router rip
My_Router(config-router)#network 10.0.0.0
My_Router(config-router)#

```

## 第2天 CSMA/CD, 交换和虚拟局域网

### CSMA/CD, Switching, and VLANs

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

## 第二天的任务

- 阅读今天的课文
- 复习昨天的课文
- 阅读 ICND1 记诵指南

思科工程师混饭吃的手艺就是安装、配置和调试（troubleshooting）交换机。而这又恰恰是这些工程师们不那么擅长的事情，这么说有些不思议吧。可能有些人靠的是交换机本身的即插即用能力，或是在问题出现后直面解决。这样的“凭感觉试试看”的风格，在出现交换相关的问题时，就会心态事与愿违了。（This "fly by the seat of your pants" mentality backfires for many engineers when there is a switching-related issue.）

我建议你在学习本书时，先粗略地过一遍，然后在回头读几次，每次都将那些重点做一下笔记或是划一下重点。

今天你将学到以下内容：

- CSMA/CD
- 虚拟局域网，VLANs
- 配置 VLANs
- VLANs 故障排除

此模块涵盖 CCNA 大纲要求的以下方面：

- 掌握以太网络中用到的技术及介质访问控制方法
- 理解基本的交换概念及思科交换机操作：
  - 冲突域
  - 广播域
  - 交换的不同类型
  - CAM 表
- 配置并验证初始交换机配置，含远程访问管理
  - 执行基本交换机设置的那些思科 IOS 命令
- 使用如 ping、Telnet 以及 SSH 等基本工具程序来验证网络状态以及交换机的运作
- 描述 VLANs 是如何创建出逻辑独立网络，以及这些网络之间的路由需求
  - 解释网络分段及基本的流量管理概念
- 配置和验证 VLANs

## 交换机基础知识

### 带有冲突检测载波侦听，多路复用，Carrier Sense, Multiple Access with Collision Detection

带有冲突检测载波侦听，多路复用(Carrier Sense, Multiple Access with Collision Detection, CSMA/CD)一词可以分解为以下几个部分，“载波侦听”的意思是线路为设备所侦听，以确定是否有信号在其上传输。如果线路正在使用中，那么以太网帧是不能发送出去的。“多路复用”的意思是网段上有多余一台的设备在使用线缆。最后的“冲突检测”是指协议运行着一套确定线路上的太网帧是否因为遇到另一个帧而已经损坏的算法。从下图 2.1 你可以看到交换机端口在监听着线路。

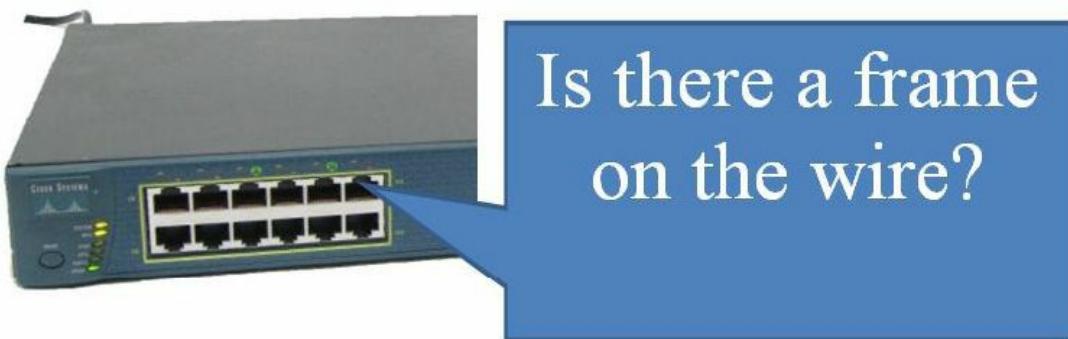


图 2.1 -- 端口监听着线路

如果线路上出现了冲突，则监听设备会发出一个拥塞信号，以通知其它设备发生了冲突，它们就不会尝试往线路上发送数据了。此时，协议算法运行起来，产生一个随机数时间间隔，在以此间隔后重传。在线路清空前，设备不会发送以太网帧。Wikipedia 上是这样解释该过程的：

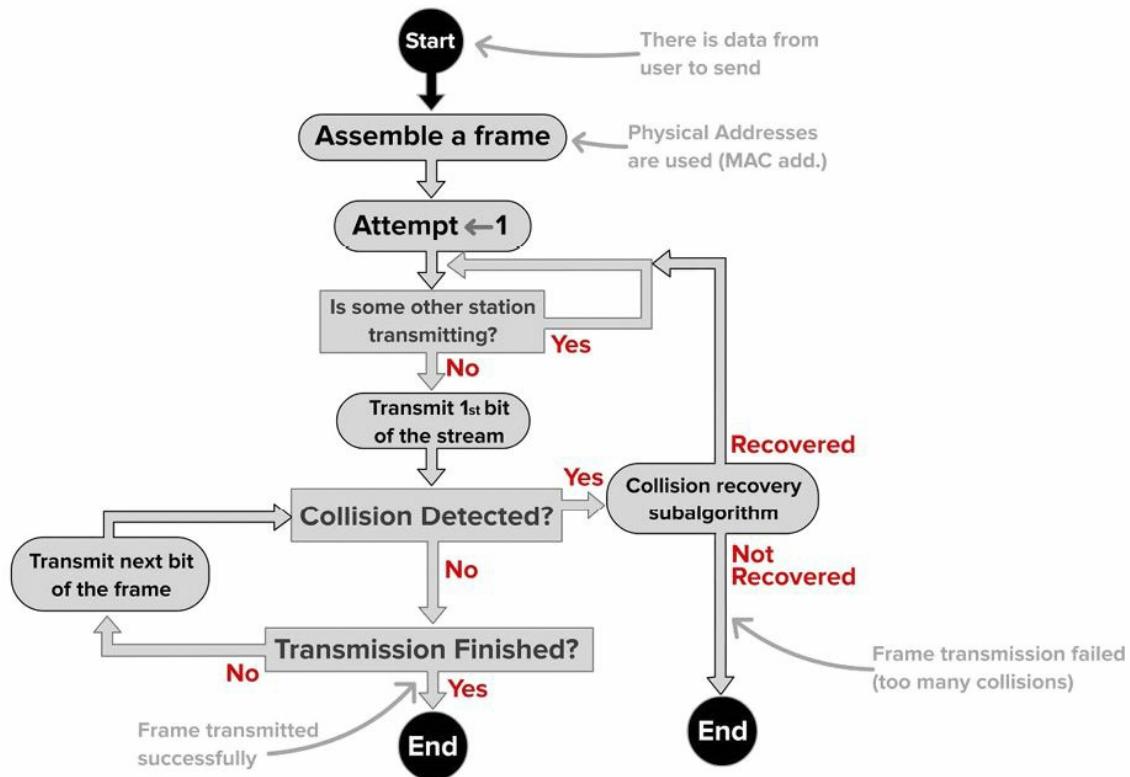


图 2.2 -- CSMA/CD 过程

Farai 指出 -- “需要注意，现代交换机使用的是全双工连接交换机，因此而不会用到 CSMA/CD。但它仍然支持该技术，而这完全是为了向后兼容性。”

## 冲突域和广播域

网络集线器的一个缺点是在线路上发生冲突时，损坏的帧会发送到所有连接的设备。现代交换机的优势之一就是交换机的每个端口都是作为一个冲突域。在冲突发生时（全双工下是不可能出现的），损坏的帧不会通过接口。图 2.3 展示了一台交换机增加到使用两台集线器的小型网络上的情形。交换机将该网络分解成两个冲突域。

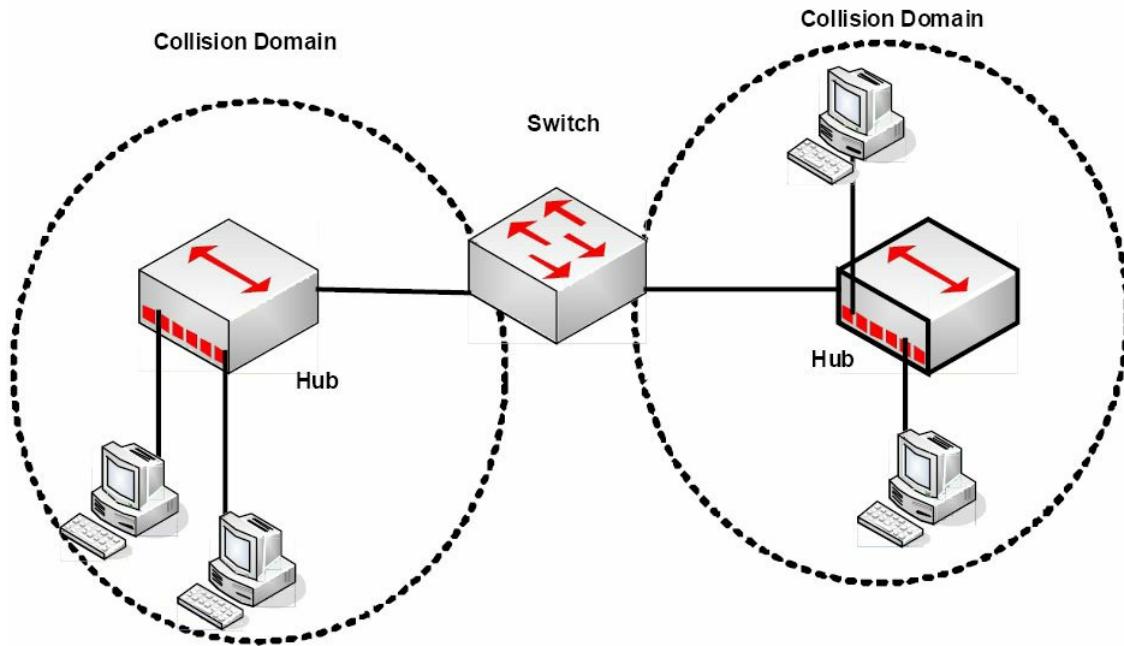


图 2.3 -- 一台交换机创建出两个冲突域

思科通常会在考试中以提问交换机是否减少冲突域数量的方式引诱你犯错。匆忙之下，你可能受导向说交换机会减少冲突域数量，但实际情况是相反的，交换机会增加冲突域的数量，而这是好事。交换机确实增加了冲突域的数量。因为集线器受限于其所采用的技术，而只能工作于半双工下，它就显得相当无用了。

图 2.4 中，4 台 PC 连接到交换机上，产生 4 个冲突域。每台 PC 都工作在全双工下，能够完全用上 100Mbps 的带宽。

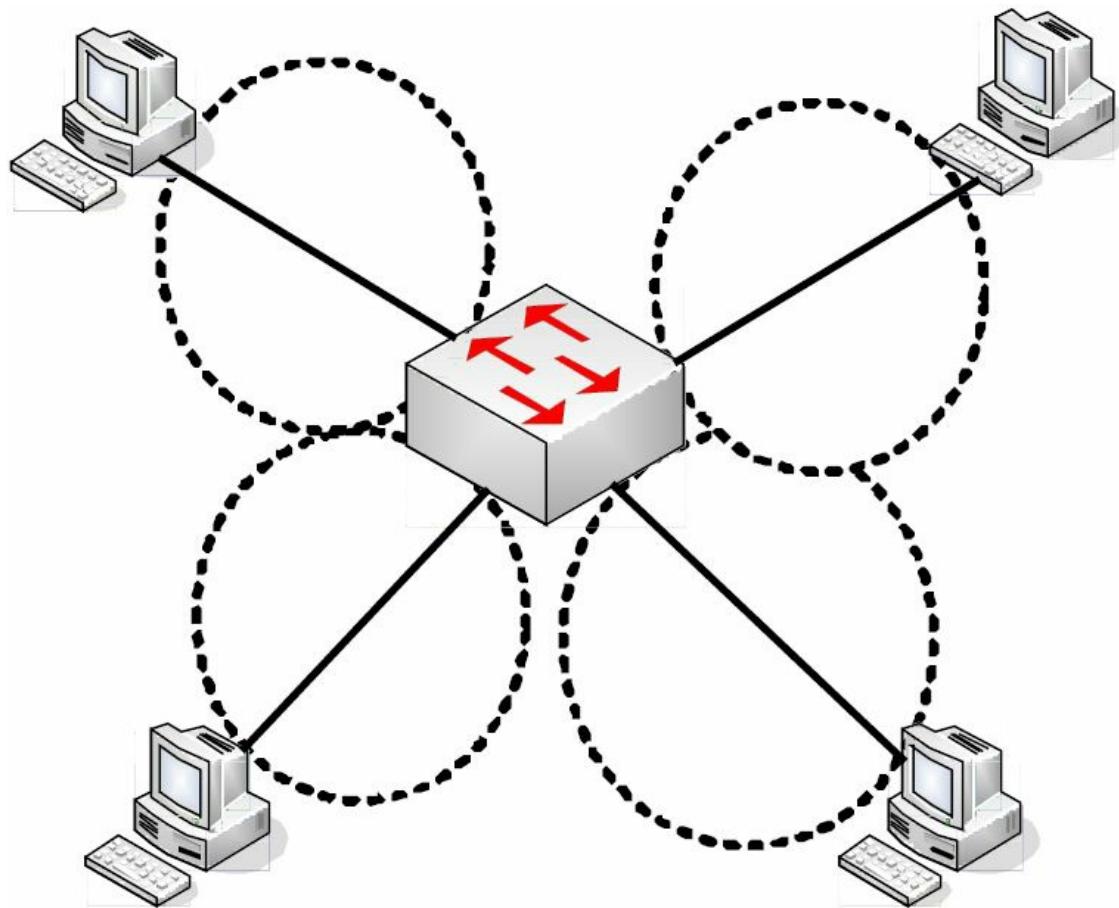


图 2.4 -- 四个冲突域

交换机（这里说的是二层交换机）不会隔离广播域，路由器会。如果交换机收到带有广播目的地址的以太网帧，就转发给所有端口，不管该帧是从哪个端口收到的。需要一台路由器来隔离广播域。图 2.5 展示了使用交换机/网桥以及一台路由器的小型网络，用以说明冲突域是如何隔离的。

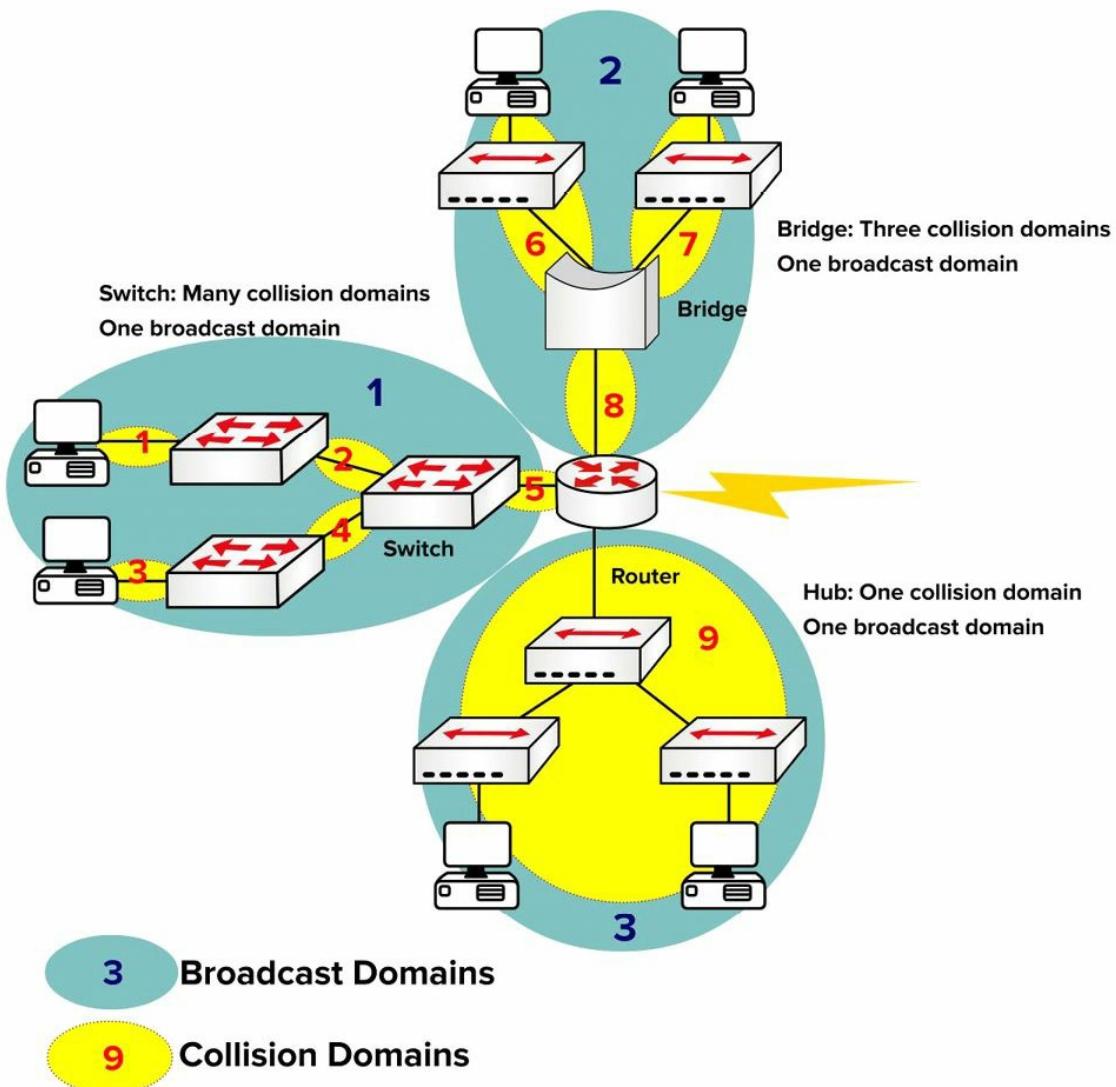


图 2.5 -- 广播域和冲突域

## 自动协商, Auto-negotiation

你已经见到在连接不同速率和双工设置时，可能会出问题。在网络的一部，你可能会经常予以升级，但因为财务预算的制约，网络的其它部分会有老旧设备。这就会导致双工和速率的不匹配，引起错误和丢帧现象的发生。后面的章节中，我们将涉及交换机故障排除的内容。

IEEE 将自动协商作为解决此问题的方案，该技术让设备在传输流量前，就双工和速率上达成一致。速率设置为低速设备的速率。在下面的输出中，速率可被手动设定为 10Mbps 或 100Mbps，或者设定为 auto-negotiation。

```
Switch(config)#int f1/0/1
Switch(config-if)#speed ?
 10 Force 10 Mbps operation
 100 Force 100 Mbps operation
 auto Enable AUTO speed configuration
```

该设置可用命令 `show interface x` 进行查看。

```

Switch#show int f1/0/1
FastEthernet1/0/1 is down, line protocol is down (notconnect)
    Hardware is FastEthernet, address is 001e.13da.c003 (bia 001e.13da.c003)
        MTU 1600 bytes, BW 10000 Kbit, DLY 1000 usec,
        reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA, Loopback not set
    Keepalive set (10 sec)
    Auto-duplex, Auto-speed, media type is 10/100BaseTX

```

请务必牢记，尽管有如此设置，**auto-negotiation** 仍可能会引起问题。这就是为何许多生产性网络仍然坚持将端口直接配置成 100/full 或者千兆以太网的 1000/full。思科这样解释的：

不合格的应用(nonconforming implementation)、硬件不兼容或者软件缺陷(software defects)三种原因，可能会导致各种自动协商问题。在网卡或厂商交换机与 IEEE 802.3u 规范不完全一致时，问题就会发生。厂商特定的一些高级特性，比如自动正负极性或线缆完整性 (auto-polarity or cable integrity) 等在 IEEE 802.3u 的 10/100Mbps 自动协商标准中没有描述的那些特性，同样会导致硬件的不兼容或其它问题。 ([Cisco.com](#))

## 帧交换，Switching Frames

交换机是为交换帧而生（也就是说，将来自某进入接口的帧传输到正确的出口接口）。广播帧被交换机所有接口（除了接收到广播帧的那个接口），带有不明目的地（目的地址不在 MAC 表中）的那些帧也一样，交换机执行下面三个动作：

- 根据目的 MAC 地址，进行帧转发或过滤(forwarding or filtering)
- 从进来的帧学习 MAC 地址
- 使用 STP 协议来阻止二层环回的发生 (STP 在 ICND2 第 31 天学习)

图 2.6 中，交换机将来自主机 A (F0/1) 以主机 C 为目的地的帧正确转发出 F0/3，而阻止其离开接口 F0/2。

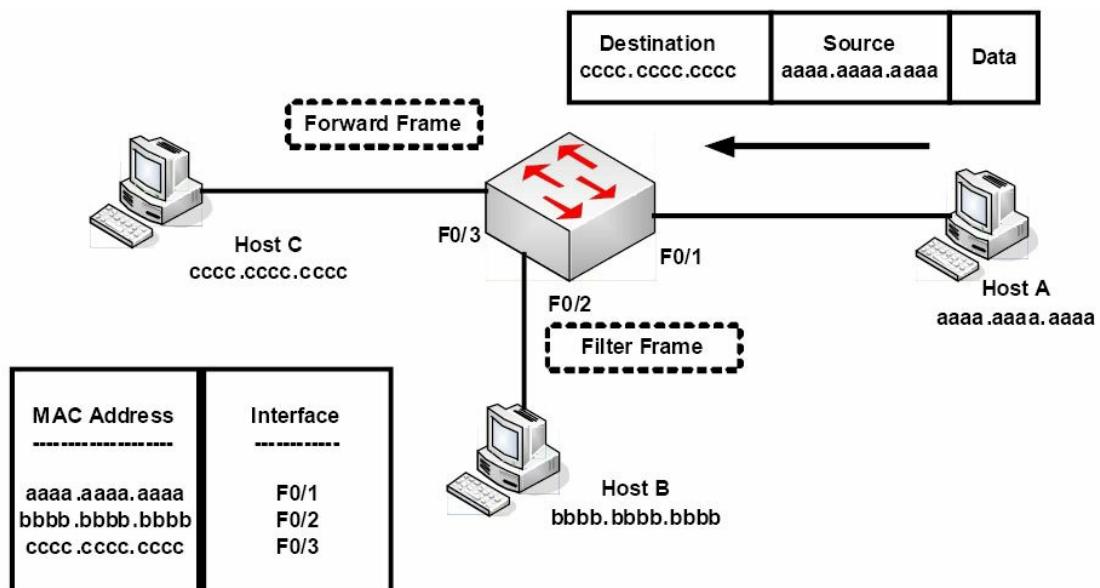


图 2.6 -- 帧过滤

如目的地址不在 MAC 地址表中，交换机将该帧泛洪(flooding)至除它收到该帧的那个接口外的所有接口上。交换机也会存储那些连接在另一台交换机上的设备的 MAC 地址；不过在地址表中它们对应的接口名称会是同一个，这样下来多个 MAC 地址与一个同样的出口接口对应，列出在 MAC 地址表中。这是一种找出网络

上你不熟悉设备的方法。图 2.7 用以说明这个概念。

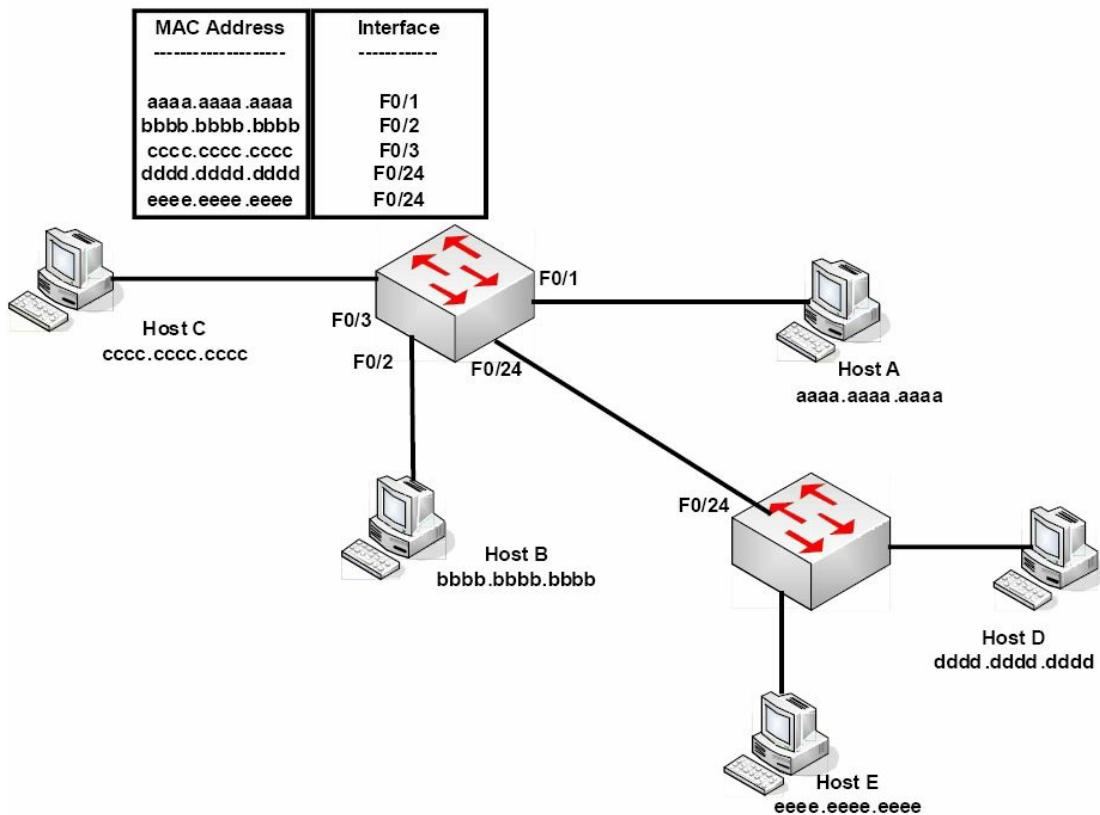


图 2.7 -- 同一接口上的多 MAC 地址

流量传输中的任何延时(delay)，都被称为传输延迟 (latency)。依据你所希望的在流量传输前对帧的检查程度，思科交换机提供了三种流量交换的方式。对帧的检查越多，引入到交换机的延迟就越多。三种可供选择的交换模式(switching modes)为：

- 直通模式，Cut-through
- 存储转发模式(交换机默认)，Store-and-forward
- Fragment-free (改进的直通模式)

### 直通模式

直通模式交换是最快交换方式，它有最低的延迟。进入交换机的帧仅读至目的地址，便做出转发决定。在获知目的地址后，交换机马上检查 CAM 表，以找到正确的端口来转发出该帧并马上发出。因为没有错误检查，此方式才能提供最低的延迟。代价是交换机会转发任何带有错误的帧。

使用一个比方来说明交换机模式是最好不过了。你作为某夜店的保安，被要求每个进入夜店的人都要有一个带照片的出入卡 -- 却并没要求你去查看照片是否与那个人一致，只是出入卡要有就行。这种方式下，人们必定能快速进入夜店了。这就是直通模式的工作原理。

### 存储转发模式

交换机读取整个帧，并将其复制到它的缓冲区。对该帧执行一次循环冗余校验 (cyclic redundancy check, CRC) 以检查其存在的任何错误。如有发现错误，该帧就被丢弃。相反，就检查交换表，并转发该帧。存储转发模式确保帧至少有 64 字节大，且不大于 1518 字节。若小于 64 字节或大于 1518 字节，交换机就会丢弃帧。

现在请再次设想你是夜店保安，此时你务必要确保照片与那个人是相吻合的，同时你要在放进那些人之前记下他们的名字和地址。这样来查验出入卡就造成了相当大的延迟，这也是存储转发交换运行的方式。

三种交换方式中有着最高延迟的就是存储转发交换，也是 2900 系列交换机默认的交换方式。

#### Fragment-free(修订的直通模/Runt-free 模式)

因为直通交换不检查错误，而存储转发模式又耗时太长，我们需要一种又快又可靠的方式。使用夜店保安的例子，设想你被要求确保每个人都有出入卡同时照片又要吻合。此方式下，你确保每个人都是其宣称的那个人，但你不必记下他们的所有信息。在交换中，这是通过采用 fragment-free 交换方式实现的，低端的 (lower-level) 思科交换机默认配置为此种模式。

Fragment-free 交换是直通交换的一个修改变种。检查帧的前 64 字节有无错误，随后传输出去。此方法背后推理 (reasoning) 是帧的错误最有可能发生在前 64 字节中。

如同上面已经提到的，以太网帧的最小尺寸是 64 字节；任何小于 64 字节的帧被称为是“侏儒 (runt) ”帧。因为转发前的帧都必须至少有 64 字节，这就会消除那些侏儒帧，这也是为何此模式又被叫做 “runt-free” 交换的原因。

## 交换基本概念，Switching Concept

### 交换机使用需求

在交换机发明前，网络上的所有设备都会接收到来自其它设备的数据。一旦探测到线路上有一个数据帧，PC 就不得不停下来查看其头部，看看自己是不是数据帧的接收者。设想一下网络上每分钟都有上千个帧吧。所有设备很快就被折腾到挂起。图 2.8 展示了网络上的所有设备；注意因为是通过仅转发的集线器连接在一起，它们都不得不共享同一带宽。

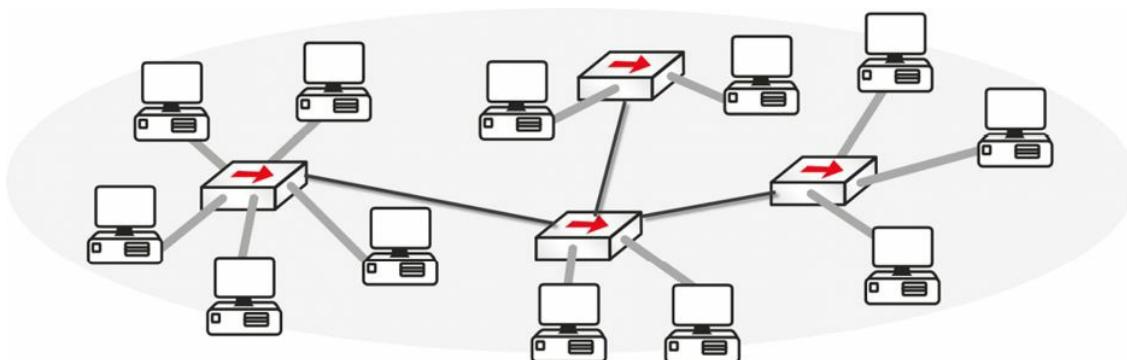


图 2.8 -- 每台设备都听着其它设备

### 集线器的问题

之前我曾提到集线器仅是简单的多端口中继器(见 2.9)。它们接收传入的信号，进行清理，然后在插线了的端口上发出。它们同时创建出一个巨大的冲突域。

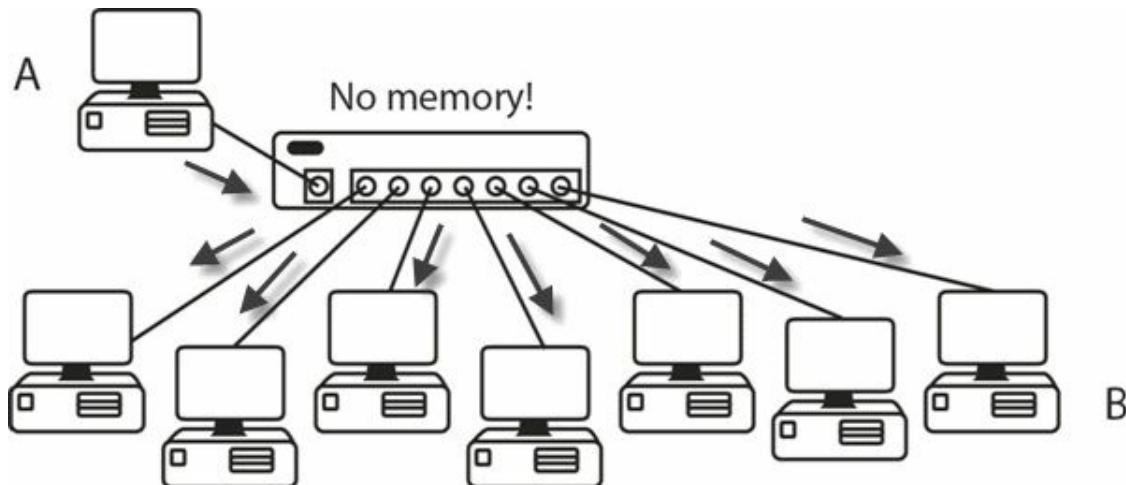


图 2.9 -- 集线器在每个端口上都发送帧

集线器是愚蠢(dumb)设备。它们没有 MAC 地址存储机制，所以在设备 A 每次往设备 B 发送一个帧时，它都会往每个端口发送。交换机就不一样，有一块叫做专用集成电路 (application-specific integrated circuit, ASIC) 的存储芯片，该芯片会建立一个设备端口表 (图 2.10)。这个表保存在内容可寻址存储器 (Content Addressable Memory, CAM) 中。

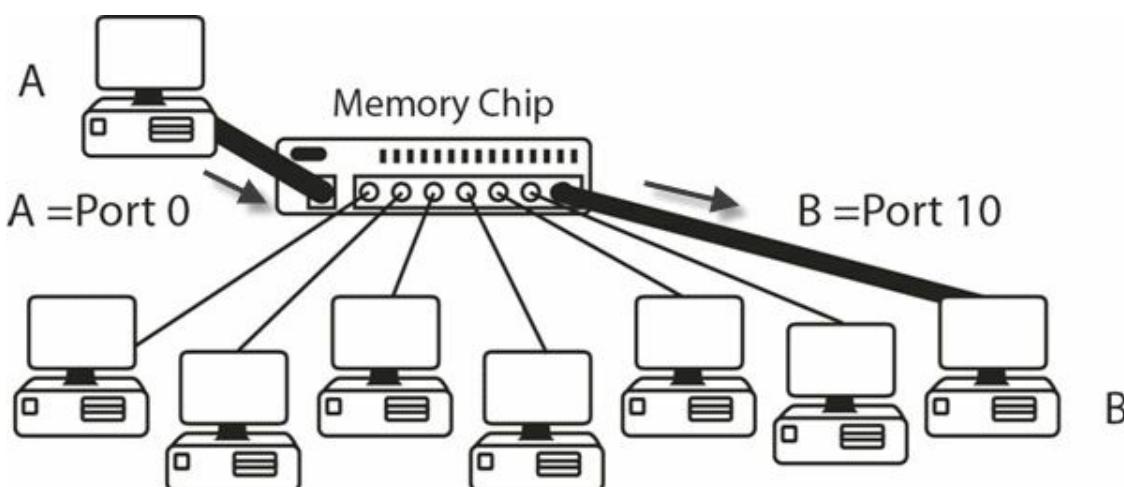


图 2.10 -- 交换机建立起一张 MAC 地址表

在首次启动时，交换机并未在其 CAM 表（思科考试又此表称为 MAC 地址表）中存储任何地址。一有帧开始传输，该表就建立起来。如果在指定时间过后没有帧从某个端口传送，这条记录就会过期。下面的输出表明，至今仍没有帧在交换机上通过。

```
Switch#show mac-address-table
  Mac Address Table
-----
Vlan   Mac Address      Type      Ports
---   -----
Switch#
```

交换机中没有记录，不过当你从一台路由器 ping 另一台时（两台都连接上交换机），表格条目建立起来。

```

Router#ping 192.168.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 62/62/63 ms
Switch#show mac-address-table
      Mac Address Table
-----
Vlan     Mac Address      Type      Ports
----  -----
1        0001.c74a.0a01  DYNAMIC    Fa0/1
1        0060.5c55.da01  DYNAMIC    Fa0/2

```

该条目的意思是，任何目的地址为连接到交换机上 FastEthernet 端口 0/1 或 0/2 的 MAC 地址的帧，都会直接发送到对应的端口。对于任何其它帧，交换机都将执行一次广播查询，看看目的设备是否插入了交换机。从上面的五次 ping 中的第一个句点可以看出。在等待交换机广播查询及收到目的路由器回应时，第一次 ping 发生了超时（80% 的成功率）。

**show mac-address-table 命令是一个非常重要的命令**，务必要记住这个命令，考试和现实工作中都是需要的。你已经注意到 MAC 地址是个什么东西了。MAC 地址指派给所有设备，以实现数据链路层的通信。在以太网卡、路由器的以太网接口及无线设备上，你都能看到各厂商分配的 MAC 地址。下面是我的笔记本的以太网卡的 MAC 地址。

```

Ethernet adapter Local Area Connection:
  Connection-specific DNS Suffix . . . . . : BigPond
  Description . . . . . : Realtek RTL8168C(P),
  igabit Ethernet NIC
  Physical Address . . . . . : 00-1E-EC-54-85-17
  Dhcp Enabled . . . . . : Yes
  Autoconfiguration Enabled . . . . . : Yes
  IP Address . . . . . : 10.0.0.11
  Subnet Mask . . . . . : 255.255.255.0
  Default Gateway . . . . . : 10.0.0.138

```

各家厂商有各自的组织唯一识别号（Organizationally Unique Identifier, OUI），该号码构成了 MAC 地址的前半段。随后他们就可以根据其各自的编号系统来自由创建地址的后半段了。一个 MAC 地址是 48 位二进制数（后面我们会涉及二进制和十六进制的知识），所以上面我的地址构成为：

OUI	厂商编号
24 位二进制数	24 位二进制数
6 位十六进制数	6 位十六进制数
00 1E EC	54 85 17

如交换机在某个接口上收到一个帧，它就将帧的源地址加入到表中。如它知道帧的目的地址，就将该转发出相应接口。如它不知道目的地址，它将把该帧广播到除了收到该的所有接口。如果交换机收到一个广播帧（也就是目的地址全 F 的帧），它也会将该帧广播到除接收到该帧的所有接口。后面我们会涉及十六进制编址。广播过程如图 2.11 所示。

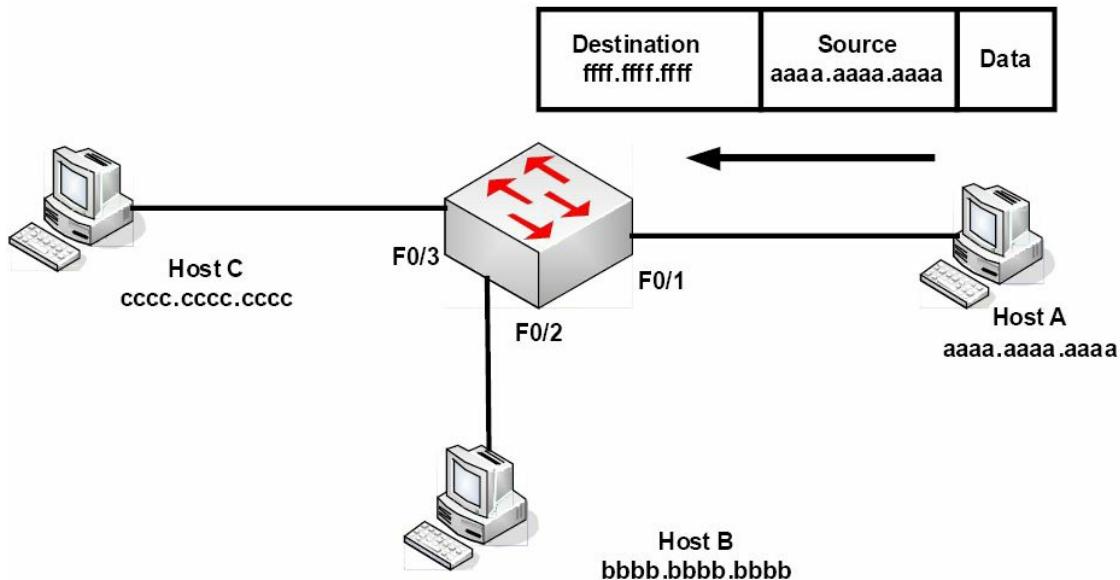


图 2.11 -- 广播帧在所有接口上发出

## 以太网帧

有四种不同类型的以太网帧：

- 以太网 802.3 帧
- 以太网 II 帧
- 以太网 802.2 SAP 帧
- 以太网 802.2 SNAP 帧

前两种以太网标准用于在网卡之间通信时的组帧方式。它们不能识别上层协议，802.2 类型帧才具备此能力。你只需要注意 802.3 类帧，此类型的帧构成如下。

Preamble	SFD	Destination address	Source address	Length	Data	FCS
----------	-----	---------------------	----------------	--------	------	-----

图 2.12 -- 以太网 802.3 帧结构

IEEE 委员会确定的 IEEE 802.3 以太网帧有以下特定字段。

- 前同步信号，preamble -- 为传入的数据对网卡进行信号同步及告知
- 帧开始界定符，start-of-frame delimiter, SFD -- 标志着帧的开始
- 目的地地址 -- 目的 MAC 地址，可以是单播 (Unicast)、广播或多播 (Multicast)
- 源地址 -- 发送主机的 MAC 地址
- 长度 -- 定义帧中数据字段的长度
- 数据 -- 帧中的载荷（就是传输中的数据）
- 帧校验序列，frame-check sequence, FCS -- 给出了帧中所有数据的循环冗余校验 (cyclic redundancy check, CRC)

## 交换机初始配置，Initial Switch Configuration

你需要先像连接全新的路由器那样，用通过控制台端口连接新的交换机。因为在你能够在通过 Telnet 或 SSH（稍后会更多地讲到）连上交换机之前，交换机上至少得已经有一两条配置才行。很多交换机的初始配置和路由器的初始配置都是一样的。

在首次连接交换机时，在任何设上都很有必要执行一下 `show version` 命令（其输出见下面）。考试中也要求你明白哪个 `show` 命令提供哪些信息。大多数情况下你不能从考试模拟器访问中得到答案，你就只有靠记忆来作答了。

`show version` 命令提供了很多有用信息，包括这些。

- 交换机运行时间，`switch uptime`
- 型号
- IOS 版本，`IOS release`
- 上次重启的原因
- 所有接口及其类型
- 所有安装的存储器
- 背板 MAC 地址，`base MAC address`

```
Switch>en
Switch#show version
Cisco IOS Software, C2960 Software (C2960-LANBASE-M), Version 12.2(25)FX, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 12-Oct-05 22:05 by pt_team
ROM: C2960 Boot Loader (C2960-HBOOT-M) Version 12.2(25r)FX, RELEASE SOFTWARE (fc4)
System returned to ROM by power-on
Cisco WS-C2960-24TT (RC32300) processor (revision C0) with 21039K bytes of memory.
24 FastEthernet/IEEE 802.3 interface(s)
2 GigabitEthernet/IEEE 802.3 interface(s)
63488K bytes of flash-simulated non-volatile configuration memory.
Base Ethernet MAC Address_____ : 0090.2148.1456
Motherboard assembly number : 73-9832-06
Power supply part number : 341-0097-02
Motherboard serial number : FOC103248MJ
Power supply serial number : DCA102133JA
Model revision number : B0
Motherboard revision number : C0
Model number : WS-C2960-24TT
System serial number : FOC1033Z1EY
Top Assembly Part Number : 800-26671-02
Top Assembly Revision Number : B0
Version ID : V02
CLEI Code Number : COM3K00BRA
Hardware Board Revision Number : 0x01
Switch Ports Model SW Version SW Image
----- -----
* 1 26 WS-C2960-24TT 12.2 C2960-LANBASE-M
Configuration register is 0xF
```

我们还没有涉及 VLANs 的知识，但现在，你可以把 VLAN 看着是一个逻辑上的局域网，在 VLAN 上的设备物理上可以在不同地方，但在能它们所能关注到的细节下（as far as they are concerned），他们都是直接连接在一台交换机下的。在下列配置中，交换机所有端口都是默认在 VLAN 1 中。

```
Switch#show vlan
VLAN      Name        Status      Ports
----      ----        -----      -----
1         default     active      Fa0/1, Fa0/2, Fa0/3, Fa0/4,
                           Fa0/5, Fa0/6, Fa0/7, Fa0/8,
                           Fa0/9, Fa0/10, Fa0/11, Fa0/12,
                           Fa0/13, Fa0/14, Fa0/15, Fa0/16,
                           Fa0/17, Fa0/18, Fa0/19, Fa0/20,
                           Fa0/21, Fa0/22, Fa0/23, Fa0/24,
```

如你打算给交换机添加一个 IP 地址（就是**管理地址**），以便通过网络连上该交换机，只需给某个 VLAN 配置 IP 地址即可；本例中就是 VLAN1。

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#interface vlan1
Switch(config-if)#ip add 192.168.1.3 255.255.255.0
Switch(config-if)# ~ hold down Ctrl+Z keys now
Switch#show interface vlan1
Vlan1 is administratively down, line protocol is down
  Hardware is CPU Interface, address is 0010.1127.2388 (bia 0010.1127.2388)
  Internet address is 192.168.1.3/24
```

而 **VLAN1 默认是关闭的**，你需要执行一个 `no shutdown` 命令来开启它。**还需告诉交换机往哪里发送所有 IP 流量，因为 2 层交换机没有建立路由表的能力**；该操作如下面的输出所示。

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#ip default-gateway 192.168.1.1
Switch(config)#

```

在网络上有许多台交换机时，你会想着去修改交换机默的主机名，这样在远程连接它们时才更容易区分（见下面的配置命令行）。设想一下通过远程 Telnet 对五台同样叫做“Switch”的交换机进行故障排除时的情形吧。

```
Switch(config)#hostname Switch1
```

如你要经由网络 Telnet（或 SSH）到某台交换机，你还需开启该协议。交换机远程访问默认是关闭的。

```
Switch1#conf t
Enter configuration commands, one per line.
Switch1(config)#line vty 0 15
Switch1(config-line)#password cisco
Switch1(config-line)#login
```

**请完成上述操作，然后从另一设备（同一子网）连上交换机来测试你的配置。**这是一道 CCNA 基础题目。

**VTYs (Virtual Teletype terminal)** 是路由器或交换机用于对其进行 Telnet 或安全 Telnet (SSH) 访问的**虚拟端口**。在你为其配置上一种认证方式之前，它们都是关闭的（最简单的方法是给它们加上一个口令，然后执行 `login` 命令）。你可以见到 0 到 4 端口、inclusive（包含）或 0 到 15 端口。要得知你有多少个可用的端口的一种方法是在编号 0 后面输入一个问号，或者使用 `show line` 命令，如下面的输出所示。

```
Router(config)#line vty ?
<1-15> Last Line number
Router#show line
  Tty Typ Tx/Rx      A Modem Roty    Acc0    AccI    Uses    Noise   Overruns   Int
*  0  CTY          - - - -       -       -      0       0      0/0
  1  AUX 9600/9600 - - - -       -       -      0       0      0/0
  2  VTY          - - - -       -       -      2       0      0/0
  3  VTY          - - - -       -       -      0       0      0/0
  4  VTY          - - - -       -       -      0       0      0/0
  5  VTY          - - - -       -       -      0       0      0/0
  6  VTY          - - - -       -       -      0       0      0/0
```

CTY 就是控制台线路，同时 VTY 线路用于 Telnet 连接，AUX 是指辅助端口。

为了获得更为安全的访问方式，你可以仅允许 SSH 连接进入交换机，这就是说流量会被加密。而要让 SSH 工作，你需要在交换机上允许安全性 IOS 镜像，如下面的输出那样。

```
Switch1(config-line)#transport input ssh
```

现在，Telnet 流量就不再被允许传入到 VTY 端口了。

请在交换机上配置一下这所有的命令。仅仅阅读它们无助于你在考试当天记起它们。

## 虚拟局域网，Virtual Local Area Networks, VLANs

就如同你已经看到的那样，交换机打破冲突域。更进一步，路由器打破广播域，这就是说现在的网络看起来像下面这样。

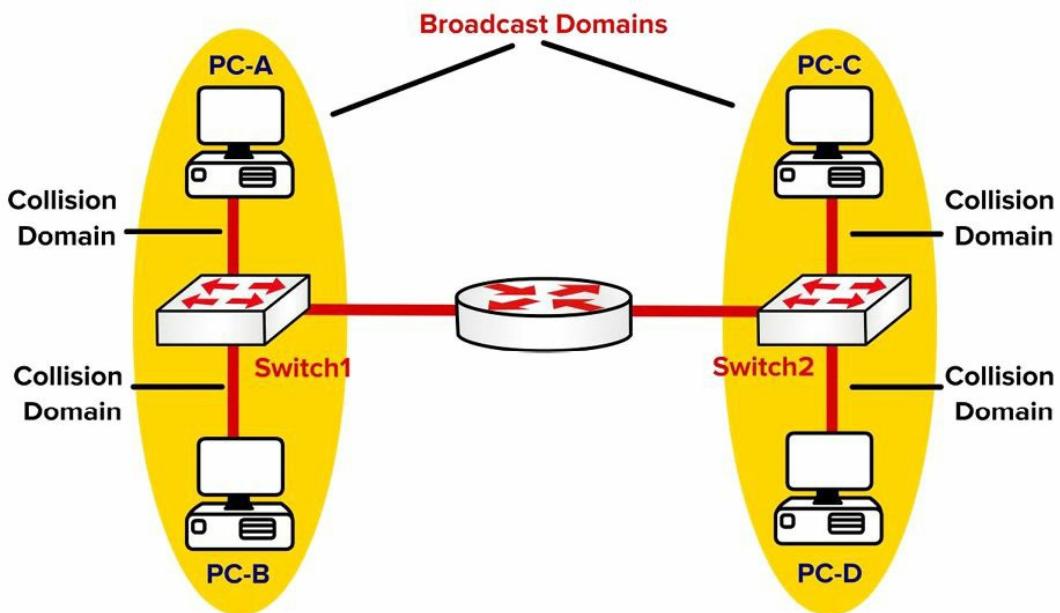


图 2.13 -- 路由器隔离广播域

在继续之前，让我们讨论一下局域网到底是什么。一个局域网本质上是一个广播域。图 2.13 中的网络上，如果 PC-A 发出一个广播包，PC-B 会接收到这个包，PC-C 和 PC-D 却不能。这是因为那台路由器打破了该广播域。现在你可以使用虚拟局域网将交换机的那些端口放入不的广播域，如下图所示。

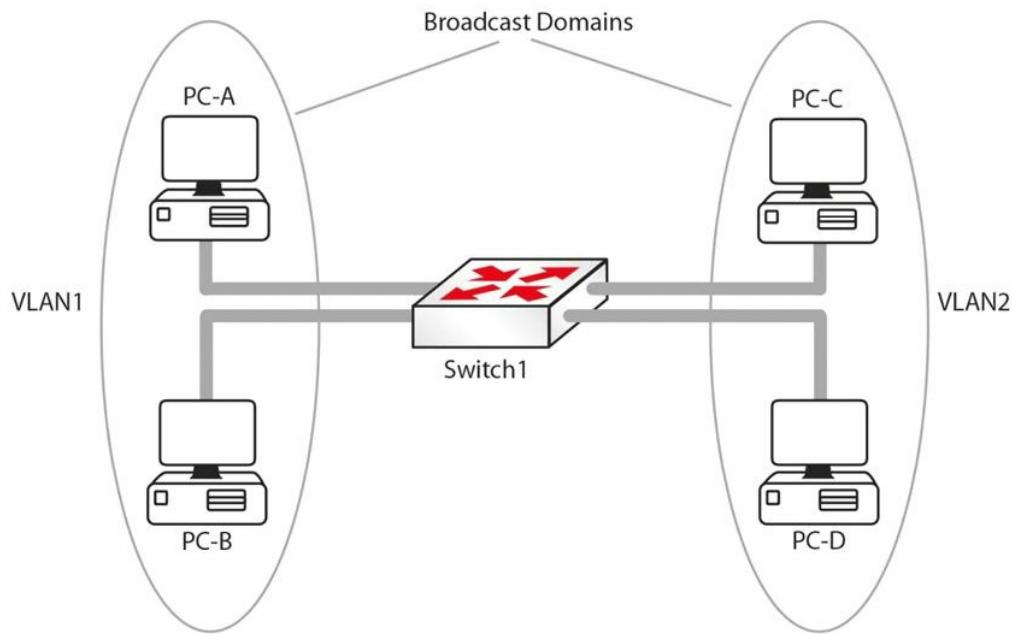


图 2.14 -- VLAN 下的广播域

图 2.14 中，该 2 层网络经由 VLANs 被划分为两个广播域。PC-A 发出的广播包为 PC-B 接收到，PC-C 和 PC-D 接收不到。如没有 VLANs，PC-C 和 PC-D 仍会收到 PC-A 发出的广播包。VLANs 的一些优点如下。

- 更小规模的设备组中的较少广播包令到网络更快
- 设备资源得以节省，因为它们只需处理少量的广播包
- 通过将设备保留在特定组别（或按特定功能分组）的一个广播域中，而提升安全性。这里的组别，可以是公司/机构的部门，或是某个安全级别等。比如开发部门或者测试室就应该与生产部门的设备分开。
- 带来在跨越任何尺度的地理位置上网络扩展的灵活性。比如，同一 VLAN 中某台 PC 在楼宇中的什么位置并不重要。它会以为自己与其它配置在同一 VLAN 中的机器在同样的网段上。图 2.15 中，VLAN 1 中的所有主机都能与其它主机通信，尽管它们不在同一楼层。对它们来说，VLAN 是透明的或不可见的。

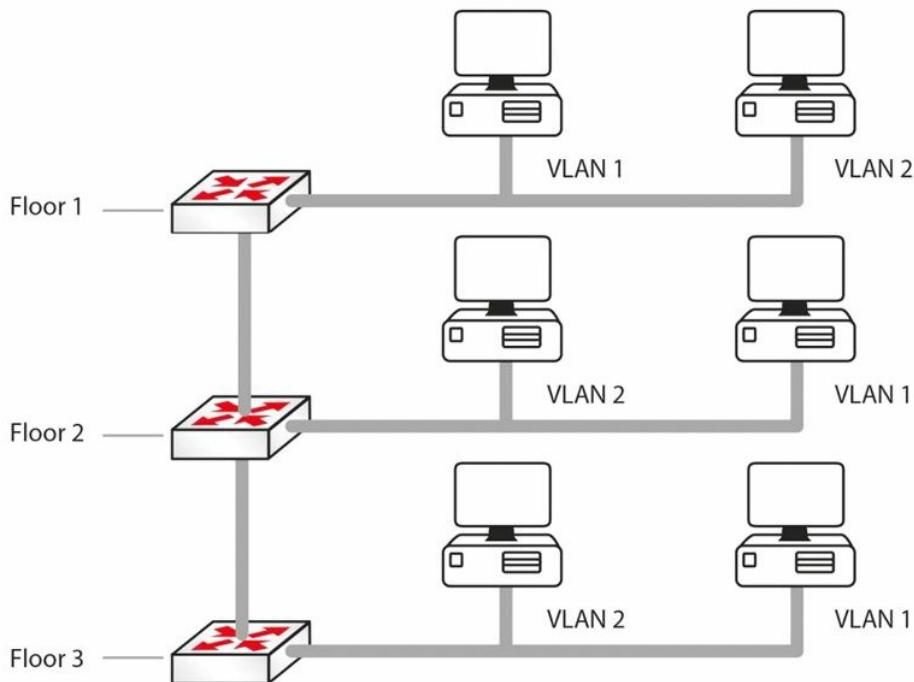


图 2.15 -- VLANs 消除局域网的物理边界

## VLAN 标记, VLAN Marking

虽然厂商在创建 VLANs 中采用其各自的方法，我们务必要小心处理一个涉及多厂商的 VLAN，以解决互操作性问题。比如思科开发的 **ISL** 标准，是通过增加一个 26 字节的头部，以及一个新的 4 字节尾部（trailer），的方式来封装原始帧。为解决兼容性问题，IEEE 开发了 **802.1Q** 标准，这是一个独立于厂商的方式，用以创建可互操作 VLANs。

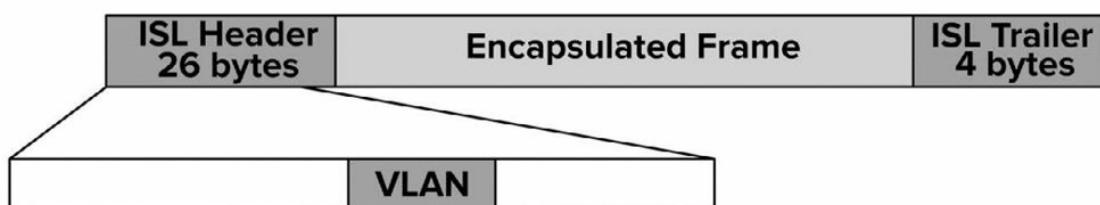


图 2.16 -- ISL 的标记方式

**802.1Q** 通常被称为“帧标记 (frame tagging)”，因为它将一个叫做标签的 32 位头部 (a 32-bit header, called a "tag")，插入到原始帧源地址后面，而不会对其它字段进行修改。紧邻源地址后两个字节，占据着一个注册以太网类型值 -- `0x8100`，它表明该帧包含了一个 **802.1Q** 头部。接着的 3 位表示 **802.1P** 用户优先级 (User Priority, UP) 字段，在服务质量 (Quality of Service, QoS) 技术中，用作服务类别 (Class of Service, CoS) 位。下一个字段是 1 位规范格式标识 (Canonical Format Indicator, CFI)，最后 12 位是 VLAN ID。所以在采行 802.1Q 标准时，我们总共能有 4096 个 VLANs。

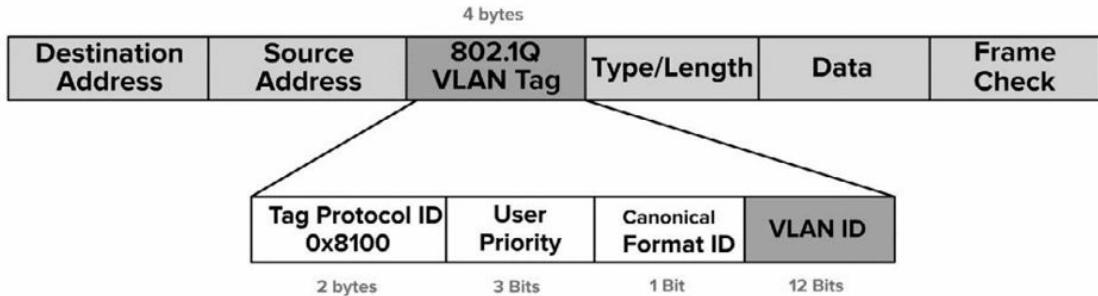


图 2.17 -- 802.1Q 的标记方式

传送来自多个 VLANs 数据的那个端口叫做干线 (trunk) 端口。trunk 端口可以使用 ISL 协议，也可以使用 802.1Q 协议。802.1Q 中的一个特别概念是“原生 VLAN (native VLAN)”。这是一种特别的 VLAN 类型，它上面的帧是没有打标签的。原生 VLAN 的目的是让交换机在某个接口上运行 802.1Q 中继（单一链上的多 VLANs），即便另一设备无法支持中继的情况下，原生 VLAN 上的流量仍能通过该链路。如交换机在一条中继链路上收到未打标签的流量，就会假定这些流量是属于原生 VLAN 上的。思科将 VLAN 1 作为默认的原生 VLAN。

## 加入虚拟局域网， VLAN Membership

有两种常用的将端口加入到 VLANs 的方式 -- 静态方式或动态方式。

通过静态 VLAN 指派或配置，交换机上的那些端口为管理员所配置在不同 VLANs 中，有关设备再连接到端口上。在某用需要搬往楼宇的其它部位时，就要求管理员改变交换机上的配置。默认情形下，所有交换机端口属于 VLAN 1。

动态指派方式令到设备可根据其 MAC 地址而加入到特定的 VLAN。该特性给予管理员在无需改变交换机配置的情况下，允许用户接入任何交换机或是在楼栋内搬动的灵活性。**通过运用一台虚拟局域网管理策略服务器 (a VIAN Management Policy Server, VMPS) 实现动态特性。**

Farai 指出“先是端口指派到 VLANs 中，随后设备插入到端口上”。

请注意，由于各个 VLAN 都是不同的广播域，这就带来以下问题。

- 默认情形下，一个 VLAN 中的主机是不能到达其它 VLAN 的
- VLAN 间通信需要一台三层设备（后面会讲到）
- 每个 VLAN 都需要它们自己的子网，比如，VLAN 1 -- 192.168.1.0/24，VLAN 2 -- 192.168.2.0/24
- 某 VLAN 中的所有主机应属于同一 VLAN

## VLAN 链路， VLAN Links

我们知道在一台交换机上可以有连接到多个 VLANs 的主机。那么在流量从一台主机前往另一主机时，发生了些什么呢？比如说，在图 2.15 中，当 VLAN 1 位于一楼的主机尝试与 VLAN 1 位于二楼的主机通信时，二层的那台主机是怎样知道该流量是属于哪个 VLAN 的呢？

我们知道交换机采用了一种叫做“帧标记 (frame tagging)”的方式，来将流量在不同 VLANs 上保持隔离。交换机把包含了 VLAN ID 的一个头部添加进帧中。在图 2.15 中，一楼交换机将会给来自 VLAN 2 的流量打上标记后，传给交换机 2，交换机 2 将会看到这个标记，从而得知该流量需要呆在 VLAN 2 中。**这样的流量只能在叫做中继链路的链路上通过。VLAN 1 通常被指定为原生 VLAN (native VLAN)，而原生 VLAN 上的流量不被标记。**关于原生 VLAN 的内容，后面会提到。

交换机端（在 CCNA 考试范围内）可分为以下三种。

- 接入端口，或接入链路，access links or ports

- 中继端口，或中链路，trunk links or ports
- 动态端口（很快就会学到这个）

## 接入链路, Access Links

被定义了作为接入链路的交换机端口，只能是唯一 VLAN 的一个成员。连接该接入链路的设备并不知晓任何其它 VLANs 的存在。在来自主机的帧进入到一个接入链路时，交换机就将一个标签加入到该帧中，在前往主机的帧从交换机接入链路出去时，交换机将该标记从帧中移除。接入链路用于连接主机，也可以用于连接路由器。

## 中继, Trunking

某个交换机端口通常既会连接网络上的某台主机，也会连接其它网络交换机、路由器或服务器。那么该链路就有可能需要传输多个 VLANs 上的流量。为实现这个目的，就需要区分每个帧都是来自于哪个 VLAN。这种区分方式就叫做“帧标记 (frame tagging)”，在经中继链路传输前，出原生 VLAN 的帧外，所有帧都已打过标签。帧中的标记包含了 VLAN ID 信息。在帧到达目的主机所在的那台交换机后，该标记被移除。

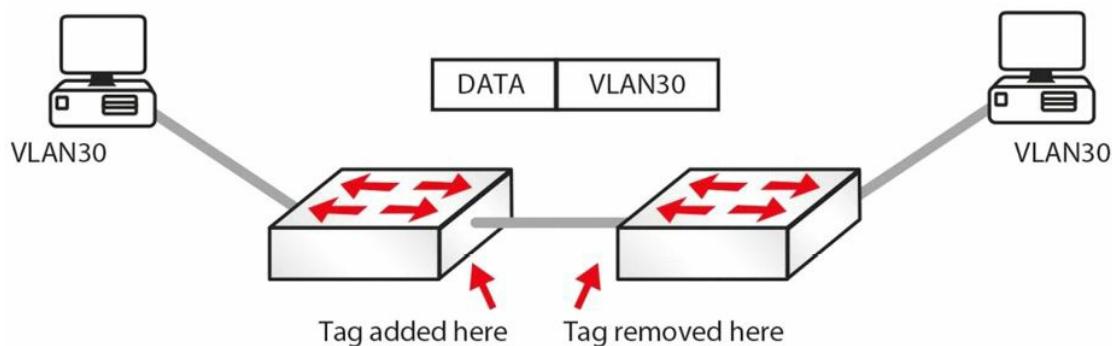


图 2.18 -- VLAN 标记法

VLAN 中继用于传输多个 VLAN 的数据。为将属于某个 VLAN 的帧与其它 VLANs 的帧有所区别，在中继链路上传输的所有帧都经特别标记过，这样目的交换机就知道该帧属于那个 VLAN。ISL 和 802.1Q 是用于确保这些 VLANs 在穿过交换机中继链路后，仍能完全分辨出来的两种主要方式。

交换机间链路 (Inter-Switch Link, ISL) 是思科专有的。尽管如此，CCNA 考试用到的型号是 2960 交换机，它只能识别 802.1Q。我们在这里提到 ISL 只是为了知识体系的完整性，以防你不得不配置老旧型号的交换机的情形。

Farai 指出 -- “所有新型号交换机默认都是采用 802.1Q，ISL 已被弃用。”

802.1Q 与 ISL 有很多的不同，最大区别是，802.1Q 可以支持最多 4096 个 VLANs, ISL 只能支持最多 1000 个。另一个大的区别是 802.1Q 中的原生 VLAN 概念。默认情形下，802.1Q 中来自所有 VLANs 的帧都被打上标签。此规则的唯一例外就是属于原生 VLAN 的帧，这些帧未被标记。

尽管如此，你要记住，在某个特定中继链路上，可以通过将某 VLAN 设为原生 VLAN，来指定其上面的帧不打标签。比如，在采用 802.1Q 时，为阻止对 VLAN 400 上的帧打标签，你需要将 VLAN 400 配置为某特定中继上的原生 VLAN。IEEE 802.1Q 原生 VLAN 配置在后面会详细介绍。

以下有关 802.1Q 特性的总结。

- 支持最多 4096 个 VLANs
- 采用帧内标记机制，修改原始帧
- 是由 IEEE 开发的开发标准协议
- 不对原生 VLAN 上的帧打标签；除此之外的所有帧都被标记

下面是一台交换机上的简短示例配置。我将 `switchport` 命令包括了进去，该命令告诉交换机将其某个端口作为二层端口，而不是三层。

```
Sw(config)#interface FastEthernet 0/1
Sw(config-if)#switchport
Sw(config-if)#switchport mode trunk
Sw(config-if)#switchport trunk encapsulation dot1q
Sw(config-if)#exit
```

当然，在 2960 系列交换机上，`encapsulation` 不被识别，因为它只有一种类型。**在将一台交换机与其它交换机连接时，你需要将其接口设置为中继接口，以令到 VLANs 都被标记上。** `switchport` 命令的作用同样如此。再次说明，我在这里提到这个，是因为在现实中，你可能要配置一台三层交换机，如我们死盯 2960 型号，你可能会感到迷惑，我们不要这个！

交换机上的某中继链路可以是下列五种模式之一。

- 开启 (On) 模式 -- 强制该端口进入永久中继模式。不管插入设备是否同意将它们之间的链路转换成中继链路，该端口都会成为一个中继端口。
- 关闭 (Off) 模式 -- 该链路不被作为中继链路使用，就算插入设备被设置成“中继”模式。
- 自动 (Auto) 模式 -- 该端口不情愿成为一条中继链路。在插入设备被设置为“开启”或“我要 (desirable)”模式时，链路就成为中继链路。当两端都被设置为“自动”模式时，链路就绝不会变成中继链路了，因为没有一方有转换成中继的意愿。
- 我要 (Desirable) 模式 -- 该端口积极尝试转换成中继链路。如另一设备被设置为“开启”、“自动”或“我要”模式，链路就会成为中继链路
- 没商量 (No-negotiate) 模式 -- 阻止端口经由协商成为中继连接。配置上此模式后，端口会强制进入接入模式或中继模式。

## 配置 VLANs, Configuring VLANs

现在你已对 VLANs 和中继链路有了了解，就让我们来配置一下图 2.19 中的网络吧。你需要将交换机配置为两个分别在 `fa0/1` 端口上的主机位于 `VLAN 5` 中，以及端口 `fa0/15` 之间的链路为中继链路。

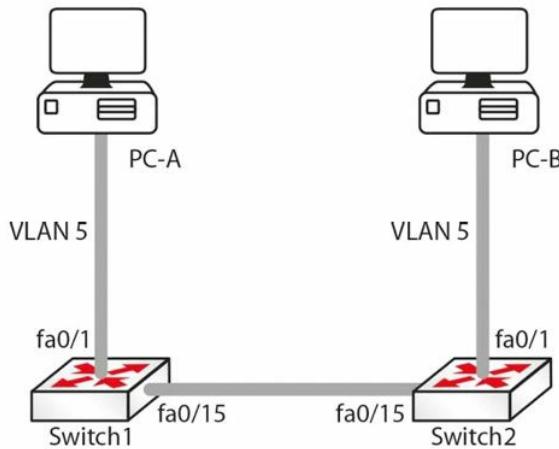


图 2.19 -- 测试网络

在将端口指派到 VLANs 之前，务必先要全局配置命令 `vlan <vlan#>` 创建出那个 VLAN。而此命令又会将你带入 VLAN 配置模式，在那里又可以为 VLANs 赋予一个描述性的名称。这里有一个示例。

```

Switch1(config)#vlan 5
Switch1(config-vlan)#name RnD
Switch2(config)vlan 5
Switch2(config-vlan)#name RnD

```

使用命令 `show vlan` 命令来查看交换上存在着哪些 VLANs。其输出与下面的相似。

```

Switch1#show vlan
VLAN      Name        Status      Ports
----      ----        -----      -----
1         default     active      Fa0/1, Fa0/2, Fa0/3, Fa0/4 Fa0/5, Fa0/6, Fa0/7, Fa0/8 Fa0/9,
          ...          ...
          [Truncated Output]
...
5         RnD         active
...
[Truncated Output]

```

我们在通过使用接口配置命令 `switchport access vlan [vlan#]`，将端口 `fa0/1` 加入到 `VLAN 5` 中去。

```

Switch1(config)#int fa0/1
Switch1(config-if)#switchport access vlan 5
Switch2(config)#int fa0/1
Switch2(config-if)#switchport access vlan 5

```

在形如 3560 这样的三层交换机上，在将某个端口加入到一个 VLAN 前，你务必要使用命令 `switchport mode access` 将端口手动设置为接入模式。现在我们来看看 `show vlan` 命令的输出。

```

Switch1#show vlan
VLAN      Name        Status      Ports
----      ----        -----      -----
1         default     active      Fa0/2, Fa0/3, Fa0/4, Fa0/5, Fa0/6, Fa0/7,
          Fa0/8, Fa0/9, Fa0/10, Fa0/11, Fa0/12, Fa0/13,
          Fa0/14, Fa0/15, Fa0/16, Fa0/17, Fa0/18
...
[Truncated Output]
...
5         RnD         active      Fa0/1
...
[Truncated Output]

```

注意 `fa0/1` 现在被指派给了 `VLAN 5`。让我们来将两台交换机的 `fa0/15` 接口配置为中继链路。这里要注意的是 3550 型号交换机端口的默认模式是我要模式（`desirable`，3560 型号的是自动模式）。动态中继协议（Dynamic Trunk Protocol, DTP）会导致两台交换机上的 `fa0/15` 接口成为 ISL 中继链路。下一节课会学 DTP 的内容，但这里会简要提到 DTP 的一些东西。这种情况可用 `show interface trunk` 命令看到。

```

Switch1#show interface trunk
Port      Mode        Encapsulation  Status      Native vlan
Fa0/15   desirable   n-isl          trunking   1

```

请注意，其模式为我要（`desirable`），封装方式是 ISL（`n` 代表 `negotiated`，协商出的）。以下输出演示了配置中继为 ISL 方式的方法。

```

Switch1(config)#interface fa0/15
Switch1(config-if)#switchport trunk encapsulation isl
Switch1(config-if)#switchport mode trunk
Switch2(config)#interface fa0/15
Switch2(config-if)#switchport trunk encapsulation isl
Switch2(config-if)#switchport mode trunk

```

`switchport trunk encapsulation` 命令设置端口的中继协议，而命令 `switchport mode trunk` 命令则是将端口设置为中继工作方式。现在的 `show interface trunk` 命令输出会是下面这样。

```

Switch2#show interface trunk
Port      Mode      Encapsulation      Status      Native vlan
Fa0/15    on        isl                trunking    1

```

取代 N-ISL 的是 ISL。这是因为此次的协议是在接口上配置的，而不是协商出的。

**重要提示：**在将交换机某端口设置为中继模式前，要先设置其中继封装方式。而这个规则又不适用于 2960 交换机（当前 CCNA 大纲中用到的型号），2960 交换机只使用 `dot1q`（`802.1Q` 的另一种叫法）封装。因此，2960 交换机上的 `switchport trunk encapsulation` 命令不工作。

与此类似，你可将交换机端口配置为 `802.1Q` 而不是 ISL，如下面的输出那样。

```

Switch1(config)#interface fa0/15
Switch1(config-if)#switchport trunk encapsulation dot1q
Switch1(config-if)#switchport mode trunk
Switch2(config)#interface fa0/15
Switch2(config-if)#switchport trunk encapsulation dot1q
Switch2(config-if)#switchport mode trunk

```

命令 `show interface trunk` 命令的输出又成了这样。

```

Switch2#show interface trunk
Port      Mode      Encapsulation      Status      Native vlan
Fa0/15    on        802.1q            trunking    1

```

请注意，原生 VLAN 是 1。这正是一个 `802.1Q` 中继上的默认原生 VLAN，同时可使用 `switchport trunk native vlan <vlan#>` 命令进行修改。**中继链路上的两个接口原生 VLAN 必须匹配**。这条命令是 CCNA 大纲的一部分，也被作为一中安全手段。

**重要提示：**交换机能存储所有 VLAN 的信息，在重启后也还在。如你打算交换机以空白配置启动，就需要在交换机上运行 `delete vlan.dat` 命令，如下面的输出所示。这仅适用于真实交换机，在诸如 Packet Tracer 等交换机模拟器是做不到的。

```

SwitchA#dir flash:
Directory of flash:/
  1  -rw-    3058048      <no date>  c2960-i6q4l2-mz.121-22.EA4.bin
  2  -rw-     676        <no date>  vlan.dat
64016384 bytes total (60957660 bytes free)
SwitchA#
SwitchA#delete vlan.dat
Delete filename [vlan.dat]?
Delete flash:/vlan.dat? [confirm]
SwitchA#dir flash:
Directory of flash:/
  1  -rw-    3058048      <no date>  c2960-i6q4l2-mz.121-22.EA4.bin
64016384 bytes total (60958336 bytes free)
SwitchA#

```

## 交换故障排除基础，Basic Switching Troubleshooting

理论上，一旦设备配置好并运行起来后，它就会一直运行下去，不过下面这些情况是常有的事，比如你要在某个不是你亲自配置的网络上做事，或者你会在轮班制工当中支持许多并不熟悉的网络，这些网络又在你不当班的时候被其他人改动过，这些状况都会导致出现多多少少的问题。我建议你在完成一些实验后，在回头看看这部分内容。

### 常见的交换机问题，Common Switch Issues

#### 无法远程登录到交换机，Can't Telnet to Switch

首先要问的是 Telnet 曾正常运行过吗？如曾正常运行过，现在却不行了，那就是有人对交换机进行了改动、重启过交换机，从而导致配置丢失，或者是网络上的某台设备阻止了 Telnet 流量。

```

Switch#telnet 192.168.1.1
Trying 192.168.1.1 ...Open
[Connection to 192.168.1.1 closed by foreign host]

```

要检查的头一件事就是交换机上的 Telnet 是否已被确实开启（见下面的输出）。网络的 80% 错误都是由于唐突或疏忽造成的，所以请不要信誓旦旦，要亲历亲为，别去相信其他人的言词。

一个简单的 `show running-config` 命令就可以将交换机的配置列出。在 `vty` 线路下，你将看到 Telnet 是否有被打开。注意你需要在 `vty` 线路下有 `login` 或者 `login local` (或者配置了 AAA，而 AAA 配置超出了 CCNA 考试范围) 命令，以及 `password` 命令。如下面所示。

```

line vty 0 4
password cisco
login
line vty 5 15
password cisco
login

```

`login local` 命令告诉交换机或路由器去查找配置在其上的用户名和口令，如下面输出的那样。

```

Switch1#sh run
Building configuration...
Current configuration : 1091 bytes!
version 12.1
hostname Switch1
username david privilege 1 password 0 football
line vty 0 4
password cisco
login local
line vty 5 15
password cisco
login local
...
[Truncated Output]

```

### Ping 不通交换机， Can't Ping the Switch

首先要弄清楚那人要 ping 交换机的原因。如你真要 ping 交换机，那么得要给交换机配置上一个 IP 地址；此外，交换机也要知道如何将流量送出（要有默认网关）。

### 不能经由交换机 ping 通其它设备， Can't Ping through the Switch

如出现经由交换机 ping 不通的情况，那就要确保那两台终端设备位于同一 VLAN 中。每个 VLAN 被看成一个网络，因此各个 VLAN 都要有与其它 VLAN 所不同的地址范围。必须要有一台路由器，以实现一个 VLAN 与其它 VLAN 之间连通。

### 接口故障， Interface Issues

默认情况下，所有路由器接口都是对流量关闭的，而交换机接口是开启的。如你发现交换机接口处于管理性关闭状态，可以通过执行接口级命令 no shutdown 来开启它。

```

Switch1(config)#int FastEthernet0/3
Switch1(config-if)#no shut

```

**二层接口可被设置成三种模式：中继、接入，或动态模式。中继模式下，交换机可与其它交换机或服务器连接。**而接入模式用于连接终端设备，比如一台 PC 或笔记本计算机。动态模式令到交换机去探测采用何种设置。

在形如 3550 型号交换机平台上，默认设置通是动态我要模式（ dynamic desirable ），你需要在 [Cisco.com](http://Cisco.com) 上去查看你的交换机型号的设置以及发行注记。**CCNA 考试中，你将被要求配置一台 2960 型号交换机。**此型号的交换机在除非你硬性设置接口为中继或接入模式的情况下，会动态选择工作模式。

```

Switch1#show interfaces switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto

```

默认设置可以方便地进行更改，如下面的输出这样。

```

Switch1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch1(config)#int FastEthernet0/1
Switch1(config-if)#switchport mode ?
  access      Set trunking mode to ACCESS unconditionally
  dynamic     Set trunking mode to dynamically negotiate access or trunk mode
  trunk       Set trunking mode to TRUNK unconditionally

Switch1(config-if)#switchport mode trunk
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down

Switch1(config-if)#^Z
Switch1#
%SYS-5-CONFIG_I: Configured from console by console
Switch1#show interfaces switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk

```

### 更多有关接口的故障， More Interface Issues

交换机端口的默认设置是双工自动侦测（ auto-detect duplex ）以及速率自动侦测（ auto-detect speed ）。如你将一台 10Mbps 的设备插入到以半双工方式运行的交换机（现在已经很难找到这样的交换机了）上，该端口就会探测到插入的设备并运作起来。然而并不是任何时候都这样的，所以一般建议将交换机端口的双工方式及速率硬性设置，如下面的输出那样。

```

Switch1#show interfaces switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto

Switch1#show interface FastEthernet0/2
FastEthernet0/2 is up, line protocol is up (connected)
  Hardware is Lance, address is 0030.f252.3402 (bia 0030.f252.3402)
  BW 100000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
  Keepalive set (10 sec)
  Full-duplex, 100Mb/s

Switch1(config)#int fast 0/2
Switch1(config-if)#duplex ?
  auto   Enable AUTO duplex configuration
  full   Force full-duplex operation
  half   Force half-duplex operation
Switch1(config-if)#speed ?
  10    Force 10Mbps operation
  100   Force 100Mbps operation
  auto  Enable AUTO speed configuration

```

双工模式不匹配的一些迹象（除开错误消息外）有接口上的输入错误以及 CRC 错误，如下面的输出所示。请同时看看 ICND1 章节的第 15 天的一层和二层故障排除部分。

```

Switch#show interface f0/1
FastEthernet0/1 is down, line protocol is down (disabled)
    Hardware is Lance, address is 0030.a388.8401 (bia 0030.a388.8401)
    BW 100000 Kbit, DLY 1000 usec,
        reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA, loopback not set
    Keepalive set (10 sec)
    Half-duplex, 100Mb/s
    input flow-control is off, output flow-control is off
    ARP type: ARPA, ARP Timeout 04:00:00
    Last input 00:00:08, output 00:00:05, output hang never
    Last clearing of "show interface" counters never
    Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
    Queueing strategy: fifo
    Output queue :0/40 (size/max)
    5 minute input rate 0 bits/sec, 0 packets/sec
    5 minute output rate 0 bits/sec, 0 packets/sec
        956 packets input, 193351 bytes, 0 no buffer
        Received 956 broadcasts, 0 runts, 0 giants, 0 throttles
        755 input errors, 739 CRC, 0 frame, 0 overrun, 0 ignored, 0 abort
        0 watchdog, 0 multicast, 0 pause input
        0 input packets with dribble condition detected
        2357 packets output, 263570 bytes, 0 underruns
        0 output errors, 0 collisions, 10 interface resets
        0 babbles, 0 late collision, 0 deferred
        0 lost carrier, 0 no carrier
        0 output buffer failures, 0 output buffers swapped out

```

### 硬件故障, Hardware Issues

和其它电子设备一样，交换机端口也会出现失效或不能全时正常运行现象，而时而正常时而故障的情况是更难于处理的。工程师经常通过将一台已知正常的设备插入到交换机的另一端口，来测试故障的接口。你也可以跳转（bounce）某端口，就是在该端口是先用 `shutdown` 命令关闭，接着用 `no shutdown` 命令开启。更换网线也是一个常见的处理步骤。图 2.20 给出了一些其它的交换机故障和处理方法。

请查阅你的交换机的文档，因为根据系统和端口 LEDs 的不同，每个端口会有闪烁的或是常亮的红色、琥珀色或者绿色的指示灯，表示功能正常或是端口、系统故障。

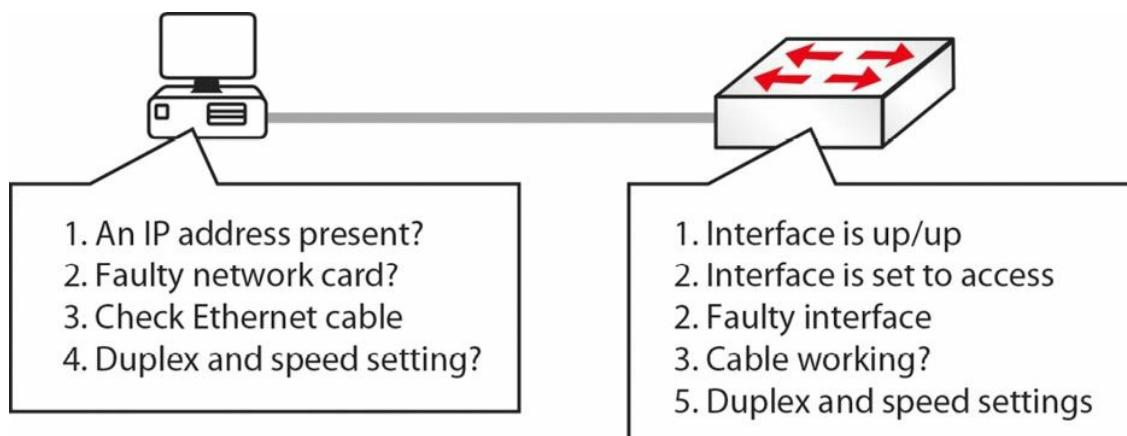


图 2.20 -- 常见交换机故障及解决方法

### VLAN 分配故障, VLAN Assignment Issues

小环境下的网络管理起来相对容易，因为只需部署少数特性，就能满足业务需求。但在企业环境中，你不会去使小型工作组交换机或是家庭办公设备的（small workgroup switches and SOHO device）。相反，你会用到高端设备，它们提供了诸多高级/复杂功能，具有流量优化能力。

此种环境下一种可能会配置到的特别特性，就是采用 VLANs 技术将不同网络区域进行逻辑隔离。在你遇到与某个 VLAN 有关的配置问题时，故障就会出现，这种故障可能会是难于处理的。一种处理方法就是去分析交换机的整个配置，并尝试找到问题所在。

VLAN 相关故障，通常是经由观察网络主机之间连通性（比如某用户不能 ping 通服务器）缺失发现的，就算一层运行无问题。**有关 VLAN 故障的一个重要特征就是不会对网络的性能造成影响。**如你配错了一个 VLAN，连接就直接不通，尤其是在考虑 VLANs 通常是用作隔离 IP 子网的情况下，只有处于同一 VLAN 的设备才能各自通信。

在排除 VLAN 故障时，首先要做的是查看设计阶段完成的网络文档和逻辑图表（网络拓扑图），如此你才能得知各个 VLAN 跨越的区域，以及相应设备和各交换机的端口情况。接着就要去检查每台交换机的配置，并通过将其与存档方案进行比较，以尝试找出问题。

你还要对 IP 地址分配方案进行查验。如你采用的是设备静态 IP 地址分配方式，你可能打算回去检查那台设备，确保其有正确的 IP 地址和子网掩码。如在 IP 分址方案上存在问题，像是将设备配置到错误的网络上，或是子网掩码错误/默认网关错误的话，即使交换机的 VLAN 配置无误，你也会遇到连通性问题。

还要确保交换机的中继配置正确。当存在多台交换机时，通常会有交换机间的上行链路，这些上行链路承载了网络上的 VLANs。这些交换机间链路常被配置为中继链路，以实现穿越多个 VLANs 的通信。如某 VLAN 中的数据需要从一台交换机发往另一交换机，那么它就必须是该中继链路组的成员，因此，你还要确保中继链路两端交换机配置正确。

最后，如你要将某设备迁往另一个 VLAN，你务必要同时更改交换机及客户端设备，因为在迁移后，客户端设备会有一个不同子网的不同 IP 地址。

如你有遵循这些 VLAN 故障排除方法，在你首次插入设备及 VLAN 间迁移时，肯定能得到预期的连通性。

## 第二天的问题

1. Switches contain a memory chip known as an \_\_\_\_\_, which builds a table listing which device is plugged into which port.
2. The \_\_\_\_\_ - \_\_\_\_\_ command displays a list of which MAC addresses are connected to which ports.
3. Which two commands add an IP address to the VLAN?
4. Which commands will enable Telnet and add a password to the switch Telnet lines?
5. How do you permit only SSH traffic into your Telnet lines?
6. What is the most likely cause of Telnet to another switch not working?
7. Switches remember all VLAN info, even when reloaded. True or False?
8. A switch interface can be in which of three modes?
9. How do you set a switch to be in a specific mode?
10. Which commands will change the switch duplex mode and speed?

## 第二天问题答案

1. ASIC.
2. `show mac-address-table`
3. The `interface vlan x` command and the `ip address x.x.x.x` command. 4.

```
Switch1(config)#line vty 0 15
Switch1(config-line)#password cisco
Switch1(config-line)#login
```

4. Use the `Switch1(config-line)#transport input ssh` command.
5. The authentication method is not defined on another switch.
6. True.
7. Trunk, access, or dynamic mode.
8. Apply the `switchport mode <mode>` command in Interface Configuration mode.
9. The `duplex` and `speed` commands.

## 第二天实验

### 交换机概念实验

请登入到一台思科交换机，并输入那些本单元课程中解释到的命令。包括：

- 在不同交换机端口上配置不同的端口速率/自动协商速率
- 使用 `show running-config` 和 `show interface` 命令，验证这些端口参数
- 执行一下 `show version` 命令，来查看硬件信息以及 IOS 版本
- 查看交换机 MAC 地址表
- 给 VTY 线路配置一个口令
- 定义出一些 VLANs 并为其指派名称
- 将一个 VLAN 指派到一个配置为接入模式的端口上
- 将某个端口配置为中继端口（ISL 以及 802.1Q），并将一些 VLANs 指派到该中继链路
- 使用 `show vlan` 命令验证 VLAN 配置
- 使用 `show interface switchport` 命令和 `show interface trunk` 命令，验证接口中继工作状态及 VLAN 配置
- 删除 `vlan.dat` 文件

## 第3天 中继、DTP 及 VLAN 间路由

### Trunking, DTP, and Inter-VLAN Routing

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第三天的任务

- 阅读今天的课文
- 回顾昨天的课程
- 完成今天的实验
- 阅读 ICND1 记诵指南

在最小规模的那些办公室里，你才会遇到仅使用一台交换机的情况，但是网络基础设施通常是由多台交换机构成的。多台交换机带来了其配置上的挑战，要求你对中继及其有关的问题有深入理解。思科将多台交换机的安装和故障排除，视为一个 CCNA 级别的基础主题。

今天你将学到。

- 中继工作方式，trunking
- 动态中继协议（Dynamic Trunking Protocol, DTP）
- VLAN 间路由

本单元覆盖 ICND1 考试大纲的以下要求。

- 在思科交换机上配置并验证中继
- DTP
- 自动协商
- VLAN 间路由的配置和验证（单臂路由，router-on-a-stick）
  - 子接口，subinterfaces
  - 上行路由，upstream routing
  - 封装，encapsulation
- 配置交换机虚拟接口，configure SVI(Switch Virtual Interface) Interfaces

## 配置并验中继链路

中继是一个可以承载多种流量类型，每种流量类型都用一个独特的 VLAN ID 做了标记，的交换机端口。在数据经由中继端口，或者说中继链路得以交换时，就被那个外出交换机中继端口（the egress switch trunk port）打上了标签（或者叫进行了着色），这样做了过后，接收交换机就能够分辨出数据是属于哪个特定的

VLAN 了。在接收交换机的进入端口（the receiving switch ingress port）上，标签会被移除，然后数据就被转发给相应的目的设备。

在思科 IOS Catalyst 交换机上部署 VLAN 中继的第一项配置任务，就是把需要的接口配置为一个二层交换端口。这是通过执行 `switchport` 接口配置命令完成的。

**注意：**该命令只在兼容三层或多层交换机上需要。在诸如 Catalyst 2960 系列这样的二层交换机上并不适用。那些支持命令 `ip routing` 的交换机才被认为是兼容三层的交换机。

接着就是要中继链路所要使用的封装协议。这是通过执行 `switchport trunk encapsulation [option]` 命令完成的。此命令可用的选项有下面这些。

```
Switch(config)#interface FastEthernet1/1
Switch (config-if)#switchport trunk encapsulation ?
dot1q - Interface uses only 802.1q trunking encapsulation when trunking
isl - Interface uses only ISL trunking encapsulation when trunking
negotiate - Device will negotiate trunking encapsulation with peer on interface
```

关键字 `[dot1q]` 强制该交换机端口使用 IEEE 802.1Q 封装方式。关键字 `[isl]` 强制该交换机端口使用思科 ISL 封装方式。而 `[negotiate]` 关键字则指明说在动态交换机间链路协议（Dynamic Inter-Switch Link Protocol, DISL）及动态中继协议（Dynamic Trunking Protocol, DTP）无法就封装格式达成一致时，ISL 作为备选格式。DISL 简化了两台互联的快速以太网设备间 ISL 中继链路的建立。在 DISL 协议下，只需链路的一端需要配置为中继端口，因此而将 VLAN 中继配置过程大大简化。

DTP 是一个思科专有的点对点协议（point-to-point protocol），它在两台交换机间协商建立起某种常见中继模式。DTP 会在稍后专门讲到。下面的输出演示了如何将某交换机端口配置为在建立起一条中继链路是采用 IEEE 802.1Q 封装方式。

```
Switch (config)#interface FastEthernet1/1
Switch (config-if)#switchport
Switch (config-if)#switchport trunk encapsulation dot1q
```

此配置可通过命令 `show interfaces [name] switchport` 进行验证，如下列输出所示。

```
Switch#show interfaces FastEthernet1/1 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic desirable
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
...
[Truncated Output]
```

中继端口配置的第三步，就是部署配置，以确保该端口已被指定为中继端口。可以通过两种方式完成。

- 手动（静态）地完成中继配置
- 使用动态中继协议（Dynamic Trunking Protocol, DTP）

## 手动（静态）中继配置

手动配置一个中继，是通过在所需要的交换机上，执行接口配置命令 `switchport mode trunk` 完成的。此命令将该端口强制变成永久（静态）中继模式。下面的配置输出演示了如何将一个端口静态地配为中继端口。

```
VTP-Server(config)#interface FastEthernet0/1
VTP-Server(config-if)#switchport
VTP-Server(config-if)#switchport trunk encapsulation dot1q
VTP-Server(config-if)#switchport mode trunk
VTP-Server(config-if)#exit
VTP-Server(config)#
```

如你使用的是一台低端交换机，就大可以忽略 `switchport` 命令，上面的输出是来自一台 Catalyst 6000 系列交换机。此配置可通过 `show interfaces [name] switchport` 命令予以验，如下面的输出所示。

```
VTP-Server#show interfaces FastEthernet0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
...
[Truncated Output]
```

尽管中继链路的静态配置强制交换机建立一条中继，动态 ISL 和动态中继协议（DTP）数据包仍能从该接口发出。这样做了后，一条静态配置的中继链路就可以与相邻的使用了 DTP 的交换机，建立起中继关系，接着的小节将会讲到。经由 `show interfaces [name] switchport` 命令的输出，便可验证这点。如下面的输出所示。

```
VTP-Server#show interfaces FastEthernet0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
...
[Truncated Output]
```

在上面的输出中，粗体文字表明尽管中继链路是静态配置的，端口仍然在发出 DTP 和 DISL 数据包。在一些场合，此特性被看成是不受欢迎的。因此，通过执行接口配置命令 `switchport nonegotiate`，在静态配置作为中继链路的端口上关闭 DISL 及 DTP 数据包发送，被认为是一种良好实践，具体操作如以下的输出。

```
VTP-Server(config)#interface FastEthernet0/1
VTP-Server(config-if)#switchport
VTP-Server(config-if)#switchport trunk encapsulation dot1q
VTP-Server(config-if)#switchport mode trunk
VTP-Server(config-if)#switchport nonegotiate
VTP-Server(config-if)#exit
VTP-Server(config)#
```

再一次，`show interfaces [name] switchport` 命令可被用作验证配置，像下面这样。

```
VTP-Server#show interfaces FastEthernet0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Off
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
...
[Truncated Output]
```

## 动态中继协议，Dynamic Trunking Protocol, DTP

DTP 是一个在两台交换机之间协商出一种常见中继模式的，思科专有的点对点协议（a Cisco proprietary point-to-point protocol）。这种动态协商不止于何种中继模式，还包括中继的封装方式。根据其平台的不同，两种交换机端口所能使用的 DTP 模式如下。

- 动态我要模式，dynamic desirable
- 动态自动模式，dynamic auto

在两台相交换机上使用 DTP 时，如交换机端口默认为动态我要状态，端口就会积极尝试变为中继端口。而如果交换机端口默认为动态自动状态，端口仅会在相邻交换机被设置为动态我要模式时，才反转为中继端口。

图 3.1 演示所有 DTP 模式组合，在两台思科 Catalyst 交换机间，这些组合有的能建立起中继链路，也有的不能建立（在这里的组合都能建立中继链路；请查看图 3.2 之后的说明）。

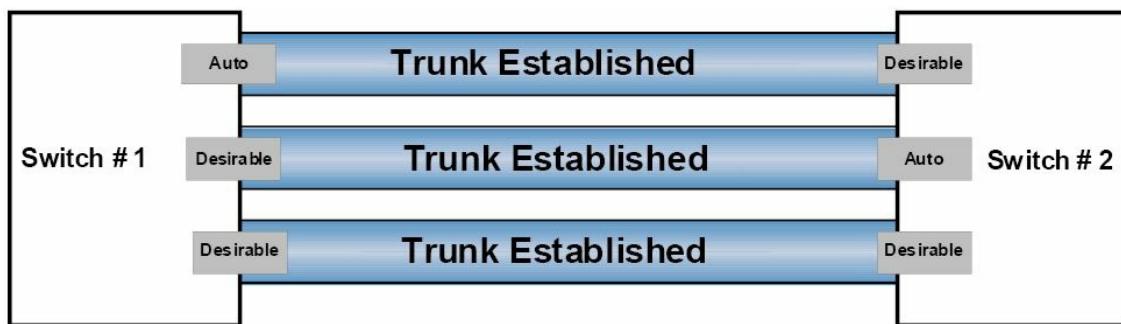


图 3.1 -- DTP 模式组合

图 3.2 示出了将会在两台相邻交换机间成功建立中继链路的有效组合 -- 一端是 DTP 另一端静态配置为中继端口。

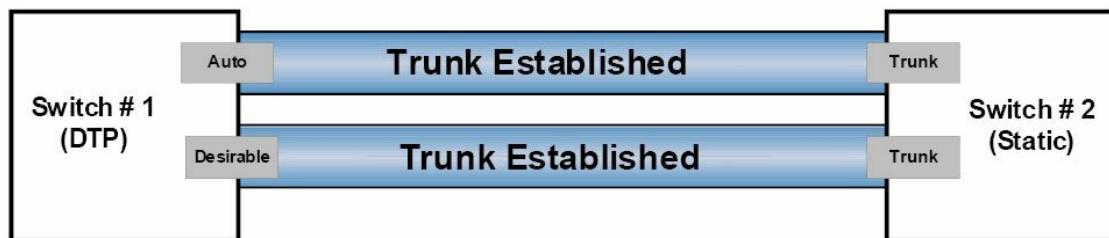


图 3.2 -- DTP 模式组合，第二部分

**注意：**如果两台交换机都设置为动态自动模式，它们是无法建立起中继链路的，知道这一点很重要。这是因为，不同于动态我要模式，动态自动模式是一种消极模式，它等待另一端发起中继建立。因此，在两个消极端口连接时，既不会发起中继建立，同时中继链路也绝不会形成。与此类，一个静态配置的交换机端口同时配置了 `switchport nonegotiate` 命令的话，它绝不会与相邻的使用 DTP 的交换机形成中继，因为这会阻止 DISL 及 DTP 数据包从那个端口发出。

在交换的局域网中应用 DTP 时，`show dtp [interface <name>]` 命令就可用来显示交换机的全局 DTP 信息以及特定接口的 DTP 信息。下面的输出给出了 `show dtp` 命令打印出的信息。

```
VTP-Server#show dtp
Global DTP information
  Sending DTP Hello packets every 30 seconds
  Dynamic Trunk timeout is 300 seconds
  4 interfaces using DTP
```

从上面的输出可以看出，交换机每 30 秒就发出一个 DTP 数据包。而 DTP 超时被设置为 300 秒（5 分钟），当前有 4 个接口正使用着 DTP。命令 `show dtp interface [name]` 会打印出特定接口的 DTP 信息，这些信息中包括了接口的类型（中继或接入）、端口当前的 DTP 配置情况、中继的封装方式，以及 DTP 数据包统计信息，如下面的输出所示。

```
VTP-Server#show dtp interface FastEthernet0/1
DTP information for FastEthernet0/1:
  TOS/TAS/TNS:                      TRUNK/ON/TRUNK
  TOT/TAT/TNT:                      802.1Q/802.1Q/802.1Q
  Neighbor address 1:                000000000000
  Neighbor address 2:                000000000000
  Hello timer expiration (sec/state): 7/RUNNING
  Access timer expiration (sec/state): never/STOPPED
  Negotiation timer expiration (sec/state): never/STOPPED
  Multidrop timer expiration (sec/state): never/STOPPED
  FSM state:                        S6:TRUNK
  # times multi & trunk:          0
  Enabled:                           yes
  In STP:                            no
  Statistics
  -----
  0 packets received (0 good)
  0 packets dropped
    0 nonegotiate, 0 bad version, 0 domain mismatches, 0 bad TLVs, 0 other
  764 packets output (764 good)
    764 native, 0 software encapsulation, 0 hardware native
  0 output errors
  0 trunk timeouts
  2 link ups, last link up on Mon Mar 01 1993, 00:00:22
  1 link downs, last link down on Mon Mar 01 1993, 00:00:20
```

## IEEE 802.1Q 原生 VLAN

昨天的课程中，你学到了 `802.1Q`，或是 VLAN 标记法，在除了原生 VLAN 的帧外的所有帧中，插入一个标签。IEEE 定义了原生 VLAN，以提供给不能明白 VLAN 标签的，原有的 `802.3` 端口以连通性。

默认情况下，`802.1Q` 中继将 `VLAN 1` 作为原生 VLAN。执行命令 `show interfaces [name] switchport` 或命令 `show interfaces trunk`，就可查看到默认原生 VLAN 是哪一个，如下面的输出所示。

```
VTP-Server#show interfaces FastEthernet0/1 switchport
Name: Fa0/1
Switchport: EnabledAdministrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
...
[Truncated Output]
```

交换机使用 VLAN 1 来承载一些特定的协议流量，比如思科发现协议（Cisco Discovery Protocol, CDP）、VLAN 中继协议（VLAN Trunking Protocol, VTP）、端口聚合协议（Port Aggregation Protocol, PAgP），以及动态中继协议（Dynamic Trunking Protocol, DTP）等的协议信息。CDP 和 PAgP 将在今天的课程后面部分详细讨论。尽管默认原生 VLAN 总是 VLAN 1，它是可以手动修改为任何不在保留 VLANs 中的任何有效 VLAN 编号的。

但是，**中继链路两端上的原生 VLAN 必须一致**，记住这点是很重要。如出现了原生 VLAN 不匹配，生成树协议（Spanning Tree Protocol, STP）就把该端口置为端口 VLAN ID (port VLAN ID, PVID) 不一致状态，且不会转发该链路。此外，CDPv2 也会在交换机间传送原生 VLAN 信息，而在出现原生 VLAN 不匹配后，将会在交换机控制台上打印错误消息。通过对所需的 802.1Q 中继链路，执行接口配置命令 `switchport trunk native vlan [number]` 可以修改其默认原生 VLAN。如下面的输出所示。

```
VTP-Server(config)#interface FastEthernet0/1
VTP-Server(config-if)#switchport trunk native vlan ?
<1-4094>    VLAN ID of the native VLAN when this port is in trunking mode
```

## VLAN 间路由，Inter-VLAN Routing

默认情况下，尽管 VLANs 能够跨越整个的二层交换网络，一个 VLAN 中的主机却是不能直接和其它 VLAN 中的主机直接通信的。为实现这个目的，必须对不同 VLANs 间的流量进行路由。这就叫做 VLAN 间路由。交换局域网（switched LANs）中的 VLAN 间路由有三种实现方式，下面有分别列出，这三种方式及其各自的优势和劣势，接下来的部分会详细介绍。

- 采用物理的路由器接口的 VLAN 间路由, Inter-VLAN routing using physical router interfaces
- 采用路由器子接口的 VLAN 间路由, Inter-VLAN routing using router subinterfaces
- 采用交换机虚拟接口的 VLAN 间路由, Inter-VLAN routing using switched virtual interfaces

### 采用物理的路由器接口的 VLAN 间路由

为实现 VLAN 间路由通信的第一种方式，需要用到带有多个接口的路由器，来作为每个单独配置 VLAN 的网关。此时路由器就能够使用这些物理的 LAN 接口，将接收自一个 VLAN 的数据包，路由到其它 VLAN 上。此种方式如图 3.3 所示。

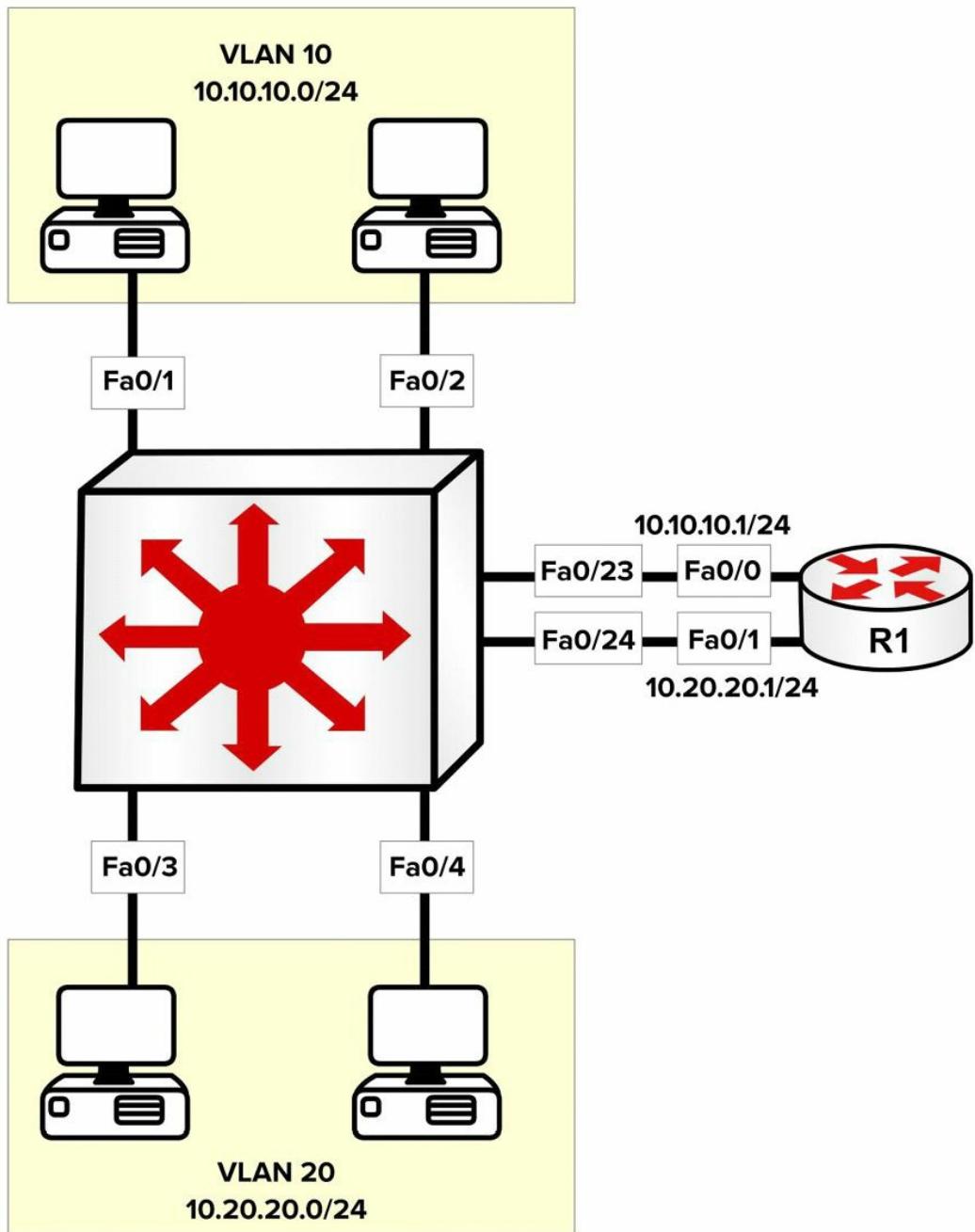


图 3.3 -- 采用多个物理路由器接口的 VLAN 间路由

图 3.3 演示了用到两个不同 VLANs 的单一 LAN，这两个 VLANs 都有分配给各自的 IP 子网。尽管图中画出的网络主机都是连接在同一物理交换机上，但因为它们处于不同的 VLANs 中，VLAN 10 中的主机与 VLAN 20 中的主机之间的数据包必须要经过路由才行，而在同样 VLAN 中的数据包只需要简单的交换即可。

这种方案的主要优势在于，它是简单的，且易于部署。而最主要的劣势在于，它不具有可扩展性。比如说，当交换机上配置了有 5 个、10 个，甚至 20 个额外的 VLANs 时，路由器上就要有相应数量的物理接口才行。在大多数情况下，这在技术上是不可行的。

在采用多物理路由器接口时，各需要的 VLAN 中到路由器的交换机链路，被配置为接入链路。然后路由器上的物理接口都配置上相应的 IP 地址，而 VLAN 上的网络主机，要么以静态方式配置上相应 VLAN 的 IP 地址，将该路由器物理接口作为默认网关，要么通过 DHCP 完成配置。图 3.3 中交换机的配置，在下面的输出中有演示。

```
VTP-Server-1(config)#vlan 10
VTP-Server-1(config-vlan)#name Example-VLAN-10
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#vlan 20
VTP-Server-1(config-vlan)#name Example-VLAN-20
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#interface range FastEthernet0/1 - 2, 23
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport access vlan 10
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#exit
VTP-Server-1(config)#interface range FastEthernet0/3 - 4, 24
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport access vlan 20
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#exit
```

2960 交换机上无需 `switchport` 命令，因为其接口已经运行于二层模式。

下面的输出又演示了图 3.3 中的路由器的配置。

```
R1(config)#interface FastEthernet0/0
R1(config-if)#ip add 10.10.10.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface FastEthernet0/1
R1(config-if)#ip add 10.20.20.1 255.255.255.0
R1(config-if)#exit
```

### 使用路由器子接口的 VLAN 间路由

采用路由器子接口实现 VLAN 间路由的方法，解决了使用多路由器物理接口方法所可能存在的伸缩性问题。有了路由器子接口，就只需要路由器有一个物理接口就行，接下来的子接口是经由在那个物理接口上的配置获得。图 3.4 演示了这种方法。

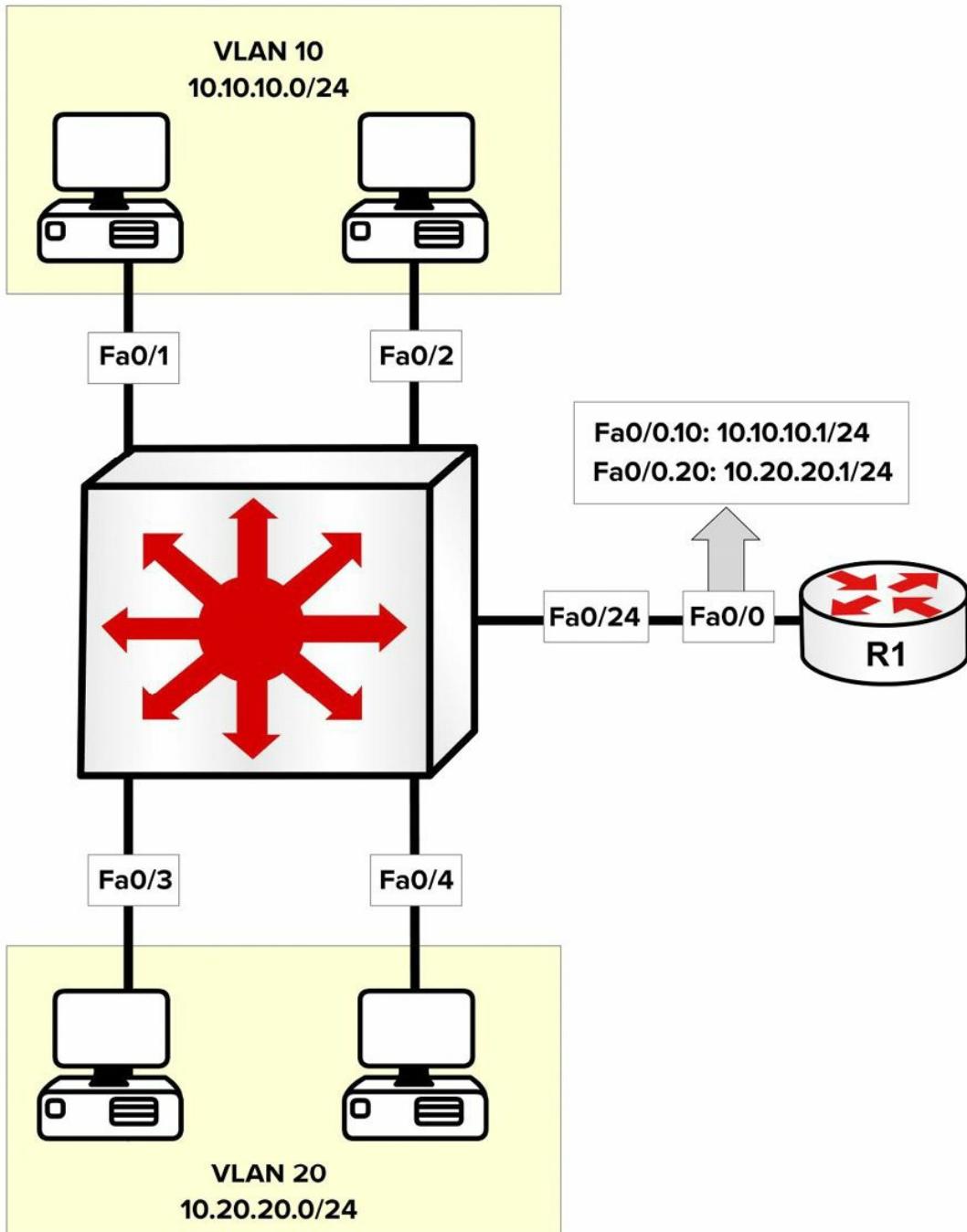


图 3.4 -- 采用路由器子接口的 VLAN 间路由

图 3.4 描绘了图 3.3 中同样的 LAN。但在图 3.4 中，仅使用了一个物理路由器接口。而为了实现一种 VLAN 间路由解决方案，使用 `interface [name].[subinterface number]` 全局配置命令，在该主要路由器接口上配置出了子接口。而通过命令 `encapsulation [isl|dot1q] [vlan]` 子接口配置命令，又将各子接口与某个特定 VLAN 关联了起来。最后一步就是给子接口配置上需要的 IP 地址。

在交换机上，那条连接路由器的单一链路，必须要配置为中继链路，这是因为**路由器不支持 DTP**。假如中继配置成 802.1Q 中继，那么在原生 VLAN 不是默认原生 VLAN 时，此中继的原生 VLAN 一定要定义。而该原生 VLAN 也要在相应的路由器接口上予以配置，配置命令为 `encapsulation dot1q [vlan] native` 子

**接口配置命令。**下面的输出演示了使用单一物理接口的 VLAN 间路由配置（又称作“单臂路由，router-on-a-stick”）。图 3.4 中绘出的两个 VLANs 在下面的输出中也有显示，同时还有一个额外的 VLAN 用于管理用；该管理 VLAN 将被配置为原生 VLAN。

```
VTP-Server-1(config)#vlan 10
VTP-Server-1(config-vlan)#name Example-VLAN-10
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#vlan 20
VTP-Server-1(config-vlan)#name Example-VLAN-20
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#vlan 30
VTP-Server-1(config-vlan)#name Management-VLAN
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#interface range FastEthernet0/1 - 2
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport access vlan 10
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#exit
VTP-Server-1(config)#interface range FastEthernet0/3 - 4
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport access vlan 20
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#exit
VTP-Server-1(config)#interface FastEthernet0/24
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport trunk encapsulation dot1q
VTP-Server-1(config-if)#switchport mode trunk
VTP-Server-1(config-if)#switchport trunk native vlan 30
VTP-Server-1(config-if)#exit
VTP-Server-1(config)#interface vlan 30
VTP-Server-1(config-if)#description 'This is the Management Subnet'
VTP-Server-1(config-if)#ip address 10.30.30.2 255.255.255.0
VTP-Server-1(config-if)#no shutdown
VTP-Server-1(config-if)#exit
VTP-Server-1(config)#ip default-gateway 10.30.30.1
```

图 3.4 中的路由器之配置如下面的输出所示。

```
R1(config)#interface FastEthernet0/0
R1(config-if)#no ip address
R1(config-if)#no shut <- 这一步相当重要，否则子接口也会处于 down down 状态
R1(config-if)#exit
R1(config)#interface FastEthernet0/0.10
R1(config-subif)#description 'Subinterface For VLAN 10'
R1(config-subif)#encapsulation dot1Q 10
R1(config-subif)#ip add 10.10.10.1 255.255.255.0
R1(config-subif)#exit
R1(config)#interface FastEthernet0/0.20
R1(config-subif)#description 'Subinterface For VLAN 20'
R1(config-subif)#encapsulation dot1Q 20
R1(config-subif)#ip add 10.20.20.1 255.255.255.0
R1(config-subif)#exit
R1(config)#interface FastEthernet0/0.30
R1(config-subif)#description 'Subinterface For Management'
R1(config-subif)#encapsulation dot1Q 30 native
R1(config-subif)#ip add 10.30.30.1 255.255.255.0
R1(config-subif)#exit
```

此方案的主要优在于，路由器上仅需一个物理接口。主要的劣势在于，该物理端口的带宽，是为所配置的多个子接口所公用的。因此，如果存在很多 VLAN 间流量时，路由器就很快会成为网络的性能瓶颈。

### 采用交换机虚拟接口的 VLAN 间路由

**多层交换机支持在物理接口上配置 IP 地址。**但要先用**接口配置命令** `no switchport` 对这些接口进行配置，以允许管理员在其上配置 IP 地址。除开使用物理接口外，多层交换机还支持交换机虚拟接口（Switch Virtual Interfaces, SVIs）技术。

SVIs 是一系列代表了 VLAN 的逻辑接口。尽管某个交换机虚拟接口代表了一个 VLAN，它也不是在某个 VLAN 在交换机上配置出来时，就自动配置出来的；它必须要管理员通过执行 `interface vlan [number]` 全局配置命令，手动加以配置。而那些诸如 IP 分址等的三层配置参数，也要与在物理接口上一样，在交换机虚拟接口予以配置。

以下输出演示了在单一交换机上实现 VLAN 间路由，做出的交换机虚拟接口配置。此输出引用了本小节前面的配置输出所用到的 VLANs。

```
VTP-Server-1(config)#vlan 10
VTP-Server-1(config-vlan)#name Example-VLAN-10
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#vlan 20
VTP-Server-1(config-vlan)#name Example-VLAN-20
VTP-Server-1(config-vlan)#exit
VTP-Server-1(config)#interface range FastEthernet0/1 - 2
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#switchport access vlan 10
VTP-Server-1(config-if-range)#exit
VTP-Server-1(config)#interface range FastEthernet0/3 - 4
VTP-Server-1(config-if-range)#switchport
VTP-Server-1(config-if-range)#switchport mode access
VTP-Server-1(config-if-range)#switchport access vlan 20
VTP-Server-1(config-if-range)#exit
VTP-Server-1(config)#interface vlan 10
VTP-Server-1(config-if)#description "SVI for VLAN 10"
VTP-Server-1(config-if)#ip address 10.10.10.1 255.255.255.0
VTP-Server-1(config-if)#no shutdown
VTP-Server-1(config-if)#exit
VTP-Server-1(config)#interface vlan 20
VTP-Server-1(config-if)#description 'SVI for VLAN 10'
VTP-Server-1(config-if)#ip address 10.20.20.1 255.255.255.0
VTP-Server-1(config-if)#no shutdown
VTP-Server-1(config-if)#exit
```

**在用到多层交换机时，交换机虚拟端口是推荐的配置方法，和实现 VLAN 间路由的首选方案。**

你可通过使用 `show interface vlan x` 命令，来验证某个交换机虚拟接口是配置恰当的（IP 分址等）。下面的输出与 `show interface x` 命令等同。

```
Switch#show interfaces vlan 100
Vlan100 is up, line protocol is down
    Hardware is EtherSVI, address is c200.06c8.0000 (bia c200.06c8.0000)
    Internet address is 10.10.10.1/24
    MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
        reliability 255/255, txload 1/255, rxload 1/255
    Encapsulation ARPA, loopback not set
    ARP type: ARPA, ARP Timeout 04:00:00
```

如你希望使用一台 2960 交换机来路由 IP 数据包，那么就需要对配置进行修改，然后进行重启。这是因为 2960 和更新型号的一些交换机进行了性能调优，实现一种明确的交换机资源分配方式。该资源管理方式叫做交换机数据库管理（Switch Database Manager, SDM）模板。你可以在以下几种 SDM 模板中进行选择。

- 默认 (default) -- 各项功能的平衡
- IPv4/IPv4 双协议支持 (dual IPv4/IPv6) -- 用于双栈环境(dual-stack environments)
- Lanbase-routing -- 支持各种单播路由 (Unicast routes)
- 服务质量 (Quality of Service, QoS) -- 提供对各种服务质量特性的支持

下面是在我的 3750 交换机上的输出。这些输出与 2960 上的选项不完全一致，但你明白了这个意思。同时，请记住，**交换机型号及 IOS 对 SDM 配置选项有影响，因此，你要查看你的型号的配置手册。**

```
Switch(config)#sdm prefer ?
access                  Access bias
default                 Default bias
dual-ipv4-and-ipv6     Support both IPv4 and IPv6
ipe                     IPe bias
lanbase-routing         Unicast bias
vlan                   VLAN bias
```

在你期望在 2960 交换机上配置 VLAN 间路由时，就需要开启 Lanbase-routing SDM 选项。同时在此变更生效前，需要重启交换机。下面是 `show sdm prefer` 命令的输出，该输出告诉你当前的 SDM 配置以及资源分配情况。

```
Switch#show sdm prefer
The current template is "desktop default" template.
The selected template optimizes the resources in
the switch to support this level of features for
8 routed interfaces and 1024 VLANs.
number of unicast mac addresses:          6K
number of IPv4 IGMP groups + multicast routes: 1K
number of IPv4 unicast routes:            8K
    number of directly-connected IPv4 hosts: 6K
    number of indirect IPv4 routes:        2K
number of IPv4 policy based routing aces:   0
number of IPv4/MAC qos aces:              0.5K
number of IPv4/MAC security aces:         1K
Switch#
```

## 虚拟局域网中继协议，VTP

虚拟局域网中继协议 (VLAN Trunking Protocol, VTP) 是一个思科专有的二层消息协议 (a Cisco proprietary Layer 2 messaging protocol)，用于管理同一个 VTP 域中交换机上 VLANs 增加、删除及重命名。VTP 允许 VLAN 信息在交换网络 (the switched network) 上宣告/扩散 (propagate)，这将减轻交换网络中的管理开销，同时使得众多的交换机能够交换 (exchange) 并维护一致的 VLAN 信息。此概念在图 3.5 中进行了演示。

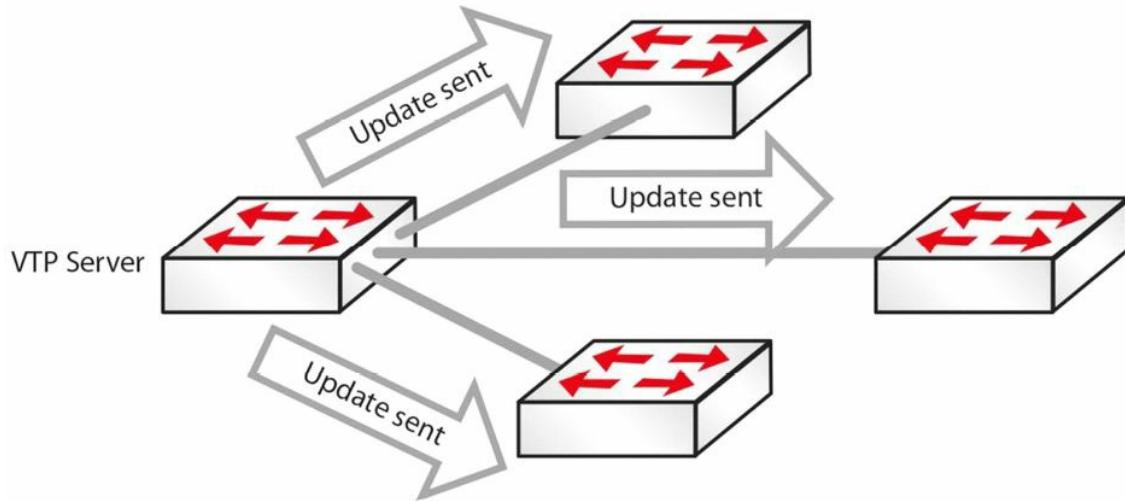


图 3.5 -- VTP 更新

采用 VTP 能够得到以下好处。

- VLANs 信息的精确监控和汇报
- 整个网络上 VLANs 的一致性
- 易于添加和移除 VLANs

## 配置 VTP, configuring VTP

要让交换机进行 VLAN 信息交换，这些交换机就必须配置在同一个 VTP 域中，如下面的输出这样。

```
Switch(config)#vtp mode server ←this is on by default
Switch(config)#vtp domain in60days
Changing VTP domain name from NULL to in60days
Switch#show vtp status
VTP Version : 2
Configuration Revision : 0
Maximum VLANs Supported Locally : 255
Number of Existing VLANs : 5
VTP Operating Mode : Server
VTP Domain Name : in60days
```

如要安全的传输 VTP 更新数据，可以为其加上一个口令，但要求 VTP 域中的每台交换机的口令都要匹配。

```
Switch(config)#vtp password Cisco321
Setting device VLAN database password to Cisco321
```

## VTP 模式, VTP Modes

VTP 以下列三种模式允许。

- 服务器模式（默认模式），server(default)
- 客户端模式, client
- 透明模式, transparent

上面的输出中，你可以看到配置中有个服务器模式。

### 服务器模式, Server Mode

在服务器模式时，该交换机被授权去建立、修改及删除整个 VTP 域上的 VLAN 信息。你对服务器所做的任何修改，都会扩散到整个域中。而 VLAN 配置是保存在位于闪存中的 VLAN 数据库文件“`vlan.dat`”中的。

### 客户端模式, Client Mode

处于客户端模式下的交换机，将会接收 VTP 信息，并根据收到的通告信息做出配置上的改变，而不能增加、移除或是改变它们的 VLAN 信息了。客户端交换机也会在它们的中继端口上，发出接收到的 VTP 数据包。记住，你是不能将客户端交换机的某个端口，添加到 VTP 服务器上不存在的 VLAN 中去的。VLAN 配置也是保存在位于闪存中的 VLAN 数据库文件“`vlan.dat`”中的。

### 透明模式, Transparent Mode

透明模式下的交换机，将在它们的中继端口上转发接收到的 VTP 信息，却不会应用通告的更新。一台 VTP 透明模式交换机（a VTP Transparent-mode switch）是可以创建、修改并移除 VLANs 的，但其 VLAN 配置变动不会通告给其它交换机。VTP 透明模式仍然需要域信息配置项。当处于 VTP 服务器与客户端之间的某台交换机，需要有不同的 VLAN 数据库时，它就需要是一台 VTP 透明交换机。而要配置上扩展的 VLAN 编号范围（the extended VLAN range）时，也要用到透明模式。

## VTP 修剪, VTP Pruning

时常会出现这样的情形，比如说，在网络的一边有 VLANs 20 到 50，另一边有 VLANs 60 到 80。而一边的那些交换机上的 VLAN 信息却又无需传送到另一边的那些交换机上。为此，交换机能够将它们的 VLAN 信息进行修剪，因此而减少广播流量，如图 3.6 所示。

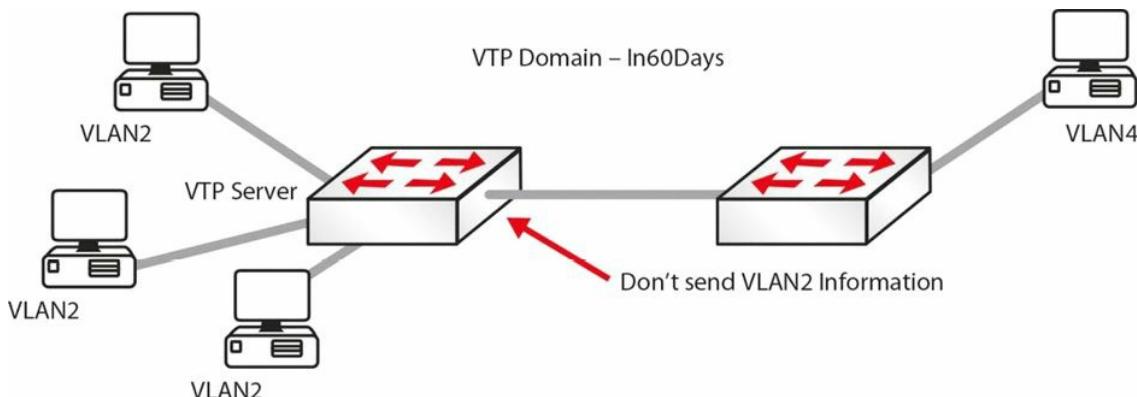


图 3.6 -- 运行中的 VTP 修剪

下面这行配置将 VTP 修剪功能加入到你的交换机。

```
Switch(config)#vtp pruning
```

而当你将一台处于其它两台交换机之间的交换机设置为透明模式时，该配置就没有任何意义了，透明模式交换机上的 VTP 修剪不会运行。

## 配置修订号, Configuration Revision Number

配置修订号（the configuration revision number）是一个 32 位整数，用以表明一个 VTP 数据包的修订级别（在上面的 `show vtp status` 输出中可以看到）。此信息用于判断收到的信息是否与当前版本要新。你每次此处于 VTP 服务器模式的交换机上做出 VLAN 变动时，配置修订号就会加 1，同时变动会通告给 VTP 客户端（处于 VTP 透明模式的交换机，其修订号会是 0，且不会随 VLAN 数据库的变动而增大）。修改 VTP 域名称，然后又改回早前的名称，可以实现交换机配置修订号重置。

**重要提示：**当以匹配的 VTP 域名，同时有着较高修订号的一台交换机被配置为 VTP 服务器，或者 VTP 客户端而接入到网络中时，它的数据库将会被通告给其它交换机，进而潜在地将它们各自现有的 VTP 数据库进行替换。这有可能会将整个局域网拖垮，所以在将一台新交换机连入到局域网时，一定要小心谨慎（总是要检查当前的 VTP 状态）。

## VLAN 故障排除基础， Basic VLAN Troubleshooting

VLANs 是一种相当直观的交换机特性，它很少需要进行故障排除。你所发现的问题，大部分都是人为配置错误。在第 15 天的课程中，我们会详细讲到二层故障排除。而一些常见的涉及 VLAN 的问题有这些。

1. VLAN 间路由无效， Inter-VLAN routing not working: 检查交换机之间的链路、路由器都是正确设置的，以及相关的 VLANs 允许通过且未被修剪（参照“VTP 修剪”部分）。 `show interface trunk` 命令将提供所需信息。还要检查路由器子接口有配置了正确的封装方式和 VLAN，同时子接口的 IP 地址是那些主机的默认网关。
2. 无法创建 VLANs， VLANs cannot be created: 检查交换机的 VTP 模式是否被设置成了“client”。在 VTP 模式为“client”时，是不能创建 VLANs 的。另一个重要原因是交换机所允许的 VLANs 编号。`show vtp status` 命令将提供所需的信息（参看下的“中继和 VTP 故障排除”部分）。
3. 同一 VLAN 中的主机之间不能通信， Hosts within the same VLAN cannot reach each other: 重要的是某 VLAN 中的主机都要有一个属于同一子网的 IP 地址。如子网不同，它们之间就无法通信。另一个需要考虑的原因是这些主机是否都是连接到同一台交换机上。如它们不是连接到同一交换机，就要确保交换机之间的中继链路工作正常，还要确保该 VLAN 未在允许清单中被排除/被修剪[ensure that the trunk links(s) between the switches is/are working correctly and that the VLAN is not excluded/not pruned from the allowed list]。`show interface trunk` 命令将给出有关该中继链路的所需信息。

## 中继和 VTP 故障排除， Troubleshooting Trunking and VTP

下面是一些问题实例机器可能的解决方法。

- 中继宕掉？
  - 接口务必要是 up/up
  - 中继链路两端的封装方式要匹配

```
SwitchA#show interface fa1/1 switchport
Name: Fa1/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: Disabled
Access Mode VLAN: 0 ((Inactive))
```

- VLAN 信息未有传输
  - 该 VLAN 在中继链路上阻塞了吗

```
Switch#show interface trunk
• VTP 信息无法到达 VTP 客户端
◦ VTP 域名称密码正确吗?

show vtp status / show vtp password
```

- 在增加一台新的交换机后，所有 VTP 信息都变动了
  - 总是客户端模式加入新的交换机（但请查看上面的有关“配置修订号（the configuration revision number）”的提示）
  - 服务器模式将通告新信息
- VTP 修剪无效
  - 中间是否有一台透明交换机？
  - 该 VLAN 允许通过该中继链路？

## VLAN 间路由故障排除， Troubleshooting Inter-VLAN Routing

VLAN 间路由故障可以多种形式出现，尤其是考虑在该过程中涉及多种设备（交换机、路由器等）。通过下面给出的适当故障排除方法论，你就能够将问题孤立在某台特定设备上，接着再其对应到一个错误配置的具体特性。

从连通性立足点上看，下面这些事情都应该检查一下。

- 检查一下终端主机连接了正确的交换机端口
- 检查一下正确的交换机端口连接了正确的路由器端口（如使用了一台路由器做 VLAN 间路由）
- 检查一下在此过程中所涉及到的每个端口承载的是正确的 VLANs
  - 连接终端站的那些端口，通常是被分配到一个特定 VLAN 的接入端口
  - 而将交换机连接至路由器的那些端口，则通常是中继端口

在确认设备之间的连通性无误后，逻辑上下一步就是检查二层配置了，**以所配置的中继端口上的封装方式开始**，这通常是作为首选的 802.1Q 封装方式。接着就要确保中继链路两端都是配置了同样的封装方式。

可用于查看封类型的一些命令有以下这些。

- `show interface trunk`
- `show interface <number> switchport`

这里有个输出示例。

```
Cat-3550-1#show interfaces trunk
Port      Mode       Encapsulation      Status      Native vlan
Fa0/1    on        802.1q            trunking    1
Fa0/2    on        802.1q            trunking    1
Port      Vlans allowed on trunk
Fa0/1    1,10,20,30,40,50
Fa0/2    1-99,201-4094
```

命令 `show interface trunk` 提供的另一重要细节是中继状态。从中继状态可以看出中继是否形成，同时在链路两端都要检查中继状态。如果接口未处于“中继”模式，那么接口的运行模式（on, auto, 等）是最重要的检查项，以弄清接口能否允许与链路另一端形成中继态（a trunking state）。

中继端口上另外一个需要检查的重要元素便是原生 VLAN。原生 VLAN 错误配置可能带来功能缺失，抑或安全问题。中继链路的两端的原生 VLAN 需要匹配。

假如在完成二层检查任务后，VLAN 间路由问题仍然存在，你就可以继续进行三层配置检查了。依据用于实现 VLAN 间路由的三层设备，可能会在下列设备上进行配置及配置检查。

- 多层交换机， multilayer switch

- 路由器 -- 物理接口, router -- physical interfaces
- 路由器 -- 子接口, router -- subinterfaces

三层设备上应该检查一下其各接口（或者交换机虚拟接口，SVI）都有分配的正确的子网，同时如有必要，你还应检查一下路由协议。通常情况下，各个 VLAN 都有分配不同的子网，所以你应确保你未曾错误配置了接口。而为检查此项，你可以对特定物理接口、子接口或是 SVI，使用 `show interface` 命令。

## 第三天的问题

1. Name four advantages of using VLANs.
2. Hosts in the same VLAN can be in different subnets. True or false?
3. An access link is part of more than one VLAN. True or false?
4. Name the two trunk link encapsulation types.
5. Which commands will configure and name a VLAN?
6. A trunk link on a switch can be in which five possible modes?
7. Which command would put your interface into VLAN 5?
8. Which command will change the native VLAN?
9. VTP Client mode allows you to configure VLANs. True or false?
10. Name three benefits of using VTP.
11. Which command configures VTP pruning on your switch?

## 第三天问题的答案

1. Containing Broadcasts within a smaller group of devices will make the network faster; saves resources on devices because they process less Broadcasts; added security by keeping devices in a certain group (or function) in a separate Broadcast domain; and flexibility in expanding a network across a geographical location of any size.
2. True, but not recommended.
3. False.
4. 802.1Q and ISL.
5. The `vlan x` and `name y` commands.
6. On, off, auto, desirable, and nonegotiate.
7. The `switchport access vlan 5` command.
8. The `switchport trunk native vlan x` command.
9. False.
10. Accurate monitoring and reporting of VLANs; VLAN consistency across the network; and ease of adding and removing VLANs.
11. The `vtp pruning` command.

## 第三天的实验

### VLAN 和中继实验

#### 拓扑图, Topology



### 实验目的, Purpose

学习如何配置 VLANs 以及中继链路。

### 实验步骤, Walkthrough

1. 你需要在每台 PC 上添加 IP 地址。可自由选择，只要求它们在同一子网上。
2. 在交换机 A 上设置主机名 (hostname) , 创建 VLAN 2, 并将连接 PC 的那个接口放到 VLAN 2 中。如你愿意，你也可以赋予 VLAN 2 一个名称。

```

Switch>en
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch(config)#hostname SwitchA
SwitchA(config)#vlan 2
SwitchA(config-vlan)#name 60days
SwitchA(config-vlan)#interface FastEthernet0/1
SwitchA(config-if)#switchport mode access
SwitchA(config-if)#switchport access vlan 2
SwitchA(config-if)#^Z
SwitchA#show vlan brief
VLAN      Name           Status      Ports
----      ----           -----      -----
1        default         active     Fa0/2, Fa0/3, Fa0/4, Fa0/5,
                                    Fa0/6, Fa0/7, Fa0/8, Fa0/9,
                                    Fa0/10, Fa0/11, Fa0/12, Fa0/13,
                                    Fa0/14, Fa0/15, Fa0/16, Fa0/17,
                                    Fa0/18, Fa0/19, Fa0/20, Fa0/21,
                                    Fa0/22, Fa0/23, Fa0/24
2        60days         active     Fa0/1
1002     fddi-default   active
1003     token-ring-default active
1004     fddinet-default active
1005     trnet-default   active
SwitchA#

```

1. 将中继链路设置为中继模式。

```

SwitchA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)#int FastEthernet0/2
SwitchA(config-if)#switchport mode trunk
SwitchA#show interface trunk
Port      Mode          Encapsulation  Status      Native vlan
Fa0/2    on            802.1q        trunking    1
Port      Vlans allowed on trunk
Fa0/2    1-1005

```

1. 如你愿意，设置在该中继链路上仅允许 VLAN 2。

```

SwitchA(config)#int FastEthernet0/2
SwitchA(config-if)#switchport trunk allowed vlan 2
SwitchA(config-if)#^Z
SwitchA#
%SYS-5-CONFIG_I: Configured from console by console
SwitchA#show int trunk
Port      Mode      Encapsulation      Status      Native vlan
Fa0/2    on        802.1q            trunking      1
Port      Vlans allowed on trunk
Fa0/2    2

```

1. 此时，如你自其中一台 PC ping 往另一台，将会失败。这是因为一边是在 VLAN 1 中，另一边在 VLAN 2 中。

```

PC>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Request timed out.
Ping statistics for 192.168.1.1:
    Packets: Sent = 2, Received = 0, Lost = 2 (100% loss)

```

1. 此时在交换机 B 上配置同样的那些命令。创建 VLAN、将交换机 PC 端口放入 VLAN 2，并将该接口设置为接入模式，还要将中继链路设置为“中继”。
2. 现在你就可以从一台 PC 实现跨越中继链路 ping 通另一 PC 了。

```

PC>ping 192.168.1.1
Pinging 192.168.1.1 with 32 bytes of data:
Reply from 192.168.1.1: bytes=32 time=188ms TTL=128
Reply from 192.168.1.1: bytes=32 time=78ms TTL=128
Reply from 192.168.1.1: bytes=32 time=94ms TTL=128
Reply from 192.168.1.1: bytes=32 time=79ms TTL=128
Ping statistics for 192.168.1.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 78ms, Maximum = 188ms, Average = 109ms

```

## VTP 实验

在一个又两台交换机组成的拓扑中，实验今天所提到的那些 VTP 配置命令。

- 将其中一台交换机配置为 VTP 服务器
- 将另一台交换机配置为 VTP 客户端
- 在两台交换机上配置同样的 VTP 域及口令 (the same VTP domain and password)
- 在服务器交换机上创建一系列的 VLANs，然后观察它们是如何彼此之间是如何通告的
- 在两台交换机上都配置 VTP 修剪 (VTP pruning)
- 在两台交换机上检查 (展示) VTP 配置
- 在两台交换机上配置不同的 VTP 域及口令，并重复上述过程；观察结果的不同

# 第4天 路由器和交换机安全

## Router and Switch Security

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第四天任务

- 阅读今天的课文（接下来的）
- 复习昨天的课文
- 完成今天的实验
- 阅读 ICND1 记诵指南

购买的交换机和路由器都是没有任何安全配置的。依据业务需求，你需要添加这些配置。令到交换机安全的那些命令和过程，与路由器的是大致一样的。现在我们就要移步到这些防止意外或是恶意登入及重新配置你的路由器，从而确保其安全的操作步骤上来。

我在思科公司的一份工作，是在核心团队。涉及在访问控制列表（access control lists, ACL）、IOS 升级、灾难恢复及相关任务等方面，的客户支持。最初打击到我的事情，就是有那么多的工程师没有采用口令将其路由器进行锁定。而这些人中很多就是用“password”或者“cisco”-- 两个最容易猜到的，作为口令，简直不敢想象。

在本书的这一章，我们看看在所有网络上，为了保护你的路由器，你应该做的几个基本步骤。

今天你会学到以下几点。

- 物理访问的防护, Protecting physical access
- 远程登入防护, Telnet access
- Enable 模式防护, Protecting Enable mode
- 路由器日志记录, Router logging
- 交换机安全加固, Securing the switch

本章覆盖以下的 CCNA 大纲要求。

- 配置和检查网络设备的以下安全特性
  - 设备口令安全, device password security
  - Enable 密码与 enable, Enable secret versus enable
  - 传入, transport
  - 关闭 Telnet, disable telnet
  - 安全壳, Secure SHell, SSH
  - 虚拟终端, VTYs

- 物理安全, Physical security
- 服务口令, service password
- 描述不同的外部认证方式, describe external authentication methods
- 配置并检查交换机端口安全特性, 比如以下这些。
  - MAC 地址粘滞, sticky MAC
  - MAC 地址限制, MAC address limitation
  - 静态/动态, static/dynamic
  - 危害模式, violation modes
  - 错误关闭, err disable
  - 关闭端口, shutdown
  - 受限保护, protect restrict
  - 关闭未使用端口, shutdown unused ports
  - 错误关闭恢复, err disable recovery
- 将未使用端口指派到一个未使用 VLAN 中, assign unused ports to an unused VLAN
- 将原生 VLAN 设置成非 VLAN 1, set native VLAN to something other than VLAN 1
- 配置并验证 NTP 客户端

## 物理访问防护, Protecting Physical Access

在某个公司因失去网络接入而出现灾难性后果时, 你常发现他们的路由器竟然实在某人的办公桌下, 这是多么的令人惊奇啊。

网络设备应该安放在一间有密码锁的安全房间里, 最起码也应该锁起来。思考路由器可是非常贵重的设备, 也是小偷炙手可热的目标。网络规模越大, 设备就越值钱, 同时数据保护与路由器配置文件的保护需求也越高。

## 控制台访问

控制台接口是设计用于提供到路由器的物理访问的, 以完成路由器的初始设置及灾难恢复。任何能取得控制台访问的人, 都能够完全擦出或是重新配置这些文件, 因此, 控制台接口应有一个口令进行保护, 可以简单地添加一个口令, 也可以为其添加一个本地用户名和口令, 如下面所示。

- 只添加一个口令即可

```
Router(config)#line console 0
Router(config-line)#password cisco
Router(config-line)#login
```

- 为控制台添加一个本地用户名和口令组合

```
Router(config)#username paul password cisco
Router(config)#line console 0
Router(config-line)#login local
```

你还可以为控制台 (以及虚拟终端) 线路创建一个超时值, 如此就可以在确定的时间过后断开连接。默认的超时是 5 分钟。

```

Router(config)#line console 0
Router(config-line)#exec-timeout ?
<0-35791> Timeout in minutes
Router(config-line)#exec-timeout 2 ?
<0-2147483> Timeout in seconds
<cr>
Router(config-line)#exec-timeout 2 30
Router(config-line)#

```

## 远程登陆访问， Telnet Access

在某人给远程登陆或者说虚拟终端线路添加了口令之前，你实际上是不能远程登陆进一台路由器的。同样，你可以给虚拟终端线路添加一个口令，或是告诉路由器去查找一个本地用户名及口令组合（该组合可以在配置文件中，或是存储在一台 RADIUS/TACACS 服务器上），如下面所示。

```

Router(config-line)#line vty 0 15
Router(config-line)#password cisco
Router(config-line)#login ~ or login local

```

下面的输出是自某台路由器到另一台的远程登陆会话。当你获得远程登陆访问时，你可以看到主机名发生了改变。注意在你输入口令时，它看不到。

```

Router1#telnet 192.168.1.2
Trying 192.168.1.2 ...Open
User Access Verification
Username: paul
Password:
Router2>

```

而如你有一个安全版 IOS 镜像，则可以将路由器配置为仅允许安全壳访问，而不是远程登陆访问。这样做的好处在于所有数据都是加密了的。如你在启用安全壳后，再次使用远程登时，连接将被终止。

```

Router1(config)#line vty 0 15
Router1(config-line)#transport input ssh
Router2#telnet 192.168.1.2
Trying 192.168.1.2 ...Open
[Connection to 192.168.1.2 closed by foreign host]

```

## 使能模式保护， Protecting Enable Mode

使能模式（enable mode）取得路由器的配置访问，因此你会想要保护该模式。你可以配置一个**使能秘密（an enable secret）** 或**使能口令（an enable password）**。实际上，使能秘密和使能口令是可以同时有的，但这是一个坏主意。

使能口令是未加密的，所以在路由器配置中可以看到。而使能秘密有 5 级加密(level 5 encryption, MD5)，难于破解。自 15.0(1) S 后的较新 IOS 版本中，还可以使用比 MD5 加密高级的 4 级加密（level 4 encryption, SHA256），5 级加密最终会不赞成使用。你可以给使能口令加上命令 `service password-encryption`，但因为此方式使用 7 级加密(level 7 encryption, 比如，低安全性；思科称其为“背后安全性，over the shoulder security”，因其仅需某人从你背后偷看并记住一个稍难的词组，便可以用网上的 7 级口令解密工具予以破解），而很容易被破解。下面的输出中可以看到 7 级与 5 级加密文本。

```

Router(config)#enable password cisco
Router(config)#exit
Router#show run
enable password cisco
Router(config)#enable password cisco
Router(config)#service password-encryption
Router#show run
enable password 7 0822455D0A16
Router(config)#enable secret cisco
Router(config)#exit
Router#show run
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0

```

记住如你忘记了使能口令，你将不得不对路由器或交换机进行一下**口令恢复操作**。请用 Google 搜索你所使用的型号，因为型号不同其口令恢复过程也不一样。对于路由器来说，涉及

- 设备重启，以及
- 在重启过程中按下指定的中断键盘按键
- 再设置配置寄存器（the configuration register）以跳过启动配置文件（通常将配置寄存器设置为 0x2142）
- 接着要执行一个 `copy start-config running-config` 命令

此时，就可以创建新的口令了。

而对交换机来说，口令恢复过程会有一点复杂（请再次用 Google 搜索你用到的具体交换机型号），但也可以通过一个小把戏实现口令恢复 -- 在给交换机上电时，按住 MODE 按钮 8 秒钟。交换机将以空白配置启动，而上一次的启动配置(the last startup configuration)将保存在 flash 中的 config.text.renamed 文件里头，所以该文件可复制用于运行配置(running configuration)，然后用其它口令对其进行修改。

## 用户访问防护，Protecting User Access

思科 IOS 提供对用户的单独用户名及口令，同时对所能够使用的命令进行清单限制的能力。这在分层次网络支持时是有用的。下列输出中给出了一个示例。

```

RouterA#config term
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#username paul password cisco
RouterA(config)#username stuart password hello
RouterA(config)#username davie password football
RouterA(config)#line vty 0 4
RouterA(config-line)#login local
RouterA(config-line)#exit
RouterA(config)#exit

```

**你可在路由器上指派不同用户帐号的访问级别**。比如，你也许打算那些初级网络团队成员仅能使用一些基本的故障排除命。你还有必要记住思科路由器有口令安全的两种模式（two modes of password security），用户模式（`Exec mode`）和特权模式（`Enable mode`）。

思科路由器有可供配置的 16 种（0 到 15）不同特权级别，其中 15 级是完全的访问权限，如下所示。

```
RouterA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#username support privilege 4 password soccer
    LINE Initial keywords of the command to modify
RouterA(config)#privilege exec level 4 ping
RouterA(config)#privilege exec level 4 traceroute
RouterA(config)#privilege exec level 4 show ip interface brief
RouterA(config)#line console 0
RouterA(config-line)#password basketball
RouterA(config-line)#login local -- password is needed
RouterA(config-line)#^z
```

支持那人在登入到路由器并尝试进入配置模式时，此命令及其它命令将不可用且无效，也不能看到。

```
RouterA con0 is now available
Press RETURN to get started.
User Access Verification
Username: support
Password:
RouterA#config t -- not allowed to use this
^
% Invalid input detected at '^' marker.
```

你可在路由器提示符下查看默认的不同特权级别(the default privilege levels)。

```
Router>show privilege
Current privilege level is 1
Router>en
Router#show priv
Router#show privilege
Current privilege level is 15
Router#
```

## 更新 IOS, updating the IOS

公认地，更新 IOS 有时会将漏洞或故障引入你的网络中，因此，如你有与思科公司有个技术支持合同（a TAC contract, Technical Assistance Centre, TAC），那么最好的做法就是依据思科公司的建议来做。一般来讲，保持 IOS 版本最新是高度推荐的做法。藉由更新 IOS，你能得到下面这些好处。

- 修正已知的软件缺陷，fixed known bugs
- 解决安全隐患，close security vulnerabilities
- 提供特性强化及 IOS 能力提升，Offers enhanced features and IOS capabilities

## 路由器日志记录，Router Logging

路由器提供事件记录的能力。它们可将日志消息照你的意愿，发送到屏幕或某台服务器。你应该记录路由器消息，而又有 8 个可用的日志记录严重程度级别（考试要求你知道这些不同的级别），如下面输出中的粗体字所示。

```

logging buffered ?
`<0-7>` Logging severity level
alerts=Immediate action needed (severity=1)
critical=Critical conditions (severity=2)
debugging=Debugging messages (severity=7)
emergencies=System is unusable (severity=0)
errors=Error conditions (severity=3)
informational=Informational messages (severity=6)
notifications=Normal but significant conditions (severity=5)
warnings=Warning conditions (severity=4)

```

而你有可以将这些日志消息发往几个不同的地方。

```

Router(config)#logging ?
  A.B.C.D      IP address of the logging host
  buffered     Set buffered logging parameters
  console      Set console logging parameters
  host         Set syslog server IP address and parameters
  on           Enable logging to all enabled destinations
  trap         Set syslog server logging level
  userinfo    Enable logging of user info on privileged mode enabling

```

日志消息通常会在你经由控制台进入到路由器时，显示在屏幕上。而这可能会在你敲入配置命令时多少有些烦人。这里就有个在我输入一个命令（加了下划线的那条）时，被一条控制台日志消息(a console logging message)给中断了的例子。

```

Router(config)#int f0/1
Router(config-if)#no shut
Router(config-if)#end
Router#
*Jun 27 02:06:59.951: %SYS-5-CONFIG_I: Configured from console <u>show ver</u>
*Jun 27 02:07:01.151: %LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up

```

此时你既可以用命令 `no logging console` 关闭日志消息输出，也可以用 `logging synchronous` 命令将它们设置为无中断 (not interrupt)，`logging synchronous` 命令会重新输入在被日志消息中断之前，你所输入的那行命令。`logging synchronous` 命令在虚拟终端线路上也是可用的。

```

Router(config)#line con 0
Router(config-line)#logging synchronous
Router(config-line)#
Router(config-line)#exit
Router(config)#int f0/1
Router(config-if)#shut
Router(config-if)#exit
Router(config)#
*Jun 27 02:12:46.143: %LINK-5-CHANGED: Interface FastEthernet0/1, changed state to
administratively down
Router(config)#exit

```

这里值得一提的是，在你经由 Telnet (或 SSH) 进入到路由器时，你是不会看到控制台输出的。如你想在此时看到日志消息，执行 `terminal monitor` 命令即可。

## 简单网络管理协议，Simple Network Management Protocol, SNMP

**SNMP 是一种可用于远程管理网络的服务。**它由一台网络管理员维护、运行了 SNMP 管理软件的中心工作站，及包括路由器、交换机及服务器等的，各台网络设备上的小文件（代理，agents）构成。

包括 HP、Cisco、IBM 及 SolarWinds 等的几家厂商，都有设计 SNMP 软件。也有很多开发源代码版本的 SNMP 软件可用。这类软件允许你监测设备的带宽及活动情况，比如登陆活动以及端口状态等。

运用 SNMP，你可以远程地配置或是关闭端口和设备。你也可以将其配置在某些条件触发时，诸如出现高带宽或是端口宕掉时，发出警告消息。我们会在第 40 天来讲 SNMP 的细节，因为 SNMP 是 ICND2 大纲的部分。

## 加固交换机，Securing the Switch

### 阻止远程登陆访问，Prevent Telnet Access

远程登陆流量以明文方式发送口令，这就是说，可轻易地在配置中读取口令，或是有人接到你的网络上，那么就能通过网络嗅探软件查看到口令。

默认情况下，远程登陆实际上是关闭的（也就是说，你需要为其设置一个口令，也可选择设置一个用户名，来让其工作）。不过，如你仍想要有对管理端口的远程访问的话，你可使用命令 `transport input ssh`，开启到交换机的 SSH 通信，这已在前面讨论过了。

Farai 说 -- “所有虚拟终端线路下，命令 `transport input all` 是默认开启的，而其它线路的 `transport input none` 命令是默认开启的。”

### 开启 SSH，Enable SSH

尽可能地采用 SSH 而不是 Telnet 及 SNMP 来访问你的交换机。SSH 表示安全壳(secure shell)，令到某网络上的两台设备之间信息的安全交换。SSH 采用公钥加密法 (public-key cryptography) 来认证连接设备。Telnet 及 SNMP 版本 1 和 2 都是未加密的，易受包嗅探 (packet sniffing) 的影响，SNMP 版本 3 提供了保密性 -- 数据包有加密以防止恶意源窃取数据 (snooping by an unauthorised source)。

要开启 SSH，你需要有一个支持加密的 IOS 版本。一种快速找出 IOS 镜像是否支持加密的方法是执行 `show version` 命令。查找镜像文件名中有无 `k9` 字样，或者在思科系统公司的安全性声明中查找有关字句。

```

Switch#sh version
Cisco IOS Software, C3560 Software (C3560-ADVIPSERVICES K9-M), Version
12.2(35)SE1, RELEASE SOFTWARE (fc1)
Copyright (c) 1986-2006 by Cisco Systems, Inc.
Compiled Tue 19-Dec-06 10:54 by antonio
Image text-base: 0x00003000, data-base: 0x01362CA0
ROM: Bootstrap program is C3560 boot loader
BOOTLDR: C3560 Boot Loader (C3560-HBOOT-M) Version 12.2(25r)SEC, RELEASE
SOFTWARE (fc4)
Switch uptime is 1 hour, 8 minutes
System returned to ROM by power-on

System image file is "flash:/c3560-advipservicesk9-mz.122-35.SE1.bin"

This product contains cryptographic features and is subject to United States and local
country laws governing import, export, transfer and use. Delivery of Cisco cryptographic
products does not imply third-party authority to import, export, distribute or use
encryption. Importers, exporters, distributors and users are responsible for compliance
with U.S. and local country laws. By using this product you agree to comply with
applicable laws and regulations. If you are unable to comply with U.S. and local laws,
return this product immediately. A summary of U.S. laws governing Cisco cryptographic
products may be found at:
http://www.cisco.com/wlc/export/crypto/tool/stqrg.html
If you require further assistance please contact us by sending email to export@cisco.com.
--More-

```

**注意:** 如你没有带有安全特性版本的 IOS, 你就必须为此付费购买。

为建立加密连接, 你需要在交换机上创建一对公钥和私钥 (a private/public key, 见下面)。在连接时, 你这边使用公钥加密数据, 交换机将会使用它的私钥来解密数据。而在认证时, 使用你所选择的用户名/口令组合。下一个问题是, 要设置交换机的主机名和域名 (hostname and domain name), 因为在创建公钥/私钥对时, 会用到主机名.域名命名法 (hostname.domainname nomenclature)。显然, 在命名主机名和域名时, 将其命名为能够代表系统的有意义名字, 是好的做法。

首先, 你要给交换机一个与默认主机名 Switch 不一样的主机名。接着, 添加其域名 (该域名通常与 Windows 活动目录的 FQDN 一致)。这时就可以创建用于秘密的密钥 (the crypto key) 了。系数/模量 (the modulus) 是指你所希望使用的密钥的长度, 取值范围是 360 到 2048, 后者具有最高的安全性; 高于 1024 位的模量就认为是安全的了。此时, 交换机上的 SSH 就已经开启了。

有一些 SSH 相关的维护命令需要输入。`ip ssh time-out 60` 命令会将任何空闲 60 秒的 SSH 连接置为超时。而命令 `ip ssh authentication-retries 2` 则会在认证失败两次的 SSH 连接重置为初始状态。此设置并不会阻止用户建立新的连接并重试认证。设置过程如下所示。

```

Switch(config)#hostname SwitchOne
SwitchOne(config)#ip domain-name mydomain.com
SwitchOne(config)#crypto key generate rsa
Enter modulus: 1024
SwitchOne(config)#ip ssh time-out 60
SwitchOne(config)#ip ssh authentication-retries 2

```

可使用命令 `ip ssh version 2` 开启 SSH 版本 2。让我们看看其中一个密钥。在这个实例中, 该密钥是为 HTTPS 生成的。因为其是在开启 HTTPS 时自动生成的, 所以其名称也会自动产生。

```
firewall#show crypto key mypubkey rsa
Key name: HTTPS_SS_CERT_KEYPAIR.server
Temporary key
Usage: Encryption Key
Key is not exportable.
Key Data:
306C300D 06092A86 4886F70D 01010105 00035B00 30580251 00C41B63 8EF294A1
DC0F7378 7EF410F6 6254750F 475DAD71 4E1CD15E 1D9086A8 BD175433 1302F403
2FD22F82 C311769F 9C75B7D2 1E50D315 EFA0E940 DF44AD5A F717BF17 A3CEDBE1
A6A2D601 45F313B6 6B020301 0001
```

要验证交换机上的 SSH 开启，输入以下命令。

```
Switch#show ip ssh
SSH Enabled - version 1.99
Authentication timeout: 120 secs; Authentication retries: 2
Switch#
```

而用一个简单的命令，就可以关闭 HTTP 访问。

```
Switch(config)#no ip http server
```

查看交换机上 HTTP 服务器的状态。

```
Switch#show ip http server status
HTTP server status: Disabled
HTTP server port: 80
HTTP server authentication method: enable
HTTP server access class: 0
HTTP server base path: flash:html
Maximum number of concurrent server connections allowed: 16
Server idle time-out: 180 secondsServer life time-out: 180 seconds
Maximum number of requests allowed on a connection: 25
HTTP server active session modules: ALL
HTTP secure server capability: Present
HTTP secure server status: Enabled
HTTP secure server port: 443
HTTP secure server ciphersuite: 3des-ede-cbc-sha des-cbc-sha rc4-128-md5 rc4-12
HTTP secure server client authentication: Disabled
HTTP secure server trustpoint:
HTTP secure server active session modules: ALL
```

还可以在 VTY 线路上应用控制列表 (an access control list, ACL)。在第 9 天的课程将会讲到。

## 设置使能秘密口令，Set an Enable Secret Password

全局配置模式允许用户对交换机或路由器进行配置，还可以擦除配置，以及重置口令。你务必要设置一个口令或秘密口令来保护此模式，而这实际上是为阻止用户闯过 (get past) 用户模式。一般口令在路由器配置文件中会显示出来，而 `enable secret` 口令则会进行加密。

上面已经提到，你实际上可以在交换机或路由器上同时设置使能口令 (a password) 和使能秘密口令 (enable secret password)，但这会带来混乱。所以请只设置使能秘密口令就好。下面的配置文件演示了通过在命令前键入 `do` 关键字，而无需回到特权模式，就可执行该命令的情形。

```

Switch1(config)#enable password cisco
Switch1(config)#do show run
Building configuration...
Current configuration: 1144 bytes
hostname Switch1
enable password cisco

```

Farai 补充道 -- “你可以使用 `service password-encryption` 命令，对使能口令 `enable password` 进行 7 级加密。”

通过在命令前加上 `no` 关键字后再次执行该命令，可以擦除配置文件中的大多数行。上面 Farai 提到的使用 `service password-encryption` 命令是毫无作用的，因为这个方法仅提供了弱加密（7 级），而下面的秘密口令（the secret password）则有着强加密（MD5）。

```

Switch1(config)#no enable password
Switch1(config)#enable secret cisco
Switch1(config)#do show run
Building configuration...
Current configuration: 1169 bytes
hostname Switch1
enable secret 5 $1$mERr$hx5rVt7rPNoS4wqbXKX7m0 [strong level 5 password]

```

## 服务，Services

你总是应该关闭那些你不会用到的服务。思科已经在关闭那些不安全和很少用到的服务和协议上做得很好了；尽管如此，你可能会要因明确这点而亲自关闭它们。同样也会有一些服务是有帮助的。多数服务可在全局配置模式中的 `service` 命令下找到。

```

Switch(config)# service ?
compress-config      Compress the configuration file
config               TFTP load config files
counters             Control aging of interface counters
dhcp                 Enable DHCP server and relay agent
disable-ip-fast-frag Disable IP particle-based fast fragmentation
exec-callback        Enable EXEC callback
exec-wait            Delay EXEC startup on noisy lines
finger               Allow responses to finger requests
hide-telnet-addresses Hide destination addresses in telnet command
linenumber           enable line number banner for each exec
nagle                Enable Nagle's congestion control algorithm
old-slip-prompts    Allow old scripts to operate with slip/ppp
pad                 Enable PAD commands
password-encryption Encrypt system passwords
password-recovery   Disable password recovery
prompt               Enable mode specific prompt
pt-vty-logging      Log significant VTY-Async events
sequence-numbers    Stamp logger messages with a sequence number
slave-log            Enable log capability of slave IPs
tcp-keepalives-in   Generate keepalives on idle incoming network
    connections
tcp-keepalives-out  Generate keepalives on idle outgoing network
    connections
tcp-small-servers   Enable small TCP servers (e.g., ECHO)
telnet-zeroidle     Set TCP window 0 when connection is idle
timestamps          Timestamp debug/log messages
udp-small-servers   Enable small UDP servers (e.g., ECHO)

```

一般来讲，有下列的这些最常见的要开启或关闭的服务。其各自的说明在中括号里。

- no service pad [数据包组装程序/分拆程序，在异步组网中有使用；很少使用到]
- no service config [阻止交换机从网络获取其配置文件]
- no service finger [关闭 finger 服务器；很少用到]
- no ip icmp redirect [组织 ICMP 重定向，而 ICMP 重定向可被用于路由器投毒]
- no ip finger [关闭 finger 服务的另一种方式]
- no ip gratuitous-arp [关闭此服务以阻止中间人攻击 (man-in-the-middle attacks) ]
- no ip source-route [关闭由用户提供到目的地的逐跳路由(user-provided hop-by-hop routing to destination)]
- service sequence-numbers [在每条日志记录中，分配给其一个编号，同时此编号序列增加]
- service tcp-keepalive-in [防止路由器将挂起的管理会话一直保持开启，prevents the router from keeping hung management sessions open]
- service tcp-keepalive-out [与 service tcp-keepalive-in 功能一样]
- no service up-small-servers [关闭 echo, chargen, discard, daytime 等功能，这些功能很少用到]
- no service tcp-small-servers [关闭 echo, chargen, discard 等功能，这些功能很少用到]
- service timestamps debug datetime localtime show-timezone [在调试模式下 (in debug mode)，将每个记录的数据包，使用本地时间，打上日期和时间的时间戳，并显示时区]
- service timestamps log datetime localtime show-timezone [在非调试模式下 (not in debug mode)，将每个记录的数据包，使用本地时间，打上日期和时间的时间戳，并显示时区 -- 这服务在查看日志文件非常有用，尤其是在时钟设置正确的情况下]

## 修改原生 VLAN， Change the Native VLAN

交换机使用原生 VLAN 来承载那些特定的协议流量，诸如思科发现协议 (Cisco Discovery Protocol, CDP)、VLAN 中继协议 (VLAN Trunking Protocol, VTP)、端口聚合协议 (Port Aggregation Protocol, PAgP)，以及动态中继协议 (Dynamic Trunking Protocol, DTP) 等协议信息。默认原生 VLAN 总是 VLAN 1；但原生 VLAN 是可以手动设置为任何有效 VLAN 编号 (除开 0 和 4096, 因为这些 VLAN 编号处于 VLANs 的保留范围)。

你可以使用下面输出中演示的命令（在每个接口下执行的），来查看原生 VLAN。

```
Switch#show interfaces FastEthernet0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
```

**将端口放入 VLAN 1 被认为是一种安全漏洞 (a security vulnerability)**，允许黑客取得网络资源的访问。为减轻此问题，避免任何主机放入 VLAN 1 是一种明智可取的做法。同时也可将所有中继接口上的原生 VLAN 修改为某个未使用的 VLAN。

```
Switch(config-if)#switchport trunk native vlan 888
```

**注意：**这是 CCNA 大纲中的一个关键目标 (one of the key objectives)，所以务必牢记。

你还可以用下面的命令，来阻止原生 VLAN 上的数据通过中继链路。

```
Switch(config-if)#switchport trunk allowed vlan remove 888
```

## 修改管理 VLAN, Change the Management VLAN

给交换机配置一个 IP 地址，以实现为管理目的而远程登陆到其上，也是可以的。这又叫做交换机虚拟接口（Switch Virtual Interface, SVI）。将该管理访问做到除 `VLAN 1` 之外的其它 VLAN 上，是一种明智的预防措施，如下面的输出所示。

```
Switch(config)#vlan 3
Switch(config-vlan)#interface vlan3
%LINK-5-CHANGED: Interface Vlan3, changed state to up
Switch(config-if)#ip address 192.168.1.1 255.255.255.0
```

## 关闭 CDP, Turn Off CDP

后面会讲到思科发现协议（Cisco Discovery Protocol, CDP），但在这里，你要知道，在大多数的路由器和交换机上的每个接口下，该协议都是打开的，这也是通行的做法，其功能是发现路由器或交换机上连接的思科设备。你可能不打算让其它思科设备看到你的网络设备的信息，那么就可以关掉这个服务，至少应该在那些你的网络边缘上的，连接其它公司或者 ISP 的那些设备上关闭 CDP。

Farai 指出 -- “CDP 在比如 ASR 路由器的所有平台上默认都未开启。”

在下面的输出中，你能看到一台连接我的交换机的路由器，在我执行 `show cdp neighbor detail` 命令时，其能看见哪些基本信息。

```
Router#show cdp neighbor detail
Device ID: Switch1
Entry address(es):
Platform: Cisco 2960, Capabilities: Switch
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/2
Holdtime: 176
Version :
Cisco Internetwork Operating System Software
IOS (tm) C2960 Software (C2960-I6Q4L2-M), Version 12.1(22)EA4, RELEASE SOFTWARE(fc1)
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 18-May-05 22:31 by jharirba
advertisement version: 2
Duplex: full
Router#
```

下面的命令将对整个设备关闭 CDP。

```
Switch1(config)#no cdp run
```

而要对某个特定接口关闭 CDP，执行以下命令。

```
Switch1(config)#int FastEthernet0/2
Switch1(config-if)#no cdp enable
```

## 添加横幅消息，Add a Banner Message

横幅消息将于某用户登入路由器或交换机时显示出来。其并不会提供任何实质性的安全，但会显示你设置的警告信息。在下面的配置中，我选择的是“Y”字母作为界定符（delimiting character），界定符用以告诉路由器，我已输完消息文字。

```
Switch1(config)#banner motd Y
Enter TEXT message. End with the character 'Y'.
KEEP OUT OR YOU WILL REGRET IT Y
Switch1(config)#

```

在我从交换机登入到路由器时，我能看到横幅消息。错在选择了“Y”作为界定符，因为它割除了我的消息文字。

```
Router#telnet 192.168.1.3
Trying 192.168.1.3 ...open
KEEP OUT OR

```

横幅消息可以是以下这些。

- 在用户看到登陆提示符之前显示出来 -- MOTD (message of the day)
- 在用户看到登陆提示符之前显示出来 -- Login
- 在登陆提示符之后显示给用户 -- Exec (在你打算对未授权用户隐藏的信息)

在本书中，横幅消息作为一些实验的组成部分。我建议你掌握全部三种类型横幅消息，并以登入路由器的方式来测试它们。依据你所用的平台和 IOS，会有不同的选择。

```
Router(config)#banner ?
LINE           c banner-text c, where 'c' is a delimiting character
exec          Set EXEC process creation banner
incoming      Set incoming terminal line banner
login         Set login banner
motd          Set Message of the Day banner
prompt-timeout Set Message for login authentication timeout
slip-ppp       Set Message for SLIP/PPP

```

## 设置 VTP 口令， Set a VTP Password

VTP 确保网络上交换机之间传输的是精确的 VLAN 信息。而为了保护 VLAN 信息的更新，你应该在交换机上加入 VTP 口令（该 VTP 域中所有交换机上的 VTP 口令都应一致），如下面输出演示的那样。

```
Switch1(config)#vtp domain 60days
Changing VTP domain name from NULL to 60days
Switch1(config)#vtp password cisco
Setting device VLAN database password to cisco
Switch1(config)#

```

## 限定 VLAN 信息， Restrict VLAN Information

默认下的交换机允许所有 VLANs 通过中继链路。你将其修改为指定 VLANs 才能通过中继链路。如下面的输出所示。

```

Switch1(config)#int FastEthernet0/4
Switch1(config-if)#switchport mode trunk
Switch1(config-if)#switchport trunk allowed vlan ?
    WORD      VLAN IDs of the allowed VLANs when this port is in trunking mode
    add       add VLANs to the current list
    all      all VLANs
    except   all VLANs except the following
    none     no VLANs
    remove   remove VLANs from the current list
Switch1(config-if)#switchport trunk allowed vlan 7-12
Switch1#show interface trunk
Port      Mode      Encapsulation      Status      Native vlan
Fa0/4    on        802.1q            trunking      1
Port      Vlans allowed on trunk
Fa0/4    7-12

```

## 端口因出错关闭后的恢复功能，Error Disable Recovery

由一系列的事件导致的，思科交换机将其端口置为一种特别的关闭模式（a special disabled mode），叫做出错关闭（err-disabled）。此特性简单来讲，由于在某个特定端口上发生某种错误后，该端口就被关闭了。错误可能有多种原因，之最常见的就是出现触发了某项端口安全策略（a port security policy）。在某个未经授权用户尝试连接到某个交换机端口时，这是通常的做法，它阻止那些违规设备访问网络。

出错关闭端口(an err-disabled port) 看起来会是这样的。

```

Switch# show interface f0/1
FastEthernet0/1 is down, line protocol is down [err-disabled]
.....

```

而为了重新使用（re-activate）某个出错关闭接口，以在该接口上执行 shutdown 及 no shutdown 命令的人工干预是必要的，网络工程师们俗称此操作为端口弹跳(a bouncing the port)。但是，某些情形要求从原端口状态自动恢复过来，而不是等到管理员手动开启该端口。此出错关闭回复模式，通过将交换机配置为在依引发通信失败事件的不同，而不同的一段时间后，自动重新打开出错关闭端口的方式，来发挥作用。区分通信失败事件，提供了出错关闭恢复功能所监测事件上的粒度（granularity）控制。

完成该功能设置的命令是 errdisable recovery cause，在全局路由器配置模式下输入。

```

Switch(config)#errdisable recovery cause ?
    all      Enable timer to recover from all causes
    bpduguard  Enable timer to recover from bpdu-guard error disable state
    dtp-flap   Enable timer to recover from dtp-flap error disable state
    link-flap   Enable timer to recover from link-flap error disable state
    pagp-flap   Enable timer to recover from pagp-flap error disable state
    rootguard   Enable timer to recover from root-guard error disable state
    udld      Enable timer to recover from udld error disable state
.....

```

errdisable recovery cause 命令依设备型号会有所不同，但最常见的参数有这些。

- all
- arp-inspection
- bpduguard
- dhcp-rate-limit
- link-flap
- psecure-violation

- security-violation
- storm-control
- udld

多数平台上端口自动恢复的默认时间是 300 秒，此时间可以用全局配置命令 `errdisable recovery interval` 手动修改。

```
Switch(config)#errdisable recovery interval ?
<30-86400> timer-interval(sec)
```

而命令 `show errdisable recovery` 命令则会提供有关出错关闭恢复功能（the err-disable recovery function）激活了的那些特性的细节信息，以及受到监测的接口，并包含了接口重新开启剩余时间。

```
Switch#show errdisable recovery
ErrDisable Reason      Timer Status
-----
arp-inspection        Disabled
bpduguard             Disabled
channel-misconfig     Disabled
dhcp-rate-limit       Disabled
dtp-flap              Disabled
gbic-invalid          Disabled
inline-power          Disabled
l2ptguard             Disabled
link-flap              Disabled
mac-limit              Disabled
link-monitor-failure Disabled
loopback               Disabled
oam-remote-failure   Disabled
pagp-flap              Disabled
port-mode-failure    Disabled
psecure-violation     Enabled
security-violation    Disabled
sfp-config-mismatch  Disabled
storm-control          Disabled
udld                  Disabled
unicast-flood          Disabled
vmps                 Disabled

Timer interval: 300 seconds
Interfaces that will be enabled at the next timeout:
Interface      Errdisable reason      Time left(sec)
-----
Fa0/0          psecure-violation      193
```

## 外部认证方式，External Authentication Methods

与本地存储不同，你可以采用一台通常运行了 AAA 或 TACACS+ 的服务器来存储用户名和口令。这么做的优势在于，你无需在每台路由器和交换机上都手动输入用户名和口令。而是将其存储在服务器的数据库中。

TACACS+ 表示“加强版终端访问控制器访问控制系统（Terminal Access Controller Access Control System Plus, TACACS+）”。它是一个思科专有协议，使用 TCP 49 号端口。**TACACS+** 提供了经由一台或多台 TACACS+ 中心服务器，对包含路由器及网络介入服务器等网络设备的访问控制。

**拨入用户远端认证服务（Remote Authentication Dial-In User Service, RADIUS）**，是一套分布式网络安全系统，用以确保网络远程访问的安全性，同时它又是一个使用 UDP 的客户端/服务器协议（a client/server protocol）。RADIUS 是开放标准。

如你拥有 TACACS+ 或者 RADIUS，那么你可能希望开启认证、授权和记账（Authentication, Authorization, and Accounting, AAA）。AAA 是安装在一台服务器上的，它监测着网络的一个用户帐号数据库。用户访问、协议、连接，以及断开原因，及其它很多特性都能被监测到。

路由器和交换机可被设置为在某用户尝试登入时查询服务器。服务器此时来验证用户。**CCNA 考试不要求你去配置这些协议。**

## 路由器时钟及 NTP， Router Clock and NTP

交换机上的时间经常被忽略；但它却是重要的。在你遇到安全入侵（security violations）、SNMP 问题（SNMP traps），或者事件记录时，会用到时间戳。如交换机上的时间不正确，就会难于找出时间发生的时间。举个例子，让我们看看下面的交换机，并检查一下它的时间。

```
Switch#show clock  
*23:09:45.773 UTC Tue Mar 2 1993
```

该时间是不准确的，所以我们要修改一下。但首先，我们要设置一些属性值。

```
clock timezone CST -6  
clock summer-time CDT recurring  
clock summer-time CST recurring 2 Sun Mar 2:00 1 Sun Nov 2:00
```

首先，我们设置时区（the time zone）。我是位于中部时区（the Central time zone），比 GMT 要早 6个小时。接着告诉交换机夏令时（时间变化，the time change）是循环的。最后设置夏令时具体是什么。此时，我们就可以设置时间和日期了。

```
Switch#clock set 14:55:05 June 19 2007  
Switch#  
1d23h: %SYS-6-CLOCKUPDATE: System clock has been updated from 17:26:01 CST  
Tue Mar 2 1993 to 14:55:05 CST Tue Jun 19 2007, configured from console.  
Switch#show clock  
14:55:13.858 CST Tue Jun 19 2007
```

请注意，**时钟设置实在使能模式（Enable mode），而不是配置模式下**。除了手动设置时钟外，你可以使用网络时间协议（Network Time Protocol, NTP）。它让你可将交换机的时钟与某台原子钟（an atomic clock）同步，保证非常精确的时间。

```
Switch(config)#ntp server 134.84.84.84 prefer  
Switch(config)#ntp server 209.184.112.199
```

使用下面的两个命令，你可以查看时钟是否已经和 NTP 源保持同步。

```
Switch#show ntp associations  
Switch#show ntp status
```

在第 40 天中，我们会涉及更多有关 NTP 的内容。

## 关闭未用到的那些端口， Shut Down Unused Ports

未使用的，或者说“空起的”那些没有任何网络设备的端口，因为某人会插入一条网线并将未授权设备连接到网络，而引发安全威胁。这会导致一些安全问题，包括。

- 网络未能如同与其的那样运作
- 网络信息暴露于外部人员

这就是为何你要关闭路由器、交换机及其它网络设备上，所有未使用端口的原因。依据具体设备，关闭状态可能是端口默认的状态，但你仍要对此进行验证。

而关闭端口是通过在**接口配置模式**下使用 `shutdown` 命令完成的。

```
Switch#conf t
Switch(config)#int fa0/0
Switch(config-if)#shutdown
```

验证某端口处于关闭状态有多种方法，其一就是使用 `show ip interface brief` 命令。

```
Router(config-if)#do show ip interface brief
Interface          IP-Address  OK? Method   Status           Protocol
FastEthernet0/0    unassigned   YES unset   administratively down   down
FastEthernet0/1    unassigned   YES unset   administratively down   down
```

请注意，**管理性关闭**状态就是说该端口是手工关闭的。验证关闭状态的另一方法是使用 `show interface` 命令。

```
Router#show interface fa0/0
FastEthernet0/0 is administratively down, line protocol is down
    Hardware is GT96k FE, address is c200.27c8.0000 (bia c200.27c8.0000)
    MTU 1500 bytes, BW 10000 Kbit/sec, DLY 1000 usec,
    ....
```

## 思科发现协议，Cisco Discovery Protocol, CDP

现在来讨论思科发现协议正是时候。

CDP 因为其在做出任何配置之前，就提供了一种发现有关网络设备信息的方法，而是一个**热门的考试考点**。它是一直非常有用的故障排除工具；但它又带来了安全威胁。

CDP 是一个思科专有协议，也就是说它只运行在思科设备上。它是一种**二层服务**，设备用它来通告和接收那些直接连接设备的基本信息。IEEE 版本的 CDP 叫做**链路层发现协议 (Link Layer Discovery Protocol, LLDP)**，CCNA 大纲并不包含此内容。

因为 CDP 是一种二层服务，所以它**并不需要配置有 IP 地址来交换信息**。只需开启接口就行。如有配置 IP 地址，该 IP 地址也会包含进 CDP 消息中。

CDP 作为非常强大的故障排除工具，考试中要求你掌握如何来使用它。图 4.1 展示了 Router 0 的 CDP 输出。请设想一下在没有拓扑图 (topology diagram) 的情况下，你要对此网络进行故障排除的情形。

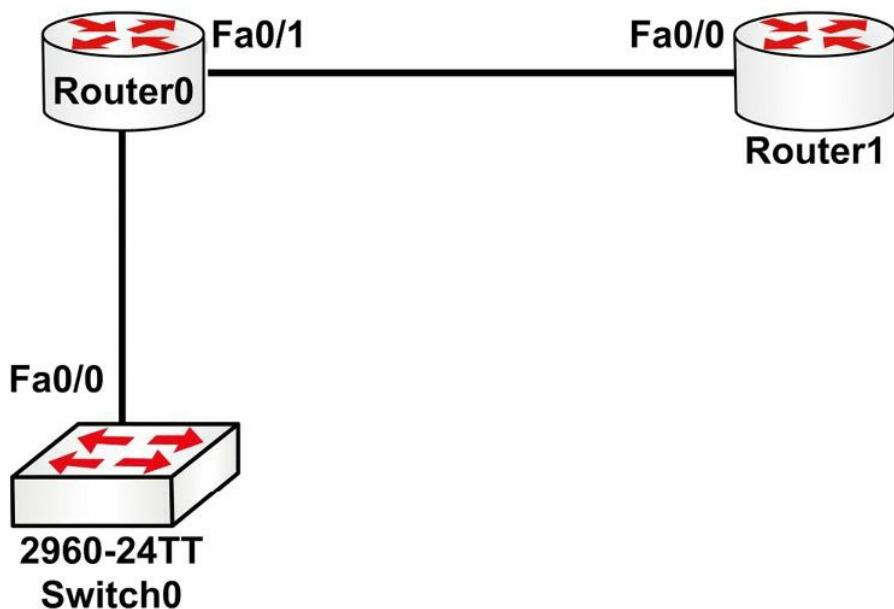


图 4.1 -- Router 0 的 CDP 输出

下列配置输入，正是图 4.1 中的。

```
Router0#show cdp neighbors
Capability Codes: R - Router, T - Trans Bridge, B - Source Route Bridge, S - Switch, H -
Host, I - IGMP, r - Repeater, P - Phone
Device ID      Local Interface Holdtime     Capability   Platform      Port
Switch        Fas 0/0          165           S            2960         Fas 0/1
Router         Fas 0/1          169           R            C1841        Fas 0/0
Router0#
```

在上述命令的后面加上 `detail` 命令，你可以看到更多信息。

```

Router0#show cdp neighbors detail
Device ID: Switch
Entry address(es):
Platform: cisco 2960, Capabilities: Switch
Interface: FastEthernet0/0, Port ID (outgoing port): FastEthernet0/1
Holdtime: 178
Version :
Cisco IOS Software, C2960 Software (C2960-LANBASE-M), Version 12.2(25)FX, RELEASE
SOFTWARE (fc1)
Copyright (c) 1986-2005 by Cisco Systems, Inc.
Compiled Wed 12-Oct-05 22:05 by pt_team
advertisement version: 2
Duplex: full
-----
Device ID: Router
Entry address(es):
IP address : 192.168.1.2
Platform: cisco C1841, Capabilities: Router
Interface: FastEthernet0/1, Port ID (outgoing port): FastEthernet0/0
Holdtime: 122
Version :
Cisco IOS Software, 1841 Software (C1841-ADVIPSERVICESK9-M), Version 12.4(15)T1, RELEASE
SOFTWARE (fc2)
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2007 by Cisco Systems, Inc.
Compiled Wed 18-Jul-07 04:52 by pt_team
advertisement version: 2
Duplex: full

```

现在你可以看到 IOS 版本、型号、IP 地址以及其它信息。记住现在你仍未在 Router 0 上配置 IP 地址。

前面我们已经讲过怎样在整台设备或仅在某个接口上关闭 CDP 了。而另两个有关命令是显示设备有关 CDP 的协议信息的 show cdp 命令，以及通过输入设备名称来查看某台具设备信息的 show cdp entry <Router> 命令。建议在今天要配置的实验中花些时间，来查看 CDP 的众多输出。

```

Router0#show cdp
Global CDP information:
    Sending CDP packets every 60 seconds
    Sending a holdtime value of 180 seconds
    Sending CDPv2 advertisements is enabled
Router0#show cdp ?
    entry      Information for specific neighbor entry
    interface  CDP interface status and configuration
    neighbors  CDP neighbor entries
    traffic    CDP statistics
    |          Output modifiers
<cr>

```

## 交换机端口安全，Switch Port Security

端口安全特性，是通过限制某个特定端口或是接口能够学习到的 MAC 地址数目，来保护交换机端口安全，并最终确保内容可寻址存储器（Content Addressable Memory, CAM）表的安全的一项，Catalyst 交换机的有力特性。具备了端安全特性，交换机就能够维护一张用于明确哪个 MAC 地址（或哪些地址），可以接入哪些本地交换机端口的表格。此外，交换机同样可以配置为仅允许在任何给定的端口上学习到指定数量的 MAC 地址。端口安全如图 4.2 所示。

**注：**关于CAM的更多信息，请参考 [CAM \(Content Addressable Memory\) VS TCAM \(Ternary Content Addressable Memory\)](#)。

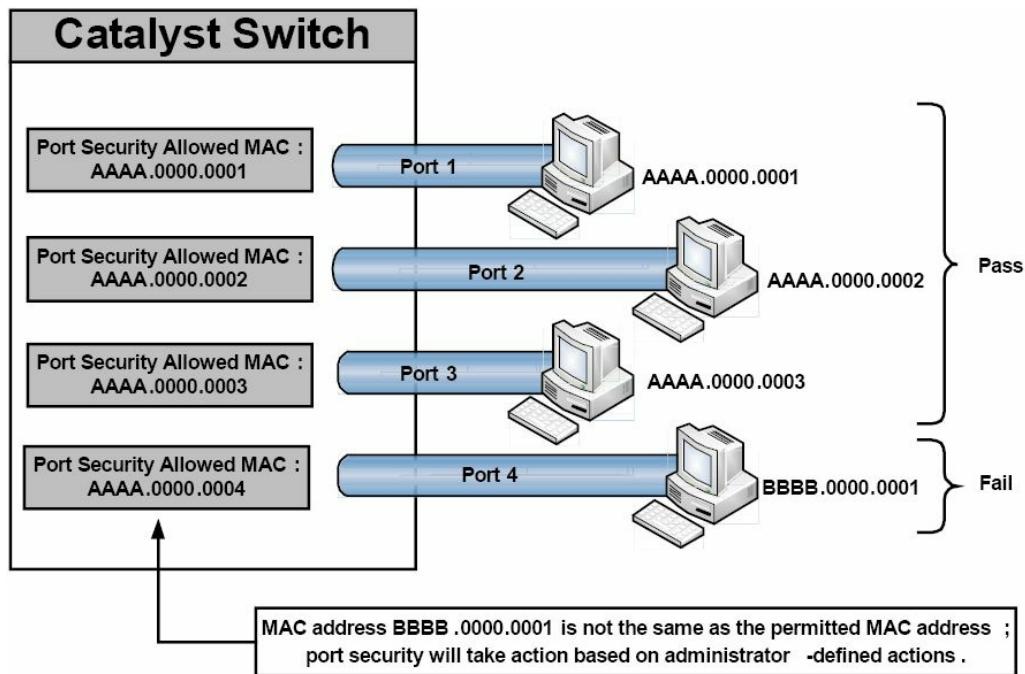


图 4.2 -- 端口安全的运作

图 4.2 展示了在某台 Catalyst 交换机上，通过端口安全特性配置的 4 个端口，它们都只允许单一 MAC 地址接入。从 1 号到 3 号端口连接的 MAC 地址与端口安全所允许的地址匹配。在没有其它过滤的情况下，这些主机就能够经由其各自交换机端口转发流量。而端口 4 上所配置的是允许 AAAA.0000.0004 MAC 地址，但所接入的 MAC 地址却是 BBBB.0000.0001。因为主机 MAC 地址与所允许的 MAC 不一样，端口安全 (port security) 将在端口上做出如同管理员所设定的适当动作。这些有效端口安全动(the valid port security actions)作将在接下来的部分详细说明。

**端口安全特性**设计用于保护交换局域网 (the switched LAN) 免受两种主要的攻击方式。这两种方式在下的小节讲到。

- CAM 表溢出攻击， CAM table overflow attacks
- MAC 欺骗攻击， MAC spoofing attacks

## CAM 表溢出攻击

交换机的 CAM 表是一些存储位置，这些存储位置包含了物理端口上的那些 MAC 地址，及其 VLAN 参数。交换机 CAM 表中动态学习到的内容，或者说 MAC 地址表，可通过命令 `show mac-address-table dynamic` 查看到，如下面的输出所示。

```
VTP-Server-1#show mac-address-table dynamic
      Mac Address Table
-----
Vlan   Mac Address        Type      Ports
----  -----
  2    000c.ce47.f3a0  DYNAMIC   Fa0/1
  2    0013.1986.0a20  DYNAMIC   Fa0/2
  6    0004.c16f.8741  DYNAMIC   Fa0/3
  6    0030.803f.ea81  DYNAMIC   Fa0/4
  8    0004.c16f.8742  DYNAMIC   Fa0/5
  8    0030.803f.ea82  DYNAMIC   Fa0/6
Total Mac Addresses for this criterion: 6
```

如同所有的计算装置一样，交换机的存储资源也是有限的。这就意味着 **CAM 表的存储空间是固定的，已分配好的**。CAM 表溢出攻击 (CAM table overflow attacks) 将此限制作为目标，用大量随机生成的无效源及目的 MAC 地址灌入交换机，直到填满 CAM 表，此时交换机就无法接收新的 CAM 表条目了。在此情况下，交换机成为了一台集线器，只能开始简单地将新近接收的帧广播到其上的所有接口（同一 VLAN 中的），就是将该 VLAN 变成了一个大的广播域。

对 CAM 表的攻击易于开展，因为有着像 MACOF 及 DSNIFF 等常见的工具可用于实施这样的行为。而增加 VLANs 的数目（此举减少了广播域的尺寸），可有助于降低 CAM 表攻击的影响，在交换机上配置端口安全特性，是推荐的安全方案。

## MAC 欺骗攻击，MAC Spoofing Attacks

MAC 地址欺骗，用于冒充某个源 MAC 地址，以达到扮演网络上的其它主机或设备的目的。冒充 (spoofing) 就是伪装成或扮着是并非你本人的某个人。MAC 地址欺骗的主要目的是扰乱交换机，令到其以为某台同一主机连接到两个端口上，这就导致交换机尝试同时将帧转发给受信任的主机和攻击者。图 4.3 展示了连接 4 台不同网络主机的交换机的 CAM 表。

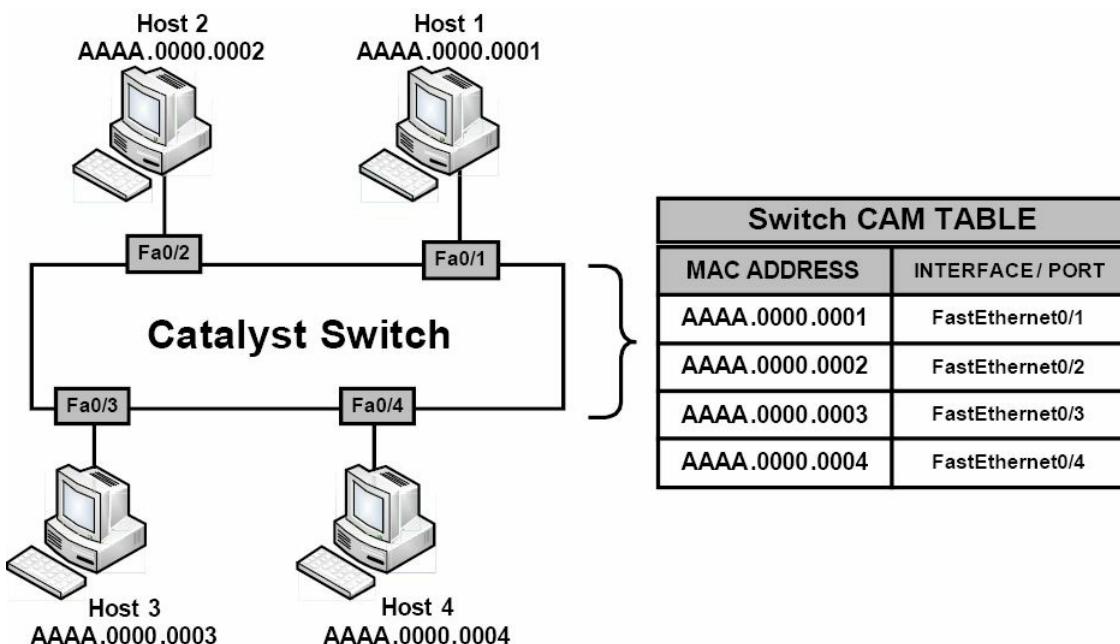


图 4.3 中，交换机允许正常，根据 CAM 表条目，它知道连接至其端口上的所有设备的 MAC 地址。而基于当前的 CAM 表，如 4 号主机想要发给 2 号主机一个帧，交换机就会简单地将该帧转发出它的 FastEthernet 0/2 接口，而前往 2 号主机。

现在，假设 1 号主机被某个想要接收所有发往 2 号主机流量的攻击者入侵了。则经由 MAC 地址欺骗，攻击者精心构建出使用 2 号主机源地址的以太网帧。在交换机收到该帧后，它记下该源地址，并重写 CAM 表中 2 号主机所对应的条目，将其指向 FastEthernet 0/1 端口，而不是 2 号主机所真正连接的 FastEthernet 0/2。此概念如图 4.4 所示。



根据图 4.4，在 3 号主机或 4 号主机尝试将帧发给 2 号主机时，交换机会将这些帧转发出 FastEthernet 0/1，到 1 号主机，因为 CAM 表已被 MAC 地址欺骗攻击投毒。在 2 号主机发出另一个帧时，交换机再次从 FastEthernet 0/2 了解其 MAC 地址，并再度将 CAM 表条目重写，以反应出该变化。结果就出现 2 号主机与 1 号主机之间就谁保有此 MAC 地址的拔河。

此外，这也将扰乱交换机，造成反复的重写 MAC 地址表条目，引发在合法主机（也就是 2 号主机）上的拒绝服务攻击（Denial of Service , DoS）。而假如假冒的 MAC 地址数目很高，MAC 地址欺骗攻击还将对持续重写其 CAM 表的交换机的性能造成严重影响。应用端口安全(implementing port security)，可以减轻 MAC 地址欺骗攻击的影响。

## 地址安全的端口安全，Port Security Secure Addresses

经由端口安全特性，我们可以指定特定 MAC 才被允许访问某个交换机端口，同时限制某个单一交换机端口所支持的 MAC 地址数目。以下本节所说明的几种端口安全应用方式。

- 静态 MAC 地址保全, Static secure MAC addresses
- 动态 MAC 地址保全, Dynamic secure MAC addresses
- 粘滞 MAC 地址保全, Sticky secure MAC addresses

静态 MAC 地址保全是由网络管理员静态配置，并存储在 MAC 地址表中，同时还保存在交换机配置文件里。当将静态 MAC 地址保全指派给某个安全端口时，交换机不会转发那些源地址与所配置的静态安全 MAC 地址不匹配的帧。

动态 MAC 地址保全，是交换机所学习到的，存储在 MAC 地址表中。与静态 MAC 地址安全不同，在交换机重启或断电后，动态安全 MAC 地址条目会从交换机中移除。在交换机再次启动时，这些地址要再次习得。

粘滞的 MAC 地址保全，是静态和动态 MAC 地址保全的结合。地址可动态习得，也可静态配置，存储在 MAC 地址表中，也保存在交换机配置文件里。这就意味着在交换机关闭或是重启后，它无需再次动态发现 MAC 地址，因为这些 MAC 地址已经保存在配置文件中了（如你有保存允许的配置）。

## 动作的端口安全，Port Security Actions

一旦开启了端口安全，管理员就可定义出在出现违反端口安全事件后，交换机所采取的动作了。思科 IOS 软件允许管理员指定 4 种在出现冲突时所采取的不同动作。

- 保护，Protect
- 关闭（默认动作），Shutdown(default)
- 限制，Restrict
- 关闭 VLAN （超出 CCNA 大纲），Shutdown VLAN(outside of the CCNA syllabus)

**保护动作选项强制端口进入受保护端口模式（Protected Port mode）**。此模式下，交换机会简单地丢弃所有源地址不明的单播和多播帧（simply discard all Unicast or Multicast frames with unknown source MAC addresses）。而在交换机被配置为保护某端口时，当其以受保护端口模式运行时，不会发出通知，这就意味着由处于此模式下的交换机端口阻止所有流量时，管理员是无法获知的。

**关闭动作选项则是在出现违反端口安全后，将某端口置于某种错误关闭状态（an err-disabled state）**。在此配置动作被用到时，交换机上相应端口的 LED 同时被关闭。而在关闭模式下，交换机发出一条 SNMP trap（[浅谈 Linux 系统中的 SNMP trap](#)），以及一条 syslog 消息，同时冲突计数器会增大。这个选项是接口配置了端口安全时的默认动作。

**限制动作选项是在安全 MAC 地址数目到达管理员为该端口所定义的最大限制时，用于丢弃那些带有不明 MAC 地址的数据包**。此模式下，交换机会持续限制额外的那些 MAC 地址发出帧，直到移除对那些安全 MAC 地址数目的限制，或直到所允许的最大地址数目得以增加。和关闭工作选项一样，交换机也会发出一条 SNMP trap 及一条 syslog 消息，同时冲突计数器会增加。

**关闭 VLAN 动作选项**跟**关闭动作选项**类似；不过此选项是**关闭某个 VLAN**，而不是**某个交换机端口**。此配置可能会应用在那些指派了多个 VLAN，诸如语音 VLAN 和数据 VLAN，的端口上，以及交换机的中继链路上。

## 端口安全配置， Configuring Port Security

在配置端口安全之前，建议将交换机端口静态配置为二层接入端口（端口安全只能配置为静态接入端口或中继端口上，不能配置在动态端口上）。此配置如下面的输出所示。

```
VTP-Server-1(config)#interface FastEthernet0/1
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport mode access
```

**注意:** 在诸如 Catalyst 2950 及 Catalyst 2960 系列的二层交换机上无需 `switchport` 命令。而在比如 Catalyst 3750、Catalyst 4500 以及 Catalyst 6500 系列等的多层交换机上，它是需要的。

默认情况下，端口安全是关闭的；但可通过接口配置命令 `switchport port-security [mac-address {mac-address} [vlan {vlan-id} | {access | voice}]] | mac-address {sticky} [mac-address | vlan {vlan-id} | {access | voice}]] [maximum {value} [vlan {vlan-list} | {access | voice}]]` 予以开启。表 4.1 说明了该命令的这些选项。

关键字	说明
<code>mac-address {mac-address}</code>	此关键字用于指定一个静态保全 MAC 地址。你还可以加入不超过配置的最大数目的其它安全 MAC 地址。
<code>vlan {vlan id}</code>	此关键字应只使用在某个中继端口上，以指定 VLAN ID 和 MAC 地址。如果没有指定 VLAN ID，就使用原生 VLAN。
<code>vlan {access}</code>	此关键字应只用在某个接入端口上，以指定该 VLAN 作为接入 VLAN。
<code>vlan {voice}</code>	此关键字应只用在某个接入端口上，用以知道该 VLAN 作为一个语音 VLAN。而只有在该特定端口上配置了语音 VLAN 时，该选项才可用。
<code>mac-address {sticky} [mac-address]</code>	此关键字用于在特定接口上开启动态或地址粘滞学习（used to enable dynamic or sticky learning），或者为其配置一个静态安全 MAC 地址。
<code>maximum {value}</code>	此关键字用于指定某个接口上可以学到的安全地址的最大数目。默认是 1。

## 静态安全地址配置， Configuring Static Secure MAC Addresses

下面的输出演示了怎样在某个接口上开启端口安全，以及在某个交换机**接入端口**上配置一个静态安全 MAC 地址 `001f:3c59:d63b`。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport mode access
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security mac-address 001f.3c59.d63b
```

下面的输出演示了怎样在某个接口上开启端口安全，并在某个交换机**中继端口**的 VLAN 5 中配置一个静态安全 MAC 地址 `001f:3c59:d63b`。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport trunk encapsulation dot1q
VTP-Server-1(config-if)#switchport mode trunk
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security mac-address 001f.3c59.d63b vlan 5
```

而下面的输出则演示了如何在某个接口上开启端口安全，并在某个交换机接入端口的 VLAN 5 (数据 VLAN) 和 VLAN 7 (语音 VLAN)，分别配置一个静态安全 MAC 地址 001f:3c59:5555 和 001f:3c59:7777。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport mode access
VTP-Server-1(config-if)#switchport access vlan 5
VTP-Server-1(config-if)#switchport voice vlan 7
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security maximum 2
VTP-Server-1(config-if)#switchport port-security mac-address 001f.3c59.5555 vlan access
VTP-Server-1(config-if)#switchport port-security mac-address 001f.3c59.7777 vlan voice
```

记住在某个同时配置了语音 VLAN 和数据 VLAN 的接口上开启端口安全时，该端口上的最大允许安全地址数应设置为 2，这一点非常重要。这又是通过包含在上面输出中的**接口配置命令** switchport port-security maximum 2 完成的。

两个 MAC 地址中的一个由 IP 电话使用，交换机在语音 VLAN 上学到此地址。另一个由可连接在 IP 电话上的主机（比如 PC）所使用。交换机将在数据 VLAN 上学到这个 MAC 地址。

## 静态安全地址配置的验证，Verifying Static Secure MAC Address Configuration

同过执行 show port-security 命令，可以验证全局端口安全配置参数（global port security configuration parameters）。下面展示了默认值下的此命令的打印输出。

```
VTP-Server-1#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
(Count) (Count) (Count)
-----
Gi0/2 1 1 0 Shutdown
-----
Total Addresses in System : 1
Max Addresses limit in System : 1024
```

如同上面的输出中所见到的那样，默认情况下，每个端口上仅允许一个安全 MAC 地址。此外，在出现冲突事件时的默认动作就是关闭端口。粗体文本表明，已知仅有一个安全地址，就是配置在接口上的静态地址。经由执行 show port-security interface [name] 亦可确认同样的参数，如下面的输出所示。

```
VTP-Server-1#show port-security interface gi0/2
Port Security : Enabled
Port status : SecureUp
Violation mode : Shutdown
Maximum MAC Addresses : 1
Total MAC Addresses : 1
Configured MAC Addresses : 1
Sticky MAC Addresses : 0
Aging time : 0 mins
Aging type : Absolute
SecureStatic address aging : Disabled
Security Violation count : 0
```

**注意：**在我们进一步学习本章内容的过程中，将会详细介绍对上面的输出中其它默认参数的修改。

而要查看该端口上具体配置的静态安全 MAC 地址，就要用到 `show port-security address` 或者 `show running-config interface [name]` 命令了。以下输出演示了 `show port-security address`。

```
VTP-Server-1#show port-security address
Secure Mac Address Table

-----  

Vlan      Mac Address        Type          Ports      Remaining Age  

          (mins)  

-----  

  1       001f.3c59.d63b    SecureConfigured   Gi0/2      -  

-----  

Total Addresses in System : 1
Max Addresses limit in System : 1024
```

## 动态安全 MAC 地址的配置，Configuring Dynamic Secure MAC Address

默认情况下，当某个端口的端口安全开启时，该端口无需管理员做任何配置，就会动态学习并将学到的 MAC 地址设为安全 MAC 地址。而要令到该端口学到不止一个 MAC 地址并将它们设为安全 MAC 地址，就要使用命令 `switchport port-security maximum [number]`。记住 `[number]` 关键字是依平台而定的，在不同的思科 Catalyst 交换机型号上会有所不同。



### 现实场景部署，Real-World Implementation

在使用思科 Catalyst 3750 交换机的生产网络中，事先确定下来某台特定交换机的用途，总是一个好主意，接着就可以通过全局配置命令（global configuration command）`sdm prefer {access | default | dual-ipv4-ipv6 {default | routing | vlan} | routing | vlan} [desktop]`，选定恰当的交换机数据库（Switch Database Management, SDM）模板。

各个 SDM 模板以支持正在或即将用到的那些特性的最优方式，来分配系统资源。默认的 SDM 模板则是尝试在各项特性间提供一种平衡。因此，会影响其它各项特性和功能的最大性能值。一个例子就是在采行端口安全时，所能学到或配置上的安全 MAC 地址最大可能数会减少。

下面的输出演示了怎样将交换机端口，接口 GigabitEthernet0/2，配置为动态学习并将至多两个 MAC 地址设为安全 MAC 地址。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport mode access
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security maximum 2
```

## 验证动态 MAC 地址保全，Verifying Dynamic Secure MAC Addressed

可用除了 `show running-config` 命令外的，在静态地址保全配置示例中用到的同样命令，来验证动态 MAC 地址保全的配置。这是因为，与静态或粘滞的 MAC 地址保全不同，所有动态学习到的地址是不保存在交换机配置文件中的，且在端口关闭后会被移除。那些同样的地址也要在端口再度开启后重新学习。下面的输出演示了 `show port-security address` 命令的输出，现实了一个配置为动态 MAC 地址保全学习的接口。

```
VTP-Server-1#show port-security address
Secure Mac Address Table
-----
Vlan     Mac Address          Type      Ports      Remaining Age
                                         (mins)
-----
1       001d.09d4.0238        SecureDynamic   Gi0/2      -
1       001f.3c59.d63b        SecureDynamic   Gi0/2      -
-----
Total Addresses in System : 2
Max Addresses limit in System : 1024
```

## 配置保全 MAC 地址粘滞，Configuring Sticky Secure MAC Addresses

下面的输出演示了如何来在某个端口上配置动态粘滞学习，以及限制端口学习到至多 10 个的 MAC 地址。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport
VTP-Server-1(config-if)#switchport mode access
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security mac-address sticky
VTP-Server-1(config-if)#switchport port-security maximum 10
```

默认情况下，基于上述配置，在接口 GigabitEthernet0/2 将会动态学到至多 10 个地址，并添加进交换机当前配置中去。在开启粘滞地址学习后，各个端口上学到的 MAC 地址被自动保存到当前配置文件，同时加入到地址表中。下面的输出显示了接口 `GigabitEthernet0/2` 上所自动学到的 MAC 地址（以粗体显示）。

```
VTP-Server-1#show running-config interface GigabitEthernet0/2
Building configuration...
Current configuration : 550 bytes
!
interface GigabitEthernet0/2
switchport
switchport mode access
switchport port-security
switchport port-security maximum 10
switchport port-security mac-address sticky
switchport port-security mac-address sticky 0004.c16f.8741
switchport port-security mac-address sticky 000c.ce47.f3a0
switchport port-security mac-address sticky 0013.1986.0a20
switchport port-security mac-address sticky 001d.09d4.0238
switchport port-security mac-address sticky 0030.803f.ea81
...
...
```

上面输出中粗体的 MAC 地址都是动态学到的，且被加入到当前配置文件中了。而无需管理员手动配置来将这些地址加入到配置文件。默认情况下，粘滞 MAC 地址保全并不是自动加入到启动配置文件（the startup configuration, NVRAM）中去的。而为确认此信息已被保存到 NVRAM 中，也就是这些地址不要在交换机重启后重新学习，就要记住执行 `copy running-config startup-config` 命令，或者命令 `copy system:running-config nvram:startup-config`，执行二者中的哪一条，取决于部署该特性的那台交换机的 IOS 版本。下面的输出演示了在配置了粘滞地址学习的端口上的 `show port-security address` 命令。

```
VTP-Server-1#show port-security address
Secure Mac Address Table
-----
Vlan   Mac Address        Type      Ports    Remaining Age
                                         (mins)
-----
1     0004.c16f.8741    SecureSticky Gi0/2    -
1     000c.ce47.f3a0    SecureSticky Gi0/2    -
1     0013.1986.0a20    SecureSticky Gi0/2    -
1     001d.09d4.0238    SecureSticky Gi0/2    -
1     0030.803f.ea81    SecureSticky Gi0/2    -
-----
Total Addresses in System : 5
Max Addresses limit in System : 1024
```

你还可以在交换机上设置一个老化时间和类型（an aging time and type），不过这是超出 CCNA 要求的。（如你愿意可以自己试试。）

## 配置端口安全冲突的动作， Configuring the Port Security Violation Action

和早前指出的那样，思科 IOS 软件允许管理员指定于出现冲突时可采取的 4 种不同动作，如下所示。

- 保护动作，Protect
- 端口关闭动作（默认），Shutdown(default)
- 限制动作，Restrict
- 关闭 VLAN 动作（CCNA 大纲不要求），Shutdown VLAN (this is outside the CCNA syllabus)

使用接口配置命令 `switchport port-security [violation {protect | restrict | shutdown | shutdown vlan}]` 来配置这些选项。如果某个端口因为一个安全冲突而关闭，它就显示为 `errdisabled`，此时需要使用 `shutdown` 和接着的 `no shutdown` 命令来将其再度开启。

```
Switch#show interfaces FastEthernet0/1 status
Port Name      Status        Vlan   Duplex   Speed    Type
Fa0/1         errdisabled     100     full     100    100BaseSX
```

思科要求你知道何种冲突动作引起发出给网络管理员一条 SNMP 消息以及产生一条日志消息，下面的表 4.2 是你所要的信息。

模式	端口动作	流量	系统日志	冲突计数器
保护模式	端口是受保护的	抛弃未知 MAC 地址的帧	不会记录	不会增加
关闭模式	端口出错关闭	关闭流量转发	记录日志并发出一条 SNMP trap 消息	计数器增加
限制模式	端口开放	超出数量的那些 MAC 流量被拒绝	记录日志并发出一条 SNMP trap 消息	计数器增加

为顺利通过考试，你务必要记住这个表！

下面的输出演示了如何在某个端口上配置粘滞地址学习最多 10 个 MAC 地址。如果端口上探测到某未知 MAC 地址（比如第 11 个 MAC 地址）时，端口将被配置为丢弃收到的那些帧。

```
VTP-Server-1(config)#interface GigabitEthernet0/2
VTP-Server-1(config-if)#switchport port-security
VTP-Server-1(config-if)#switchport port-security mac-address sticky
VTP-Server-1(config-if)#switchport port-security maximum 10
VTP-Server-1(config-if)#switchport port-security violation restrict
```

## 对端口安全冲突动作的验证，Verifying the Port Security Violation Action

是通过命令 `show port-security` 命令，来对所配置的端口安全冲突动作进行验证的，如下面的输出所示。

```
VTP-Server-1#show port-security
Secure Port      MaxSecureAddr      CurrentAddr      SecurityViolation      Security Action
                  (Count)          (Count)          (Count)
Gi0/2            10                5                0                    Restrict
Total Addresses in System : 5
Max Addresses limit in System : 1024
```

如交换机上开启了日志记录，同时配置了限制模式（Restrict mode）或关闭模式（Shutdown mode），类似于下面输出的这些消息将会在控制台打印出来，并记录到本地缓存或者发往某台日志服务器。

```
VTP-Server-1#show logging
...
[Truncated Output]
...
04:23:21: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 0013.1986.0a20 on port Gi0/2.
04:23:31: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 000c.cea7.f3a0 on port Gi0/2.
04:23:46: %PORT_SECURITY-2-PSECURE_VIOLATION: Security violation occurred, caused by MAC
address 0004.c16f.8741 on port Gi0/2.
```

最后要说明的一点是在 Packet Tracer 上可以配置交换机安全，但许多命令及 `show` 命令不会工作。

## 第四天的问题，Day 4 Questions

1. Write out the two ways of configuring console passwords. Write the actual commands.
2. Which command will permit only SSH traffic into the VTY lines?
3. Which command will encrypt a password with level 7 encryption?
4. Name the eight levels of logging available on the router.
5. Why would you choose SSH access over Telnet?
6. Your three options upon violation of your port security are protect, \_\_\_\_\_, and \_\_\_\_\_.
7. How would you hard set a port to accept only MAC 0001.c74a.0a01?
8. Which command turns off CDP for a particular interface?
9. Which command turns off CDP for the entire router or switch?
10. Which command adds a password to your VTP domain?
11. Which command would permit only VLANs 10 to 20 over your interface?

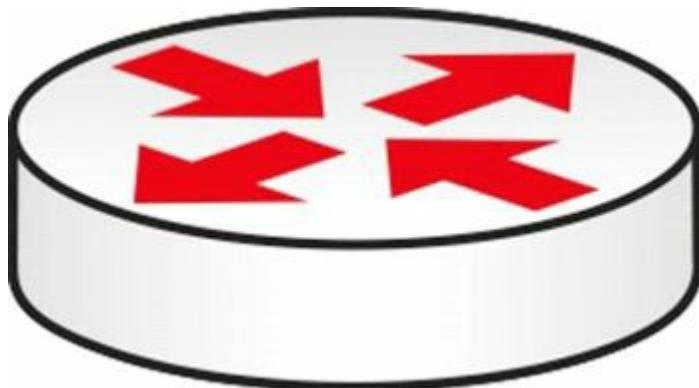
## 第四天问题的答案，Answers

1. The `password xxx` and `login local` commands (username and password previously configured).
2. The `transport input ssh` command.
3. The `service password-encryption` command.
4. Alerts, critical, debugging, emergencies, errors, informational, notifications, and warnings.
5. SSH offers secure, encrypted traffic.
6. Shutdown and restrict.
7. Issue the `switchport port-security mac-address x.x.x.x` command.
8. The `no cdp enable` command.
9. The `no cdp run` command.
10. The `vtp password xxx` command.
11. The `switchport trunk allowed vlan 10-20` command.

## 第四天实验，Day 4 Labs

### 路由器安全基础实验，Basic Router Security Lab

拓扑图，Topology



## 实验目的, Purpose

学习一些给路由器上锁所需的基本步骤。

## 实验步骤, Walkthrough

1. 使用某个启用秘密口令 (an enable secret password) , 登入使用保护启用模式 (Protect Enable mode) 。通过登出特权模式 (Privileged mode) 并再度登入来进行测试。

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#enable secret cisco
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#exi
Router con0 is now available
Press RETURN to get started.
Router>en
Password:
Router#
```

1. 设置一个启用口令 (enable password) , 接着加入口令加密服务 (service password encryption) 。此操作在实际路由器上很少执行, 因为这是不安全的做法。

```
Router(config)#no enable secret
Router(config)#enable password cisco
Router(config)#service pass
Router(config)#service password-encryption
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show run
Building configuration...
Current configuration: 480 bytes
!version 12.4
no service timestamps log datetime msec
no service timestamps debug datetime msec
service password-encryption
!
hostname Router
!
enable password 7 0822455D0A16
```

1. 对 Telnet 线路进行保护。建立一个本地用户名及其口令, 并令到用户在登入路由器时, 使用此用户名和口令。

```
Router(config)#line vty 0 ?
<1-15>
Last Line number
<cr>
Router(config)#line vty 0 15
Router(config-line)#login local
Router(config-line)#exit
Router(config)#username in60days password cisco
Router(config)#
```

之前你已经测试过 Telnet 了, 但请无需担心在加入一台 PC 及 Telnet 到路由器, 会受到要求用户名和口令的提示。

1. 用一个口令来保护控制台。只需在控制台端口上直接设置一个口令就行。

```
Router(config)#line console 0
Router(config-line)#password cisco
```

通过将控制台线从路由器拔出，并再次插入路由器，就可以对此进行测试。同样，如有一个替代端口，也可为其设置口令进行保护。

```
Router(config)#line aux 0
Router(config-line)#password cisco
```

1. 通过仅允许 SSH 流量进入，来保护 Telnet 线路。还可以仅允许 SSH 流量发出。该命令需要一个安全镜像（a security image）才能工作。

```
Router(config)#line vty 0 15
Router(config-line)#transport input ssh
Router(config-line)#transport output ssh
```

1. 添加一个今日横幅消息（a banner message of the day, MOTD）。将告知路由器已结束输入的字符设为 "X"（界定符，the delimiting character）。

```
Router(config)#banner motd X
Enter TEXT message.
End with the character 'X'.
Do not use this router without authorization. X
Router(config)#
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Exit
Router con0 is now available
Press RETURN to get started.
Do not use this router without authorization.
Router>
```

1. 关闭整个路由器的思科发现协议。还可以使用命令 `no cdp enable interface`，只关闭某个接口上的思科发现协议。

```
Router(config)#no cdp run
```

可通过在关闭思科发现协议前，连接一台交换机或路由器到该路由器，并执行 `show cdp neighbor (detail)` 命令，来测试上面的命令是否起作用。

1. 设置路由器将日志消息发送到网络上的某台主机。

```
Router#conf t
Enter configuration commands, one per line.
End with CNTL/Z.

Router(config)#logging ?
  A.B.C.D      IP address of the logging host
  buffered     Set buffered logging parameters
  console      Set console logging parameters
  host         Set syslog server IP address and parameters
  on           Enable logging to all enabled destinations
  trap         Set syslog server logging level
  userinfo    Enable logging of user info on privileged mode enabling

Router(config)#logging 10.1.1.1
```

## 交换机安全基础实验，Basic Switch Security Lab

### 拓扑图，Topology



### 实验目的，Purpose

学习一些给交换机上锁的基本步骤。

### 实验步骤，Walkthrough

1. 连接一台 PC 或笔记本计算机到交换机。另外为后面的配置建立一个控制台连接。连接 PC 的那个以太网端口，将会作为本实验中配置安全设置的那个端口。我所选择的是交换机的 `FastEthernet0/1` 端口。
2. 登入 VTY 线路，并建立使用本地用户名和口令的远程登陆访问（Telnet access referring to a local username and password）。

```
Switch#conf t
Enter configuration commands, one per line. End with CNTL/Z.

Switch(config)#line vty 0 ?
  <1-15>  Last Line number
  <cr>
Switch(config)#line vty 0 15
Switch(config-line)#
Switch(config-line)#login local
Switch(config-line)#exit
Switch(config)#username in60days password cisco
Switch(config)#

```

1. 为交换机上的 `VLAN 1` 添加一个 IP 地址（所有端口都自动在 `VLAN 1` 中）。此外，将 `192.168.1.1` 加到 PC 的 `FastEthernet` 接口上。

```
Switch(config)#interface vlan1
Switch(config-if)#ip address 192.168.1.2 255.255.255.0
Switch(config-if)#no shut
%LINK-5-CHANGED: Interface Vlan1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan2, changed state to up
Switch(config-if)#^Z -- press Ctrl+Z keys
Switch#
Switch#ping 192.168.1.1 -- test connection from switch to PC
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 31/31/32 ms
Switch#
```

1. 通过从 PC 远程登陆到交换机来测试远程登陆。

### Command Prompt

```
PC>telnet 192.168.1.2
Trying 192.168.1.2 ...Open
```

```
User Access Verification
```

```
Username: in60days
```

```
Password:
```

```
Switch>
```

1. IT 经理改变主意，要仅使用 SSH 访问，那么就在 VTY 线路上修改配置。仅有那些确定的交换机型号和 IOS 版本才支持 ssh 命令。

```
Switch(config)#line vty 0 15
Switch(config-line)#transport input ssh
```

1. 现在从 PC 尝试登入交换机。因为仅允许 SSH，此连接将失败。

### Command Prompt

```
Packet Tracer PC Command Line 1.0
```

```
PC>telnet 192.168.1.2
```

```
Trying 192.168.1.2 ...Open
```

```
[Connection to 192.168.1.2 closed by foreign host]
```

```
PC>
```

1. 在交换机上为 FastEthernet 端口设置端口安全。如你未将端口设置为接入模式（而是动态模式或者中继模式）的话，此操作将失败。

```

Switch(config)#interface FastEthernet0/1
Switch(config-if)#switchport port-security
Command rejected: FastEthernet0/1 is a dynamic port.
Switch(config-if)#switchport mode access
Switch(config-if)#switchport port-security
Switch(config-if)#

```

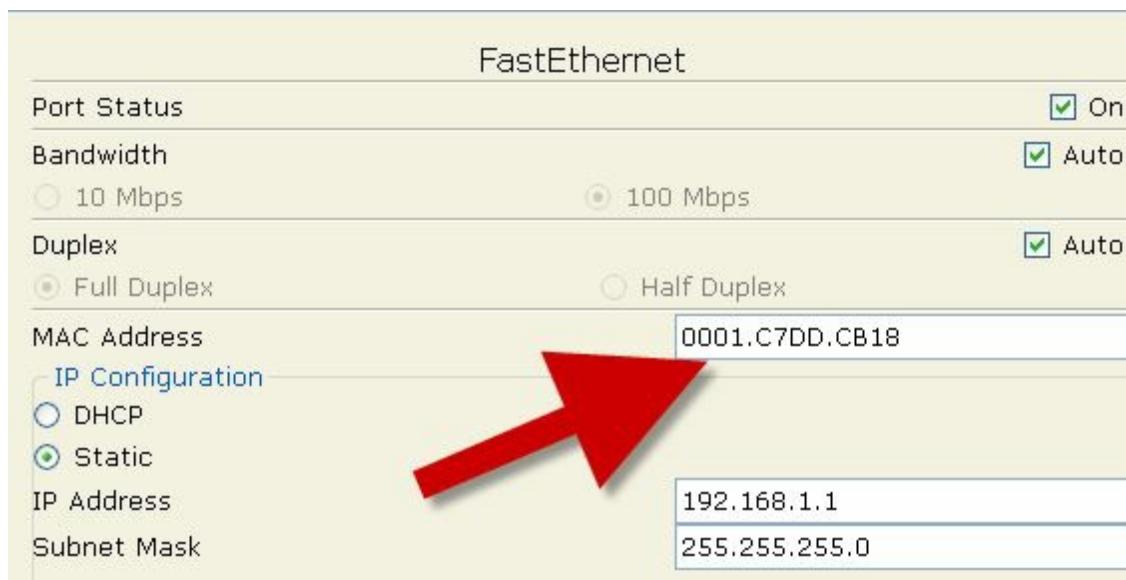
1. 硬性设置 PC 的 MAC 地址为该端口的允许地址。在 PC 的命令行上使用命令 `ipconfig/all` 来查看其 MAC 地址。再就要检查端口安全的状态和设置了。

```

Switch(config-if)#switchport port-security mac-address 0001.C7DD.CB18
Switch(config-if)#^Z
Switch#show port-security int FastEthernet0/1
Port Security          : Enabled
Port Status             : Secure-up
Violation Mode         : Shutdown
Aging Time              : 0 mins
Aging Type              : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses   : 1
Total MAC Addresses     : 1
Configured MAC Addresses: 0
Sticky MAC Addresses    : 0
Last Source Address:Vlan : 0001.C7DD.CB18:1
Security Violation Count: 0

```

1. 修改 PC 的 MAC 地址，如你无法修改，可以将另一台设备插入该交换机端口。这将会令到该端口关闭，因为破坏了安全设置。下面的屏幕截图展示了 Packet Tracer 中修改 MAC 地址的地方。



1. 你将看到 FastEthernet 端口立即宕掉。

```
Switch#
%LINK-5-CHANGED: Interface FastEthernet0/1, changed state to administratively down
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed state to down
%LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to down
Switch#
%SYS-5-CONFIG_I: Configured from console by console
Switch#show port-security interface FastEthernet0/1
Port Security : Enabled
Port Status : Secure-shutdown
Violation Mode : Shutdown
Aging Time : 0 mins
Aging Type : Absolute
SecureStatic Address Aging : Disabled
Maximum MAC Addresses : 1
Total MAC Addresses : 0
Configured MAC Addresses : 0
Sticky MAC Addresses : 0
Last Source Address:Vlan : 0001.C7DD.CB19:1
Security Violation Count : 1
```

**注意：**请重复本实验，直到理解这些命令，并在不看上述实验步骤的情况下输入这些命令为止（本书的其它实验也要这样做）。

# 第5天 IP 地址分配

## IP Addressing

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第五天的任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读 ICND1 记诵指南
- 花 15 分钟浏览 [subnetting.org](http://subnetting.org) 网站

欢迎来到今天的学习，许多人都发现今天的内容是 CCNA 大纲中最难掌握的部分之一。为理解 CCNA 考试的 IP 分址，我们必须涵盖二进制运算及十六进制计数系统 (binary mathematics and the hexadecimal numbering system)、地址类别(classes of addresses)、2 的指数 (powers of two) 和诸如零号子网 (subnet zero) 等规则，以及广播地址与网络地址，还有用于计算子网地址和主机地址的公式。

尽管有这些难点，但请无需担心；这都是有个过程的，而不是一蹴而就的，所以请跟上课文，就一定会发现在本书中屡次回顾到这些概念。

今天会学到这些内容。

- IP 分址（采用二进制和十六进制），IP addressing (using binary and hexadecimal)
- IP 地址的使用，Using IP addresses
- 子网划分，Subnetting
- 简易子网划分，Easy subnetting
- 网络规划设计，Network design
- 采用 VLSM，Using VLSM
- 切分网络，Slicing down networks

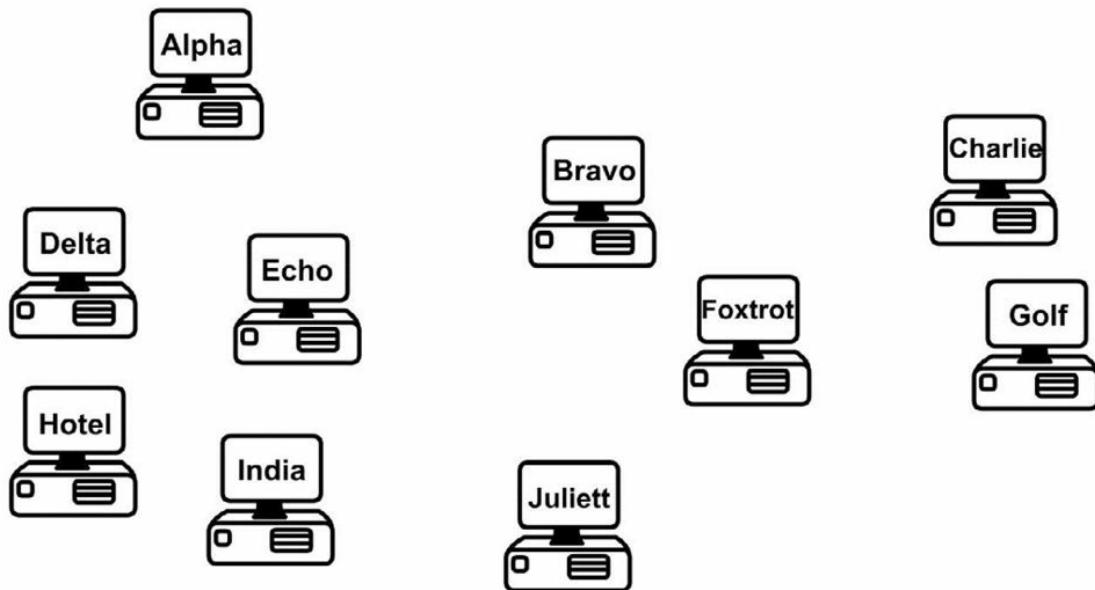
本模块对应的是 CCNA 大纲要求的以下部分。

- 描述 IPv4 分址中使用私有和公共 IP 地址的做法和必要性，Describe the operationg and necessity of using private and public IP addresses for IPv4 addressing
- 找出采行 VLSM 和汇总技术，用以满足某个 LAN/WAN 环境中分址要求的恰当 IPv4 分址方案，Identify the appropriate IPv4 addressing scheme using VLSM and summarisation to satisfy addressing requirements in a LAN/WAN environment
- 对有关 IP 分址和主机配置有关的故障进行排除和修正，Troubleshoot and correct common problems associated with IP addressing and host configurations

思科已经将一些 VLSM 要求加入到 ICND1 和 ICND2 考试中了。而在 ICND2 考试中看起来考得更多一些，不过两个考试都需要你做好解答问题的准备。在掌握 VLSM 前，你需要先理解 IP 分址和子网划分。

## IP 分址，IP addressing

网络上的所有设备，都需要某种方法来将其识别为某台特定主机。早期网络简单地采用某种命名格式，同时服务器上维护着一张 MAC 地址与主机名称的映射图。服务器上的表格迅速增长，伴随其产生诸如一致性及准确性（consistency and accuracy）问题（如图 5.1）。IP 分址有效地解决了此问题。



## IP 版本 4, IP Version 4

IP 版本 4(IPv4) 设计用于解决设备命名问题。IPv4 使用二进制在网络设备上应用一个地址。它使用 32 位二进制数，分成每 8 位的 4 组。下面是二进制的 IPv4 地址的一个实例。

11000000.10100011.11110000.10101011

以十进制来看，就是。

192.168.240.171

每一个二进制位表示一个十进制数，你可以在相应的列中，依据该列是 1 还是 0，而使用或不用其对应的十进制数。下面是 8 个列。

128	64	32	16	8	4	2	1
1	1	0	0	0	0	0	0

从上表中可以看到，仅有前两个十进制数被用到（下方有 1 的两个），这就产生出数值  $128+64=192$ 。

## 二进制，Binary

为理解 IP 寻址(IP addressing)的工作原理，你需要理解二进制计数法（binary mathematics）。计算机及网络设备是不理解十进制的。我们使用 10 进制，是由于它是一种使用了 10 个数字的计数系统，由很久很久以前的穴居人类在意识到他有 10 个手指头，可以在有恐龙经过洞口时用来数恐龙时发明的。

计算机和网络设备只明白电信号。而电信号不是开就是关，唯一可用的计数系统就是二进制。二进制只用到两个数字，`0` 和 `1`。`0` 表示线路上没有电脉冲，而 `1` 就表示线路上有一个脉冲。

使用二进制值，可以生成任何数字。加的二进制数值越多，得到的数量就越大。所加入的每个二进制值，其下一个数字都要是它的两倍（也就是，`1` 到 `2` 到 `4` 到 `8` 到 `16`，以致无穷），从右往左。如有两位，最多可计到 `3`。只需将 `0` 或 `1` 放入到表格的列中，以确定是否要使用该列的值。

我们从仅有两位的二进制数开始。

<code>2</code>	<code>1</code>
<code>0</code>	<code>0</code>

$$0+0=0$$

<code>2</code>	<code>1</code>
<code>0</code>	<code>1</code>

$$0+1=1$$

<code>2</code>	<code>1</code>
<code>1</code>	<code>0</code>

$$2+0=2$$

<code>2</code>	<code>1</code>
<code>1</code>	<code>1</code>

$$2+1=3$$

如你使用 8 位二进制数（也就是一个八位字节），你能取得如何从 `0` 到 `255` 之间的数值。而你可以看到，这些位数自右往左移动。

<code>128</code>	<code>64</code>	<code>32</code>	<code>16</code>	<code>8</code>	<code>4</code>	<code>2</code>	<code>1</code>

在往各列中填入 `0` 时，就有了十进制的 `0`。

<code>128</code>	<code>64</code>	<code>32</code>	<code>16</code>	<code>8</code>	<code>4</code>	<code>2</code>	<code>1</code>
<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>	<code>0</code>

而将 `1` 填入各列，就得到了十进制的 `255`。

<code>128</code>	<code>64</code>	<code>32</code>	<code>16</code>	<code>8</code>	<code>4</code>	<code>2</code>	<code>1</code>
<code>1</code>	<code>1</code>	<code>1</code>	<code>1</code>	<code>1</code>	<code>1</code>	<code>1</code>	<code>1</code>

不信吗？

$$128+64+32+16+8+4+2+1=255$$

如此，逻辑使然，你实际上可以通过将 `0` 或 `1` 放入不同的列，而生成 `0` 到 `255` 之间的任何数值。比如。

<code>128</code>	<code>64</code>	<code>32</code>	<code>16</code>	<code>8</code>	<code>4</code>	<code>2</code>	<code>1</code>
<code>0</code>	<code>0</code>	<code>1</code>	<code>0</code>	<code>1</code>	<code>1</code>	<code>0</code>	<code>0</code>

$$32+8+4=44$$

**上面的基础知识，是IP寻址和子网划分的基础。** 下面的表5.1对你现在所掌握的进行了总结。这些值可用作任意子网掩码，所以请留心一下。

**表 5.1 -- 二进制值, Binary Values**

二进制, Binary	十进制, Decimal
1000 0000	128
1100 0000	192
1110 0000	224
1111 0000	240
1111 1000	248
1111 1100	252
1111 1110	254
1111 1111	255

构造一些你自己的二进制数，确保你完全地掌握了这个概念。

## 十六进制, Hexadecimal

十六进制 (hex) 是另一个替代的计数系统。比起以2或10来计数，它用到16个数字或字母。十六进制从0开始知道F，如下面所示。

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

每位十六进制数实际上代表的是 4 位二进制数，如表5.2所示。

**表5.2 -- 十进制、十六进制和二进制位数, Decimal, Hex, and Binary Digits**

十进制, Decimal	`0`	`1`	`2`	`3`	`4`	`5`	`6`	`7`
十六进制, Hex	`0`	`1`	`2`	`3`	`4`	`5`	`6`	`7`
二进制, Binary	`0000`	`0001`	`0010`	`0011`	`0100`	`0101`	`0110`	`0111`
十进制, Decimal	`8`	`9`	`10`	`11`	`12`	`13`	`14`	`15`
十六进制, Hex	`8`	`9`	`A`	`B`	`C`	`D`	`E`	`F`
二进制, Binary	`1000`	`1001`	`1010`	`1011`	`1100`	`1101`	`1110`	`1111`

将二进制转换成十六进制及十进制，是相当简单的，如表5.3所示。

**表5.3 -- 二进制到十六进制、十进制的转换**

十进制, Decimal	`13`	`6`	`2`	`12`
十六进制, Hex	`D`	`6`	`2`	`C`
二进制, Binary	`1101`	`0110`	`0010`	`1100`

相对于二进制，十六进制对人类来讲更易于掌握，其又能够近似于二进制那样为计算机和网络设备所使用。任何的数都可由十六进制构造出来，这点跟二进制和十进制一样；如下面的例子一样，只需计算  $_{16}$  的乘积。

```
1 x 16 = 16
16 x 16 = 256
16 x 16 x 16 = 4096
```

如此等等。

十六进制, Hex	`4096`	`256`	`16`	`1`
			`1`	`A`

在以十六进制数数时，就要像这样， 0 1 2 3 4 5 6 7 8 9 A B C D E F 10 11 12 13 14 15 16 17 18 19 1A 1B 1F 1E 1F 20 21 22，等等，一直到无穷。比如上面的 1A，就是在 1 的列上有个 A，在 16 的列上有个 1，那么：  $1A = 10 + 16 = 26$ 。

在将二进制转换成十六进制时，如你将 8 位的二进制数分为 4 位一组的两组，就变得相当容易了。如此一来，二进制数 11110011 就成了 1111 0011。1111 就是  $8 + 4 + 2 + 1 = 15$ ，而 0011 就是  $2 + 1 = 3$ 。15 就是十六进制的 F，3 就是 3，所以答案就是 F3。你可以通过表 5.2 来验证这点。

而十六进制到二进制的转换，其过程与此一致。比如，7C 可分解为 7，也就是二进制的 0111，及 C（十进制的 12），它是二进制的 1100。答案就是 01111100。

## 转换练习，Converting Exercise

这里有些你可以试做的例题。在进行计算前，先写出上面的表格（也就是显示 1 这列，接着是 16 这列，再是 256 列，等等）。

1. 将 1111 转换成十六进制和十进制。
2. 将 11010 转换成十六进制和十进制。
3. 将 10000 转换成十六进制和十进制。
4. 将 20 转换成二进制和十六进制。
5. 将 32 转换成二进制和十六进制。
6. 将 101 转换成二进制和十六进制。
7. 将 A6 从十六进制转换成十进制和二进制。
8. 将 15 从十六进制转换成十进制和二进制。
9. 将 B5 从十六进制转换成十进制和二进制。

在考试中，写出表5.2，有助于你完成三种进制之间的转换。

IP 地址分配的规则有：网络上的每个地址，都要是其主机所唯一的（也就是说 IP 地址不能共用）。一些地址不能用作主机地址。这将在后面的章节涉及，但在这里，要知道为整个网络保留的那个地址，也就是广播地址，以及保留用于测试目的那些地址，此外，有三组保留的用于内部网络的地址（此举正是为节省 IP 地址），是不能使用的。

由于网络规模的迅速增长，每个IP地址就必须与一个子网掩码配合使用。子网掩码是要告诉网络设备，怎样来使用IP地址中的数字。而此举的用意，就是可以借用地址中的主机位，将网络切割为更小的子网。

这里有个带子网掩码的IP地址实例， 192.168.1.1 255.255.255.0。

## 地址类别，Address Classes

你是要掌握这个的，却没有掌握吧。我知道我是不能帮你太多的，但地址类别实际上是明显过时的了，所以作为一名思科工程师，当你在见到这种老规矩时，总是会感到迷惑，却还要把这些规则用到网络设计中去。

现在我们仍然将IP地址组别叫做类（classes），但随着子网掩码和变长子网掩码（Variable-Length Subnet Masking, VLSM）概念的引入，地址类实际上已不再适用于网络设计了。掌握地址类别仍然是有用的，因为类别的不同可以让我们清楚，在小型网络（子网）中，可以使用哪些IP地址，而不能使用另一些。

在IPv4刚推出时，其地址就分成了不同类别。不同地址类别依其需求而分配给各家机构。机构越大，地址类别就越大。不同地址类别又指定了相应字母，从 A 到 E。A 类地址保留给最大的一些网络。而 A 类地址的前 8 个二进制位可以是从 1 到 126 的数。此举的原因在于其前8位的首位必须是 0。而当前 8 位中有了第一位的 0 时，那么剩下的值就只能是 1 到 126 了。也就是下面这样。

0000 0001 = 1

0111 1111 = 127

**在网络中，是不可以有全 0 地址的。**在加入其它三个 8 位二进制数后，就可以看到 A 类地址的全貌了。就像下面的那样。

10.1.1.1

120.2.3.4

126.200.133.1

这些都是 A 类地址，因为它们都是在 1 到 126 的范围内。**127 不是IP地址所允许的数字； 127.0.0.1 实际上用于在设备上测试TCP/IP是否正常。**

B 类地址前 8 位二进制数的头两位则必须是10。这就意味着前 8 位二进制数值处于 128 到 191 之间，也就是下面这样。

1000 0000 = 128

1011 1111 = 191

对于 C 类地址来说，前 8 位二进制数的头三位必需为 110，那么地址就在 192 到 223 之间，也就是下面这样。

1100 0000 = 192

1101 1111 = 223

D 类地址用于多播（multicasting, directed broadcasting, 受导向的广播），而 E 类地址则仅用于实验用途。

## 子网掩码初步，Subnet Mask Primer

先前提到过IP地址用于区分网络的部分以及用于区分网络上主机地址的部分。子网掩码的作用就是确立此两部分。难点就在于并不总是能仅仅看一眼子网掩码，就能知道IP地址的网络部分和主机部分。这需要实践，且对于那些更难的地址，你就必须要动手计算出来（或是使用某个子网计算程序来作弊）。

就算未曾将网络划分成更小的部分，你仍需采用为用到的每个地址应用一个子网掩码。而上面提到的地址类，它们都有一个默认的子网掩码，如同下面这样。

A 类地址 = 255.0.0.0 B 类地址 = 255.255.0.0 C 类地址 = 255.255.255.0

在二进制位开启时，网络就知道该位是用作网络地址，而不是网络上的主机地址，如下表所示。

192	168	12	2
255	255	255	0
网络位	网络位	网络位	主机位

上面的地址表明 192.168.12 是网络地址， 2 是该网络上的一台主机。再者，任何以 192.168.12 开头的IP 地址，都是在同一网络上的。而在看看前 8 位的数字，以及该默认的子网掩码，就知道这是一个 c 类网络。

请记住早前提到的规则：主机所不能使用的那些网络号，那么下面的这些网络号就不能为设备所使用了。

10.0.0.0

174.12.0.0

192.168.2.0

另一规则是你不能使用各个网络或子网上的广播地址。某广播地址是前往网络上所有设备，那么，逻辑上就不能为设备所使用了。广播地址就是将所有主机位开启的地址，像下面这样。

10.255.255.255

192.168.1.255

在上面的例子中，主机部分的所有二进制位都是打开的。

## IP地址的使用，Using IP Addresses

接下来就是IP地址使用实务了，在这里我们要探讨一下哪些可以使用，哪些又不能使用。

你知道在过去二十年中计算机的使用曾有一个大暴发。个人计算机曾是十分昂贵的物品，以致只有少数人才买得起；因此只有那些有钱的机构才会保有使用。今天，几乎每个家庭都有那么一台或几台计算机了。

问题就在于IPv4实在仅有少数设备投入使用时发明的，且那时未曾预期到会有如此大的变化。在地址分配时，就意识到了如今的增长率，我们将很快用完可用的地址。

## 私有IP地址，Private IP Addresses

几种解决方案之一就是保留一些类别的地址给那些要用的人，同时这些地址不再国际互联网上使用。这些地址就是私有IP地址，而此方案是由 1918 和 4193 两个RFC所构建的。

下面就是私有地址的几个范围。

10.x.x.x -- 以10开头的地址

172.16.x.x 到 172.31.x.x -- 172.16 到 172.31 中的那些地址

192.168.x.x -- 以 192.168 开头的那些地址

## 子网划分，Subnetting

子网划分让我们可以从一般用于网络上的主机位的那些IP地址位中，进行借用。此时就可以自较大的网络空间，划出一些更小的网络了，这些较小的网络，就被成为子网（subnetworks, 简写为subnets）。

在对三类可用地址应用默认子网掩码时，你会发现不能用于划分子网的地址部分，如下面的表格所示。

A 类 -- 255	0	0	0
不能使用	可以使用	可以使用	可以使用
B类 -- 255	255	0	0
不能使用	不能使用	可以使用	可以使用
C类 -- 255	255	255	255
不能使用	不能使用	不能使用	可以使用

比如，如你将某个C类网络以默认子网掩码方式使用，那么就是这样的。

IP地址	`192`	`168`	`1`	`0`	
子网掩码	`255`	`255`	`255`	`0`	
二进制形式	`1111 1111`	`1111 1111`	`1111 1111`	0000 0000	

在从后 8 位二进制数借用 2 位后，就会得到下面的子网，每个子网有 62 台主机。

网络号	网络号	网络号	子网号	主机	广播地址
192	168	1	0	1-62	63
192	168	1	64	65-126	127
192	168	1	128	129-190	191
192	168	1	192	193-254	255

在较大的网络中，你原来可以使用到 1 至 254 的主机号，这样看来，在进行了子网划分后，可用的主机号减少了，但得到的是更多的网络数。下面的表说明了 4 个子网是怎么得来的。

128	64	32	16	8	4	2	1	子网号
0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	64
1	0	0	0	0	0	0	0	128
1	1	0	0	0	0	0	0	192

考虑二进制数学，你能发现对主机位的头两位使用，就可以使用 00、01、10 和 11 四种组合，在将其写出后，在子网列就得到子网 0、64、128 以及 192 四个子网号。为更明了的表示，头两列的子网号标记为灰色，而剩下的 6 位，就用作每个子网的主机号。

如你现在觉得很绕，这是正常的。我恐怕你会有点时间来适应的。

## 简易子网划分，Easy Subnetting

在考试时，或是在现场网络上进行故障排除时，你会希望快速精确地得到结果。那么我就给出一直简易子网划分方法，是我的 Kindle 电子书“子网划分秘密/Subnetting Secrets”的内容。你无需通读该书，这里就要相关内容。

我所创建的网站 [www.subnetting.org](http://www.subnetting.org)是一个非常有用的资源，有着一些有个子网划分和网络设计的习题。

## 无类域间路由, Classless Inter-Domain Routing, CIDR

无类域间路由是由互联网工程任务组 (Internet Engineering Task Force, IETF) 创建的，作为一种分配IP地址块及路由IP数据包的方法。这里要考察的CIDR主要特性，就是以斜杠 (/) 地址表示法 (slash address notation)，来表示子网掩码。因为这可以节省时间，所以该方法是较为重要的，在现实中也要用到，而除此之外，还会考到有关CIDR地址的问题。

有了CIDR，你要将所用到的二进制位的树木写下来，以取代之前用到完整子网掩码。比如对于 255.255.0.0，它用到了两个 8 位二进制数，那么就会用 /16 来表示。又比如 255.255.240.0，使用到 8 + 8 + 4 个二进制位，就是 /20 了。

在网际互联是对子网掩码或是网络掩码的叫法，应该读作“斜杠16”或“斜杠20”，如此来与同事配合工作，而他们就能明白你说的是一个CIDR掩码了。

## 子网划分秘笈，The Subnetting Secrets Chart

此秘笈将从几个星期的子网划分纠葛中将你拯救。我（原作者）的这本秘笈，已为全世界上万的CCNA及CCNP学员所采用，他们用其通过考试，或是在工作面试中获得成功。

多年前，在我在为CCNA考试学习时偶然发现这个简易方法前，学员们都不得不将网络地址的二进制形式写下来，或是要进行痛苦地计算，来得到正确答案。

要写出秘笈所要用到的图表，你需要一只铅笔和一张纸。在考试中，因为只会给你一块白板用于计算，你需要凭记忆将该图表画出来。而在工作面试中，你是可以使用铅笔和白纸的。

在白纸的顶部右边，写下 1，再往左依次写下乘以 2 的结果，分别是 2、4、8，并一直乘以 2，直到数字 128。那么就有了一组8位二进制数了。

-	-	-	-	-	-	-	-
128	64	32	16	8	4	2	1

在 128 这个数往下，写下第一个格子里的数的和（128 的那个格子）。接着再写下到第二个格子里的数的和（64），接着到第三个（32），直到将所有 8 个格子的数加完为止。

128
192
224
240
248
252
254
255

在将两部分放在一起后，就得到了秘笈图表的上半部分了。

二进制位数	128	64	32	16	8	4	2	1
子网号								
128								
192								
224								
240								
248								
252								
254								
255								

顶上的行表示子网掩增量，而左侧的列则表示子网掩码。使用这个图表后，你就可以在数秒内回答任何子网划分的问题了。而那个可指明任何网络设计问题，诸如在以某子网掩码  $x$  划分网络时，可得到多少个子网和主机这样的问题，的答案的图表部分，只需加入 " $2$  的幂" 就行了。

其中一列会是 " $2$  的幂"，另一列就是 " $2$  的幂减去  $2$ "。减去的  $2$  的意思是要除去不能使用的两个地址，一个是网络号，另一个是广播地址。以数字  $2$  开始，乘以  $2$ ，一直到回答问题所需要的大小为止。

二进制位数	128	64	32	16	8	4	2	1
子网号								
128								
192								
224								
240			为计算出主机所在的子网 是哪一个					
248								
252								
254								
255								
	子网数	主机数 - 2						
2			为计算出有多少个子网以及每个子网有多少台主机					
4								
8								
16								
32								
64								

通过直接切入一个考试问题，可以更好的学习到子网划分。

`192.168.100.100/26` 是在那个子网中？

那么，你知道这是一个C类地址，而C类地址的默认掩码是24个二进制位，或写着 `255.255.255.0`。而这里是26位，所以有两位被借用来产生子网了。我们只需简单地在上面的秘笈图表中的顶上一行，从左往右勾上两个位置。这样就揭示出子网个数了。接着在子网号那列往下勾上两个位置，来揭示出所存在的子网掩码。

二进制位数	128	64	32	16	8	4	2	1
子网号	<input type="radio"/>	<input checked="" type="radio"/>						
128	<input checked="" type="radio"/>							
192	<input checked="" type="radio"/>							
224								
240								
248								
252								
254								
255								
	子网数	主机数 - 2						
2								
4								
8								
16								
32								
64								

现在所知道的有两件事，子网号以 `64` 递增（可将 `0` 用作首个子网号），同时子网掩码 `/26` 以 `192` 结束，那么，该结束子网掩码的完整形式为 `255.255.255.192`。

`192.168.100.0` 是第一个子网 `192.168.100.64` 是第二个子网 `192.168.100.128` 是第三个子网

`192.168.100.192` 是第四个子网

是不可以比实际的子网号有更多的了，也就是这里的 `192`。不过记住问题是要你找出主机 `100`。我们轻易地就看出子网 `64` 就是主机 `100` 所在的子网，因为下一子网是 `128`，那太高了。

下面为了知识的完整性，我加入了主机地址及广播地址。去下一子网号再减去1，就可以很快算出广播地址来。

子网	首台主机	最后的主机	广播地址
<code>192.168.100.0</code>	<code>192.168.100.1</code>	<code>192.168.100.62</code>	<code>192.168.100.63</code>
<code>192.168.100.64</code>	<code>192.168.100.65</code>	<code>192.168.100.126</code>	<code>192.168.100.127</code>
<code>192.168.100.128</code>	<code>192.168.100.129</code>	<code>192.168.100.190</code>	<code>192.168.100.191</code>
<code>192.168.100.192</code>	<code>192.168.100.193</code>	<code>192.168.100.254</code>	<code>192.168.100.255</code>

考虑到IP地址是 `0` 到 `255` 之间的任何值。（不翻译了，太简单！）

## 路由汇总，Route Summarisation

国际互联网上有数百万条路由。如果这些路由都不得不单独存储，因特网在好多年前就会停摆了。路由汇总，也就是常说的超网（supernetting），是在 RFC 1338 中提出的，点击 RFC -- [www.faqs.org/rfcs/rfc1338.html](http://www.faqs.org/rfcs/rfc1338.html) 可以读到这个 RFC。

如你要阅读一份更详尽的路由汇总文档，那么去找一本 Jeff Doyle 的卓越的思科书 *Routing TCP/IP Volume 1*，现在该书出了第二版。

### 邮编，ZIP Codes

美国邮政局用邮编来提升美国内信件到地址的路由效率（见图 5.2）。邮编的第一位表示一组的美国州份，第二和三位表示那组中的一个区域。这个想法在于可以将信件或包裹经由机器或人工快速地路由到正确的州份，并转发到相应州份。在邮件到达该州时，又可以正确地路由到相应区域。从该区域有可以正确地路由到相应城市等等，直到邮件装入当地邮政投递人员的正确邮报为止。

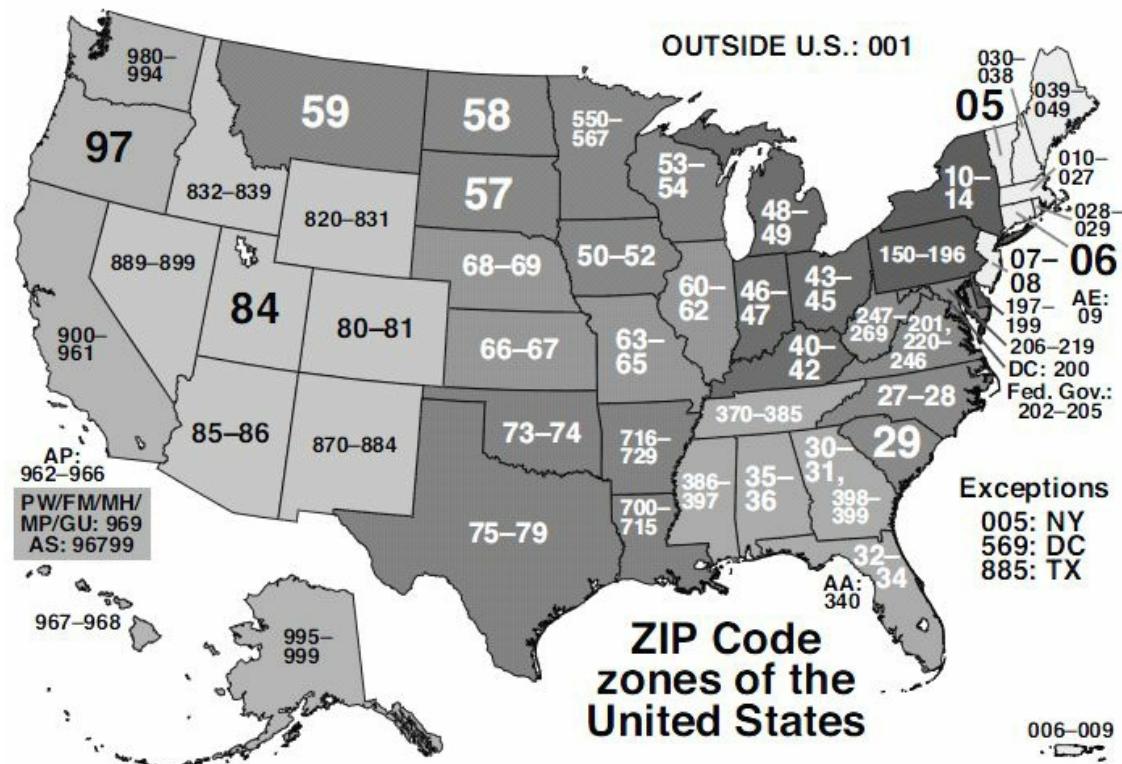


图5.2 -- 美国邮政编码

### 路由汇总的前提，Route Summarisation Prerequisites

为在网络上应用路由汇总，需要使用某种无类协议（a classless protocol，这将在后面涉及），比如 RIPv2、EIGRP、或者OSPF。同时还需以层次化顺序方式设计网络（design your network in a hierarchical order），这就需要仔细规划和设计。这就意味着你—不能随机地任意地在网络中给路由器或局域网分配网络。

### 应用路由汇总，Applying Route Summarisation

我们来看看一个实例网络，如不采用路由汇总，会有什么问题。在此实例中，说的就是在某网络上的IP地址范围内，汇总是如何工作的。图5.3中的路由器连着一些网络。头一种解决办法是将这些网络都通告给下一跳的路由器（the next-hop router）。而替代的做法是汇总这8个网络到一条路由，并将汇总结果发送给下一跳路由器，这样做可降低带宽、CPU和内存的需求。

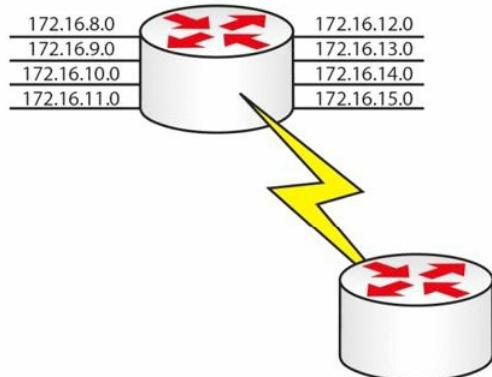


图5.3 -- 路由汇总的一个实例

抱歉的是，计算出汇总路由（a summary route）的唯一方法是将IP地址转换成二进制形式。如你不这样做，就无法知道是否通告了正确的汇总路由，而这将导致网络故障。

首先，写出完整的所有网络地址，接着在右边写出其二进制形式，如下面所示。

网络号	二进制形式
172.16.8.0	<b>10101100.00010000.00001</b> 000.00000000
172.16.9.0	<b>10101100.00010000.00001</b> 001.00000000
172.16.10.0	<b>10101100.00010000.00001</b> 010.00000000
172.16.11.0	<b>10101100.00010000.00001</b> 011.00000000
172.16.12.0	<b>10101100.00010000.00001</b> 100.00000000
172.16.13.0	<b>10101100.00010000.00001</b> 101.00000000
172.16.14.0	<b>10101100.00010000.00001</b> 110.00000000
172.16.15.0	<b>10101100.00010000.00001</b> 111.00000000
匹配的位	<b>10101100.00010000.00001</b> = 21 位

我将每个网络地址中匹配的位进行了加粗。你可以看到各个网络地址的前 21 位是匹配的，所有汇总路由可由下面的 21 位反应出来。

172.16.8.0 255.255.255.248.0

运用路由汇总的另一个显著优势在于，如某个本地网络宕掉，汇总网络仍然可以通告出去。这就是说网络的其它部分无需更新其各自路由表（routing table），甚至在更糟的情况下，无需去处理一条抖动路由（那种迅速起来又宕掉的路由）。下面有两个路由汇总的练习。

**练习一：**写出下面的地址的二进制形式，并找出匹配的位。我已写出了它们的前两个 8 位，以节省你的时间。

网络号	二进制形式
172.16.50.0	<b>10101100.00010000.0</b> 0110010.00000000
172.16.60.0	<b>10101100.00010000.0</b> 0111100.00000000
172.16.70.0	<b>10101100.00010000.0</b> 1000110.00000000
172.16.80.0	<b>10101100.00010000.0</b> 1010000.00000000
172.16.90.0	<b>10101100.00010000.0</b> 1011010.00000000
172.16.100.0	<b>10101100.00010000.0</b> 1100100.00000000
172.16.110.0	<b>10101100.00010000.0</b> 1101110.00000000
172.16.120.0	<b>10101100.00010000.0</b> 1111000.00000000

通告的汇总地址会是什么呢？

那就是 172.16.50.0 255.255.128.0，或者 /17。

**练习二：**下面述及的机构有3台连接到公司总部的路由器。他们需要将通告自伦敦 1、2、3 号路由器的路由进行汇总。

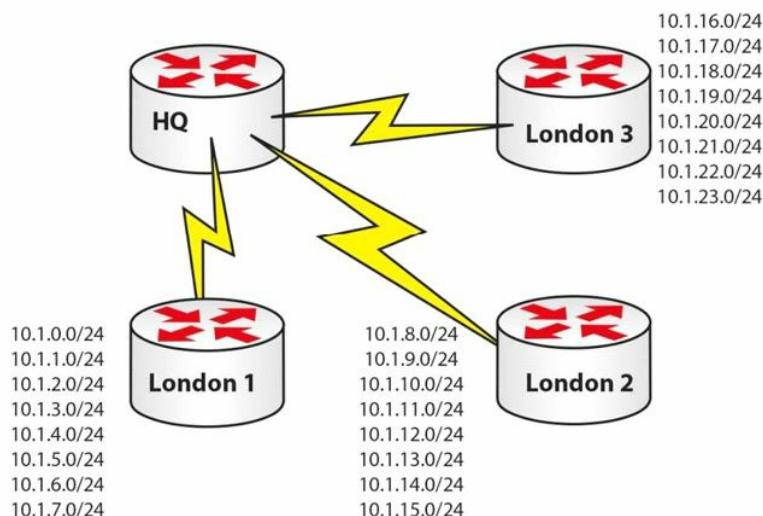


图5.4 -- 通告自伦敦 1、2、3 号路由器的汇总路由

我们先从伦敦 1 号路由器开始。

网络号	二进制形式
10.1.0.0	00001010.00000001.00000000.00000000
10.1.1.0	00001010.00000001.00000001.00000000
10.1.2.0	00001010.00000001.00000010.00000000
10.1.3.0	00001010.00000001.00000011.00000000
10.1.4.0	00001010.00000001.00000100.00000000
10.1.5.0	00001010.00000001.00000101.00000000
10.1.6.0	00001010.00000001.00000110.00000000
10.1.7.0	00001010.00000001.00000111.00000000

有 21 位匹配，所以伦敦 1 号路由器可向总部路由器通告  $10.1.0.0/21$ ，这个汇总路由。

再看伦敦 2 号路由器。

网络号	二进制形式
10.1.8.0	00001010.00000001.00001000.00000000
10.1.9.0	00001010.00000001.00001001.00000000
10.1.10.0	00001010.00000001.00001010.00000000
10.1.11.0	00001010.00000001.00001011.00000000
10.1.12.0	00001010.00000001.00001100.00000000
10.1.13.0	00001010.00000001.00001101.00000000
10.1.14.0	00001010.00000001.00001110.00000000
10.1.15.0	00001010.00000001.00001111.00000000

伦敦 2 号也有 21 位匹配位，所以它可以往总部路由器通告  $10.1.8.0/21$ 。

再看伦敦 3 号路由器。

网络号	二进制形式
10.1.16.0	00001010.00000001.00010000
10.1.17.0	00001010.00000001.00010001
10.1.18.0	00001010.00000001.00010010
10.1.19.0	00001010.00000001.00010011
10.1.20.0	00001010.00000001.00010100
10.1.21.0	00001010.00000001.00010101
10.1.22.0	00001010.00000001.00010110
10.1.23.0	00001010.00000001.00010111

伦敦 3 号路由器同样有 21 位匹配位，因此它可以上游的中心路由器通告  $10.1.16.0/21$ 。

CCNA 考试要求你掌握路由汇总。如你能快速地算出同样的位，那么就可以快且准地回答问题了。

## 变长子网掩码，Variable Length Subnet Masking, VLSM

## 变长子网掩码的使用，Using VLSM

先看看下面这个网络。

- 192.168.1.0/24 = 这是一个有 254 台主机的网络

当然这会很好地工作，那么如果你的网络需要多于一个的子网呢？或者你的那些子网无需 254 台这么多的主机呢？此两种情形，都需要做出一些改变。如你取而代之的是用一个 /26 的掩码，就可以得到这样的结果。

- 192.168.1.0/26 = 4 个有 62 台主机的子网

如这样不适当，那么来个 /28 的掩码如何？

- 192.168.1.0/28 = 16 个有 14 台主机的子网

对子网划分秘笈的设计部门的再度引用，可帮你计算出如何来在网络中应用变长子网掩码，或是有助于解答考试问题。在使用 /26 掩码时，你可以发现将会得到多少个子网及每个子网有多少台主机。

二进制位数	128	64	32	16	8	4	2	1
子网号	<input type="radio"/>	<input checked="" type="radio"/>						
128	<input type="radio"/>							
192	<input type="radio"/>							
224								
240								
248								
252								
254								
255								
	子网数	主机数-2						
2	<input type="radio"/>	<input checked="" type="radio"/>						
4	<input type="radio"/>	<input checked="" type="radio"/>						
8		<input checked="" type="radio"/>						
16		<input checked="" type="radio"/>						
32		<input checked="" type="radio"/>						
64		<input checked="" type="radio"/>						

因为必须从主机位借用两位，所以得到 4 个子网，每个子网有 62 台主机。

## 网络切分，Slicing Down Networks

变长子网掩码的关键在于取得网络块并令到这些网络块满足特定的网络需求（take your network block and make it work for your particular network needs）。拿典型的网络地址 192.168.1.0/24 来说，在使用VLSM 时，你可以使用掩码 /26，实现这样的划分。

192.168.1.0/26	子网	主机数
192.168.1.0	1	62
192.168.1.64 -- 使用中	2	62
192.168.1.128 -- 使用中	3	62
192.168.1.192 -- 使用中	4	62

在发现基础设施中有着两个仅需 30 台主机的较小网络之前，这么做是没有问题的。那么在已经使用了 3 个较小子网（标为“使用中”），而仅剩下 1 个（也就是 192.168.1.0）时呢？变长子网掩码就可以让你用上任何已划分出的子网，对其再进行划分。**唯一的规则就是IP地址仅能使用一次，而与其掩码无关。**

如你使用子网划分秘笈图表，那么就可以看到哪个掩码带来 30 台主机的子网。

	子网数	主机数-2					
2	<input type="radio"/>	<input type="radio"/>					
4	<input type="radio"/>	<input type="radio"/>					
8		<input type="radio"/>					
16		<input type="radio"/>					
32		<input type="radio"/>					
64							

该图表的上面部分（这里没有显示）告诉我们在左边列勾选了 3 个位置，这就给出掩码 /24 或者是 /27（借用了 3 位）。

192.168.1.0/27	子网	主机数
192.168.1.0	1	30
192.168.1.32	2	30
192.168.1.64	不能使用	不能使用

是不可以使用 .64 子网的，因为该子网已被使用了。现在就可以使用其余两个子网了。如你只需使用一个，那么就还可以将剩下的那个进行进一步划分，得到更多的子网，只是每个子网中的主机数更少而已。

## IP分址故障排除，Troubleshooting IP Addresses Issues

### 子网掩码及网关故障的排除

在出现IP分址、子网掩码或网关问题时，你会看到多种现象。一些问题会如同下面这样。

- 网络设备可在其本地子网通信，却无法与本地网络之外的设备通信。这通常表明有着与网关配置或运行相关类型的问题。
- 没有任何类型的IP通信，不管是内部的还是远程的。这通常表明存在大问题，可能涉及相应设备上功能的缺失。

- 还有这种能与某些IP地址通信，却无法与存在的全部IP地址通信的情形。这通常是最难解决的故障，因为其可能有很多原因。

在处理这些问题的过程中，首先要做的就是对设备上所配置的IP地址、子网掩码及默认网关进行反复检查。同时还要查看设备文档，来验证相应信息。大量的故障都是由错误配置造成的。

如你正在首次安装一些网络设备，多半要手动输入一些IP地址、子网掩码和默认网关等信息。建议在进行提交前进行检查，因为这方面人所犯的错误是难免的。许多企业都有关于将新设备引入网络的手册，包括网关测试及到SNMP服务器的可达能力。

如需在故障排除过程中收集信息，可能需要做一下包捕获，以此来观察设备间发送了哪些数据包。如果看到有来自其它网络上主机的包，就可能存在某种VLAN错误配置问题。如怀疑子网掩码不正确，就要检查在网络上其它设备的参数。如果其它机器工作良好，就要在该设备上使用如预期一样无法工作的同一子网掩码，并再行测试。

在使用了动态主机分址（DHCP）来为网络上的设备分配包括子网掩码和网关的地址信息时，就要检查DHCP服务器配置，因为此时问题可能发生在另一方面了。DHCP服务器错误配置或者DHCP服务已阻塞，都是可能的，所以在故障排除时包含这一步是必要的。务必还要记住从DHCP地址池中排除一些保留地址，因为这些地址通常会分配给服务器及路由器接口。

另一些有助于找出网络故障发生所在之处的故障排除工具有traceroute和ping。在本书及本书实验中会有涉及。

## 第五天的问题，Day 5 Questions

- Convert 192.160.210.177 into binary (without using a calculator).
- Convert 10010011 into decimal.
- What is the private range of IP addresses?
- Write out the subnet mask from CIDR /20 .
- Write out the subnet mask from CIDR /13 .
- 192.168.1.128/26 gives you how many available addresses?
- What is the last host of the 172.16.96.0/19 network?
- Starting with 192.168.1.0/24 , with VLSM, you can use a /26 mask and generate which subnets?
- In order to use route summarisation on your network, you need to use what?
- Write down the subnets 172.16.8.0 to 172.16.15.0 , and work out the common bits and what subnet mask you should use as a summary. Don't look in the book before working this out.

## 第五天问题的答案

- 11000000.10100000.11010010.10110001 .
- 147 .
- 10.x.x.x – any address starting with a 10 . 172.16.x.x to 172.31.x.x – any address starting with 172.16 to 172.31 , inclusive. 192.168.x.x – any address starting with 192.168 .
- 255.255.240.0 .
- 255.248.0.0 .
- 62 .
- 172.16.127.254 .
- 192.168.1.0/26 , 192.168.1.0.64/26 , 192.168.1.0.128/26 , and 192.168.1.0.192/26 .
- A classless protocol.
- 172.16.8.0/21 (mask: 255.255.248.0 ).

## 课文中进制转换的答案

1. Convert 1111 to hex and decimal

```
Hex = F  
Decimal = 15
```

1. Convert 11010 to hex and decimal

```
Hex = 1A  
Decimal = 26
```

1. Convert 10000 to hex and decimal

```
Hex = 10  
Decimal = 16
```

1. Convert 20 to binary and hex

```
Binary = 10100  
Hex = 14
```

1. Convert 32 to binary and hex

```
Binary = 100000  
Hex = 20
```

1. Convert 101 to binary and hex

```
Binary = 1100101  
Hex = 65
```

1. Convert A6 from hex to binary and decimal

```
Binary = 10100110  
Decimal = 166
```

1. Convert 15 from hex to binary and decimal

```
Binary = 10101  
Decimal = 21
```

1. Convert B5 from hex to binary and decimal

```
Binary = 10110101  
Decimal = 181
```

## 第五天的实验

## 路由器上的IP分址实验

### 拓扑图, Topology



路由器上的IP分址实验拓扑图

### 实验目的, Purpose

学习如何熟练地在路由器上配置IP地址，并经由某个串行接口执行ping操作。

### 实验步骤, Walkthrough

- 先是明确路由器上的串行借口编号，你的路由器与上面拓扑图中的可能有所不同。同时，还要明确串行链路的哪一端连接的是DCE线，因为在该端是需要 `clock rate` 命令的。

```
Router>en
Router#sh ip interface brief
Interface      IP-Address      OK?      Method      Status          Protocol
FastEthernet0/0  unassigned     YES       unset      administratively down      down
FastEthernet0/1  unassigned     YES       unset      administratively down      down
Serial0/1/0      unassigned     YES       unset      administratively down      down
Vlan1           unassigned     YES       unset      administratively down      down
Router#
Router#show controllers Serial0/1/0
M1T-E3 pa: show controller:
PAS unit 0, subunit 0, f/w version 2-55, rev ID 0x2800001, version 2
idb = 0x6080D54C, ds = 0x6080F304, ssb=0x6080F4F4
Clock mux=0x30, ucmd_ctrl=0x0, port_status=0x1
line state: down
DCE cable, no clock rate
```

- 在一侧为路由器加上主机名及IP地址，如该侧是DCE，就为其加上时钟速率 (the clock rate)。

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname RouterA
RouterA(config)#interface s0/1/0
RouterA(config-if)#ip add 192.168.1.1 255.255.255.0
RouterA(config-if)#clock rate 64000
RouterA(config-if)#no shut
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to downRouterA(config-if)#

```

- 为另一侧加上主机名和IP地址。同时使用 `no shut` 命令将该接口开启。

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#hostname RouterB
RouterB(config)#int s0/1/0
RouterB(config-if)#ip address 192.168.1.2 255.255.255.0
RouterB(config-if)#no shut
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to down
RouterB(config-if)#^Z
RouterB#
%LINK-5-CHANGED: Interface Serial0/1/0, changed state to up
```

1. 用 ping 命令测试连接。

```
RouterB#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 = 31/31/32 ms
```

**注意：**如ping不工作，就要反复检查，确保在正确的路由器上加上了 clock rate 命令。还要确保正确插入了线缆，并使用命令 show controllers serial x/x/x，这里的接口编号是你的路由器上的。

## 二进制转换及子网划分练习， Binary Conversion and Subnetting Practice

请将今天所剩下的时间，用来做下面这些重要的练习。

- 十进制到二进制的转换（随机数字）
- 二进制到十进制的转换（随机数字）
- IPv4 子网划分（随机网络和场景）

# 第6天 网络地址转换

## Network Address Translation

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第六天的任务

- 阅读今天的课文
- 回顾昨天的课文内容
- 完成今天的实验
- 阅读 ICND1 记诵指南
- 在网站[subnetting.org](http://subnetting.org) 上花15分钟

网络地址转换是另一个生僻内容(another strange subject)，思科把网络地址转换拆分到ICND1和ICND2两个大纲中了。

今天你会学到下面这些知识。

- NAT基础
- 对NAT的配置和验证
- NAT故障排除

今天的课程涵盖了ICND1大纲的以下要求。

- 弄清NAT的基本操作
  - NAT的目的
  - NAT地址池
  - 静态NAT
  - 一对一的NAT
  - NAT过载，Overloading
  - 源地址NAT
  - 单向NAT
- 按需求配置并验证NAT

## NAT基础，NAT Basics

想象一下如果网络不是以IP地址运行，而是按颜色来运作。蓝色和黄色有无限的供应，其它颜色却是短缺的。网络分开成使用蓝色和黄色的许多用户，因为这两种颜色可以随意使用。而蓝色用户需要频繁地前往外部网络，那么就需要去买点绿色凭据，在蓝色用户需要与外部网络上的主机通信时，路由器可以用其将

蓝色用户的凭据进行替换。路由器此时会像下面这样做。

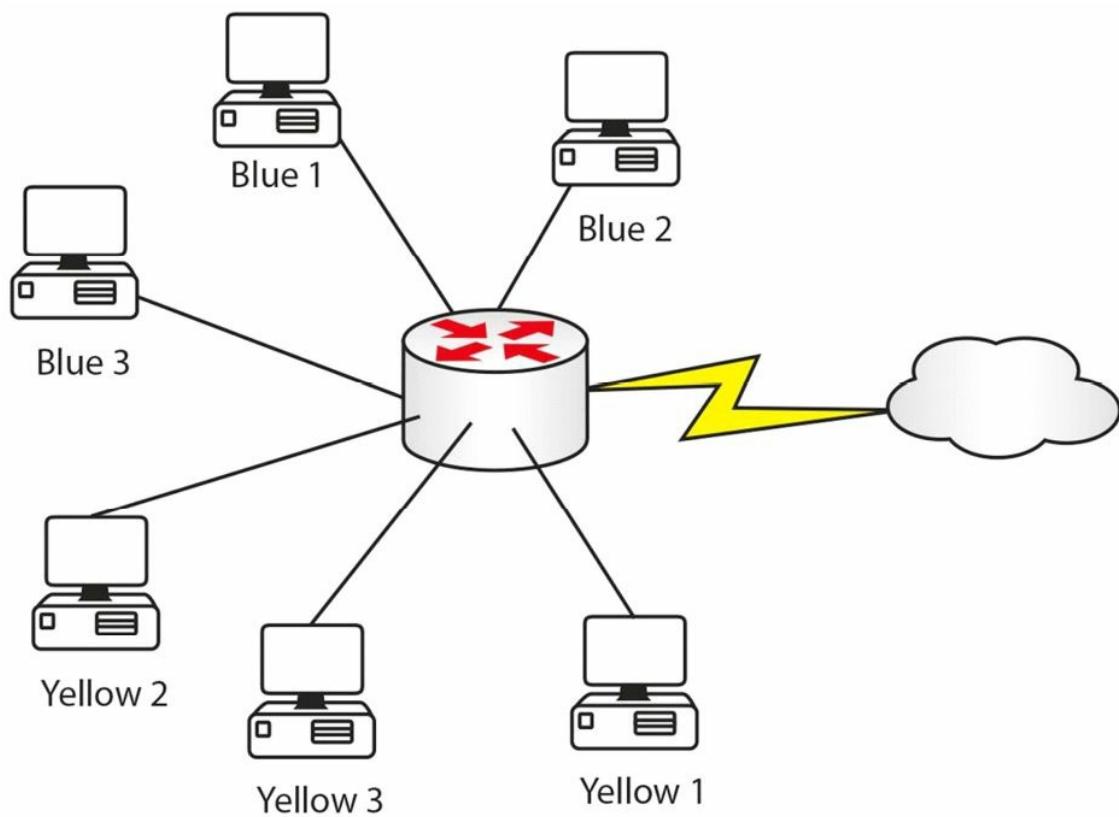


图6.1 -- 内部凭据被替换成了外部凭据

内部凭据	外部凭据
蓝色1号	绿色1号
蓝色2号	绿色2号
蓝色3号	绿色3号

在各台蓝色设备完成与外部的连接后，对应的绿色凭据会释放给其它蓝色设备使用。这么做的好处在于外部设备无法看到内部凭据编号，且有助于留下互联网上十分有限的可用凭据。

我们看到，NAT不仅保护了网络IP地址，同时也是节约地址的另一种方法。**NAT是在路由器或者防火墙上实现的**，那么，代替上面的颜色，你会看到下面这样的情况。

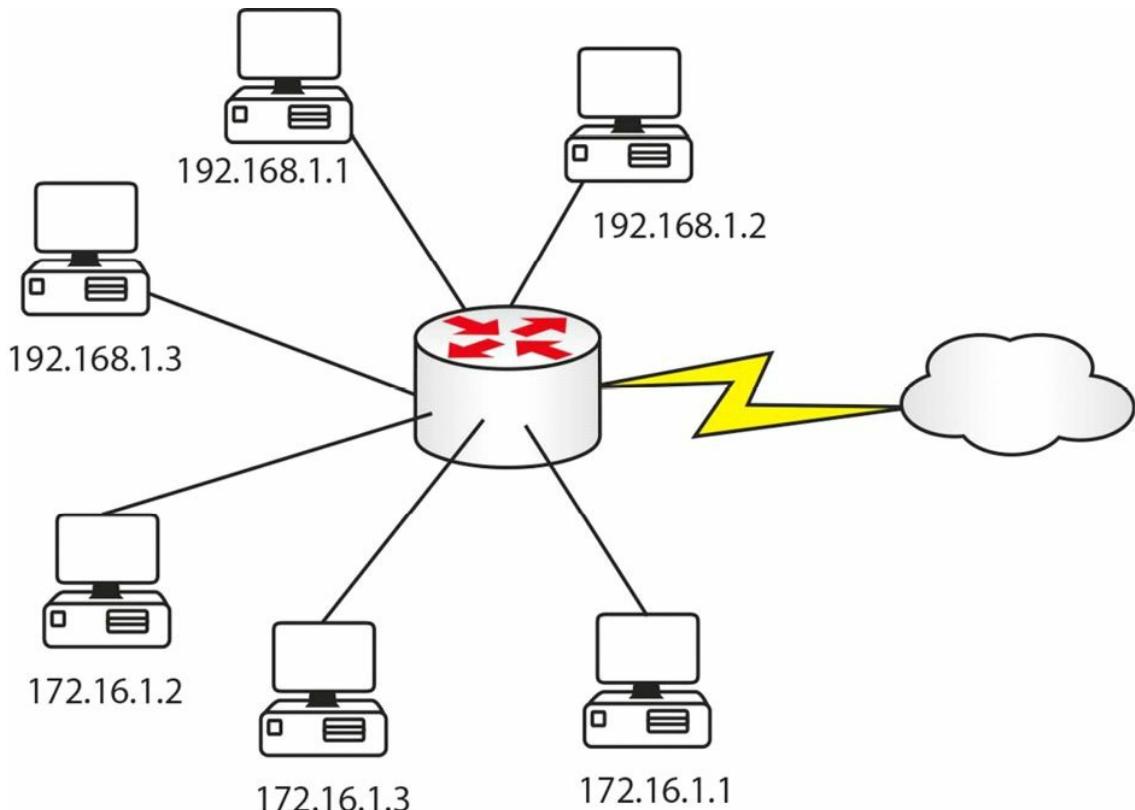


图6.2 -- 内部地址被替换成外部地址

内部地址	外部地址
192.168.1.1	200.100.1.5
192.168.1.3	200.100.1.7

根据特定的需求，在路由器上配置NAT有三种方式。CCNA考试要求你掌握所有三种方式。

为了配置NAT，要先告诉路由器，有哪些内部和外部接口。这是因为事实上可以将众多内部地址替换成某个NAT地址池中的地址（a pool of NAT addresses），或者最起码替换成单一的NAT地址，并在路由器上的两个以太网接口之间完成NAT。

正如前面说的，对于考试和现实需求来说，通常都会将私有互联网地址转换成互联网上的可路由的地址（routable addresses on the Internet）。这在家庭宽带路由器上就能见到，其通常会给笔记本电脑一个192.168.1范围的IP，而在连接到ISP的接口上有着一个可路由的地址。

NAT令到私有网络上的主机可以访问互联网上的资源，或是可以访问到其它公共网络。NAT是一个IETF标准，其让局域网的内部流量使用一个IP地址集合，这些地址通常就是RFC 1918中所定义的私有地址空间，对于外部流量，又使用另一个地址集合，这些地址通常是公开注册的IP地址空间。

NAT为进入和发出的流量去改装数据包的头部，并对每个会话进行跟踪。理解NAT的关键，同时也是NAT故障排除的关键，就是对NAT的有关术语有扎实理解。你应熟悉下面这些NAT名词。

- NAT内部接口
- 内部本地地址
- 内部全球地址
- NAT外部接口
- 外部本地地址
- 外部全球地址

上面NAT术语中的**内部接口**，是指由该组织所控制的管理域的边界接口（the border interface of the administrative domain controlled by the organization）。而并非得要是内部网络上的主机所使用的默认网关。

而**内部本地地址**则是某台内部网络上的主机的IP地址。在多数情况下，**内部本地地址都是一个 RFC 1918 地址**（也就是不可路由地址，比如 192.168.x.x 或 172.16.x.x 等等）。该地址被转换成外部全局地址，那么**外部全局地址通常就是来自一个公开分配的或是经注册的地址池了**。要记住的是，尽管如此，**内部本地地址也可以是一个公网地址**。

**内部全局地址**，则是内部主机在其呈现在外部世界时的地址。一旦内部IP地址被转换过后，对公网或是其它任何外部网络及主机来说，它就成为了一个内部全局地址了。

与内部接口对应，**外部接口是指不受该组织所控制的管理域的边界**。换句话说，外部接口是连接外部网络的，连接的网络可以是互联网或其它任何的外部网络，比如友商网络等。任何处于外部接口外侧的主机，都不属于本地组织的管理之下。

**外部本地地址**是某台外部主机呈现给内部主机的IP地址。最后，**外部全局地址又是一个合法的、可在互联网上使用的公网地址**。外部本地地址和外部全局地址都是分配自一个全球可路由网络地址空间。

为搞清楚这些概念，图6.3表示了两台主机之间的一个会话中各种地址的使用。中间的网关上开启了NAT。

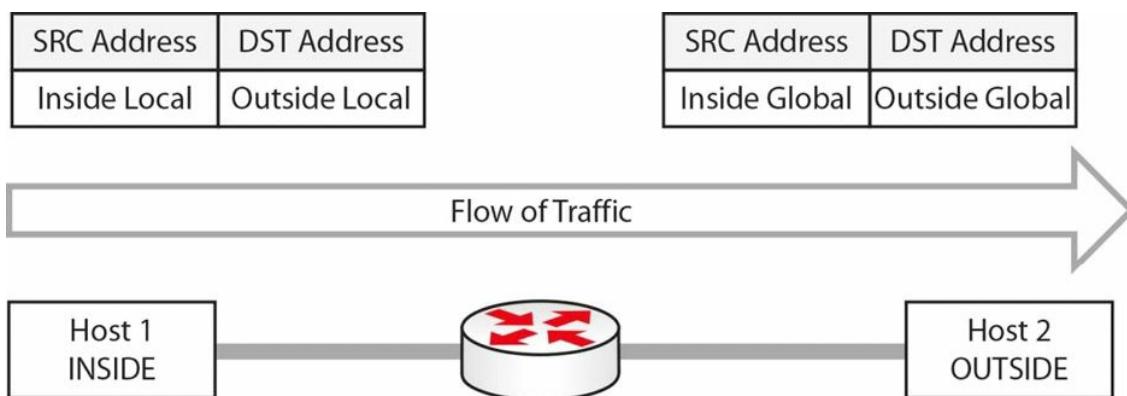


图6.3 -- 理解NAT的各种内部外部地址

NAT内部和外部的分址，是一个经典的考试问题，所以还需在回头看几次这里的内容。

## 配置并验证NAT，Configuring and Verifying NAT

在思科IOS上对网络地址转换的配置和验证是一个简单的事情。在配置NAT时，要执行下面这些操作。

- 使用接口配置命令 `ip nat inside` 将一个或多个的接口指定为内部接口。
- 使用接口配置命令 `ip nat outside` 将某个接口指定为外部接口。
- 配置一条访问控制清单（access control list, ACL），其将匹配所有需要转换的流量。此访问控制清单可以是标准、扩展的命名ACL或编号ACL（a standard or an extended named or numbered ACL）。
- 作为可选项，使用全局配置命令 `ip nat pool <name> <start-ip> <end-ip> [netmask <mask> | prefix-length <length>]`，配置一个全局地址池(a pool of global addresses)。这会定义出一个内部本地地址将会转换成的内部全局地址池。
- 使用全局配置命令 `ip nat inside source list <ACL> [interface | pool] <name> [overload]`，全局性地配置上NAT。

Farai 指出 -- “请看看命令 ip nat inside source static , 可以在 [www.howtonetwork.net/public/698.cfm](http://www.howtonetwork.net/public/698.cfm) 免费查阅。 ”

下面的输出给出了一种思科IOS软件下配置NAT（动态NAT）的方式。可以看出，该配置使用了可用的 `description` 和 `remark` 两种特性，来帮助管理员更容易地对网络进行管理和故障排除。

```
R1(config)#interface FastEthernet0/0
R1(config-if)#description 'Connected To The Internal LAN'
R1(config-if)#ip address 10.5.5.1 255.255.255.248
R1(config-if)#ip nat inside
R1(config-if)#exit
R1(config)#interface Serial0/0
R1(config-if)#description 'Connected To The ISP'
R1(config-if)#ip address 150.1.1.1 255.255.255.248
R1(config-if)#ip nat outside
R1(config-if)#exit
R1(config)#access-list 100 remark 'Translate Internal Addresses Only'
R1(config)#access-list 100 permit ip 10.5.5.0 0.0.0.7 any
R1(config)#ip nat pool INSIDE-POOL 150.1.1.3 150.1.1.6 prefix-length 24
R1(config)#ip nat inside source list 100 pool INSIDE-POOL
R1(config)#exit
```

按照这个配置，命令 `show ip nat translations` 就可以用来对路由器上具体进行的转换进行查看，如下面的输出所示。

```
R1#show ip nat translations
Pro      Inside global    Inside local     Outside local    Outside global
icmp    150.1.1.4:4        10.5.5.1:4       200.1.1.1:4      200.1.1.1:4
icmp    150.1.1.3:1        10.5.5.2:1       200.1.1.1:1      200.1.1.1:1
tcp     150.1.1.5:159      10.5.5.3:159     200.1.1.1:23     200.1.1.1:23
```

在路由器上配置NAT时，通常有以下三个选择。

- 对一个内部地址，用一个外部地址进行替换（静态NAT， static NAT）
- 对多个内部地址，用两个以上的外部地址进行替换（动态NAT， dynamic NAT）
- 将多个内部地址，用多个外部端口进行转换（这就是**端口地址转换**，或者叫**单向NAT**， Port Address Translation or one-way NAT）

## 静态NAT

### Static NAT

在网络内部一些有一台web服务器时，就要将某个特定内部地址，替换成另一个外部地址了。如此时仍然进行动态分址，就没有办法到达该特定目的地址，因为它总是变动的。

Farai指出，“对那些需要经由互联网可达的所有服务器，比如e-mail或FTP服务器，都要使用静态NAT（如下面的图6.4所示）”

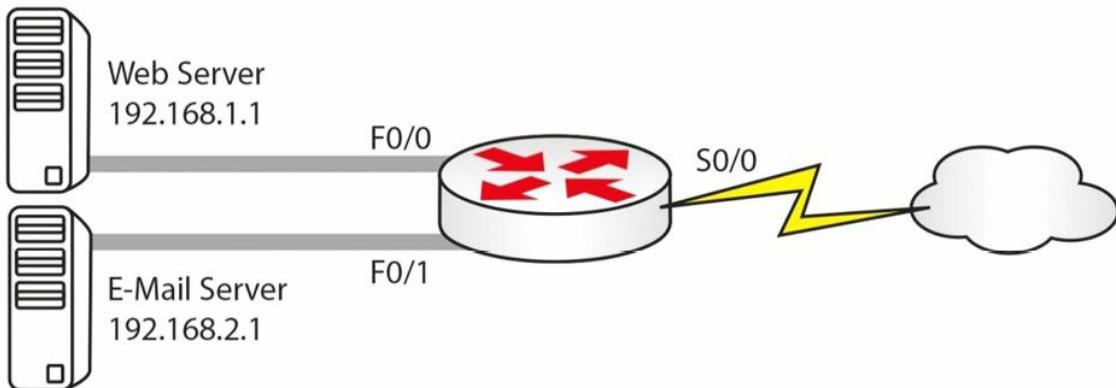


图6.4 -- 在用的静态NAT

内部地址	外部NAT地址
192.168.1.1	200.1.1.1
192.168.2.1	200.1.1.2

对上面的网络，配置应像下面这样。

```

Router(config)#interface f0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#ip nat inside
Router(config)#interface f0/1
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#ip nat inside
Router(config)#interface s0/0
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#ip nat inside source static 192.168.1.1 200.1.1.1
Router(config)#ip nat inside source static 192.168.2.1 200.1.1.2

```

命令 `ip nat inside` 和 `ip nat outside`，告诉路由器哪些是内侧NAT接口，哪些是外侧的NAT接口。而命令 `ip nat inside source` 命令，就定义了那些静态转换，想要多少条就可以有多少条的该命令，那么就算你掏钱买的那些公网IP地址有多少个，就写上多少条吧。在思科公司，笔者曾解决有关此类问题的大量主要的配置错误，就是找不到 `ip nat inside` 及 `ip nat outside` 语句！考试中可能会碰到那些要求找出配置错误的问题。

强烈建议将上述命令敲入到某台路由器中去。本书中有很多的NAT实验，但是在阅读理论章节的同时，你敲入得越多，那么这些信息就能更好地进入你的大脑。

## 动态NAT或NAT地址池

通常会用到一组可路由地址，或是一个可路由地址池。一对一的NAT映射，有其局限性，首当其冲的就是成本高，其次路由器上有许多行的配置。动态NAT允许为内部主机配置一或多个的公网地址组。

路由器会维护一个内部地址到外部地址对应的清单，而最后该表格中的转换会超时(Your router will keep a list of the internal addresses to external addresses, and eventually the translation in the table will time out)。可以修改此超时值，但请找Cisco 技术支持工程师 (a Cisco TAC engineer) 的建议去修改。

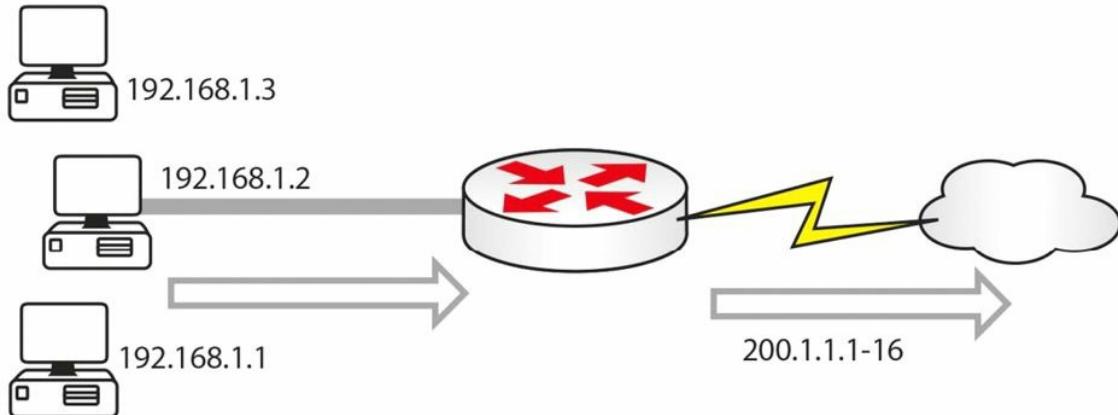


图6.5 -- 到一个NAT公网可路由地址池的内部私有地址

当路由器上的内部主机发出到外部的连接时，如执行命令 `show ip nat translations`，就会看到下面的包含类似信息的图表。

内侧地址	外侧NAT地址
192.168.1.3	200.1.1.11
192.168.1.2	200.1.1.14

在上面的图6.5中，让内部地址使用的是一个从 200.1.1.1 到 200.1.1.16 的地址池。下面是要实现该目的的配置文件。这里就不再给出路由器接口地址了。

```
Router(config)#interface f0/0
Router(config-if)#ip nat inside
Router(config)#interface s0/1
Router(config-if)#ip nat outside
Router(config)#ip nat pool poolname 200.1.1.1 200.1.1.16 netmask 255.255.255.0
Router(config)#ip nat inside source list 1 pool poolname
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

该ACL用于告诉路由器哪些地址要转换，哪些地址不要转换。而该子网掩码实际上是反转的，叫做反掩码，在第九天会涉及。所有NAT地址池都需要一个名字，而在本例中，它简单地叫做“poolname”。源列表引用自那个ACL (the source list refers to the ACL)，经译者在GNS3上测试，动态NAT仍然是一对一的地址转换。

## NAT Overload/端口地址转换/单向NAT

### NAT Overload/Port Address Translation/One-Way NAT

IP地址处于紧缺之中，在有着成千上万的地址需要路由时，将花一大笔钱（静态NAT、动态NAT都无法解决此问题）。在此情况下，可以使用**NAT overload方案**（如图6.6），该方案又被思科叫做**端口地址转换（Port Address Translation, PAT）**或**单向NAT**。PAT巧妙地允许将某端口号加到某个IP地址，作为与另一个使用该IP地址的转换区分开来的方式。每个IP地址有多达 65000 个可用端口号。

尽管这是超出CCNA考试范围的，但了解PAT如何处理端口号，会是有用的。在每个思科文档中，都将每个公网IP地址的可用端口号分为 3 个范围，分别是 0-511、512-1023 和 1024-65535。PAT给每个UDP和TCP会话都分配一个独特的端口号。它会尝试给原始请求分配同样的端口号值，但如果原始的源端口号已被使用，它就会开始从某个特别端口范围的开头进行扫描，找出第一个可用的端口号，分配给那个会话。

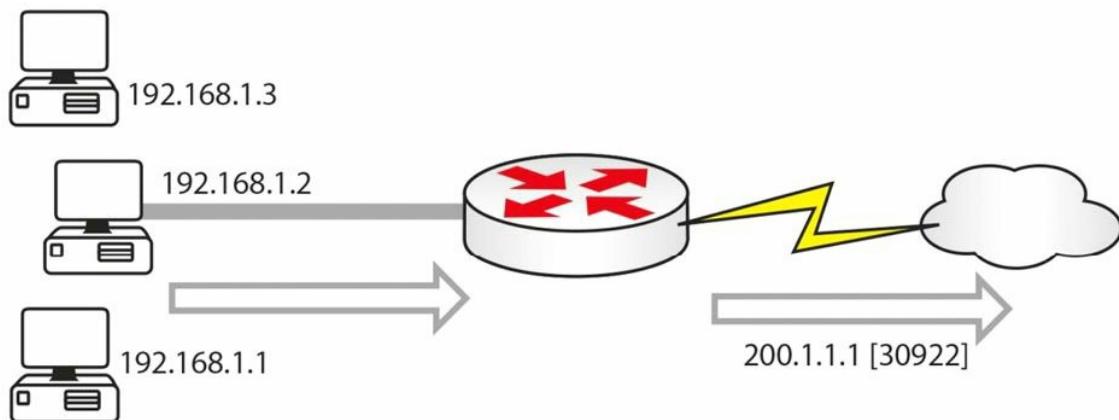


图6.6 -- NAT Overload

此时，命令 `show ip nat translations` 给出的表格，将会显示下面这样的IP地址及端口号。

内侧地址	外侧NAT地址（带有端口号）
192.168.1.1	200.1.1.1:30922
192.168.2.1	200.1.1.2:30975

而要配置PAT，需要进行如同动态NAT的那些同样配置，还要在地址池后面加上关键字 `overload`。

```
Router(config)#interface f0/0
Router(config-if)#ip nat inside
Router(config)#interface s0/1
Router(config-if)#ip nat outside
Router(config)#ip nat pool poolname 200.1.1.1 200.1.1.1 netmask 255.255.255.0
Router(config)#ip nat inside source list 1 pool poolname overload
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
```

这该很容易记住吧！

Farai指出：“以多于一个IP方式使用PAT，就是对地址空间的浪费，因为路由器会使用第一个IP地址，并为每个随后的连接仅增大端口号。这就是为何通常将PAT配置为该接口上的超载(overload)。”

## NAT故障排除

### Troubleshooting NAT

NAT故障中十次有九次，都是由于路由器管理员忘记了把 `ip nat outside` 或 `ip nat inside` 命令加到路由器接口上。事实上，几乎总是存在这个问题！接下来最频繁的错误包括不正确的ACL，以及某个拼写错误的地址池名称（地址池是区分大小写的）。

使用命令 `debug ip nat [detailed]`，可以在路由器上对NAT转换进行调试，又可以使用命令 `sh ip nat translations`，来查看NAT地址池。

## 第六天问题

1. NAT converts the \_\_\_\_\_ headers for incoming and outgoing traffic and keeps track of each session.
2. The \_\_\_\_\_ address is the IP address of an outside, or external, host as it appears to inside hosts.

3. How do you designate inside and outside NAT interfaces?
4. Which show command displays a list of your NAT table?
5. When would you want to use static NAT?
6. Write the configuration command for NAT 192.168.1.1 to 200.1.1.1 .
7. Which command do you add to a NAT pool to enable PAT?
8. NAT most often fails to work because the \_\_\_\_\_ command is missing.
9. Which `debug` command shows live NAT translations occurring?

## 第六天问题的答案

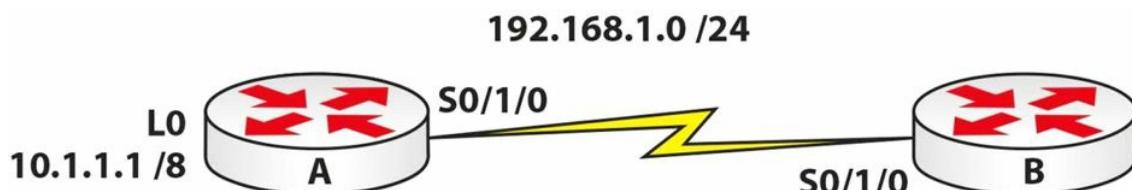
1. Packet.
2. Outside local.
3. With the `ip nat inside` and `ip nat outside` commands.
4. The `show ip nat translations` command.
5. When you have a web server (for example) on the inside of your network.
6. `ip nat inside source static 192.168.1.1 200.1.1.1` .
7. The `overload` command.
8. The `ip nat inside` OR `ip nat outside` command.
9. The `debug ip nat [detailed]` command.

## 第六天的实验

### 静态NAT实验

#### Static NAT Lab

#### 拓扑图



静态NAT实验拓扑图

#### 实验目的

学习如何配置静态NAT。

#### 实验步骤

1. 将IP地址 192.168.1.1 255.255.255.0 加入到路由器 A ， 并修改 `hostname` 为 Router A 。把IP地址 192.168.1.2 255.255.255.0 加入到路由器 B 。在正确的一侧加上时钟速度( `clock rate` )，然后分别自 A 往 B 和自 B 往 A 进行 ping 测试。如需提示，请回顾先前的那些实验。
2. 在路由器 A 上需要加入一个IP地址，以模拟LAN上的一台主机。通过一个环回接口，可以实现这个目的。

```
RouterA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#interface Loopback0
RouterA(config-if)#ip add 10.1.1.1 255.0.0.0
RouterA(config-if)#
```

1. 为进行测试，需要告诉 Router B 将发往任何网络的任何流量，都发往 Router A。通过一条静态路由完成这个。

```
RouterB#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/1/0
RouterB(config)#
```

1. 要测试该条静态路由是否工作，通过从 Router A 上的环回接口对 Router B 进行 ping 操作。

```
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms
RouterA#
```

1. 在 Router A 上配置一个静态NAT条目。使用NAT，将地址 10.1.1.1，在其离开该路由器时，转换成 172.16.1.1。同样需要告诉路由器哪个是NAT的内部接口，哪个是外部接口。

```
RouterA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#int Loopback0
RouterA(config-if)#ip nat inside
RouterA(config-if)#int Serial0/1/0
RouterA(config-if)#ip nat outside
RouterA(config-if)#
RouterA(config-if)#ip nat inside source static 10.1.1.1 172.16.1.1
RouterA(config)#
```

1. 打开NAT调试，如此就可以看到转换的进行。此时再执行另一个扩展 ping 操作（自 Lo 接口的），并查看NAT表。因为IOS的不同，你的输出可能与我的不一样。

```

RouterA#debug ip nat
IP NAT debugging is on
RouterA#
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [11]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [11]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [12]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [12]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [13]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [13]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [14]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [14]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [15]
!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/46/110 ms
RouterA#
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [15]
RouterA#show ip nat translations
      Inside global      Inside local      Outside local      Outside global
  icmp    172.16.1.1:10    10.1.1.1:10    192.168.1.2:10    192.168.1.2:10
  icmp    172.16.1.1:6    10.1.1.1:6     192.168.1.2:6    192.168.1.2:6
  icmp    172.16.1.1:7    10.1.1.1:7     192.168.1.2:7    192.168.1.2:7
  icmp    172.16.1.1:8    10.1.1.1:8     192.168.1.2:8    192.168.1.2:8
  icmp    172.16.1.1:9    10.1.1.1:9     192.168.1.2:9    192.168.1.2:9
  ---        172.16.1.1          10.1.1.1       ---           ---
RouterA#

```

- 记住，路由器随后很快就会清除该NAT转换，为其它IP地址使用这个/这些NAT地址而对其进行清理。

```

NAT: expiring 172.16.1.1 (10.1.1.1) icmp 6 (6)
NAT: expiring 172.16.1.1 (10.1.1.1) icmp 7 (7)

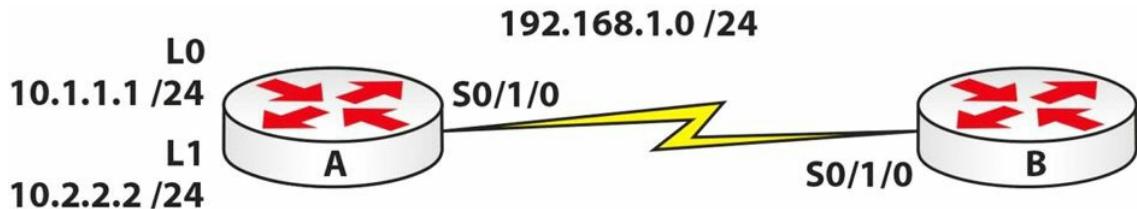
```

译者注：通过本实验，要注意三个问题：一是可路由地址可以是外部接口同一网段的地址，也可以不是；二是NAT超时问题，该参数可以设置；三是环回接口的使用，常用来模拟LAN中的计算机。

## NAT地址池/动态NAT实验

### NAT Pool Lab

#### 拓扑图



NAT地址池/动态NAT实验拓扑图

### 实验目的

学习如何配置一个NAT地址池（动态NAT）。

### 实验步骤

1. 将IP地址 192.168.1.1 255.255.255.0 加入到路由器 A，并修改 hostname 为 Router A。把IP地址 192.168.1.2 255.255.255.0 加入到路由器 B。在正确的一侧加上时钟速度( clock rate )，然后分别自 A 往 B 和自 B 往 A 进行 ping 测试。如需提示，请回顾先前的那些实验。
2. 需要给 RouterA 添加两个IP地址来模拟LAN上的主机。通过两个环回接口，可以达到这个目的。这两个IP地址将位处不同子网，但都以 10 地址开头。

```
RouterA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#interface Loopback0
RouterA(config-if)#ip add 10.1.1.1 255.255.255.0
RouterA(config-if)#int l1 - short for Loopback1
RouterA(config-if)#ip address 10.2.2.2 255.255.255.0
RouterA(config-if)#

```

1. 为了进行测试，需要告诉 RouterB 将到任何网络的任何流量，都发往 RouterA。用一条静态路由完成这点。

```
RouterB#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterB(config)#ip route 0.0.0.0 0.0.0.0 Serial0/1/0
RouterB(config)#

```

1. 在 RouterA 上，从环回接口向 RouterB 发出 ping 操作，以此来测试该静态路由是否工作。

```

RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms
RouterA#

```

- 在 RouterA 上配置一个NAT地址池。在本实验中，使用地址池 172.16.1.1 到 172.16.1.10。任何以 10 开头的地址，都将成为一个NAT。记住你必须指定NAT的内部和外部接口，否则NAT就不会工作。

```

RouterA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
RouterA(config)#int lo
RouterA(config-if)#ip nat inside
RouterA(config)#int l1
RouterA(config-if)#ip nat inside
RouterA(config-if)#int Serial0/1/0
RouterA(config-if)#ip nat outside
RouterA(config-if)#exit
RouterA(config)#ip nat pool 60days 172.16.1.1 172.16.1.10 netmask 255.255.255.0
RouterA(config)#ip nat inside source list 1 pool 60days
RouterA(config)#access-list 1 permit 10.1.1.0 0.0.0.255
RouterA(config)#access-list 1 permit 10.2.1.0 0.0.0.255
RouterA(config)#

```

命令 `ip nat pool` 创建出地址池。需要给地址池一个自己选择的名称。而命令 `netmask` 告诉路由器应用到地址池上的网络掩码。

命令 `source list` 告诉路由器查看的ACL。该条ACL告诉路由器哪些网络将与NAT地址池进行匹配和转换。

- 打开NAT调试，如此才可以看到转换的发生。接着执行扩展 `ping`（自 `L0` 和 `L1` 发出的），并查看 NAT表。因为IOS平台的不同，你的输出可能和下面的不一样。将会看到NAT地址池中的两个地址正在用到。

```

RouterA#debug ip nat
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.1.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.1.1.1
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [26]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [16]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [27]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [17]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [28]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [18]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [29]
!
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [19]
NAT: s=10.1.1.1->172.16.1.1, d=192.168.1.2 [30]
!
Success rate is 100 percent (5/5), round-trip min/avg/max = 17/28/32 ms
RouterA#
NAT*: s=192.168.1.2, d=172.16.1.1->10.1.1.1 [20]
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.2
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 10.2.2.2
Type of service [0]:
Set DF bit in IP header? [no]:Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.2, timeout is 2 seconds:
Packet sent with a source address of 10.2.2.2
NAT: s=10.2.2.2->172.16.1.2, d=192.168.1.2 [31]
!
NAT*: s=192.168.1.2, d=172.16.1.2->10.2.2.2 [21]
NAT: s=10.2.2.2->172.16.1.2, d=192.168.1.2 [32]
!
NAT*: s=192.168.1.2, d=172.16.1.2->10.2.2.2 [22]
NAT: s=10.2.2.2->172.16.1.2, d=192.168.1.2 [33]
!
NAT*: s=192.168.1.2, d=172.16.1.2->10.2.2.2 [23]
NAT: s=10.2.2.2->172.16.1.2, d=192.168.1.2 [34]
!
NAT*: s=192.168.1.2, d=172.16.1.2->10.2.2.2 [24]
NAT: s=10.2.2.2->172.16.1.2, d=192.168.1.2 [35]
!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms

```

```

RouterA#
NAT*: s=192.168.1.2, d=172.16.1.2->10.2.2.2 [25]
RouterA#show ip nat trans
Pro      Inside global     Inside local     Outside local     Outside global
icmp    172.16.1.1:16    10.1.1.1:16    192.168.1.2:16   192.168.1.2:16
icmp    172.16.1.1:17    10.1.1.1:17    192.168.1.2:17   192.168.1.2:17
icmp    172.16.1.1:18    10.1.1.1:18    192.168.1.2:18   192.168.1.2:18
icmp    172.16.1.1:19    10.1.1.1:19    192.168.1.2:19   192.168.1.2:19
icmp    172.16.1.1:20    10.1.1.1:20    192.168.1.2:20   192.168.1.2:20
icmp    172.16.1.2:21    10.2.2.2:21    192.168.1.2:21   192.168.1.2:21
icmp    172.16.1.2:22    10.2.2.2:22    192.168.1.2:22   192.168.1.2:22
icmp    172.16.1.2:23    10.2.2.2:23    192.168.1.2:23   192.168.1.2:23
icmp    172.16.1.2:24    10.2.2.2:24    192.168.1.2:24   192.168.1.2:24
icmp    172.16.1.2:25    10.2.2.2:25    192.168.1.2:25   192.168.1.2:25
RouterA#

```

## NAT Overload实验

### NAT Overload Lab

重复先前的实验。这次，在引用地址池时，将 `overload` 命令加到该配置行的后面。这会指示路由器使用 PAT。去掉 `Loopback1`。请注意，正如Farai指出的那样，在真实世界中，地址池通常只会有一个地址，否则在外部接口上会超载（Please note that as Farai says, in the real world, your pool will usually have only one address or you will overload your outside interface）。

```
RouterA(config)#ip nat inside source list 1 pool 60days overload
```

我已经为方便而使用思科Packet Tracer，完成了上面的实验，所以你通常会碰到与我的输出所不一致的输出。下面是一个PAT实验的示例输出。从中可以看出，路由器给每个转换都加上了一个端口号。不幸的是，在NAT地址池实验中，会看到相似的编号，这是一个PAT的混淆之处。

```

RouterA#show ip nat tran
Inside global     Inside local     Outside local     Outside global
10.0.0.1:8759    172.16.1.129:8759   192.168.1.2:8759   192.168.1.2:8759

```

# 第7天 互联网协议版本6

## Internet Protocol version 6, IPv6

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第七天任务

- 阅读下面的课文理论部分
- 阅读ICND1记诵指南

IPv6已经开发了很多年，且已在全世界网络中投入使用（与IPv4共同运行）。许多网络工程师在面对不得不学习一种新的分址方式时，表现出了他们的恐惧，笔者也曾听他们中的许多人说希望在IPv6成为一项必备技能之前能够退休。

恐惧是站不住脚的，是没有有事实依据的。IPv6是一种对用户友好格式，一旦对其熟悉了，就会发现其是IPv4的改良，而你可能会优先选用IPv6。**CCNA考试中IPv6占了很大部分**；为此，需要**掌握其工作原理，及如何配置IPv6地址，掌握其有关标准，并应用IPv6来满足网络的各项需求**。

今天将会学到下面这些知识点。

- IPv6的历史
- IPv6分址格式
- 应用IPv6
- IPv6子网划分

本模块对应了以下CCNA大纲要求。

- 拿出恰当的IPv6分址方案，以满足某个LAN/WAN环境的分址要求
- 正确描述IPv6的各种地址
  - 全球单播地址, Global Unicast addresses
  - 多播地址, Multicast addresses
  - 本地链路地址, Link-Link addresses
  - 本地唯一地址, Unique-Local addresses
  - 扩展唯一识别符, Extended Unified Identifier 64, EUI-64
  - 自动配置地址 (autoconfiguration)

## IPv6历史

### History of IPv6

## 满足目标吗？

### Fit for Purpose?

在 Tim Berners-Lee 爵士于 1989 年发明 WWW 时，他无法预测到该技术对世界的巨大影响。个人计算机曾经贵得高攀不起，此外，除非能够负担得起昂贵的 WAN 连接费用，否则就没有方便的长距离通信方法。那时也没有大家共同遵循的通信模型。

那时，某些事需要一些变化，以 IP 这种新型分址标准的形式，变革发生了。业界从犯下的大量失误中终有收获，并在对商业需求的回应下，IETF 早在 1998 年就发布了众多 IPv6 标准中最早的一些标准。

并不会有一个日期，能够整个地从 IPv4 转变为 IPv6；而是网络将会逐渐地变为同时运行 IPv4 和 IPv6，并最终 IPv4 会滚粗。当下，全部互联网流量的近 1% 运行在 IPv6 上（来源：Yves Poppe, IPv6 -- A 2012 Report Card）。

## 为何要迁移

### Why Migrate?

笔者已经指出，在 IPv4 发明时，互联网不是由普罗大众所使用的，也没有使用的必要。那时还没有网站，没有电子商务，没有移动网络，没有社交媒体。就算买得起 PC，拿来也干不了什么事。现在的情况是几乎所有人都在线上了。我们使用互联网来完成日常工作，很多业务都依赖互联网而存在。很快我们又会使用移动装置来管理我们的汽车及家庭安防，来打开咖啡机，设置空调，设定电视录制爱看的电视剧等等。

这些事情已经在发生当中，不光在欧洲和美国，在那些有着数十亿人口的快速发展中国家，比如印度和中国，都在发生着。IPv4 就是不能胜任了，就算勉强可以，也没有足够的地址来满足需求。

下面是迁移到 IPv6 所能带来的一些好处。

- 简化了的 IPv6 数据包头部
- 更大的地址空间
- IPv6 层次化的分址方法
- IPv6 的扩展性扩充性
- IPv6 消除了广播
- 无状态的自动配置
- 集成移动能力
- 集成了安全增强

我喜欢从其数据包层的探究，来分析 IPv6，同时也会去探究 IPv6 中可用的许多种类型的包头部，但限于篇幅，同时考试中也不会考到这两点，所以就不包含这两方面的内容了。而着重在为考试和成为一名思科工程师，所需要掌握的内容上。

## 十六进制计数

### Hex Numbering

这里很有必要回顾一下有关十六进制计数的内容。

我们知道十进制数有着从 0 到 9 的 10 个数字。二进制则有从 0 到 1 的 2 个数字。那么十六进制就有从 0 到 F 的 16 个数字。这些地址分别叫做基数 10、基数 2 和基数 16 的地址。

可以发现各个计数系统都是从 0 开始的，就像下面这样。

十进制 -- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9    二进制 -- 0, 1    十六进制 -- 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

在写下这些地址时，可能不会意识到是在使用那些从右往左的列；最右边的列是权重为  $1$  的列，接下来的列，是权重为计数基数的前一列序号次幂的列。如同下表所示。

计数基数	$N$ 乘计数基数的 $3$ 次幂	$N$ 乘计数基数的 $2$ 次幂	$N$ 乘计数基数的 $1$ 次幂	$N$ 乘 $1$
$10$ -- 十进制	1000	100	10	1
$2$ -- 二进制	8	4	2	1
$16$ -- 十六进制	4096	256	16	1

可以看出每一位都从其右边的那位继承了数值。十进制基数是  $10$  乘  $1$ 。二进制是  $1$ ，同时  $1$  乘了计数系统的  $2$ 。如对三种计数系统的最后一个十六进制数位进行比较，就会发现将十六进制作为IPv6分址首选格式的原因了。

十进制	二进制	十六进制
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

为提供足够的地址来满足我们在今后许多年的需求，IPv6已被设计成可以提供数以百亿亿的地址。为做到这点，计数范围从  $32$  位二进制数，扩展到  $128$  位。每  $4$  位可用一个十六进制数位表示（这可从上面的图表看出）。逻辑上推断就是  $2$  个十六进制位给出的是  $8$  位二进制数，也就是一个字节。

一个IPv6地址有  $128$  位长，又被分为  $8$  组的  $16$  位，在以完整格式写出时，用冒号将每组分开。每  $4$  位十六进制数的范围是  $0000$  到  $FFFF$ ，其中F是十六进制计数方法中最高的数。

第 8 组	第 7 组	第 6 组	第 5 组	第 4 组	第 3 组	第 2 组	第 1 组
0000	0000	0000	0000	0000	0000	0000	0000
to							
FFFF							

## IPv6 分址

### IPv6 Addressing

我们已经知道，IPv6用到 128 位的地址。因为这种地址格式不同于我们所熟悉的IPv4地址格式，在初次见到时通常会犯迷糊。但是，一旦掌握了，那么就知道其逻辑和结构都十分简单。这些 128 位的IPv6地址，使用了十六进制数值（也就是说，0 到 9 以及字母 A 到 F）。而在IPv4中，子网掩码既可以用CIDR表示法表示（比如 /16 或 /32），也可以用点分十进制表示法表示（dotted-decimal notation，比如 255.255.0.0 或 255.255.255.255），但在IPv6中，子网掩码只用CIDR表示法表示，因为IPv6地址的长度很长。全球范围内的 128 位IPv6地址，由下面 3 部分组成。

- 由服务商分配的前缀，the provider-assigned prefix
- 站点前缀，the site prefix
- 接口或主机ID，the interface or host ID

所谓服务商分配的前缀，也被称作**全球地址空间**(the global address space)，是一个**48位**的前缀，又被分为下面的 3 部分。

- 16 位保留的IPv6全球前缀，the 16-bit reserved IPv6 global prefix
- 16 位服务商持有的前缀，the 16-bit provider-owned prefix
- 16 位服务商分配给其客户的前缀，the 16-bit provider-assigned prefix

**IPv6 全球前缀**，用于表示**IPv6 全球地址空间**（the IPv6 global address space）。所有**IPv6 全球互联网地址**，都位于从 2000::/16 到 3FFF::/16 的范围。而 16 位服务商持有的IPv6前缀，是IANA分配给服务商，且归其所有的。ISP持有前缀，处于 0000::/32 到 FFFF::/32 范围。

接下来的 16 位，表示由实际服务提供商从其分到前缀地址空间中，再分配给某个组织的IPv6前缀。该前缀处于 0000::/48 到 FFFF::/48 范围。于是，前 48 位就共同构成了IPv6地址第一部分 -- 服务提供商分配的前缀，如下图7.1所示。

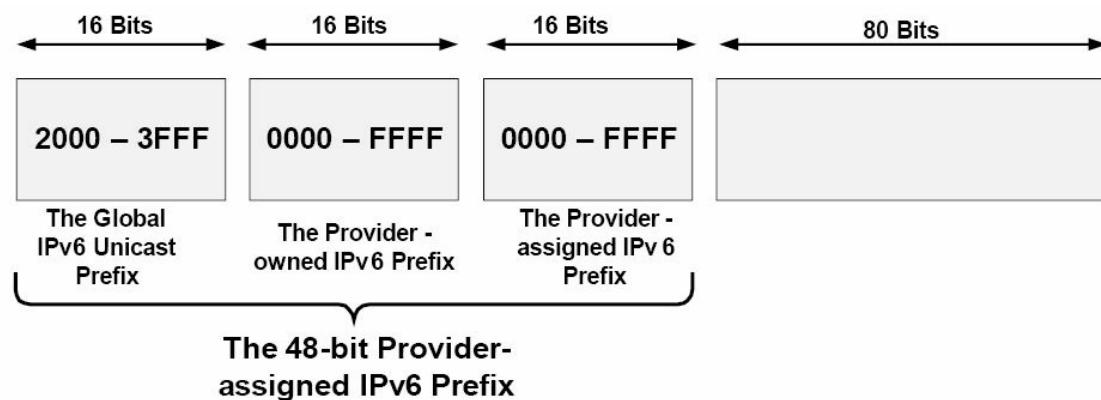


图7.1 -- 48 位服务提供商分配的IPv6前缀

在 48 位服务商分配的前缀之后，紧接着的 16 位就是站点前缀。站点前缀的子网掩码长度是 /64，该子网掩码已经包括了之前的 48 位服务商分配的前缀。此前缀长度允许在每个站点前缀中有  $2^{16}$  次幂个地址。图7.2演示了该 16 位站点前缀。

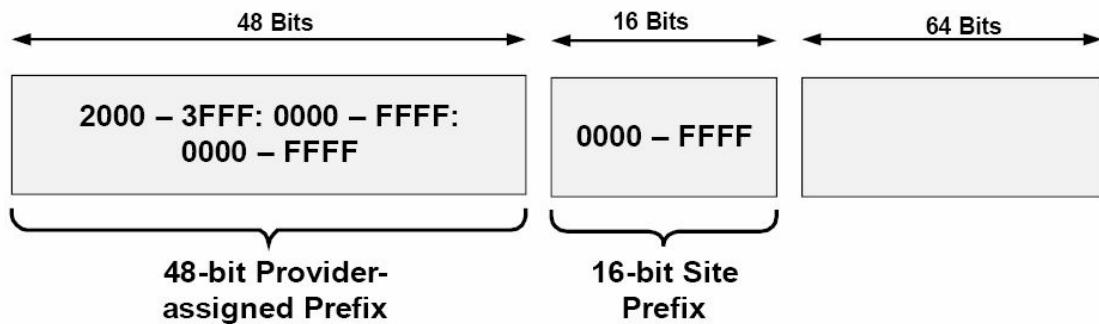


图7.2 -- 16 位的IPv6站点前缀

而在站点前缀之后，接下来的 64 位就用于接口或主机的分址了。IPv6 地址的接口或主机 ID 部分，表示了某个 IPv6 子网上的某台网络设备或主机。至于确定接口或主机地址的不同方式，在今天的课程稍后会详细讲到。图7.3说明了IPv6的这些前缀是如何分配的。

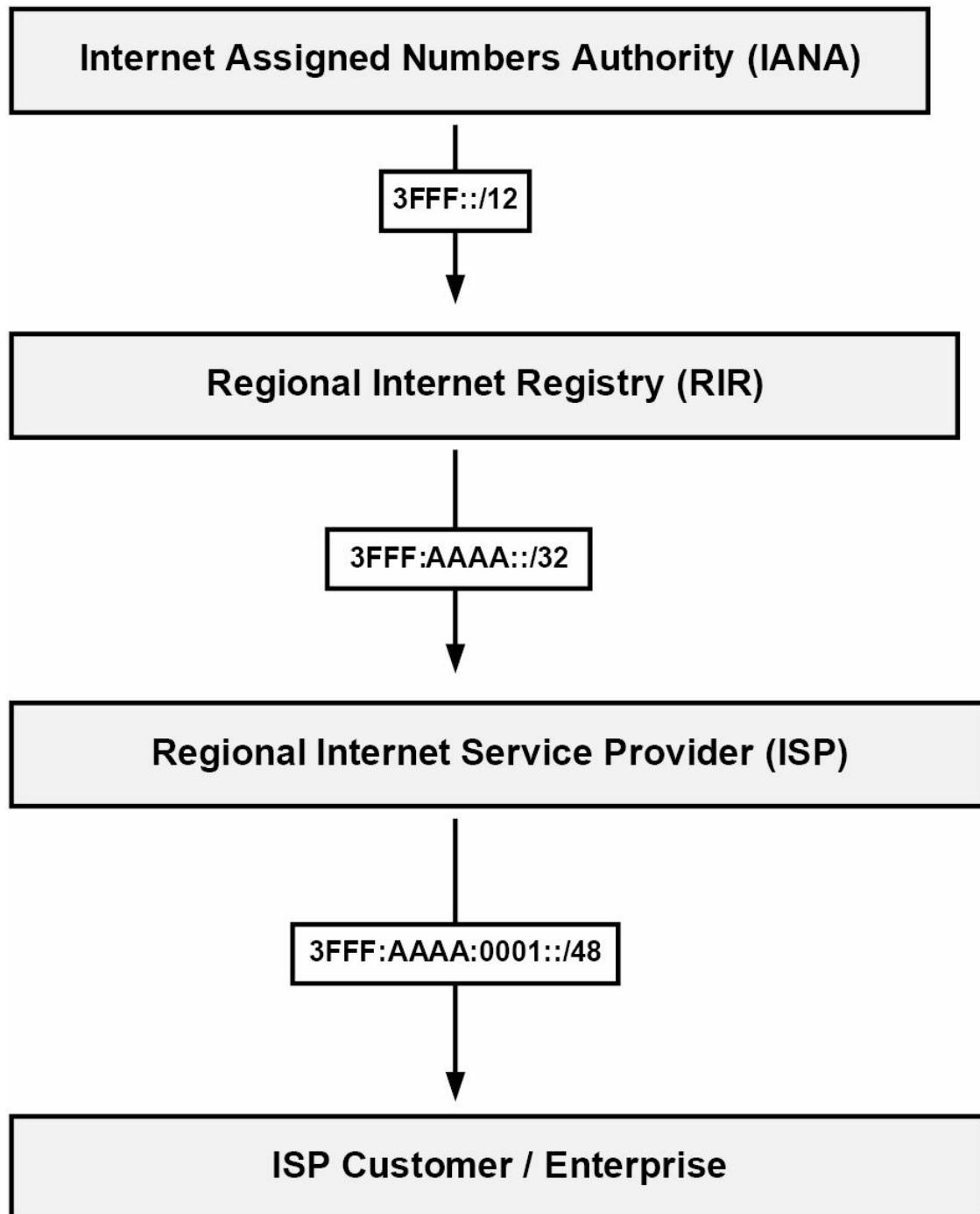


图7.3 -- IPv6前缀的分配

参考图7.3, 客户一旦收到由ISP提供的 /48 前缀, 就可以该前缀范围内, 对站点前缀和主机或接口地址进行自由分配了。基于可用的地址空间全部容量, 任何单一机构客户, 只需一个的服务商分配前缀, 机构网络上的所有设备就保证可以分配到一个唯一IPv6全球地址。因此, IPv6绝对不需要NAT这样的技术。

## IPv6 地址表示法

### IPv6 Address Representation

IPv6地址可像下面这三种方式进行表示。

- 首选的或者说完整地址表示/形式

- 压缩的表示法
- 带有一个嵌入了IPv4地址的IPv6地址

尽管在以文本格式表示 128 位IPv6地址时，首选形式或表示法是最常用的方式，熟悉其它两种IPv6地址表示法也很重要。下面会对这三种方式进行说明。

## 首选形式

### The Preferred Form

**IPv6地址的首选表示法**(the preferred representation for an IPv6 address)，有着最长的格式，又被称作**IPv6地址的完整形式**(the complete form of an IPv6 address)。此格式表示法使用 32 个十六进制字符，以构成一个IPv6地址。通过将某地址写作共八组的十六进制字段，用冒号将这 8 个字段分开（比如，`3FFF:1234:ABCD:5678:020C:CEFE:FEA7:F3A0`）。

每个 16 位字段，由4个十六进制字符表示，那么每个字符就表示了 4 位。每个 16 位十六进制字段，可以是 `0x0000` 和 `0xFFFF` 之间的值，但就如同今天后面讲到的那样，第一组的一些数值已被保留，那么所有可能的数值都不被使用 (as will be described later in this module, different values have been reserved for use in the first 16 bits, so all possible values are not used)。在书写IPv6地址时，十六进制字符不区分大小写。也就是说，`2001:ABCD:0000` 和 `2001:abcd:0000` 是完全一样的。IPv6地址表示法的完整形式，在下图7.4中有演示。

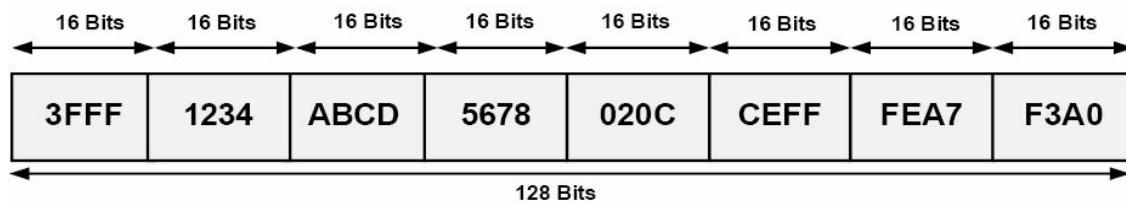


图7.4 -- IPv6地址表示法的首选形式

下面的这些IPv6地址，是完整形式下的有效IPv6地址实例。

- `0000:0000:0000:0000:0000:0000:0000:0001`
- `2001:0000:0000:1234:0000:5678:af23:bcd5`
- `3FFF:0000:0000:1010:1A2B:5000:0B00:DE0F`
- `fec0:2004:ab10:00cd:1234:0000:0000:6789`
- `0000:0000:0000:0000:0000:0000:0000:0000`

## 压缩的表示法

### Compressed Representation

压缩的表示法，允许以两种压缩方式之一，对IPv6地址进行压缩。第一种压缩方式，允许使用一对冒号（`::`），对一个有效IPv6地址中的那些由 `0s` 构成的连续 16 位字段的连续的 `0` 值，或者IPv6地址中前面的 `0s`，进行压缩。在使用这种方式时，务必要记住，双冒号在一个IPv6地址中，只能使用一次。

在用到压缩格式时，各个节点及各台路由器，负责去对双冒号两侧的位数进行计数，以判断出该双冒号究竟表示了多少个 `0s`。表7.1显示了那些IPv6地址的首选形式及其压缩表示法。

表7.1 -- 首选和压缩形式下的完整IPv6地址

完整IPv6地址表示法	压缩的IPv6地址表示法
0000:0000:0000:0000:0000:0000:0000:0001	::0001
2001:0000:0000:1234:0000:5678:af23:bcd5	2001::1234:0:5678:af23:bcd5
3FFF:0000:0000:1010:1A2B:5000:0B00:DE0F	3FFF::1010:1A2B:5000:B00:DE0F
FEC0:2004:AB10:00CD:1234:0000:0000:6789	FEC0:2004:AB10:CD:1234::6789
0000:0000:0000:0000:FFFF:172.16.255.1	::FFFF:172.16.255.1
0000:0000:0000:0000:0000:172.16.255.1	::172.16.255.1
0000:0000:0000:0000:0000:0000:0000:0000	::

跟前面指出的那样，在单个的IPv6地址中，双冒号不能多于一次地使用。比如说，如要对这个完整IPv6地址 2001:0000:0000:1234:0000:0000:af23:bcd5 以压缩形式表示，那么你就只能使用双冒号一次，就算在该地址中有两组连续的 0 字符串。那么，在尝试将该地址压缩成 2001::1234::af23:bcd5，就被看成是非法的；但是此IPv6地址既可以压缩成 2001::1234:0:0:af23:bcd5，也可以压缩成 2001:0:0:1234::af23:bcd5，取决于自己喜好。

第二种IPv6压缩地址表示法，对于单个的 16 位字段，及前导 0s，可从该IPv6地址中省略成单个的 0。在使用该方法时，如某个 16 位字段都是 0，那么就必须用一个 0 来表示此字段。在这种情况下，并非所有的 0 都能省略。表7.2中展示了首选形式的IPv6地址，以及它们怎样通过第二种IPv6压缩形式表示法进行压缩。

表7.2 -- 以替代的压缩形式表示的完整IPv6地址

完整IPv6地址表示法	压缩IPv6地址表示法
0000:0123:0abc:0000:04b0:0678:f000:0001	::123:abc:0:4b0:678:f000:1
2001:0000:0000:1234:0000:5678:af23:bcd5	2001::1234:0:5678:af23:bcd5
3FFF:0000:0000:1010:1A2B:5000:0B00:DE0F	3FFF::1010:1A2B:5000:B00:DE0F
fec0:2004:ab10:00cd:1234:0000:0000:6789	fec0:2004:ab10:cd:1234::6789
0000:0000:0000:0000:FFFF:172.16.255.1	::FFFF:172.16.255.1
0000:0000:0000:0000:0000:172.16.255.1	::172.16.255.1
0000:0000:0000:0000:0000:0000:0000:0000	::

这里就有了两种以压缩形式表示完整IPv6地址的方法，要记住，**两种方法之间并不互相排斥**。也就是说，在表示一个IPv6地址时，可以同时使用这两种方法。当某个完整IPv6地址既包含了连续 0s 字符串，又在其它字段中有前导 0s 时，这经常会用到。表7.3展示了一些既包含了连续 0s 字符串，又有前导 0s 的一些IPv6地址的完整形式，以及如何将这些地址表示成压缩形式。

表7.3 -- 使用了两种压缩格式方法的完整IPv6地址

完整IPv6地址表示法	压缩IPv6地址表示法
0000:0000:0000:0000:1a2b:000c:f123:4567	::1a2b:c:f123:4567
FEC0:0004:AB10:00CD:1234:0000:0000:6789	FEC0:4:AB10:CD:1234::6789
3FFF:C00:0000:1010:1A2B:0000:0000:DE0F	3FFF:c00:0:1010:1A2B::DE0F
2001:0000:0000:1234:0000:5678:af23:00d5	2001::1234:0:5678:af23:d5

## 带有一个嵌入的IPv4地址的IPv6地址

### IPv6 Addresses with an Embedded IPv4 Address

这是第三种IPv6地址表示法，用于在IPv6地址内部使用一个IPv4地址。尽管这也是有效的IPv6地址，但请记住这种方法是不赞成的做法，同时也在考虑废弃这种方法，因为该方法仅适用于从IPv4到IPv6的过渡。

## IPv6地址的不同类型

### The Different IPv6 Address Types

IPv4支持4中不同类别的地址，分别是任意播（Anycast）、广播(Broadcast)、多播(Multicast)及单播(Unicast)地址。尽管在本教程之前的模块中并未用到任意播一词，但要记住，**任意播地址并非特殊类型的地址**。相反，一个任意播地址简单地就是一个分配给多个接口的IP地址。常见的使用了任意播的技术包括IP多播应用(IP Multicast implementations)，以及 6to4 中继应用( 6to4 relay implementation)。

**注意：** 6to4 是一种IPv4迁移到IPv6的过渡机制。对于CCNA考试来说，只需知道有这么个东西就行了。

在任意播寻址方式下，设备使用从路由协议度量值上看离它们最近的那个公共地址(the common address)。假如该主要地址不可达时，就会使用下一个最近的地址 (with Anycast addressing, devices use the common address that is closest to them based on routing protocol metric. The next closest address is then used in the event that the primary address is no longer reachable)。此概念在下图7.5中进行了演示。

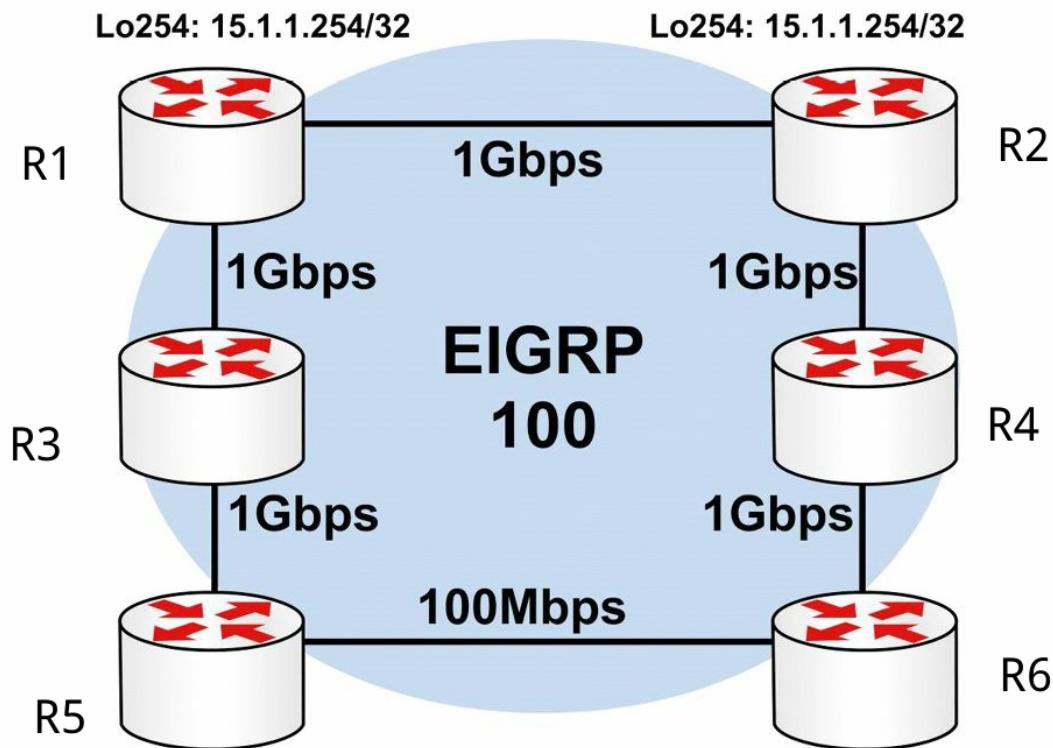


图7.5 -- 理解任意播寻址方式

在图7.5中，R1 和 R2 都有一个配置了公共地址 15.1.1.254/32 的环回接口 Loopback 254。该前缀此时会经由EIGRP进行通告。默认情况下，R1 和 R2 都会经由它们各自的相应环回接口，优先选择 15.1.1.254/32 前缀，因为该前缀是一个直接连接的子网。因此，两台路由器上所使用的公共地址绝不会发生冲突。

假定是在一般EIGRP度量值计算下，则 R<sub>3</sub> 和 R<sub>5</sub> 都会优先选择 R<sub>1</sub> 通告的那个任意播地址（the Anycast address），这是由于其有着较小的内部网关协议（Interior Gateway Protocol, IGP）度量值(due to the lower IGP metric)。同样 R<sub>4</sub> 和 R<sub>6</sub> 则会优先选择R<sub>3</sub>通告的那个任意播地址，也是由于其有着较小的IGP度量值。要是 R<sub>1</sub> 或 R<sub>3</sub> 中的某台失效，网络中的路由器就会使用由剩下的那台路由器通告的任意播地址了。某个组织在应用任意播分址时，既可以使用 RFC 1918 中定义的地址空间中的某个单播地址(私有地址)，也可以使用其公网地址块中的某个单播地址。

**注意：**当前的CCNA考试并不要求你采用任何的任意播分址或解决方案。但熟悉此概念是必要的。在完成路由章节的学习后，你将更为明白。译者注：关于任意播，可以参看 [wikipedia.org/wiki/Anycast](https://en.wikipedia.org/wiki/Anycast)，简单地说，任意播是一种冗余方法，可用来做负载均衡、加快访问速度。

在CCNA层次，IPv4的广播、多播及单播地址都无需更为详尽地阐述，本课程及本模块都不会对它们进行更为详细的说明。与IPv4支持这四种类型的地址相比，IPv6废除了广播地址，同时取而代之的仅支持以下类型的地址。

- 本地链路地址，Link-Local addresses
- 站点本地地址，Site-Local addresses
- 可聚合全球单播地址，Aggregatable Global Unicast addresses
- 多播地址，Multicast addresses，已被废除，取而代之的是本地唯一地址（Unique-Local addresses, ULAs）
- 任意播地址，Anycast addresses
- 环回地址，Loopback addresses
- 未指明的地址，Unspecified addresses， ::/128

## 本地链路地址

### Link-Local Addresses

IPv6本地链路地址只能用在本地链路上（也就是一个设备间所共享的网段），是在某个接口上开启了IPv6时，自动分配给接口的。这些地址分配自本地链路前缀（the Link-Local prefix） FE80::/10。记住 FE80::/10 等价于 FE80:0:0:0:0:0:0:0 /10，又可以表示为 FE80:0000:0000:0000:0000:0000:0000:0000/10。为了构成该地址，从第 11 到 64 位被设置为 0，同时接口的 EUI-64 (Extended Unique Identifier 64, 64位扩展唯一标识) 给追加到本地链路地址上去，作为下一顺序的 64 位（the lower-order 64 bits）。EUI-64 是由 IEEE 分配给接口厂商的 24 位 ID(Organization Unified Identifier, OUI)，以及厂商分配给其产品的 40 位值构成。本模块稍后会更为详细地说明 EUI-64 分址。图7.6演示了本地链路地址的格式。

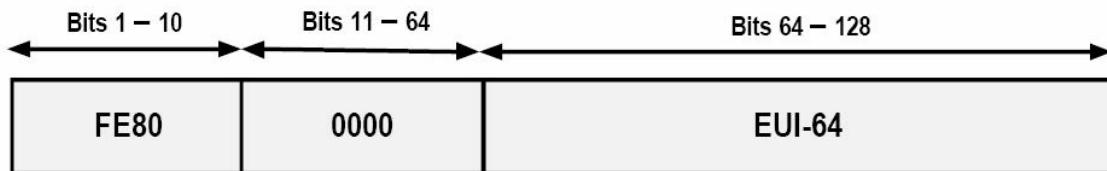


图7.6 -- IPv6本地链路分址

本地链路地址是唯一的，一旦分配给了某个接口，就不再改变。这就是说，某个接口在分配了一个公网IPv6地址后（比如， 2001:1000::1/64），就算该公网IPv6前缀发生改变(变成 2001:2000::1/64 )，本地链路地址也是不会改变的。这允许主机或路由器在IPv6全球互联网地址改变时，对其邻居始终保持可达。而 IPv6 路由器是不会转发那些以本地链路地址作为源或目的地址的数据包，到其它IPv6路由器的。

## 站点本地地址

### Site-Local Addresses

站点本地地址是那些仅在某个站点内部使用的地址。与本地链路地址不同，必须在网络设备上手动为其配置站点本地地址。这些地址就是在IPv6中，与RFC 1918所定义的私有IPv4地址等价的地址，对于那些没有可全球路由IPv6地址空间的组织，可以使用这些地址。在IPv6互联网上，这些地址是不可路由的。

尽管在IPv6上进行NAT是可能的，但绝不建议这么做。理由就是有着大得多的IPv6地址（hence, the reason for the much larger IPv6 addresses）。站点本地地址是由 `FEC0::/10` 前缀、该前缀之后的54位子网ID，以及同样的为本地链路地址所用到的EUI-64格式的接口ID组成。与本地链路地址中设置为0的54位相比，站点本地地址中的54位，被用于构建不同的IPv6前缀（最多2的54次幂个）。下图7.7演示了站点本地地址的格式。

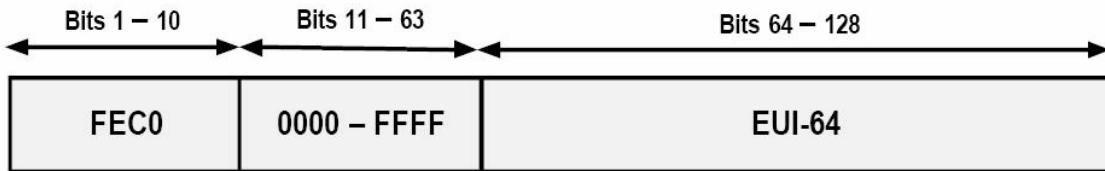


图7.7 -- IPv6站点本地分址

尽管在本章节中有对IPv6站点本地地址进行说明，同时在思科IOS软件中仍有对其的支持，但要知道这些地址已被RFC 3879（废弃站点本地地址，Deprecating Site Local Addresses）所废弃。与此同时，RFC 4193（唯一本地IPv6单播地址，Unique Local IPv6 Unicast Addresses）又阐述本地唯一地址（Unique-Local addresses, ULAs），本地唯一地址提供了站点本地地址的功能，它们在IPv6全球互联网上也是不可路由的，仅能在某个站点内部路由。

本地唯一地址分配自 `FC00::/7` 这个IPv6地址块，该地址块又被划分成两个 /8 的地址块，分别作为分配组和随机组（the assigned and random groups）。那么这两组就分别是 `FC00::/8` 和 `FD00::/8` 了。`FC00::/8` 这个地址块是由一个分配机构（an allocation authority）管理其使用到的 /48s，同时 `FD00::/8` 地址块则是通过在其后追加上随机生成的 40 位字符串，得到的一个有效 /48 地址块的。

### 可聚合全球单播地址

#### Aggregatable Global Unicast Addresses

可聚合全球单播地址，就是那些用于一般IPv6流量传输、IPv6互联网的IPv6地址了。这些地址与IPv4中用到的公网地址相似。而从网络分址角度看，每个IPv6全球单播地址都是由三个主要部分构成的：自服务商处收到的前缀（48位长）、站点前缀（16位长），以及主机部分（64位长）。这就构成了IPv6中用到的 128 位地址了。

如同本模块前面提到的，服务商分配的前缀，是由IPv6服务提供商分配给作为其客户的某家组织的。默认情况下，这些前缀用到 /48 的前缀长度。此外，这些前缀又是从该服务提供商所拥有的IPv6地址空间中分配的（也就是 /32 前缀长度）。每家服务提供商都将有着其自己的IPv6地址空间，同时由一家服务提供商分配的IPv6前缀，不能在另一家的网络上使用。

而在某个站点内部，管理员此时就能通过用于子网划分的第 49 到 64 位，将服务提供商分配的 48 位前缀，划分成 64 位的站点前缀，从而可以得到 65535 个不同的，可在其网络中使用的子网。IPv6地址的主机部分表示该IPv6子网上的某台网络设备或主机。而这又是通过IPv6地址的低 64 位表示的（this is represented by the low-order 64 bits of the IPv6 address）。

IPv6的可聚合全球单播地址，是由互联网号码分配局（the Internet Assigned Numbers Authority, IANA）分配的，这些地址处于IPv6前缀 2000::/3 中。此前缀允许的可聚合全球单播地址范围是从 2000 到 3FFF，如下表7.4所示。

表7.4 -- IPv6可聚合全球单播地址

说明	地址
范围中的第一个地址	2000:0000:0000:0000:0000:0000:0000:0000
范围中的最后一个地址	3FFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF:FFFF
二进制标记	高位序的三位被设置为 001

在本模块编写时，2000::/3 IPv6地址块中，仅分配使用了 3 个子网。这三个子网如下表7.5所示。

表7.5 -- 由IANA所分配的IPv6可聚合全球单播地址

IPv6全球前缀	二进制表示法	说明
2001::/16	0010 0000 0000 0001	全球IPv6互联网(单播)
2002::/16	0010 0000 0000 0000	6to4 迁移前缀
3FFE::/16	0010 1111 1111 1110	6bone 前缀

注意：6to4迁移地址和6bone前缀将在本课程的后面说明。

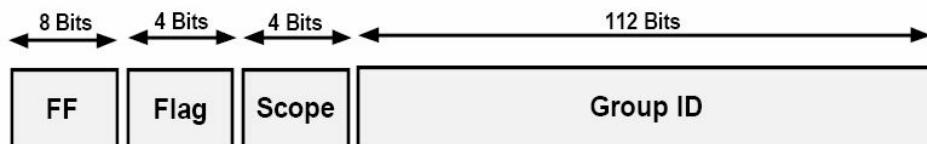
在IPv6全球可聚合单播地址范围，保留了一个叫做ORCHID（RFC 4843 中定义的覆盖可路由加密散列标识、Overlay Routable Cryptographic Hash Identifiers defined in RFC 4843）特别的实验范围。ORCHID是用于加密散列标识的不可路由IPv6地址。这些地址使用IPv6前缀 2001:10::/28。关于ORCHID地址的细节，是超出当前CCNA考试要求范围的，本模块或本课程都不会包含。

## 多播地址

### Multicast Addresses

IPv6中用到的多播地址，是从 FF00::/8 这个IPv6前缀中得到的。IPv6中的多播和IPv4中的多播，运行的方式是不一样的。**IPv6中重度使用到IP多播**，并用IP多播替换了一些诸如地址解析协议（Address Resolution Protocol, ARP）这样的IPv4协议。此外，IPv6中还用多播来完成前缀通告及其重编号（prefix advertisements and renumbering），以及重复地址侦测（Duplicate Address Detection, DAD）等。本模块后面会对这些概念进行说明。

**IPv6中的多播数据包**，不是通过使用TTL值来将其限制在本地网段上。代之以**使用多播地址内部的范围字段（the Scope field）**，**定义出其范围**。网段上的IPv6节点，都侦听着多播包，甚至也会发出多播包来交换信息。这样，IPv6网段上所有节点，都知道在其同一网段上所有其它邻居节点了。下图7.8中演示了IPv6网络中用到的多播地址格式。



### 图7.8 -- IPv6多播分址

如同图7.8中所演示的那样，IPv6多播地址格式与其它之前学到的IPv6地址略有不同。IPv6多播地址的前8位表示多播前缀 `FF::/8`。IPv6多播地址的标志字段（the Flag field）用于指明多播地址类型 -- 是永久的还是临时的。

**IPv6永久多播地址是由IANA分配的，而IPv6临时地址则可用于多播预部署的测试(Permanent IPv6 Multicast addresses are assigned by IANA, while temporary IPv6 Multicast addresses can be used in pre-deployment Multicast testing)。** 标志字段所包含的值可以是表7.6中所示的两个。

表7.6 -- IPv6永久及临时多播地址

多播地址类型	二进制表示法	十六进制值
永久	0000	0
临时	0001	1

多播地址中接下来的4位表示**多播范围**。在IPv6多播分址中，该字段是一个**用于限制多播数据包发往网络其它区域的强制字段**（this field is a mandatory field that restricts Multicast packets from being sent to other areas in the network）。该字段本质上提供了与IPv4中所用到的TTL字段一样的功能。但是，在**IPv6中**，范围的类型有好几种，下表7.7中列出了这些类型。

表7.7 -- IPv6多播地址范围的类型

范围类型	二进制表示法	十六进制值
本地接口, Interface-Local	0001	1
本地链路, Link-Local	0010	2
本地子网, Subnet-Local	0011	3
本地管理域范围, Admin-Local	0100	4
本地站点范围, Site-Local	0101	5
组织范围, Organization	1000	8
全球范围, Global	1110	E

在这些IPv6多播前缀中，又**保留了一些地址**。这些保留地址称作**多播指定地址**（Multicast Assigned addresses），如下表7.8中所示。

表7.8 -- 保留的IPv6多播地址

地址	范围	说明
<code>FF01::1</code>	主机	所有在本地接口范围内的主机
<code>FF01::2</code>	主机	所有在本地接口范围内的路由器
<code>FF02::1</code>	本地链路	所有在本地链路范围内的主机
<code>FF02::2</code>	本地链路	所有在本地链路范围内的路由器
<code>FF05::2</code>	站点	所有在本地站点范围内的路由器

除了这些地址外，对路由器接口和网络主机上配置的每个单播和任意播地址，都自动启用了一个节点询问多播地址（a Solicited-Node Multicast address）。此地址有着一个本地链路范围，就是说该地址绝不会超出本地网段之外（this address has a Link-Local scope, which means that it will never traverse farther than the local network segment）。**节点询问多播地址用于以下两个目的：取代IPv4的ARP和DAD。**

由于IPv6不会用到ARP，那么节点询问多播地址就被网络主机和路由器用于获悉邻居设备的数据链路地址（the Data Link address）。这样就可以实现IPv6数据包向帧的转换，并将帧发往IPv6主机和路由器了。DAD是IPv6邻居发现协议（Neighbor Discovery Protocol, NDP）的一部分，在本模块的稍后会详细说明这个协议。DAD就是在设备在采用自动配置方法时，将某个IPv6地址配置为其自己的地址之前，检查该地址是否在本地网段上已被使用的方法。本质上，DAD提供与IPv4中用到的无故ARP（Gratuitous ARP）相似的功能。这些**节点询问多播地址**，是由IPv6前缀 FF02::1:FF00:0000/104 定义出来的。它们的构成为前缀 FF02::1:FF00:0000/104，与单播或任意播地址低位序的 24 位结合而成。图7.9演示了这些节点询问多播地址的格式。

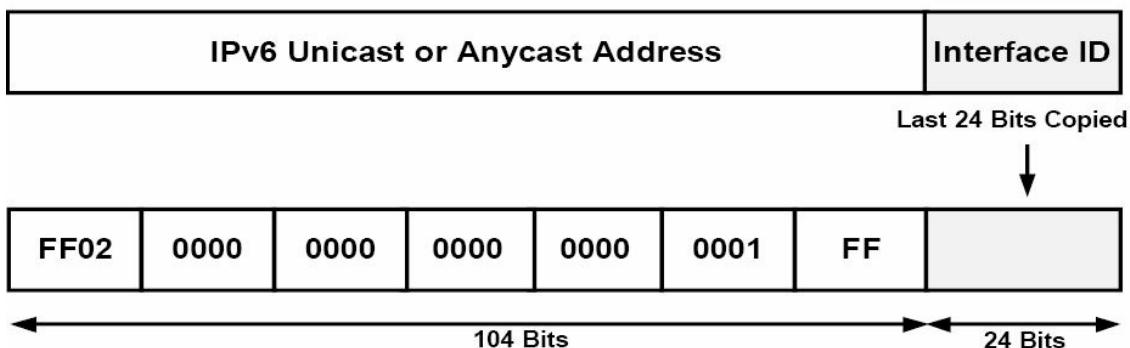


图7.9 -- IPv6 节点询问多播地址

而作为与IPv4到二层以太网的多播映射的一个类似方案，**IPv6提供了一种独特的方法，来将三层IPv6多播地址，映射到二层多播地址**。IPv6中的多播映射是通过在某多播地址的后 32 位加上一个 16 位前缀 33:33，这个前缀就是IPv6网络中定义的多播以太网前缀（the defined Multicast Ethernet prefix for IPv6 Networks）。其在下图7.10中，演示了所有位于本地接口范围前缀 FF02::2 上的路由器的以太网映射多播地址。

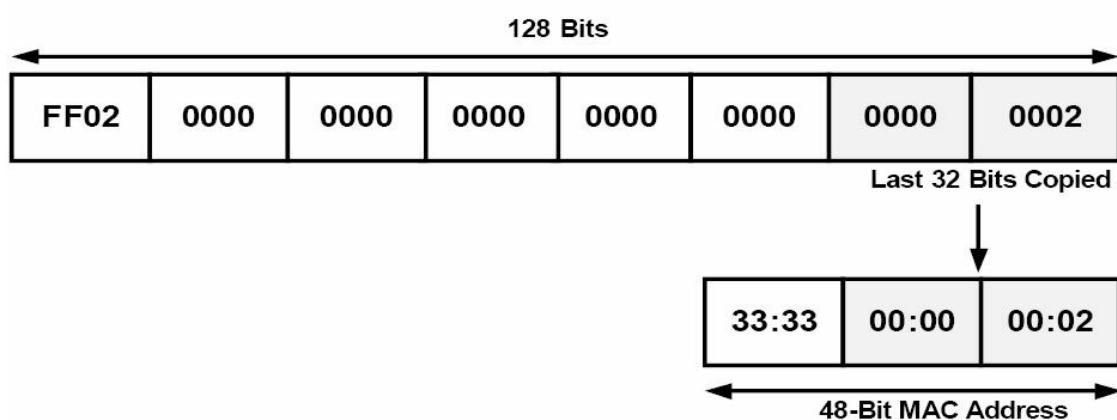


图7.10 -- IPv6多播地址

## 任意播地址

### Anycast Addresses

本章节的早前引入了任意播，其可被简单地说成是一对最近的通信（one-to-nearest communication），这是因为基于路由协议度量值的那个最近的公共地址（the nearest common address），总是会为本地设备所优先选用。在IPv6中，并无为任意播特别分配的地址范围，因为任意播地址使用的是全球单播地址、站点本地地址，甚或本地链路地址。尽管如此，仍然保留一个作为特殊用途的任意播地址。该特别地址被称为子网路由器任意播地址(the Subnet-Router Anycast address)，是由前面的孩子网 64 位单播前缀，及将后 64 位全部设置为 0（比如 2001:1a2b:1111:d7e5::）构成的。任意播地址是绝对不能作为某个IPv6数据包的源地址的。它们典型地用于诸如移动IPv6(Mobile IPv6) 等的协议中，任意播地址的用途，超出CCNA 考试范围。

## 环回地址

### Loopback Address

IPv6中的环回地址，用法和IPv4中的一样。与IPv4中用到的环回地址 127.0.0.1 相比，每台设备也都有一个IPv6环回地址，且该地址有设备自身使用。IPv6环回地址用的是前缀 ::1，用首选地址格式表示为 0000:0000:0000:0000:0000:0000:0001。也就是说，在环回地址中，除了最后一位总是 1 外，其它所有位都设置为0。当设备开启IPv6时，总是会自动分配上这些地址，且这些地址绝不会发生变化。

## 未指明地址

### Unspecified Addresses

在IPv6分址里，未指明地址就是那些没有指派到任何接口上的单播地址。这些地址表明设备缺少一个IPv6 地址，同时这些地址还用于某些诸如IPv6 DHCP和DAD等的用途。未指明地址是以IPv6地址中的全0值表示的，可以使用前缀 :: 进行书写。在首选格式下，这些地址表示为 0000:0000:0000:0000:0000:0000:0000。

## 一些IPv6的协议和机制

### IPv6 Protocols and Mechanisms

尽管互联网协议版本 6 与版本 4 是相似的，但在具体运作上，前者与后者相比仍然有着显著的不同。本节对以下的一些IPv6协议和机制进行了说明。

- IPv6的ICMP
- IPv6邻居发现协议（the IPv6 Neighbor Discovery Protocol, NDP）
- IPv6的有状态自动配置机制（IPv6 stateful autoconfiguration）
- IPv6的无状态自动配置机制（IPv6 stateless autoconfiguration）

## IPv6下的ICMP

### ICMP for IPv6

ICMP用于将有关发往预期目的主机的IP数据的错误和其他信息，汇报给源主机。在 RFC 2463 中，作为 58 号协议定义的ICMPv6，支持ICMPv4的各种报文，还包含了ICMPv6的一些额外报文。ICMPv6作为一个如同TCP一样的，属较高级别的协议，意味着在IPv6数据包中，ICMPv6是放在所有尽可能的扩展头部之后的。下图7.11演示了ICMPv6数据包中所包含的字段。

```

Internet Protocol Version 6
Internet Control Message Protocol v6
Type: 128 (Echo request)
Code: 0
Checksum: 0dbe0 [correct]
ID: 0x1afa
Sequence: 0x0000
Data (52 bytes)
Data: 000102030405060708090A0B0C0D0E0F1011121314151617...

```

图7.11 -- ICMPv6数据包头部

在ICMPv6数据包头部，其 8 位**类型字段**（the 8-bit Type field）**用于表明或区分ICMPv6报文类型**。该字段用于提供错误报文和信息性报文。表7.9列出并说明了一些可在此字段发现的常见值。

表7.9 -- ICMPv6报文类型

ICMPv6 类型	说明
1	目的主机不可达
2	数据包太大
3	发生了超时
128	Echo请求
129	Echo回应

注意：ICMPv4也是使用的这些报文类型。

紧接着类型字段的 8 位**代码字段**（the 8-bit Code field），提供了有关发出的报文细节信息。表7.10演示了该字段的常用值，也是ICMPv4所共用的。

表7.10 -- ICMPv6代码

ICMPv6代码	说明
0	Echo回应
3	目的主机不可达
8	Echo
11	发生了超时

在代码字段后面的 16 位**校验和字段**（the 16-bit Checksum field），包含一个用于检测ICMPv6中数据错误的运算值。ICMPv6数据包的最后，就是报文或数据二选一的字段（the Message or Data field is an optional），它是一个可变长度字段，包含了由类型及代码字段指明的报文类型特定数据。在用到报文或数据字段时，该字段提供了发送给目的主机的信息。

**ICMPv6是IPv6的一个核心部件**。在IPv6中，ICMPv6有以下用途。

- 重复地址检测，Duplicate Address Detection, DAD
- ARP的替代，the replacement of ARP
- IPv6无状态自动配置，IPv6 stateless autoconfiguration
- IPv6前缀重新编号，IPv6 prefix renumbering
- 路径MTU发现，Path MTU Discovery, PMTUD

**注意：**在上述用途中，DAD和无状态自动配置会在本章的稍后进行说明。PMTUD是超出当前CCNA考试要求范围的，在本模块及本教程中不会对其进行任何细节上的说明。

## IPv6邻居发现协议

### The IPv6 Neighbor Discovery Protocol, NDP

**IPv6邻居发现协议带来IPv6的即插即用特性。**它是在 RFC 2461 中定义的，是IPv6的一个必不可少的组成部分。**NDP运行在链路层，负责发现链路上的其它节点、确定其它节点的链路层地址、发现可用的路由器，以及维护有关到其它邻居节点路径的可达性信息。**NDP实现了IPv6的类似于IPv4的ARP（这正是其取代的功能）、**ICMP路由器发现(ICMP Router Discovery)**以及**路由器重定向协议（Router Redirect Protocols）**等功能。尽管如此，要记住NDP提供了比起IPv4中用到的诸多机制，都更为了不起的功能。在与ICMPv6配合使用时，NDP可以完成以下任务。

- 动态邻居和路由器发现，dynamic neighbor and router discovery
- 取代ARP，the replacement of ARP
- IPv6无状态自动配置，IPv6 stateless autoconfiguration
- 路由器重定向，router redirection
- 主机参数发现，host parameter discovery
- IPv6地址解析，IPv6 address resolution
- 确定下一跳路由器，next-hop router determination
- 邻居不可达检测，Neighbor Unreachability Detection, NUD
- 重复地址检测，Duplicate Address Detection, DAD

**注意：**并不要求对上面列出的每个优势进行细节上的探究。

邻居发现协议又定义了五种ICMPv6数据包类型，在下表7.11中有列出和说明。

表7.11 -- ICMPv6邻居发现报文类型

ICMPv6类型	说明
133	用于路由器询问报文，used for Router Solicitation(RS) messages
134	用于路由器通告报文，used for Router Advertisement(RA) messages
135	用于邻居询问报文，used for Neighbor Solicitation(NS) messages
136	用于邻居通告报文，used for Neighbor Advertisement(NA) messages
137	用于路由器重定向报文, used for Router Redirect messages

**路由器询问报文**（Router Solicitation messages）由主机在其接口开启IPv6时所发出。这些报文用于请求本地网段上的路由器立即生成RA报文，而不要等到下一个计划的RA时间间隔才生成RA报文。下图7.2演示了一条在线路上捕获到的RS报文。

```

Internet Protocol Version 6
Internet Control Message Protocol v6
Type: 133 (Router solicitation)
Code: 0
Checksum: 0x6e61 [correct]
ICMPv6 Option (Source link-layer address)
Type: Source link-layer address (1)
Length: 8
Link-layer address: 00:24:e8:f5:7e:a2

```

图7.12 -- IPv6路由器询问报文

路由器收到该RS报文后，便使用RA报文通告其存在，RA报文通常包含了本地链路的前缀信息，及所有诸如建议跳数限制等额外配置。RA中包含的信息在下图7.13中进行了演示。

```

- Internet Protocol Version 6
- Internet Control Message Protocol v6
  Type: 134 (Router advertisement)
  Code: 0
  Checksum: 0x17ed [correct]
  Cur hop limit: 64
- Flags: 0x00
  Router lifetime: 1800
  Reachable time: 0
  Retrans timer: 0
- ICMPv6 Option (Source link-layer address)
- ICMPv6 Option (MTU)
- ICMPv6 Option (Prefix information)
- ICMPv6 Option (Prefix information)

```

图7.13 -- IPv6路由器通告报文

这里重申一点，RS和RA报文，都是路由器到主机(route-to-host)或主机到路由器(host-to-router)的信息交换，如下图所示。

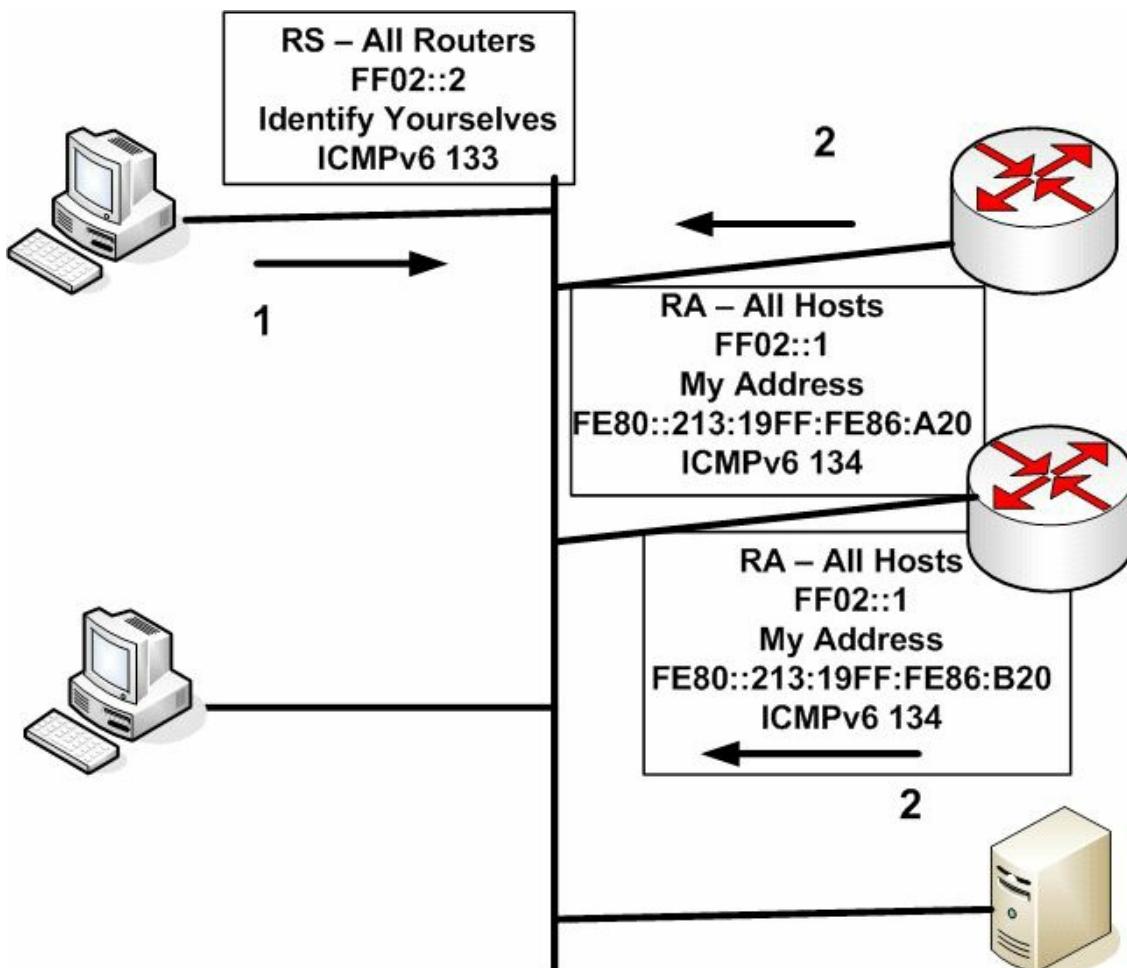


图7.14 -- IPv6的RS和RA报文

IPv6的**邻居询问报文**(Neighbor Solicitation messages)，则是本地网段上的IPv6路由器所发出的多播报文，用于确定某个邻居的数据链路地址，或是用于检查某个邻居是否仍然可达（因而NS报文取代的是ARP的功能）。这些报文也用于重复地址检测(DAD)目的。尽管对NS报文的深入探究超出了CCNA考试要求的范围，下面的图7.15仍然演示了一个在线路上捕获到的IPv6邻居询问报文数据包。

```
Internet Control Message Protocol v6
Type: 135 (Neighbor solicitation)
Code: 0
Checksum: 0x3f71 [correct]
Target: fe80::213:19ff:fe86:a20
ICMPv6 Option (Source link-layer address)
Type: Source link-layer address (1)
Length: 8
Link-layer address: 00:24:e8:f5:7e:a2
```

图7.15 -- IPv6邻居询问报文

而**邻居通告报文** (Neighbor Advertisement messages) 通常也是由本地网段上的路由器发出，用于对收到的NS报文进行回应。此外，在一个**IPv6前缀改变时，路由器也会发出一条无询问的NS报文**，以此来告知本地网络网段上的其它设备，发生了这个变化。在NA报文上，对NA报文中的格式或包含的字段的细节探究，也是超出CCNA考试要求范围之外的。图7.16和图7.17演示了一条在线路上捕获的邻居通告报文，**邻居通告报文也是通过IPv6多播发出的**。

```
Internet Control Message Protocol v6
Type: 136 (Neighbor advertisement)
Code: 0
Checksum: 0x909f [correct]
Flags: 0xa0000000
  1... .... .... .... .... .... = Router
  .0.. .... .... .... .... .... = Not advertised
  ..1. .... .... .... .... .... = Override
Target: fe80::20c:ceff:fea7:f3a0
ICMPv6 Option (Target link-layer address)
Type: Target Link-Layer address (2)
Length: 8
Link-layer address: 00:0c:ce:a7:f3:a0
```

图7.16 -- IPv6邻居通告报文

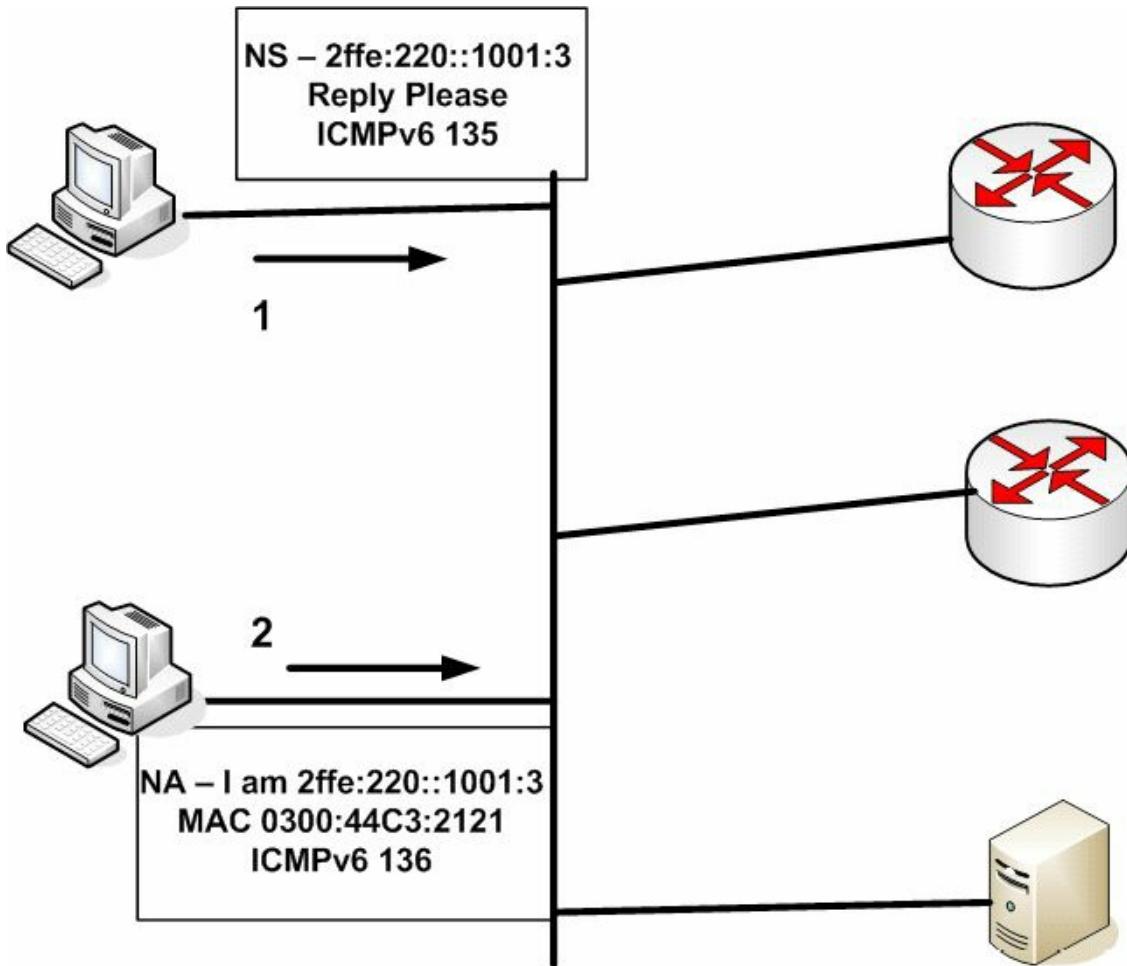


图7.17 -- IPv6邻居通告报文

最后，路由器重定向（router redirect）使用报文类型为137的ICMPv6重定向报文（ICMPv6 Redirect messages），路由器重定向用于告知网络主机，网络上存在一台路由器，该路由器有着前往预计目的主机的更优路径。ICMPv6的路由器重定向与ICMPv4的工作方式一样，而ICMPv4的路由器重定向就是用来对当前IPv4网络中的流量进行重定向的。

## IPv6的有状态自动配置

### IPv6 Stateful Autoconfiguration

如同本模块先前指出的那样，有状态自动配置允许网络主机从某台网络服务器（比如通过DHCP）上收到其地址信息。IPv4和IPv6都支持这种方式。在IPv6网络中，使用DHCPv6来为IPv6主机提供有状态（及无状态）自动配置服务。在IPv6的部署中，当某台IPv6主机收到来自本地网段上的路由器RA报文后，该主机就会检查这些数据包，以判定是否可以使用DHCPv6。RA报文通过将那些 M（受管理的， Managed）或 O（其它方式， other）位设置为 1 的方式，提供是否可以使用DHCPv6的信息。

在DHCP下，客户端设定为从DHCP服务器取得有关信息。而在DHCPv6下，客户端却并不知道从哪里得到这些信息，因为既可以从SLAAC，也可以从有状态的DHCPv6，抑或从结合了SLAAC及DHCPv6两种的方式取得。

RA报文中的M位，指的是受管理的地址配置标志位（the Managed Address Configuration Flag bit）。在此位设置了时（也就是说该位的值为 1 时），它指示IPv6主机要取得一个由DHCPv6服务器所提供有状态的地址，并忽略之后的 O 位。而路由器通告报文中的 O 位，指的是其它有状态配置标志位（the Other

Stateful Configuration Flag bit）。当该位设置了（也就是说该位的值为 1）后，指示IPv6主机要使用 DHCPv6，来取得更多的设置项，比如DNS及WINS服务器等。

如某台主机未曾配置一个IPv6地址，它就可以采用下面的三种方法之一，来获得一个IPv6地址，及诸如 DNS服务器地址等的其他网络设置。

- SLAAC -- 无状态自动配置（StateLess Address AutoConfiguration），`m` 和 `o` 位设置为 0。也就是没有DHCPv6信息。主机从一条RA收到所有必要信息。
- 有状态DHCPv6 -- `m` 标志位设置为 1，告诉主机使用DHCPv6取得所有地址和网络信息。
- 无状态DHCPv6 -- `m` 标志位设置为 0，`o` 标志位设置为 1，意味着主机将采用SLAAC来得到地址（从一条RA），而同时从DNS服务器取得其它信息。

尽管SLAAC能力是IPv6的一项优势，有状态自动配置仍然有着许多好处，包括以下这些。

- 相较SLAAC所提供的那些项目，有状态自动配置有着更大的控制权
- 在SLAAC网络上，同样可以使用有状态自动配置
- 在缺少路由器的情形下，仍然可以为网络主机提供分址
- 通过分配新的前缀给主机，而用来对网络重新编号
- 可用于将全部子网发布给用户侧设备（can be used to issue entire subnets to customer premise equipment，稍后会有说明）

## IPv6无状态自动配置

### **IPv6 Stateless Address Autoconfiguration, SLAAC**

IPv6容许设备为自己配置一个IP地址，以便进行主机到主机的通信。有状态自动配置需要一台服务器来分配地址信息，对于IPv6来说，就要用到DHCPv6。有状态就是说，信息交换的细节在服务器（或路由器）上是有保存的，那么无状态就说的是没有服务器来保存这些细节了。DHCPv6既可以是有状态的，也可以是无状态的。

在IPv6中，SLAAC允许主机依据本地网络网段上的路由器发出的前缀通告，自己配置其单播IPv6地址。所需的其它信息（比如DNS服务器地址等）则可从DHCPv6服务器获取。IPv6中SLAAC用到的三种机制，如下所示。

- 前缀通告，prefix advertisement
- 重复地址检测，DAD
- 前缀重编号，prefix renumbering

### 前缀通告

#### **prefix advertisement**

IPv6地址前缀通告用到了ICMPv6 RA报文，而ICMPv6 RA是发往链路上的所有主机（all-hosts-on-the-local-link）的，带有多播地址 `FF02::1` 的ICMPv6数据包。根据IPv6的设计，仅有路由器才被允许在本地链路上通告前缀。在采行SLAAC后，就务必要记住，所用到的前缀长度，必须是64位（比如 `2001:1a2b::/64`）。

在前缀配置之后，SLAAC用到的RA报文还包含了以下信息。

- IPv6前缀，the IPv6 prefix
- 生命期，the lifetime
- 默认路由器信息，default router information
- 标志和/或选项字段，Flags and/or Options fields

就像刚才指出的那样，**IPv6前缀必须是64位**。此外，**本地网段上还可以通告多个的IPv6前缀**。在该网络网段上的主机收到IPv6前缀后，就将它们的MAC地址以 EUI-64 格式，追加到前缀后面，从而自动地配置上他们的IPv6单播地址，这在本模块的先前部分已有说明。这样就为该网段上的每台主机，都提供了一个唯一的 128 位IPv6地址。

SLAAC RA报文也提供了每个通告前缀的生命期数值给这些节点，生命期字段可以是从 0 到无穷的值。节点在收到前缀后，就对该前缀的生命期值进行验证，从而在生命期数值到 0 时停用该前缀。此外，如收到生命期值为无穷的某个特定前缀，网络主机就绝不会停用那个前缀。每个通告前缀又带有两个生命期值：**有效生命期值及首选生命期值**（the valid and preferred lifetime value）。

有效生命期值用于确定出该主机地址将保持多长时间的有效期。在该值超时后（也就是说到值为 0 时），带有该前缀的主机地址就成为无效地址。而首选生命期值则用于确定经由SLAAC方式配置的某个地址将保持多长时间的有效期。此值必须小于或等于在有效生命期值，同时该值通常用于前缀的重编号。

SLAAC RA的默认路由器，提供了其本身IPv6地址的存在情况和生命期。默认情况下，用于默认路由器的那个地址是本地链路地址（`FE80::/10`）。这样做就可以在全球单播地址发生改变时，也不会像在IPv4中那样，在某个网络被重新编号时，导致网络服务中断。

最后，一些标志和选项字段可被用作指示网络主机采行SLAAC或有状态自动配置。这些字段在图7.13中的RA线路捕获中有包含。

## 重复地址检测

### Duplicated Address Detection, DAD

重复地址检测（DAD）是一种用在SLAAC中，在某网段上主机启动时，用到的NDP机制。DAD要求某台网络主机启动期间，在永久地配置它自己的IPv6地址之前，先要确保没有别的网络主机已经使用了它打算使用的那个地址。

DAD通过使用邻居询问（`135` 类型的ICMPv6）及节点询问多播地址（Solicited-Node Multicast addresses），来完成这个验证。主机使用一个未指明IPv6地址（an unspecified IPv6 address, 也就是地址 `::`）作为报文数据包的源地址，并将其打算使用的那个IPv6单播地址，作为目的地址，在本地网段上发送一个邻居询问ICMPv6报文数据包。如有其它主机使用着该地址，那么主机就不会自动将此地址配置为自己的地址；而如没有其他设备使用这个地址，则该主机就自动配置并开始使用这个IPv6地址了。

## 前缀重编号

### prefix renumbering

最后，前缀重编号（prefix renumbering）机制允许IPv6网络从一个前缀变为另一个时，进行前缀透明重编号。与IPv4中同样的全球IP地址可由多个服务提供商进行通告不同，IPv6地址空间的严格聚合阻止了服务提供商对不属于其组织的前缀进行通告（Unlike in IPv4, where the same global IP address can be advertised by multiple providers, the strict aggregation of the IPv6 address space prevents providers from advertising prefixes that do not belong to their organization）。

在网络发生从一家IPv6服务提供商迁移至另一家时，IPv6前缀重编号机制，就提供了一种自一个前缀往另一前缀平滑和透明的过渡。前缀重编号使用与在前缀通告中同样的ICMPv6报文和多播地址。而前缀重编号可经由运用RA报文中包含的时间参数完成。

在思科IOS软件中，路由器可配置为通告带有被减少到接近0的有效和首选生命期当前前缀，这就令到这些前缀能够更快地成为无效前缀。此时再将这些路由器配置为在本地网段上通告新前缀。这样做将允许旧前缀和新前缀在同一网段上并存。

迁移期间，本地网段上的主机用着两个单播地址：一个来自旧的前缀，一个来自新的前缀。那些使用旧前缀的当前连接仍被处理着；但所有自主机发出的新连接，则都使用新前缀。在旧前缀超时后，就只使用新前缀了。

## 配置无状态DHCPv6

### Configuring Stateless DHCPv6

为在某台路由器上配置无状态的DHCPv6，需要完成一些简单的步骤。

- 创建地址池名称和其它参数，create the pool name and other parameters
- 在某个借口上开启它，enable it on an interface
- 修改RA设置，modify Router Advertisement settings

一个身份关联是分配给客户端的一些地址（an Identity Association is a collection of addresses assigned to the client）。使用到DHCPv6的每个借口都必须要有至少一个的身份关联（IA）。这里不会有CCNA考试的配置示例。

## 在思科IOS软件中开启IPv6路由

现在，你对IPv6基础知识有了扎实掌握，本模块剩下的部分将会专注于思科IOS软件中IPv6的配置了。默认下，思科IOS软件中的IPv6路由功能是关闭的。那么就必须通过使用 **ipv6 unicast-routing** 这个全局配置命令来开启IPv6路由功能。

在全局开启IPv6路由之后，接口配置命令 **ipv6 address [ipv6-address/prefix-length | prefix-name sub-bits/prefix-length | anycast | autoconfig <default> | dhcp | eui-64 | link-local]** 就可以用于配置接口的IPv6分址了。关键字 **[ipv6-address/prefix-length]** 用于指定分配给该接口的IPv6前缀和前缀长度。下面的配置演示了如何为一个路由器接口配置子网 **3FFF:1234:ABCD:5678::/64** 上的第一个地址。

```
R1(config)#ipv6 unicast-routing
R1(config)#interface FastEthernet0/0
R1(config-if)#ipv6 address 3FFF:1234:ABCD:5678::/64
R1(config-if)#exit
```

按照此配置，**show ipv6 interface [name]** 命令就可用于验证配置的IPv6地址子网（即 **3FFF:1234:ABCD:5678::/64**），如下面的输出所示。

```
R1#show ipv6 interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
    IPv6 is enabled, link-local address is FE80::20C:CEFF:FEA7:F3A0
    Global unicast address(es):
        3FFF:1234:ABCD:5678::1, subnet is 3FFF:1234:ABCD:5678::/64
    Joined group address(es):
        FF02::1
        FF02::2
        FF02::1:FF00:1
        FF02::1:FFA7:F3A0
    ...
[Truncated Output]
```

就如在本模块早先指出的那样，IPv6允许在同一接口上配置多个前缀。而如过在同一借口上配置了多个前缀，**show ipv6 interface [name] prefix** 命令，就可以用来查看所有分配的前缀，以及它们各自的有效和首选生命期数值。下面的输出显示了在一个配置了多个IPv6前缀的路由器接口上，该命令所打印出的信息。

```
R1#show ipv6 interface FastEthernet0/0 prefix
IPv6 Prefix Advertisements FastEthernet0/0
Codes: A - Address, P - Prefix-Advertisement, O - Pool
       U - Per-user prefix, D - Default
       N - Not advertised, C - Calendar
default [LA] Valid lifetime 2592000, preferred lifetime 604800
AD 3FFF:1234:ABCD:3456::/64 [LA] Valid lifetime 2592000, preferred lifetime 604800
AD 3FFF:1234:ABCD:5678::/64 [LA] Valid lifetime 2592000, preferred lifetime 604800
AD 3FFF:1234:ABCD:7890::/64 [LA] Valid lifetime 2592000, preferred lifetime 604800
AD 3FFF:1234:ABCD:9012::/64 [LA] Valid lifetime 2592000, preferred lifetime 604800
```

**注意：** 和早前指出的一样，有效和首选生命期数值可自默认值进行修改，以实现在应用前缀重编号时的平滑过渡。但此配置是超出CCNA范围的，所以本教程不会对其进行演示。

跟着接口配置命令 `ipv6 prefix` 的使用之后，关键字 `[prefix-name sub-bits/prefix-length]` 用于配置一个通用前缀（a general prefix），通用前缀指定要配置到该接口上的子网的那些前导位。这个配置也是超出当前 CCNA 考试要求的，本模块不会对其进行演示。

关键字 `[anycast]` 用于配置一个IPv6任意播地址。和先前指出的那样，任意播分址允许将同一个**公共地址**（the same common address）分配到多个路由器接口。主机使用从路由协议度量值上看离它们最近的任意播地址。任意播配置超出CCNA考试要求范围，不会在本模块进行演示。

`[autoconfig <default>]` 关键字开启SLAAC。如用到该关键字，路由器将动态学习链路上的前缀，之后将 EUI-64 地址加到所有学习到的前缀上。`[default]` 关键字是一个允许安装一条默认路由的可选关键字（the `<default>` keyword is an optional keyword that allows a default route to be installed）。下面的配置样例，演示了如何在某个路由器接口上开启无状态自动配置，同时额外地允许安装上默认路由。

```
R2(config)#ipv6 unicast-routing
R2(config)#interface FastEthernet0/0
R2(config-if)#ipv6 address autoconfig default
R2(config-if)#exit
```

按照这个配置，路由器 R2 将会监听 `FastEthernet0/0` 接口所在本地网段上的RA报文。该路由器将会对每个学习到的前缀，动态地配置一个 EUI-64 地址，并接着安装上指向该RA通告路由器本地链路地址的默认路由。使用 `show ipv6 interface [name]` 命令，即可对动态地址配置进行验证，如下面的输出所示。

```
R2#show ipv6 interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE86:A20
  Global unicast address(es):
    3FFF:1234:ABCD:3456:213:19FF:FE86:A20, subnet is 3FFF:1234:ABCD:3456::/64 [PRE]
      valid lifetime 2591967 preferred lifetime 604767
    3FFF:1234:ABCD:5678:213:19FF:FE86:A20, subnet is 3FFF:1234:ABCD:5678::/64 [PRE]
      valid lifetime 2591967 preferred lifetime 604767
    3FFF:1234:ABCD:7890:213:19FF:FE86:A20, subnet is 3FFF:1234:ABCD:7890::/64 [PRE]
      valid lifetime 2591967 preferred lifetime 604767
    3FFF:1234:ABCD:9012:213:19FF:FE86:A20, subnet is 3FFF:1234:ABCD:9012::/64 [PRE]
      valid lifetime 2591967 preferred lifetime 604767
    FEC0:1111:1111:E000:213:19FF:FE86:A20, subnet is FEC0:1111:1111:E000::/64 [PRE]
      valid lifetime 2591967 preferred lifetime 604767
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF86:A20
  MTU is 1500 bytes
...
[Truncated Output]
```

在上面的输出中，注意到尽管接口上没有配置显式的IPv6地址，还是动态地为经由侦听RA报文所发现的子网，配置了一个 EUI-64 地址。每个这些前缀的计时器，都继承自通告RA报文的那台路由器。为了进一步验证无状态自动配置，可以使用 `show ipv6 route` 命令，来验证到首选通告路由器本地链路地址的默认路由，如下面所演示的那样。

```
R2#show ipv6 route ::/0
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
       U - Per-user Static route
       I1 - ISIS L1, I2 - ISIS L2, IA - ISIS inter area, IS - ISIS summary
       O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
       ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S    ::/0 [1/0]
      via FE80::20C:CEFF:FEA7:F3A0, FastEthernet0/0
```

在命令 `ipv6 address` 之后，关键字 `[dhcp]` 用于配置该路由器接口使用有状态自动配置（也就是DHCPv6），来请求该接口的分址配置。在此配置下，有着一个额外的关键字，`[rapid-commit]`，同样可以追加到此命令之后，以开启地址分配及其它配置信息的二报文交换快速方式（the two-message exchange method）。

再回到讨论主题，在 `ipv6 address` 命令下，关键字 `[eui-64]` 用于为某个接口配置一个IPv6地址，并在地址的低 64 位使用一个 EUI-64 地址而在该接口上开启IPv6处理。默认情况下，本地链路、站点本地以及IPv6 SLAAC都用到 EUI-64 格式来构造其各自的IPv6地址。EUI-64 分址将 48 位MAC地址扩展到一个 64 位地址。通过两步实现该扩展，这两步将在下一段进行说明。该过程就叫作SLAAC。

构造 EUI-64 地址的第一步，将值 `FFFF` 插入到MAC地址中间，就将 12 个十六进制字符的 48 位MAC地址扩展到 16 个十六进制字符的 64 位了。下图7.18演示了 48 位MAC地址到 64 位EUI地址的转换。

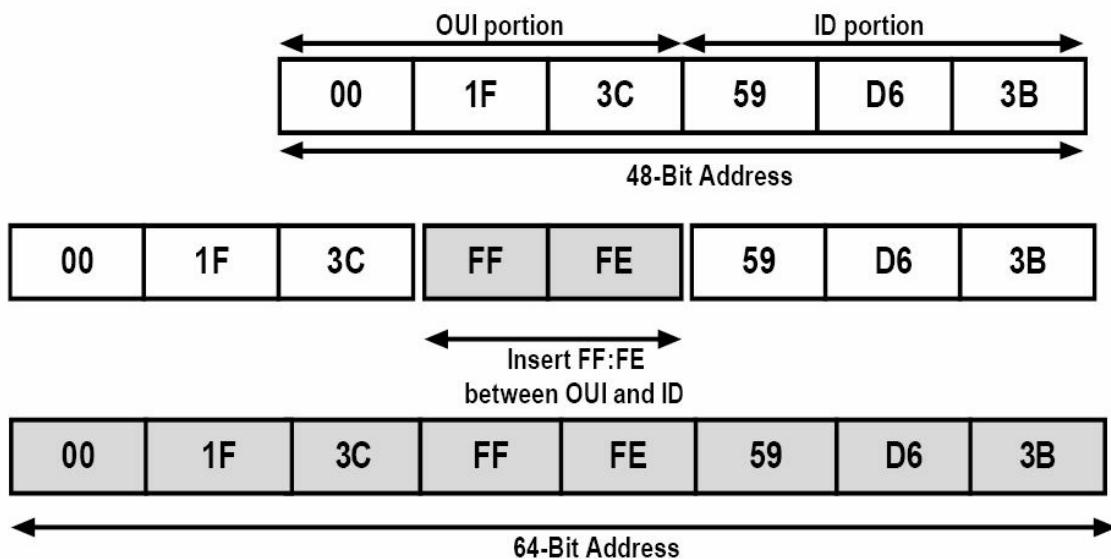


图7.18 -- 创建 EUI-64 地址

EUI-64 分址的下一步，涉及 64 位的第 7 位设置。此第 7 位用于区分该MAC地址是否是唯一的。如该位设置为 1，就表明该MAC地址是一个全球受管理MAC地址（a globally managed MAC address）-- 也就是说该MAC地址是有某厂商分配的。如该位设置为 0，就表明该MAC地址是本地分配的--就意味着该MAC地址有可能是由管理员添加的。为进一步搞清楚此声明，MAC地址实例 `02:1F:3C:59:D6:3B` 就被认为是一个全球分配的MAC地址（a globally-assigned MAC address），而MAC地址 `00:1F:3C:59:D6:3B` 则被看作是一个本地地址。下图7.19有演示。

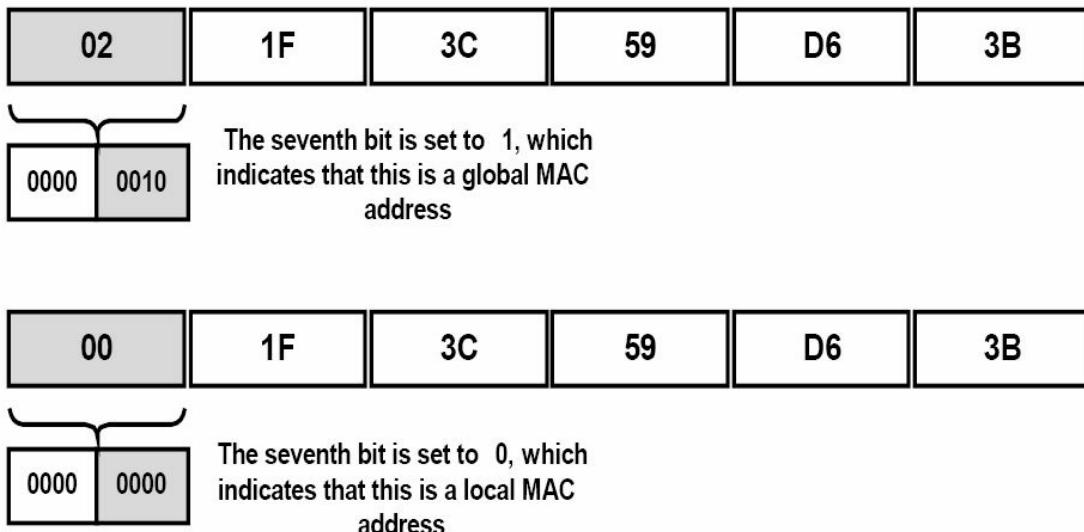


图7.19 -- 确定本地及全球MAC地址

按照这样的配置，命令 `show ipv6 interface` 就可用于验证验证分配到接口 `FastEthernet0/0` 上的IPv6接口ID，如下面的输出所示。

```
R2#show ipv6 interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80::213:19FF:FE86:A20
  Global unicast address(es):
    3FFF:1A2B:3C4D:5E6F:213:19FF:FE86:A20, subnet is 3FFF:1A2B:3C4D:5E6F::/64 [EUI]
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF86:A20
  MTU is 1500 bytes
...
[Truncated Output]
```

要验证该 EUI-64 地址的构造过程，同样可以通过使用 `show interface` 命令，查看指定接口的MAC地址的方式，来检查该完整的IPv6地址。

```
R2#show interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0013.1986.0a20 (bia 0013.1986.0a20)
  Internet address is 10.0.1.1/30
```

从上面的输出可以看出，该 EUI-64 地址实际上是有效的，且是基于该接口的MAC地址。此外，该地址是全球地址，因为那个第七位是开启的（也就是改为包含的是一个非零值）。

最后的 `[link-local]` 关键字用于分配给接口一个本地链路地址。一定要记住在默认情况下，对于动态地创建出一个本地链路地址来说，接口上并不是非得要启用一个IPv6前缀。而是当在某个接口下执行了接口配置命令 `ipv6 enable` 时，就会以 EUI-64 分址方式，自动创建出那个接口的一个本地链路地址。

而如果要手动配置一个本地链路地址，就必须分配一个本地链路地址块 `FE80::/10` 中的地址。下面的配置实例，演示了如何在某接口上配置一个本地链路地址。

```
R3(config)#interface FastEthernet0/0
R3(config-if)#ipv6 address fe80:1234:abcd:1::3 link-local
R3(config-if)#exit
```

按照该配置，就可用 `show ipv6 interface [name]` 命令验证这个手动配置的本地链路地址，如下面的输出所示。

```
R3#show ipv6 interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  IPv6 is enabled, link-local address is FE80:1234:ABCD:1::3
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF00:1111
  MTU is 1500 bytes
...
[Truncated Output]
```

**注意：**在进行手动配置本地链路地址时，如思科IOS软件侦测到另一主机正在使用一个它的IPv6地址，控制台上就会打印出一条错误消息，同时该命令将被拒绝。所以在手动配置本地链路地址时，要小心仔细。

## IPv6子网划分

### Subnetting with IPv6

如你已经学到的，IPv6地址分配给机构的是一个前缀。而IPv6地址的主机部分总是 64 位的 EUI-64，同时**标准的前缀通常又是 48 位或 /48**。那么剩下的 16 位，就可由网络管理员自主用于子网划分了。

在考虑网络分址时，因为同样的规则对IPv4和IPv6都是适用的，那就是**每个网段只能有一个网络**。不能分离地址而将一部分主机位用在这个网络，另一部分主机位用在其它网络。

如你看着下面图表中的分址，就能更清楚这个情况。

| 全球路由前缀 | 子网ID | 接口ID || 48 位或 /48 | 16 位 ( 65535 个可能的子网) | 64 位 |

绝不用担心会用完每个子网的主机位，因为每个子网有超过 2 的 64 次幂的主机。任何组织要用完这些子网都是不大可能的，而就算发生了这种情况，也可以轻易地从ISP那里要一个前缀。

比如我们说分得了全球路由前缀 (the global routing prefix) 0:123:abc/48。该地址占用了一个完整IPv6地址的三个区段，而每个区段或4位16进制字符 (quartet) 则是 16 位，那么到目前为止就用了 48 位。主机部分则需要 64 位，留下 16 位用于子网的分配。

可以简单的从零（子网零也是合法的）开始以十六进制数下去。对于主机来说，也可以这样做，除非比如说想要将头几个地址留给网段上的服务器。

用一个更简单的前缀来打比方吧 -- 2001:123:abc/48。第一个子网就是全零，当然，每个子网上的第一台主机也可以是全零，这也是合法的（只要不保留IPv6中的全 0s 和全 1s 地址）。又会将全零主机表示为缩写形式的 ::。那么这里就有开头的几个子网及主机地址。

全球前缀	子网	第一个地址
2001:123:abc	0000	::
2001:123:abc	0001	::
2001:123:abc	0002	::
2001:123:abc	0003	::
2001:123:abc	0004	::
2001:123:abc	0005	::
2001:123:abc	0006	::
2001:123:abc	0007	::
2001:123:abc	0008	::
2001:123:abc	0009	::
2001:123:abc	000A	::
2001:123:abc	000B	::
2001:123:abc	000C	::
2001:123:abc	000D	::
2001:123:abc	000E	::
2001:123:abc	000F	::
2001:123:abc	0010	::
2001:123:abc	0011	::
2001:123:abc	0012	::
2001:123:abc	0013	::
2001:123:abc	0014	::
2001:123:abc	0015	::
2001:123:abc	0016	::
2001:123:abc	0017	::

我肯定你已经注意到这与IPv4分址规则有所不同，不同之处就在与**可以使用全零子网，同时子网的第一个地址总是全零**。请看看下面这个简单的网络拓扑，可以照这种方式进行子网分配。

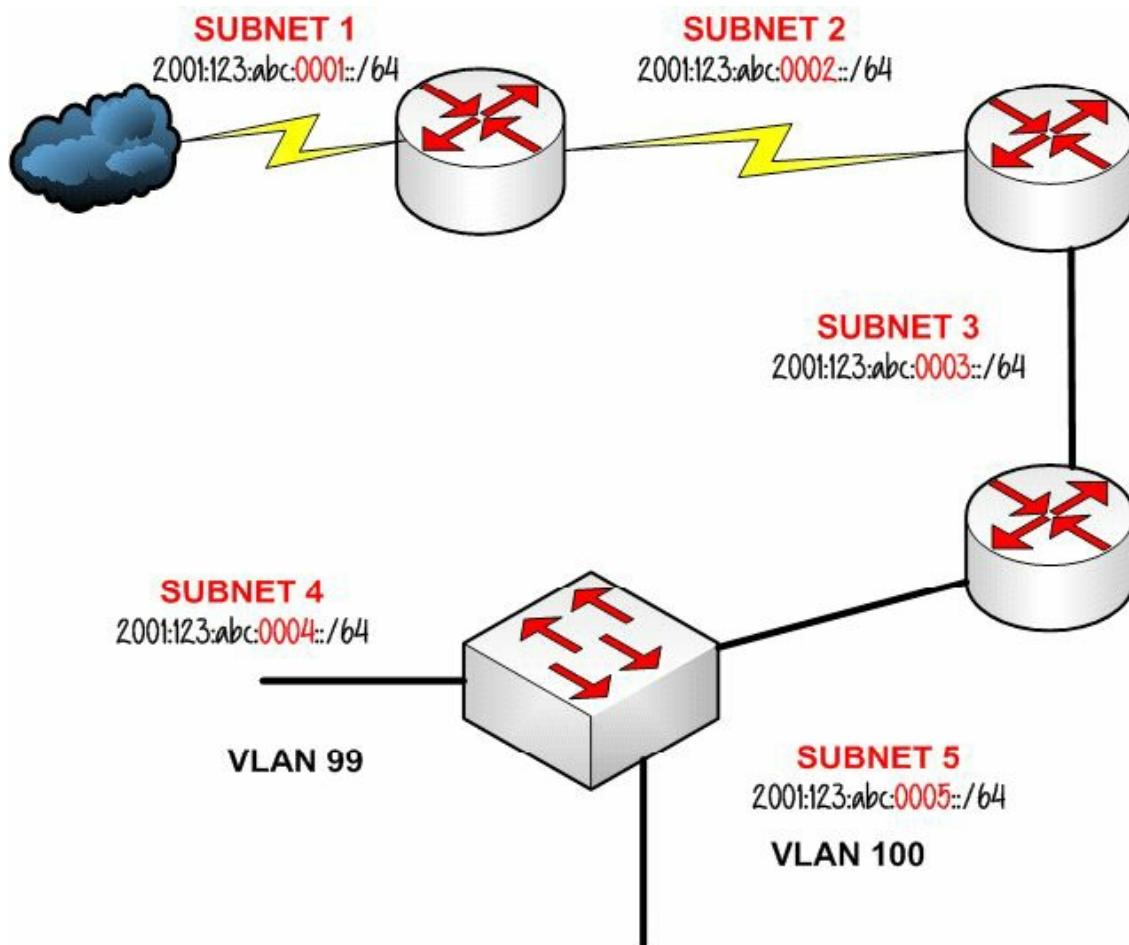


图7.20 -- IPv6子网分配

就是那么容易吗？如回忆一下IPv4子网划分章节，要完成子网划分，以及算出有多少主机多少子网并记住要排除一些地址，简直就是一场噩梦。**IPv6子网划分就容易得多**。你分配到的不一定是一个 48 位前缀，可能是一个用于家庭网络的 /56 或更小的前缀，但原则是一样的。也可以自位界限以外进行子网划分，但这是很少见的，且如果思科要你用考试中的很段时间完成那么深的细节，也是不公平的（*You can also subnet off the bit boundary, but this would be most unusual and unfair of Cisco to expect you to go into that amount of detail in the short amount of time you have in the exam*）。还好的是，考试不是要你考不过，但谁又知道呢（*Hopefully, the exam won't be a mean attempt to catch you out, but you never know*）。为以防万一，这里给出一个有着 /56 前缀长度的地址示例。

`2001:123:abc:8bbc:1221:cc32:8bcc:4231/56`

该前缀是 56 位，转换一下就是 14 个十六进制数位（ $14 \times 4 = 56$ ），那么就知道了该前缀将带到一个 4 位字节（quartet）的中间。**这里有个坑**。在前缀终止前，必须要将该 4 位字节的第 3 和 4 位置为零。

`2001:123:abc:8b00:0000:0000:0000:0000/56`

上面对位界限分离的地方进行了加粗（I've made the quartet bold where the bit boundary is broken）。在匆忙中及考试中时间上的压力下，可能会完全忘记这重要的一步。请记住也要将下面这个地址（第一个子网上的第一台主机）写作这样。

`2001:123:abc:8b00::/56`

如他们硬要在考试中把你赶出去，就可能会试着让你把那两个零从位界限分离处之前的 4 位字节中去掉（If they do try to catch you out in the exam, it would probably be an attempt to have you remove the trailing zeros from the quartet before the bit boundary is broken）。

```
2001:123:abc:8b::/56
```

那么上面这个缩写就是非法的了。

也可以从主机部分借用位来用于子网划分，但绝没有理由这么做，同时这么做也会破坏采行发明IPv6而带来的可资利用的那些众多特性的能力，包括SLAAC (You can steal bits from the host portion to use for subnets, but there should never be a reason to do so and it would break the ability to use many of the features IPv6 was invented to utilise, including stateless autoconfiguration)。

## IPv6和IPv4的比较

### IPv6 Compared to IPv4

一名网络工程师应有一幅IPv6比起IPv4所带来众多优势的图景。看着IPv6的增强，可以总结出下面这些优势。

- IPv6有着一个扩展的地址空间，从 32 位扩展到了 128 位, IPv6 has an expanded address space, from 32 bits to 128bits
- IPv6使用十六进制表示法，而不是IPv4中的点分十进制表示法, IPv6 uses hexadecimal notation instead of dotted-decimal notation(as in IPv4)
- 因为采用了扩充的地址空间，IPv6地址是全球唯一地址，从而消除了NAT的使用需求, IPv6 addresses are globally unique due to the extended address space, eliminating the need for NAT
- IPv6有着一个固定的头部长度（40 字节），允许厂商在交换效率上进行提升, IPv6 has a fixed header length(40 bytes), allowing vendors to improve switching efficiency
- IPv6通过在IPv6头部和传输层之间放入扩展头部，而实现对一些增强选项（这可以提供新特性）的支持, IPv6 supports enhanced options(that offer new features)by placing extension headers between the IPv6 header and the Transport Layer header
- IPv6具备地址自动配置的能力，提供无需DHCP服务器的IP地址动态分配, IPv6 offers address autoconfiguration, providing for dynamic assignment of IP addresses even without a DHCP server
- IPv6具备对流量打标签的支持, IPv6 offers support for labeling traffic flows
- IPv6有着内建的安全功能，包括经由 IPSec 实现的认证和隐私保护功能等, IPv6 has security capabilities built in, including authentication and privacy via IPSec
- IPv6具备在往目的主机发送数据包之前的路径MTU发现功能，从而消除碎片的需求, IPv6 offers MTU path discovery before sending packets to a destination, eliminating the need for fragmentation
- IPv6支持站点多处分布, IPv6 supports site multi-homing
- IPv6使用ND（邻居发现，Neighbor Discovery）协议取代ARP，IPv6 uses the ND protocol instead of ARP
- IPv6使用AAAA DNS记录，取代IPv4中的A记录, IPv6 uses AAAA DNS records instead of A records (as in IPv4)
- IPv6使用站点本地分址，取代IPv4中的 RFC 1918， IPv6 uses Site-Local addressing instead of RFC 1918(as in IPv4)
- IPv4和IPv6使用不同的路由协议, IPv4 and IPv6 use different routing protocols
- IPv6提供了任意播分址, IPv6 provides for Anycast addressing

## 第七天的问题

1. IPv6 addresses must always be used with a subnet mask. True or false?
2. Name the three types of IPv6 addresses.
3. Which command enables IPv6 on your router?

4. The `0002` portion of an IPv6 address can be shortened to just 2. True or false?
5. How large is the IPv6 address space?
6. With IPv6, every host in the world can have a unique address. True or false?
7. IPv6 does not have natively integrated security features. True or false?
8. IPv6 implementations allow hosts to have multiple addresses assigned. True or false?
9. How can the broadcast functionality be simulated in an IPv6 environment?
10. How many times can the double colon ( `::` ) notation appear in an IPv6 address?

## 第七天问题答案

1. False.
2. Unicast, Multicast, and Anycast.
3. The `ipv6 unicast-routing`
4. True.
5. 128 bits.
6. True.
7. False.
8. True.
9. By using Anycast.
10. One time.

## 第七天实验

### IPv6 概念实验

#### IPv6 概念实验

在一对直接连接的思科路由器上，对在本模块中提到的IPv6概念和命令，进行测试。

- 在两台路由器上都开启IPv6全局单播路由
- 在每个连接的接口上手动配置一个IPv6地址，比如下面这样。
  - 在路由器R1的连接接口上配置 `2001:100::1/64`
  - 在路由器R2的连接接口上配置 `2001:100::2/64`
- 使用命令 `show ipv6 interface` 和 `show ipv6 interface prefix` 对配置进行验证
- 测试直接 `ping` 的连通性
- 使用IPv6无状态自动配置 (`ipv6 address autoconfig default`) 进行重新测试
- 使用 EUI-64 地址 (IPv6地址 `2001::/64` EUI-64) 进行重新测试
- 硬编码一个借口本地链路地址: `ipv6 address fe80:1234:adcd:1::3 link-local`
- 查看IPv6路由表

## 十六进制转换及子网划分练习

### Hex Conversion and Subnetting Practice

请把今天剩下的时间用于练习这些重要的题目上。

- 将十进制转换成十六进制 (随机数字)
- 将十六进制转换成十进制 (随机数字)
- IPv6子网划分 (随机网络和场景)

# 第8天 IPv4与IPv6共存的网络环境

## Integrating IPv4 and IPv6 Network Environments

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第8天任务

- 阅读下面的理论课文
- 阅读ICND1记诵手册

如同在前一课程模块中所学到的那样，通过从IPv4迁移到IPv6，就能收到数不清的好处。回顾一下，这些优势包括：

- 简化了的IPv6数据包头部
- 更大的地址空间
- IPv6寻址层次，IPv6 addressing hierarchy
- IPv6的扩展能力，IPv6 extensibility
- IPv6消除了广播，IPv6 Broadcast elimination
- 无状态的自动配置，Stateless autoconfiguration
- 集成了移动性，Integrated mobility
- 集成了安全增强，Integrated enhanced security

此课程模块对应了一下CCNA大纲要求：

- 掌握IPv6与IPv4共同运作，比如双栈部署时，所需的技术要求（Describe the technological requirements for running IPv6 in conjunction with IPv4, such as dual-stack implementation）。

在有关IPv6的第7天课程中，专门着重于一个纯IPv6环境，并掌握了IPv6运作原理，还学习了如何配置不同路由协议，来支持IPv6的路由，且在思科IOS软件中进行了验证。尽管对IPv6本身的扎实掌握是重要的，但现实情形在于IPv4仍是当前所使用的占主导地位的互联网协议。因此，在考虑完全迁移到纯IPv6环境时，对如何将两种不同协议栈共同运作，是必要的。

尽管迁移到IPv6环境将提供到前面提到的那些优势，当下的情况在于并非所有可寻址的设备，都支持IPv6，那么为了那些运行不同协议栈的网络设备，使用上同一网络设施，就要求在同一网络中IPv6与IPv4共存。IPv4与IPv6的集成与共存策略，可被分为以下三个大的类别：

- 双栈部署，Dual-stack implementation

- 隧道技术， Tunnelling
- 协议转换， Protocol translation

在网际网络设备和主机同时用到两种协议栈（也就是IPv4和IPv6）时，就要求采用双栈部署策略。双栈部署令到主机能够同时使用IPv4或IPv6与其它主机建立端到端的IP会话（Dual-stack implementation is required when internetwork devices and hosts use both protocol stacks(i.e., IPv4 and IPv6). Dual-stack implementation allows the hosts to use either IPv4 or IPv6 to establish end-to-end IP sessions with other hosts）。

**注意：** 双栈部署并不是说那些仅支持IPv4或仅支持IPv6的主机，具备与其它主机通信的能力。要实现此特性，就需要其它的协议与机制。双栈仅指主机（或设施）能够同时支持IPv4协议栈及IPv6协议栈。

在双栈部署无法应用的情形中，就有可能将IPv6数据包经由IPv4网络进行隧道处理（in situations where Dual-stack implementation can not be used, it is possible to tunnel the IPv6 packets over IPv4 networks），使用一些隧道来将IPv6数据包封装在IPv4数据包中，以令到这些IPv6数据包进行跨越尚无或是还没有原生支持IPv6的网络部分。这样做允许一些IPv6“岛”（IPv6 "islands"）通过现行的IPv4设施之间相互通信。

**注意：** 在运用隧道技术时，为将IPv6数据包经由IPv4设施进行隧道化传输，节点或互联网络的设备必须支持双栈（with tunnelling, nodes or internetwork devices must support dual-stack in order to tunnel IPv6 packets over the IPv4 infrastructure）。

最后，在某些情况下，有可能出现某些仅IPv4的环境，需要与仅IPv6的环境进行通信，以及反过来的情况。那么在此种情形下，双栈技术或隧道技术部署都不能用到，因此就必须启用IPv4与IPv6之间的协议转换。尽管此中方案也是支持的，但对于集成IPv4与IPv6网络时，其是最不可选的（while supported, this is the least desirable method of integrating IPv4 and IPv6 networks）。不过因为此方案仍被支持，那么掌握如何实现此种方案，仍是重要的。

这个课程模块的剩余部分，将详细地介绍集成IPv4与IPv6网络的**双栈部署与隧道技术**。包括特定于思科IOS软件的一些配置示例。

## IPv4与IPv6的双栈部署

在双栈部署方案下，尽管某些主机有着采用IPv4和IPv6两种协议栈的能力，但在确定何时采用IPv6而不是IPv4协议栈时，这些主机仍需一些帮助。幸运的是，有两种方法可以实现，这两种方法如下：

- 第一种方法需要用户手动配置。如用户知道目的IPv6主机的IPv6地址，就可以用此IPv6地址，自其双栈主机手动建立一此IPv6会话。尽管此方式可以良好运作，但要记住多台主机的IPv4及IPv6地址，会十分繁琐。
- 那么第二种方式就需要使用某种命名服务，比如DNS。使用此种方法，就要同时使用IPv4和IPv6地址，来配置完全合格的域名（Full Qualified Domain Names, FQDNs），比如 [www.howtonetwork.com](http://www.howtonetwork.com)。FQDN是由一个IPv4协议栈的 A 记录（an A record for the IPv4 protocol stack），以及一个IPv6协议栈的 AAAA 记录表示的，这样的FQDN就令到DNS服务器既可使用IPv4，又可使用IPv6进行查询了。

## 在思科IOS路由器中部署双栈支持

尽管对那些不同厂商的具备双栈部署支持的不同类型主机的不同配置方式的讨论，是超出CCNA考试要求范围的。但作为一名未来的网络工程师，掌握如何在思科IOS软件下部署各种双栈方案，是强制性的

(imperative to understand how to implement dual-stack solutions in Cisco IOS software)。在思科IOS路由器中，双栈运作的启用，通过简单地在路由器接口上配置好IPv4及IPv6即可。

通过在接口配置命令 `ip address [address] [mask]` 后添加 `[secondary]` 关键字，就可以为接口指定多个的 IPv4 地址。对于 IPv6 来说，是不需要 `[secondary]` 关键字的，因为使用第 7 天课程中所介绍的接口配置命令 `ipv6 address`，就可以为每个接口配置多个前缀。下面的配置示例，演示了如何在单一的路由器接口上配置多个IPv4地址和IPv6地址及前缀：

```
R3(config)#ipv6 unicast-routing
R3(config)#interface FastEthernet0/0
R3(config-if)#ip address 10.0.0.3 255.255.255.0
R3(config-if)#ip address 10.0.1.3 255.255.255.0 secondary
R3(config-if)#ip address 10.0.2.3 255.255.255.0 secondary
R3(config-if)#ipv6 address 3fff:1234:abcd:1::3/64
R3(config-if)#ipv6 address 3fff:1234:abcd:2::3/64
R3(config-if)#ipv6 address 3fff:1234:abcd:3::3/64
R3(config-if)#ipv6 enable
R3(config-if)#exit
```

**注意：** 尽管在思科IOS软件中IPv4路由默认是开启的，但IPv6路由却是默认关闭的，所以必须显式地开启。

依据这些IPv4与IPv6地址的配置，就可以通过简单地对查看路由器配置，来验证这些配置，如下面的输出所示：

```
R3#show running-config interface FastEthernet0/0
Building configuration...
Current configuration : 395 bytes
!
interface FastEthernet0/0
ip address 10.0.1.3 255.255.255.0 secondary
ip address 10.0.2.3 255.255.255.0 secondary
ip address 10.0.0.3 255.255.255.0
ipv6 address 3FFF:1234:ABCD:1::3/64
ipv6 address 3FFF:1234:ABCD:2::3/64
ipv6 address 3FFF:1234:ABCD:3::3/64
ipv6 enable
end
```

而要查看具体的IPv4及IPv6接口参数，只需使用思科IOS软件的 `show ip interface [name]` 或 `show ipv6 interface [name]` 命令即可。下面是 `Fastethernet0/0` 接口上 `show ip interface` 的输出：

```
R3#show ip interface FastEthernet0/0 | section address
Internet address is 10.0.0.3/24
Broadcast address is 255.255.255.255
Helper address is not set
Secondary address 10.0.1.3/24
Secondary address 10.0.2.3/24
Network address translation is disabled
```

下面的输出则演示了上一示例中用到的同样的 `Fastethernet0/0` 接口的 `show ipv6 interface` 命令，所打印出的信息：

```
R3#show ipv6 interface FastEthernet0/0 | section address
  IPv6 is enabled, link-local address is FE80::213:19FF:FE86:A20
  Global unicast address(es):
    3FFF:1234:ABCD:1::3, subnet is 3FFF:1234:ABCD:1::/64
    3FFF:1234:ABCD:2::3, subnet is 3FFF:1234:ABCD:2::/64
    3FFF:1234:ABCD:3::3, subnet is 3FFF:1234:ABCD:3::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::5
    FF02::6
    FF02::9
    FF02::1:FF00:3
  Hosts use stateless autoconfig for addresses.
```

## 思科IOS软件中配置静态IPv4及IPv6主机地址

思科IOS软件通过使用全局配置命令 `ip host [name] [v4-address]` 及 `ipv6 host [name] [v6-address]`，而提供了对相应的静态IPv4与IPv6主机地址配置的支持。下面的示例演示了在思科IOS软件中，如何配置静态IPv4及IPv6的主机名字与地址：

```
R1(config)#ip host TEST-HOST 10.0.0.3
R1(config)#ipv6 host TEST-HOST 3FFF:1234:ABCD:1::3
```

该静态IPv4与IPv6主机配置可使用 `show hosts` 命令进行验证，下面打印出了改命令的输出：

```
R1#show hosts
...
[Truncated Output]
...
Host      Port  Flags      Age Type      Address(es)
TEST-HOST  None  (perm, OK)  0   IP        10.0.0.3
TEST-HOST  None  (perm, OK)  0   IPV6     3FFF:1234:ABCD:1::3
```

在同一主机同时配置一个IPv4及IPv6地址时，思科IOS软件将使用IPv6地址。如有使用DNS，那么在主机同时配置了IPv6及IPv4 DNS服务器时，该双栈主机将先搜寻 AAAA (IPv6) 记录，并（在查询不到时）回滚到 A 记录 (IPv4) （If DNS is used, the dual-stack host will first search AAAA (IPv6) records and then fall back to the A records(IPv4) when configured with both IPv6 and IPv4 DNS servers）。可想下面这样通过执行一次简单的到先前配置的静态主机 `TEST-HOST` 的 `ping` 操作，对此默认行为进行验证：

```
R1#ping test-host repeat 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 3FFF:1234:ABCD:1::3, timeout is 2 seconds:
!!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 0/1/4 ms
```

## 在思科IOS软件中配置IPv4及IPv6的DNS服务器

思科IOS软件中IPv4与IPv6 DNS服务器的配置，都依然是使用全局配置命令 `ip name-server [address]`。不过这条命令现在已修改为允许将一个IPv4或IPv6地址，指定为DNS服务器的IP地址。下面的示例演示了如何将路由器配置为同时使用一台IPv4及IPv6 DNS服务器：

```
R1(config)#ip name-server ?
  A.B.C.D Domain server IP address (maximum of 6)
  X:X:X::X Domain server IP address (maximum of 6)
R1(config)#ip name-server 3FFF:1234:ABCD:1::2
R1(config)#ip name-server 192.168.1.2
```

**注意：**正如先前提到的，当在同一路由器上同时配置了IPv4及IPv6 DNS服务器时，路由器将首先查找 AAAA 记录（也就是IPv6）。在如果未找到 AAAA 记录，主机就会查找一条 A 记录，以与该主机名进行通信。

## 经由IPv4网络对IPv6数据报进行隧道传输

### Tunnelling IPv6 Datagrams across IPv4 Networks

通道技术，也就是集成IPv4与IPv6网络的第二种方法，是将IPv6数据包进行封装并通过IPv4网络发送的。为了对本小节中即将说到的几种不同隧道机制进行支持，思科IOS的边缘路由器（Cisco IOS edge routers）必须具有双栈部署（译者注：要配置IPv4和IPv6地址，并开启IPv6路由），以令到IPv6数据包能够以IPv4数据包方式进行封装，且在终端路由器（the terminating router）处被解封装。需要注意的是，中途的那些路由器（intermediate routers）是无需运行IPv6的。也就是说，这些路由器只需简单的仅IPv4路由器。下图8.1演示了一个典型的隧道技术部署：

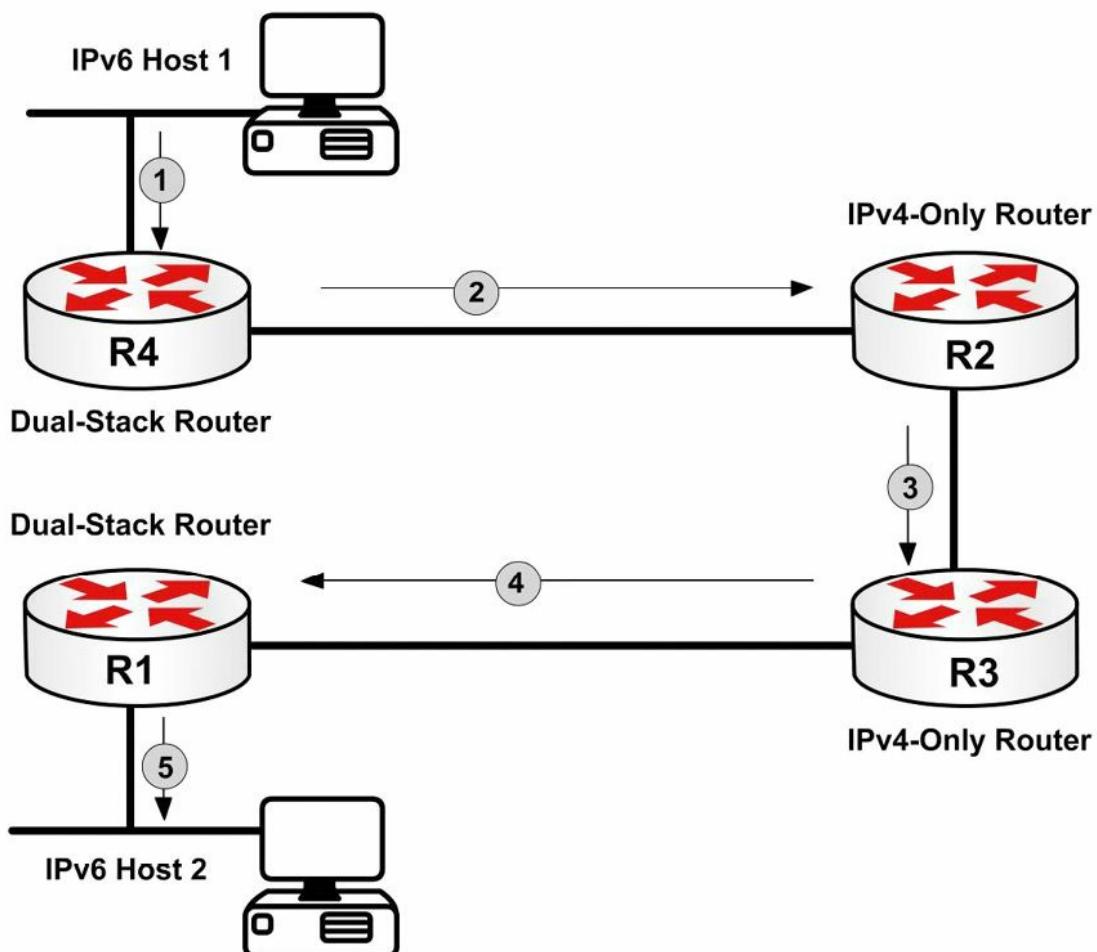


图 8.1 -- 经由IPv4网络进行IPv6数据包的隧道传输

参考图8.1, 假设那台IPv6 Host 1 正在将数据报发送给IPv6 Host 2 , 那么在这些数据包通过该网络时, 会顺序发生下面这些事件:

1. IPv6 Host 1 将那些以IPv6 Host 2 为目的的IPv6数据包, 发送到其默认网关, 也就是路由器 R4 。这些数据包是原生的、在数据包头部包含了IPv6源地址及目的地址的IPv6数据包。
2. R4 是一台双栈路由器。其LAN接口已开启了IPv6, 同时其WAN接口已开启了IPv4。R4有着一个配置在其WAN接口和 R1 的WAN接口, R1 也是一台双栈路由器, 之间的隧道。在收到IPv6数据包后, R4 就将它们以IPv4数据包格式进行封装, 并转发给 R2 。这些数据包的目的地址被设置为 R1 , 同时该路由器 ( R4 ) 将IPv4头的值设置为41, 以表明IPv6数据包是以IPv4数据包形式封装的。
3. R2 收到这些IPv4数据包, 并简单地运用IPv4头部中的目的地址, 将它们路由或转发到其最终目的地。
4. R3 从 R2 处收到这些IPv4数据包, 并简单地运用IPv4头部中的目的地址, 将它们路由或转发到其最终目的地。
5. 最终路由器 R1 , 也是该隧道的出口, 接收到这些原生的IPv4数据包并将其解封装, 从而仅剩下IPv6数据包。于是该路由器就将这些IPv6数据包转发给 Host 2 。

封装与解封装过程, 对这两台主机以及隧道两端 (the tunnel endpoints) 之间的中间那些路由器来说, 是透明的。在以IPv4数据包进行IPv6数据包的隧道传输上, 有着多种方式 (参见后面)。因为超出了CCNA考试要求, 这里不会涉及任何具体配置。

下面列出了一些其它的隧道方式。思科可能会期望你知道有着这些方式, 但不会出有关这些方式如何运作的题目。

- 静态的 (手动配置的) IPv6隧道传输, Static(manually configured) IPv6 tunnelling
- 6to4 隧道技术, 6to4 tunnelling
- 自动的兼容IPv4隧道技术, Automatic IPv4-compatible tunnelling
- ISATAP 隧道技术, ISATAP tunnelling
- 通用路由封装隧道技术, Generic Routing Encapsulation tunnelling

## 第8天问题

1. Name three IPv4 to IPv6 transition mechanism classes.
2. \_\_\_\_\_ implementation is required when internetwork devices and hosts use both protocol stacks (i.e., IPv4 and IPv6) at the same time.
3. With dual-stack implementation, name two methods that help hosts decide when to use the IPv6 protocol stack instead of the IPv4 protocol stack.
4. While IPv4 routing is enabled by default in Cisco IOS software, IPv6 routing is disabled by default and must be explicitly enabled. True or false?
5. Name a command that will provide IPv6 interface parameters.
6. The static IPv4 and IPv6 host configuration can be validated using the \_\_\_\_\_ command.
7. Which command is used to configure an IPv6 DNS server?
8. \_\_\_\_\_ entails encapsulating the IPv6 packets or datagrams and sending them over IPv4 networks.

## 第8天答案

1. Dual-stack implementation, tunnelling, and protocol translation.

2. Dual-stack.
3. Manual configuration and naming service.
4. True.
5. The `show ipv6 interface` command.
6. `show hosts`.
7. The `ip name-server` command.
8. Tunnelling.

## 第8天实验

### IPv4 - IPv6 基础集成实验

在两台直连的思科路由器上，对本课程模块中讲到的一些IPv6概念与命令进行测试：

- 在设备上开启IPv6单播路由，并在直连接口上同时配置IPv4及IPv6地址
- 使用命令 `show interface` 及 `show ipv6 interface`，对该配置进行验证
- 为远端接口地址配置IPv4及IPv6主机，configure IPv4 and IPv6 hosts for remote interface addresses
- 在设备上验证这些主机配置（`show` 命令）
- 在设备之间通过这些主机名字，进行 `ping` 操作
- 在两台路由器上配置IPv4及IPv6的DNS服务器

### IPv4 - IPv6 隧道技术实验

在家庭网络环境下，重现“通过IPv4进行IPv6的隧道传输”小节的场景（包括所有的机制）。要依循该小节中所呈现的事件顺序。可访问[www.in60days.com](http://www.in60days.com)，看看作者是如何完成这个实验的。

# 第9天 访问控制清单

## Access Control Lists

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第九天的任务

- 阅读今天的课文
- 完成今天的实验
- 阅读ICND1记诵指南
- 在[subnetting.org](https://subnetting.org)上花15分钟

和子网划分及VLSM一样，访问控制清单（access control lists, ACL）对于新CCNA学员来说，也是一大难点（one of the bugbear subjects）。有关ACL的问题包括，学习相关的IOS配置命令、理解ACL规则（包括隐式的 deny all 规则），以及掌握端口号及协议类型。

与其它CCNA科目一样，应该在学习过程中一次完成一个小部分，在路由器上使用所见到的每个命令，并完成许多实验。

今天你将学到以下内容。

- ACL基础
- 标准ACLs, 扩展ACLs, 以及命名ACLs
- ACL 规则
- 反掩码 (wildcard masks)
- ACL的配置
- ACLs 故障排除

本课程对应了以下CCNA大纲要求。

- 描述ACLs的类型、特性及应用
  - 标准ACLs, standard
  - 序列号, sequence numbers
  - ACLs的编辑, editing
  - 扩展的ACLs, extended
  - 命名的ACLs, named
  - 编号的ACLs, numbered
  - 日志选项, log option
  - 在某个网络环境下配置并验证ACLs

## ACL 基础

### ACL Basics

ACLs 用于过滤那些通过路由器的流量。没有那个网络是会让任何流量都进入或流出该网络的。

在流量过滤的同时，ACLs 还可用于对NAT地址池的引用，及对调试命令进行过滤（filter your debugging commands），以及对路由地图进行过滤（这是超出CCNA大纲要求的）。

依据所配置的ACL类型，可实现基于源网络/IP地址的过滤、基于目的网络/IP地址的过滤、基于协议或基于端口号的过滤。可在路由器的任何接口，包括Telnet端口，上应用ACLs。

下面是3中主要的ACLs类型。

- 标准的编号ACLs
- 扩展的编号ACLs
- 标准或扩展的命名ACLs

**标准的编号ACLs**是可以应用到路由器上的最为基本的ACL形式。它们是最易于配置的，因此其可用的过滤有着最大的限制。它们仅能依据源IP地址或源网络进行过滤。识别标准ACL的方法就是看配置行的前导数字；标准ACLs的该数字为 1 到 99。

**扩展的编号ACLs**可以有多得多的粒度，但配置和故障排除起来会更难应付。它们可以对某个目的或源IP地址或网络、某种协议类型以及某个端口号进行过滤（they can filter a destination or source IP address or network, a protocol type, and a port number）。可用于配置扩展ACLs的编号为 100 到 199（包含 100 和 199）。

**命名ACLs**允许给某过滤清单一个名称，而不是编号。这就令到在路由器配置中更易于区别这些ACLs了。命名ACLs可以是标准及扩展ACLs；在该ACLs的初始化配置行处，可以选择其作为标准ACL还是扩展ACL。

为在CCNA考试中取得成功，并成为一名思科工程师，你需要理解以下内容。

- 端口号，port numbers
- ACL规则，ACL rules
- ACLs的命令语法，command syntax for ACLs

## 端口号，Port Numbers

如要通过CCNA考试，以及要在实际网络上工作，就必须记住这些常见的端口号。在客户盯着你做事时，去查一下常见端口号是不可能的。这里有些你会碰到且需掌握的一些最常见的端口号。

端口	服务	端口	服务
`20`	FTP数据	`80`	HTTP
`21`	FTP控制	`110`	POP3
`22`	SSH	`119`	NNTP
`23`	Telnet	`123`	NTP
`25`	SMTP	`161/162`	SNMP
`53`	DNS	`443`	HTTPS(带有SSL的HTTP)
`69`	TFTP		

## 访问控制清单规则，Access Control List Rules

这是最难掌握的部分之一。我从没有在哪本思科手册中见到里面曾写过一条完整的规则清单。仅有一些手册对其简单概过或是稍加解释，另外一些则完全不讲。难点就在于这些规则一直都在用，但到目前为止你都是通过试误法发现的它们（the difficulty is that the rules always apply but until now, you found them only by trial and error）。下面就是你需要知道的这些规则了。

### ACL规则一 -- 在每个接口的每个方向，只使用一条ACL

**Use only one ACL per interface per direction**

这么做是很明智的。在同一接口上，有多条ACLs去做不同的事情，大概不是你想要的。简单地配置一条ACL，来完成需要完成的事情，而不是将过滤器分散到两条或多条的清单中。本应将“每个协议（per protocol）”加入到此规则中，因为这里是可以包含IPX的访问控制清单的，不过在现代网络中，IP已成为唯一的协议了。

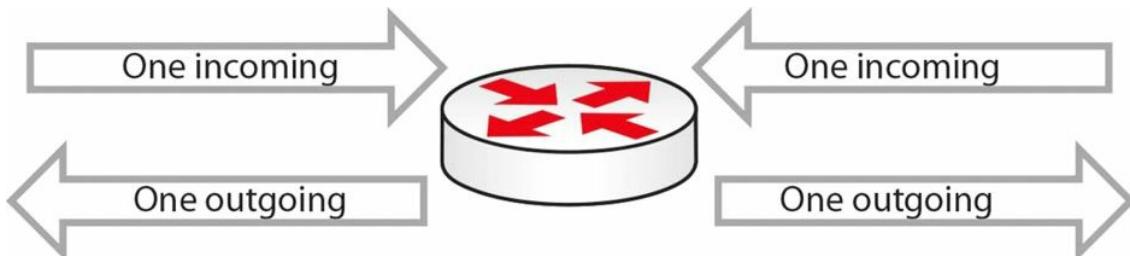


图9.1 -- 接口上的一个方向仅做一条ACL

### ACL规则二 -- ACLs行自顶向下进行处理

**the lines are processed top-down**

某些工程师在他们的ACL未如预期那样运行时感到迷惑。路由器会看看ACL的顶行，在发现匹配后，就会停在那里且不再对其他行进行检查了。为此，需要将**最明确的(最小的)那些条目放在ACL的顶部**（you need to put the most specific entries at the top of the ACL）。比如在利用ACL来阻挡主机 172.16.1.1 时的做法。

`Permit 10.0.0.0`		没有匹配的
`Permit 192.168.1.1`		没有匹配的
`Permit 172.16.0.0`	‘O’	匹配了-放行
`Permit 172.16.1.0`		不会处理了
`Deny 172.16.1.1`		不会处理了

在本例中，应该将 `Deny 172.16.1.1` 这行，放到顶部，或至少应在语句（statement） `Permit 172.16.0.0` 之前。

### ACL规则三 -- 在每条ACL的底部，都有一句隐式的“deny all”

**There is an implicit "deny all" at the bottom of every ACL**

这条规则另很多工程师为难。在每条ACL的底部，有着一条看不见的命令。该命令设置为拒绝尚未匹配的所有流量。而阻止此命令起作用的唯一方法，就是在底部手动配置一条 `permit all` 命令。在取得来自IP地址 `172.20.1.1` 的某个进入的数据包时的做法。

`Permit 10.0.0.0`	无匹配项
`Permit 192.168.1.1`	无匹配项
`Permit 172.16.0.0`	无匹配项
`Permit 172.16.1.0`	无匹配项
`[Deny all]`	匹配 -- 丢弃数据包

你实际上想要路由器放行该数据包，但却拒绝了。原因就在于那条隐式的 `deny all` 命令了，而该命令实际上是一种安全手段。

## ACL规则四 -- 路由器是不能过滤自己产生的流量的

**The router can't filter self-generated traffic.**

这在某个实际网络上于部署ACL前进行测试时，会造成混乱。路由器不会过滤其自身产生的流量。在图9.2中有演示。

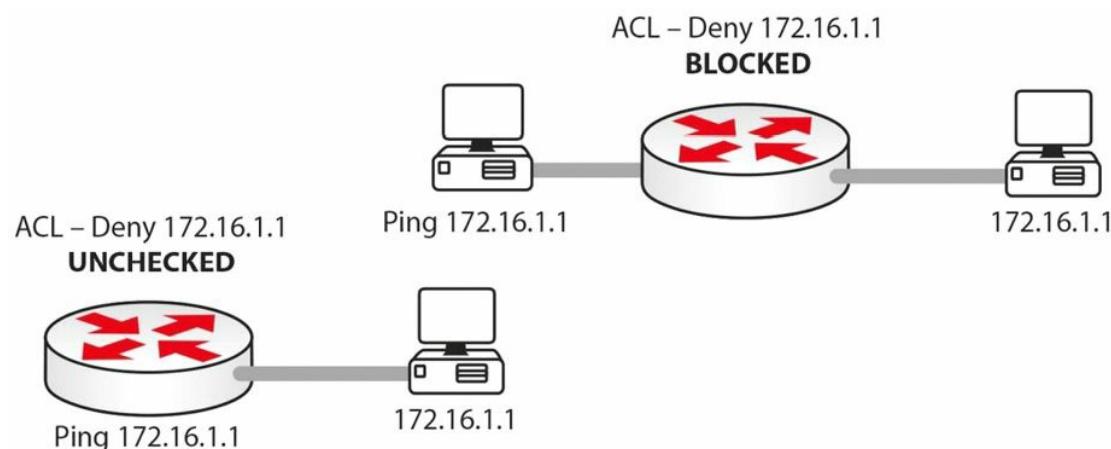


图9.2 -- 对自身流量的ACL 测试

## ACL规则五 -- 不能对运行中的ACL进行编辑

**You can't edit a live ACL.**

实际上，在 `iOS 12.4` 之前的版本中，只能对命名ACL进行编辑，而不能对标准ACL或扩展ACLS两种进行编辑。这曾是ACL架构的一个局限（this was a limitation of ACL architecture）。在 `iOS 12.4` 之前，如想要编辑标准ACL或扩展ACL，就必须按照以下步骤进行（这里使用 `list 99` 作为例子）。

1. 使用命令 `no ip access-group 99 in`，在接口上停用ACL流量（stop ACL traffic on the interface with the `no ip access-group 99 in` command）。
2. 将该条ACL复制粘贴到文本编辑器，并在那里编辑好。
3. 进入到ACL模式，将新的ACL粘贴上去。
4. 再次将该ACL应用到接口。

在实际的路由器上，执行下面的这些命令。

在接口上已创建并应用的ACL。

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 1 permit 172.16.1.1
Router(config)#access-list 1 permit 172.16.2.1
Router(config)#interface FastEthernet0/0
Router(config-if)#ip access-group 1 in
```

现在从接口上卸下。

```
Router(config)#int FastEthernet0/0
Router(config-if)#no ip access-group 1 in
Router(config-if)#^Z
```

查看那些ACLs。将其复制并粘贴到文本编辑器，并进行修改。

```
Router#show run - or show ip access lists
access-list 1 permit host 172.16.1.1
access-list 1 permit host 172.16.2.1
```

实际上还需在配置行之间加入一个叹号（如是将其粘贴到路由器上的情况下），来告诉路由器执行一次确认（you actually need to add an exclamation mark in-between each line of configuration, if you are pasting it in, to tell the router to do a carriage return）[wikipedia: 回车符](#)。

```
access-list 1 permit host 172.16.1.1
!
access-list 1 permit host 172.16.2.2
```

下面是正被粘贴到路由器配置中的那些行。要先删除早先的ACL，再粘贴进新版本。

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#no access-list 1
Router(config)#access-list 1 permit host 172.16.1.1
Router(config)#
Router(config)#access-list 1 permit host 172.16.2.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
show ip access
Router#show ip access-lists
Standard IP access list 1
    permit host 172.16.1.1
    permit host 172.16.2.2
Router#
Router(config)#int FastEthernet0/0
Router(config-if)#ip access-group 1 in - reapply to the interface
```

如使用的是Packet Tracer，那么这些命令可能不会工作。同时，请一定在某台路由器上尝试这些命令，因为它们是考试考点。记住在编辑ACL前要先在接口上关闭它（此时它就不再是活动的了），以避免一些奇怪或是不可预期的行为发生。而在ios 12.4 及以后的版本中，如何来编辑ACLs，会在后面演示。

## ACL规则六 -- 在接口上关闭ACL

#### Disable the ACL on the interface.

在打算短时间对ACL进行测试或是撤销ACL时，许多工程师都会将其完全删除掉。这是不必要的。如你要停止ACL运行，只需简单地将其从所应用到的接口上移除即可。

```
Router(config)#int FastEthernet0/0  
Router(config-if)#no ip access-group 1 in  
Router(config-if)#^Z
```

### ACL规则七 -- 可重用同一ACL

#### You can reuse the same ACL.

这是我在实际网络中经常见到的。整个网络通常都有着同样的ACL策略。与其配置多条ACLs，只需简单地引用同一ACL，然后在所需要的那些接口上应用该ACL即可。图9.3演示了此概念。

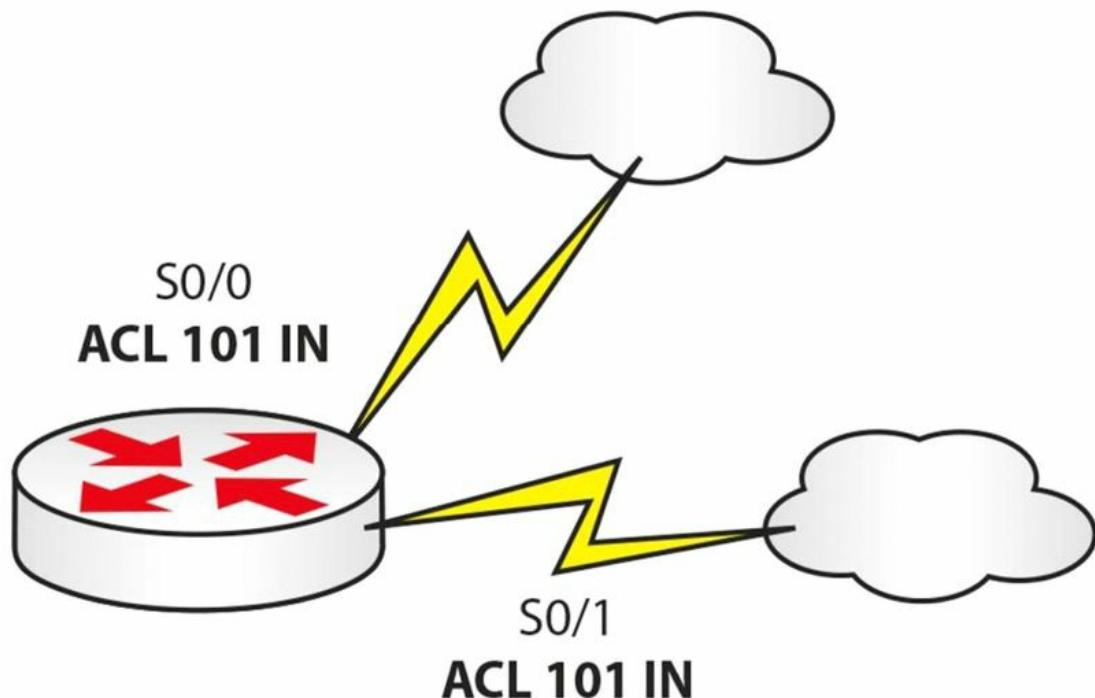


图9.3 -- ACL 的重用

### ACL规则八 -- ACL应保持简短

ACLs的基本规则就是保持简短且只专注于做一件事情。许多新手的思科工程师，将其ACL延伸到数行那么长，最后，经深思熟虑后，就可以紧缩到少数几行的配置。前面提到的将那些最为特定的（最小的）行放在ACL的顶部。这是好的做法，从而可以节约路由器CPU的执行周期。

优良的ACL配置技能，来自于知识和操练。

### ACL规则九 -- 尽可能将ACL放在接近源的地方

思科文档建议将扩展ACL尽量放在离源近的地方，而将标准ACL尽量放在离目的近的地方，因为这可以避免不必要的开销，又能放行那些合法流量。

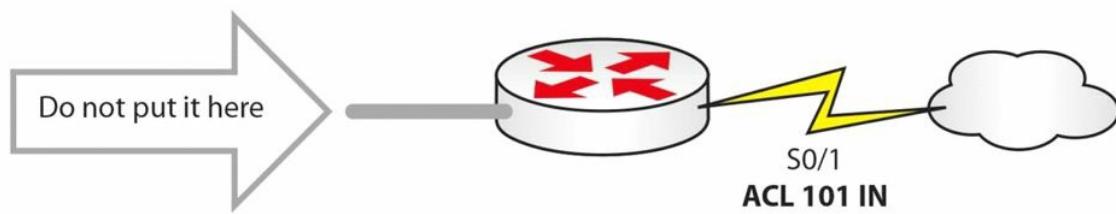


图9.4 -- 将ACL尽量放在离源近的地方

Farai 指出 -- “思科官方建议是扩展ACL尽量离源近，而标准ACL尽量离目的近”。

## 反掩码

### Wildcard Masks

因为在ACLS及某些路由协议的配置中，反掩码是命令行的组成部分，所有有必要学习反掩码。之所以存在反掩码，是因为需要有某种方式来告诉路由器，我们要去匹配IP地址或网络地址的哪些部分。

匹配实在二进制层面完成的，却可以采用与子网掩码相同的表示法，来容易地配置反掩码。一个二进制 1 告诉路由器忽略对应的位，0 则是要匹配的位。

CCNA考试中反掩码计算的一种简易方法，就是把一个数与子网掩码相加，确保它们的和为 255。那么如果子网掩码的某个 8 位值为 192 时，需要加上 63 才等于 255。而如果子网掩码的某个 8 位值为 255，则需要加的就是 0。看看下面的例子吧。

子网掩码	`255`	`255`	`255`	`192`
反掩码	`0`	`0`	`0`	`63`
相加之和	`255`	`255`	`255`	`255`

子网掩码	`255`	`255`	`224`	`0`
反掩码	`0`	`0`	`31`	`255`
相加之和	`255`	`255`	`255`	`255`

子网掩码	`255`	`128`	`0`	`0`
反掩码	`0`	`127`	`255`	`255`
相加之和	`255`	`255`	`255`	`255`

在想要ACL与匹配某个子网或是整个网络时，就需要输入一个反掩码。比如，要匹配 172.20.1.0 255.255.224.0，就需要输入下面的命令。

```
Router(config)#access-list 1 permit 172.20.1.0 0.0.31.255
```

而要匹配子网 192.200.1.0 255.255.255.192，就需要下面的命令。

```
Router(config)#access-list 1 permit 192.200.1.0 0.0.0.63
```

在OSPF中应用网络语句时也要当心(be careful when applying network statements with OSPF)，那位那也要用到反掩码。

在有着一个仅有两位主机位的网络时，也要当心，因为需要输入一条ACL来匹配这些主机位。比如，要匹配子网 192.168.1.0 255.255.255.252，或 /30 的话，需要输入下面的命令。

```
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.3
```

这里剔除了一些配置，是为展示出对应的部分。上面的命令将匹配 192.168.1.0 网络上的 1 号和 2 号主机。而如果要匹配 192.168.1.4/30 网络上的 5 号和 6 号主机，则需输入下面的命令。

```
Router(config)#access-list 1 permit 192.168.1.4 0.0.0.3
```

请阅读子网划分和VLSM部分的课文，以更好地掌握此概念。

## 访问控制清单的配置

### Configuring Access Control Lists

熟能生巧，对于任何技能都是适用的。如同前面提到的，你应该在路由器上输入这里给出的每个例子，完成尽可能多的实验，并构建出自己的实例。在考试和现实世界中，你都需要精准快速的设计ACL。

接下来的章节中出现的标准和扩展ACLs都是编号ACLs。它们是配置ACLs的经典方法。命名ACLs是配置ACLs的另一种方式，将在其后的部分出现。

### 标准ACLs

#### Standard ACLs

标准的编号ACLs是最易于配置的，所以拿它来作为开端是最好的。标准ACLs只能实现依据源网络或源IP地址的过滤。

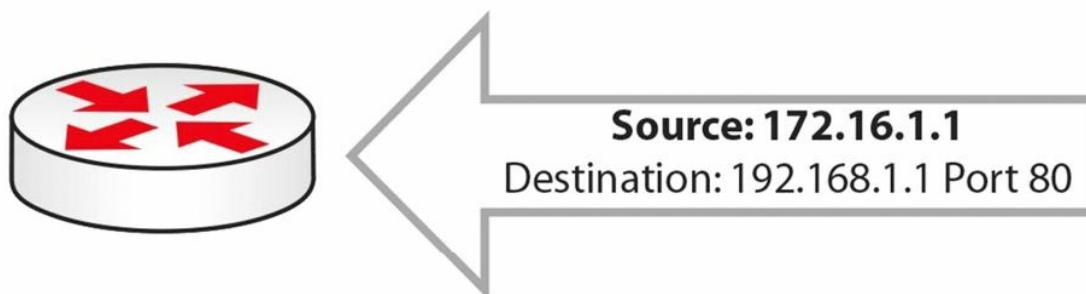
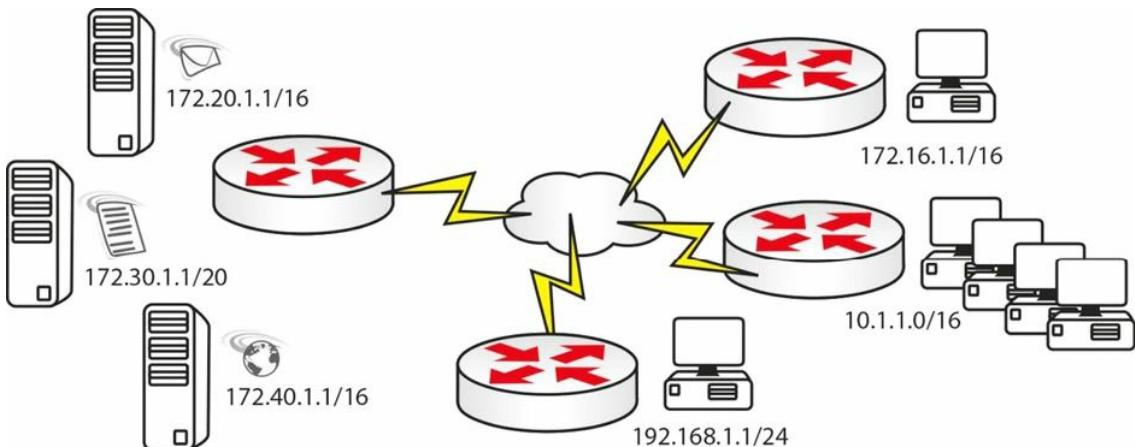


图9.5 -- 带有源和目的地址的进入数据包

在上面的图9.5中，进入的数据包有着一个源和目的地址，但标准ACL只会查看源地址。该ACL会放行会拒绝该源地址（参见图9.6）。



```
Router(config)#access-list 1 permit host 172.16.1.1
Router(config)#access-list 1 permit host 172.16.1.1
Router(config)#access-list 1 permit host 192.168.1.1
Router(config)#access-list 1 permit 10.1.0.0 0.0.255.255
```

此ACL应在服务器侧的路由器上应用。又记得在清单的底部有一条隐式的 `deny all`，所以其它流量都会给阻止掉。

## 扩展ACLs

### Extended ACLs

**扩展的编号ACLs中可以构建出细得多的粒度。**而正是由于有了细得多的粒度，令到扩展的编号ACLs变得诡异起来。藉由扩展的编号ACLs，可以对源或目的网络地址、端口、协议及服务进行过滤。

一般来说，你可以看看扩展的ACLs配置语法，就像下面这样。

```
access list# permit/deny [service/protocol] [source network/IP] [destination network/IP] [port#]
```

比如下面这样。

```
access-list 101 deny tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq telnet
access-list 100 permit tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq ftp
access-list 100 permit icmp any any
```

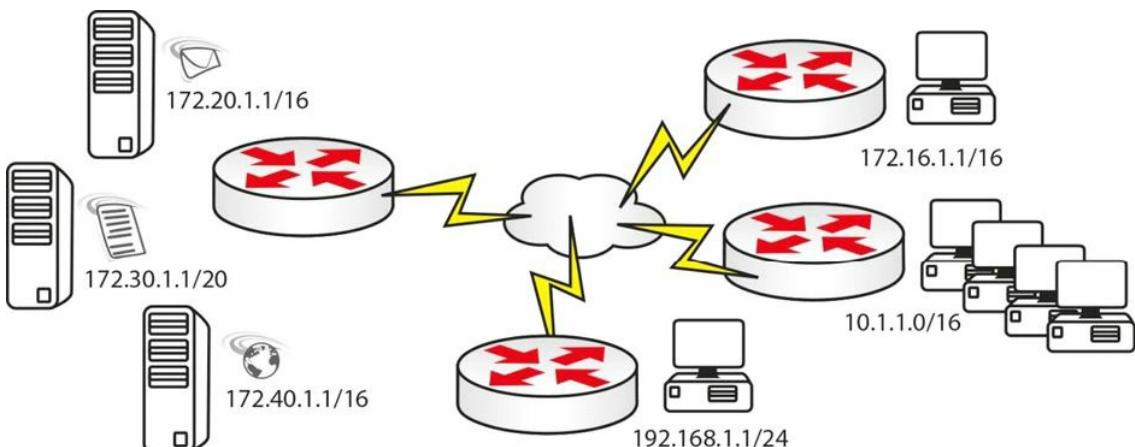


图9.7 -- 阻止服务器访问实例

可为上面的网络配置一条ACL，以e-mail、web和文件服务器为例，可以像下面这样（应用在服务器侧的）。

```
access-list 100 permit tcp host 172.16.1.1 host 172.20.1.1 eq smtp
access-list 100 permit tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq ftp
access-list 100 permit tcp host 192.168.1.1 host 172.40.1.1 eq www
```

而如有不同要求，就也可以像下面这条ACL。

```
access-list 101 deny icmp any 172.20.0.0 0.0.255.255
access-list 101 deny tcp 10.1.0.0 0.0.255.255 host 172.30.1.1 eq telnet
```

或者也可以像下面这样。

```
access-list 102 permit tcp any host 172.30.1.1 eq ftp established
```

关键字 [established] 告诉路由器仅放行在网络内部的主机所发起的流量。三次握手标志（ACK或RST位）将表明这点（the three-way handshake flags, ACK or RST bit, will indicate this）。

## 命名ACLs

### Named ACLs

与编号ACLs不同，命名ACLs可由其描述性名称容易地区分，而这在一些大型的配置中尤其有用。引入命名ACLs就是为增加灵活性及ACLs的易于管理的。命名ACLs可以看作是配置增强的提升，因为它并未对ACLs结构进行修改（仅改变了引用ACL的方式而已）。

其语法跟编号ACLs是相似的，主要的不同就是使用名称而不是编号来区分ACLs。和编号ACLs一样，可以配置标准的或扩展的命名ACLs。

在配置命名ACLs时的另一不同之处，就是必须一直使用命令 ip access-list，这与编号ACLs可以只使用简单的 access-list 命令，是不一样的。

```
Router(config)#access-list ?
<1-99>          IP standard access list
<100-199>        IP extended access list
<1100-1199>      Extended 48-bit MAC address access list
<1300-1999>      IP standard access list (expanded range)
<200-299>        Protocol type-code access list
<2000-2699>      IP extended access list (expanded range)
<700-799>        48-bit MAC address access list
dynamic-extended   Extend the dynamic ACL absolute timer
rate-limit         Simple rate-limit specific access list
Router(config)#ip access-list ?
extended          Extended access list
log-update         Control access list log updates
logging            Control access list logging
resequence         Resequence access list
standard           Standard access list
R1(config)#ip access-list standard ?
<1-99>    Standard IP access-list number<1300-1999> Standard IP access-list number (expanded range)
WORD      Access-list name
R1(config)#ip access-list extended ?
<100-199>  Extended IP access-list number
<2000-2699> Extended IP access-list number (expanded range)
WORD      Access-list name
```

命名ACLs在语法上与其它类型的ACLs（也就是标准和扩展的编号ACLs）有着轻微的不同。同时也可以编辑活动的命名ACLs，这是一个有用的特性。只需简单地告诉路由器要配置一条命名ACL，而不管它是标准的还是扩展的。在较新的IOS版本上，也可以编辑编号ACLs，所以请检查所用的平台。

在使用 `ip access-list` 命令常见一条命名ACL时，思科IOS会将你带入ACL配置模式，在那里就可以输入或是移除ACL条目了（就是那些拒绝或放行的访问条件）。图9.8展示了一条命名ACL的实例，以及相应的输出。

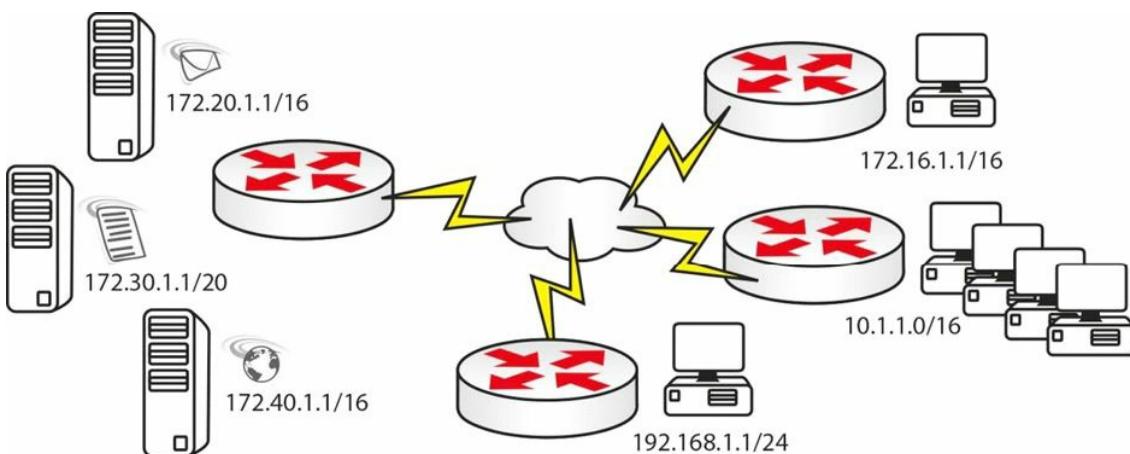


图9.8 -- 命名ACL

```
Router(config)#ip access-list extended BlockWEB
Router(config-ext-nacl)#
Ext Access List configuration commands:
<1-2147483647>    Sequence Number
default              Set a command to its defaults
deny                Specify packets to reject
dynamic              Specify a DYNAMIC list of PERMITs or DENYs
evaluate             Evaluate an access list
exit                Exit from access-list configuration mode
no                  Negate a command or set its defaults
permit              Specify packets to forward
remark              Access list entry comment
Router(config-ext-nacl)#deny tcp any any eq 80
Router(config-ext-nacl)#permit ip any any
```

命名ACL的验证，可通过下面的命令完成。

- `show ip access-list` : 显示设备上所创建的所有ACLs
- `show ip access-list <acl_name>` : 显示某条特定的命名ACL

```
Router(config)#do show ip access-lists
Standard IP access list test
 30 permit 10.1.1.1
 20 permit 192.168.1.1
 15 permit 172.20.1.1
 10 permit 172.16.1.1
```

要知道如何来增加或是删除某条命令ACL中的条目，请参考下面的“ACL序号（ACL Sequence Numbers）”小节。

## 应用ACLs

### Applying ACLs

为让ACLs发挥效果，就必须将ACL应用到路由器的某个接口或端口上。之所以这样讲，是因为我曾见到许多的新手思科工程师在敲入了ACL后，就想为什么它不工作！或者他们配置了ACL，却将错误的ACL编号或命名应用到相应的接口上。

如要应用在某条线路上，就必须使用 `access-class` 命令来指定它，而如果是应用在某个接口上，就要用 `ip access-group` 命令。思科这么做的原因，我也不知道。

这里有应用ACLs到端口或接口上的三个实例。

接口上的应用。

```
Router(config)#int FastEthernet0/0
Router(config-if)#ip access-group 101 in
```

线路上的应用。

```
Router(config)#line vty 0 15
Router(config-line)#access-class 101 in
```

接口上的应用。

```
Router(config)#int FastEthernet0/0
Router(config-if)#ip access-group BlockWEB in
```

## ACL序号

### ACL Sequence Numbers

自 12.4 往后，你会发现思科IOS给每个ACL条目添加了序号。那么现在就可以创建一条访问控制清单，并在其后从它里面一处一行了。

```
Router(config)#ip access-list standard test
Router(config-std-nacl)#permit 172.16.1.1
Router(config-std-nacl)#permit 192.168.1.1
Router(config-std-nacl)#permit 10.1.1.1
Router(config-std-nacl)#
Router(config-std-nacl)#exit
Router(config)#exit
Router#
*Jun 6 07:38:14.155: %SYS-5-CONFIG_I: Configured from console by console access
Router#show ip access-lists
Standard IP access list test
    30 permit 10.1.1.1
    20 permit 192.168.1.1
    10 permit 172.16.1.1
```

注意到在路由器运行配置中，序号并不会显示出来。要查看它们，必须执行一个 `show [ip] access-list` 命令。

## 加入一个ACL行

### Add an ACL Line

要加入一个新的ACL行，只需简单地输入新的序号并接着输入该ACL语句。下面的例子展示如何往现有的 ACL中加入行 15。

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access
Router(config)#ip access-list standard test
Router(config-std-nacl)#15 permit 172.20.1.1
Router(config-std-nacl)#
Router(config-std-nacl)#do show ip access
Router(config-std-nacl)#do show ip access-lists
Standard IP access list test
    30 permit 10.1.1.1
    20 permit 192.168.1.1
    15 permit 172.20.1.1
    10 permit 172.16.1.1
Router(config-std-nacl)#

```

### 移除一个ACL行

#### Remove an ACL Line

要移除某个ACL行，只需简单地敲入 `no <seq_number>` 命令即可，就如同下面的例子中行 20 被删除掉了。

```
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access
Router(config)#ip access-list standard test
Router(config-std-nacl)#no 20
Router(config-std-nacl)#
Router(config-std-nacl)#do show ip access
Router(config-std-nacl)#do show ip access-lists
Standard IP access list test30 permit 10.1.1.1
    15 permit 172.20.1.1
    10 permit 172.16.1.1
Router(config-std-nacl)#

```

### 为某条ACL重新编号

#### Resequence an ACL

要对某条ACL重新编号，使用 `ip access-list resequence <acl_name> <starting_seq_number> <step_to_increment>` 命令。该命令的行为可由下面的例子进行检验。

```
Router(config)#ip access-list resequence test 100 20
Router(config)#do show ip access-lists
Standard IP access list test
    100 permit 10.1.1.1
    120 permit 172.20.1.1
    140 permit 172.16.1.1
Router(config-std-nacl)#

```

命令 `resequence` 则会创建新的序号，自 `100` 开始，每个新行增加 `20`。译者注：在更新的IOS版本中，此命令可指定开始序号及步进序号。

## ACL日志

### ACL Logging

默认情况下，通过那些为某个接口的数据包所匹配上的ACL条目，会创建出一个不断增大的计数器，该计数器可使用 `show ip access-list` 命令进行查看，如下面的例子所示。

```
Router#show ip access-lists
Extended IP access list test
  10 deny tcp any any eq 80 (10 matches)
  20 permit ip any any (56 matches)
```

而如果需要更详细的有关那些为ACL条目所匹配的流量信息，可以给相关的ACL条目配置 `log` 或 `log-input` 参数。

```
Router(config)#ip access-list extended test
Router(config)#no 10
Router(config)#10 deny tcp any any eq 80 log
Router#show ip access-lists
Extended IP access list test
  10 deny tcp any any eq 80 log
  20 permit ip any any (83 matches)
```

在上面的配置样例中，配置了test ACL的10号条目的ACL日志。在某个数据包与那个条目匹配时，该ACL计数器就会增加，与此同时路由器也会生成一条包含了该特定ACL匹配的详细日志记录。

```
%SEC-6-IPACCESSLOGP: list test denied tcp 10.10.10.2(24667) -> 10.10.10.1(80), 1 packet
```

而如果你仍需要更多有关该事件（transaction）的细节，就要用 `log-input` 参数替代 `log` 参数了，就像下面这样。

```
Router(config)#ip access-list extended test
Router(config)#no 10
Router(config)#10 deny tcp any any eq 80 log-input
Router#show ip access-lists
Extended IP access list test
  10 deny tcp any any eq 80 log-input
  20 permit ip any any (125 matches)
```

这时，当有该特定ACL条目匹配时，路由器就会生成一条更为详细的消息，当中包含了进入的接口以及源MAC地址。

```
%SEC-6-IPACCESSLOGP: list test denied tcp 10.10.10.2(14013) (FastEthernet0/0 00aa.aabb.ccdd) -> 10.10.10.1(80)
```

**ACL日志在查看到底那些数据包被丢弃或放行的故障排除中，会是非常有用的**，但在现实世界情形中（此内容超出CCNA考试范围）不得不提的是：包含 `[log]` 或 `[log-input]` 关键字的ACL条目是为路由器进行线程交换的，与之相反，现代路由器中，默认都是经由CEF交换的（`ACL entries that contain [log] or [log-input]` keyword are process-switched by the router, as opposed to being CEF-switched, which is the default in modern routers）。这需要更多的路由器CPU周期，因而导致在有大量与被记录的ACL条目匹配时，出现问题。

## 使用ACLs来限制Telnet和SSH访问

### Using ACLs to Limit Telnet and SSH Access

除了在接口级别过滤流量外，ACLs可与其他设备特性配合使用，包括过滤VTY线路上的流量。在前面的课程中，我们曾学过如何利用 `line vty` 命令，配置Telnet和SSH以实现对某台设备的访问（比如路由器或交换机）。

有时，我们可能不想接受到设备或自设备发出的所有Telnet/SSH连接。而为实现此操作，就必须定义一条ACL，以指定在VTY线路上所允许或拒绝的流量类型。该ACL可以是编号ACL或命名ACL。通过命令 `access-class <acl> | [in|out]`，将该ACL加入到想要的VTY线路上。

下面的例子定义了一条允许来自主机 `10.10.10.1` 的Telnet流量，该ACL随后被应用到VTY线路的进入方向。

```
Router(config)#ip access-list extended VTY_ACCESS
Router(config-ext-nacl)#permit tcp host 10.10.10.1 any eq telnet
Router(config-ext-nacl)#deny tcp any any
Router(config-ext-nacl)#exit
Router(config)#
Router(config)#line vty 0 4
Router(config-line)# access-class VTY_ACCESS in
Router(config-line)#

```

使用以下命令对配置进行验证。

```
Router#show run | sect line vty
line vty 0 4
access-class VTY_ACCESS in
....
```

## ACLs故障排除和验证

### Trubleshooting and Verifying ACLs

相信有了对配置命令和规则的深入理解，在访问控制清单上就不会有问题了。在ACL不工作的时候，首先要通过ping操作，检查有没有基本的IP连通性问题。接着看看有没有应用该ACL，看看在ACL中有没有什么文字错误，以及你是否需要允许任何IP流量通过（记住那个隐式的 `deny all` 条目）。而一些在ACL故障排除过程中最重要的检查点包括下面这些。

- 查看ACL统计信息
- 检查所允许的网络
- 检查应用ACL的接口及方向

### 查看ACL统计信息

在成功配置一条ACL并将其应用到某个接口上之后，某种可以验证该ACL正确行为的手段非常重要，尤其是某个ACL条目被使用到的次数。基于匹配次数，就可以对过滤策略进行调整，或者对ACLs进行增强，以实现整体安全性的提升。而根据需求的不同，可以在全局层面或者单个接口上（从 `ios 12.4` 开始）查看ACL统计信息。

#### ACL全局统计信息

#### Global ACL Statistics

可使用命令 `show ip access-list` 或 `show access-list` 命令，查看ACL全局统计信息，这两个命令又可以仅查看某个特定编号ACL或命名ACL的全局统计信息。

```
Router#show ip access-lists
Extended IP access list test
  10 deny tcp any any eq 80 (10 matches)
  20 permit ip any any (56 matches)
```

在将某同一ACL重用到不同接口上时，这种方式并不会提供到十分特定的信息，因为它给出的是整体统计信息。

### 单个接口上的ACL统计信息

#### Per Interface ACL Statistics

在想要查看单个接口上的ACL匹配情况，不管是进还是出方向时，可以使用命令 `show ip access-list interface <interface_name> [in|out]`，如下面所示。

```
Router#show ip access-list interface FastEthernet0/1 in
Extended IP access list 100 in
  10 permit ip host 10.10.10.1 any (5 matches)
  30 permit ip host 10.10.10.2 any (31 matches)
```

如未有指定方向，则应用到该特定接口上的任何进或出方向的ACL都将显示出来。此特性也叫做“ACL可管理能力（ACL Manageability）”，自 `ios 12.4` 开始可用。

## 检查那些放行的网络

#### Verifying the Permitted Networks

有的时候，特别实在那些必须配置很多ACLs的大型网络中，在配置ACL条目是就会犯下一些书写错误，而这就会导致不同接口上有错误的流量被阻止。为了检查那些正确的ACL条目（也就是permit及deny语句），可以照前面章节中讲到的那样，使用 `show run | section access-list` 或者 `show ip access-list` 命令。

## 检查ACL的接口和方向

#### Verifying the ACL Interface and Direction

在将某条ACL应用到某个接口上时，一个常见的错误就是将其应用到了错误的方向，也就是本应在进方向的，却应用到了出方向，或者本应在出方向的，却应用到了进方向。这会导致功能上和安全方面的很多问题。于是在ACL故障排除上的最先几步之一，就是检查ACL应用到正确的接口及正确的方向。

为此，可以使用多种命令，包括 `show run` 及 `show ip access-list interface <interface> | [in|out]` 命令。

## 第九天的问题

1. You can have a named, extended, and standard ACL on one incoming interface. True or false?
2. You want to test why your ping is blocked on your Serial interface. You ping out from the router but it is permitted. What went wrong? (Hint: See ACL Rule 4.)
3. Write a wildcard mask to match subnet mask 255.255.224.0 .
4. What do you type to apply an IP access control list to the Telnet lines on a router?
5. How can you verify ACL statistics per interface (name the command)?
6. How do you apply an ACL to an interface?

## 第九天问题的答案

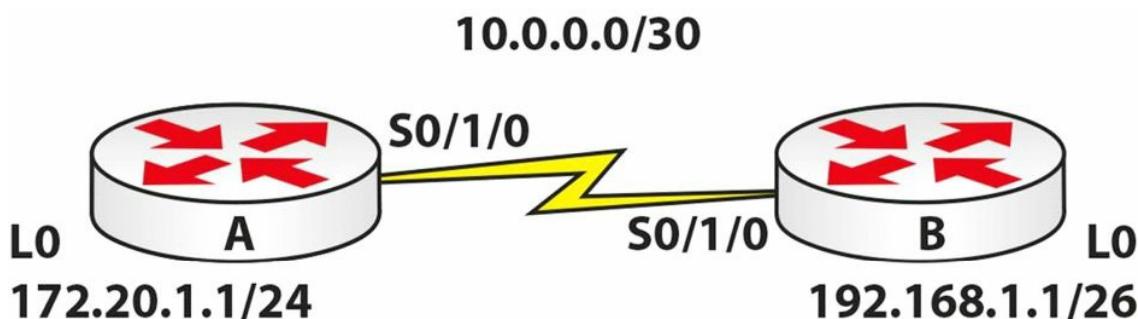
1. False. You can only configure a single ACL on an interface per direction.
2. A router won't filter traffic it generated itself.
3. `0.0.31.255`.
4. `access-class`.
5. Issue the `show ip access-list interface` command.
6. Issue the `ip access-group <ACL_name> [in|out]` command.

## 第九天的实验

### 标准ACL实验

#### Standard ACL Lab

##### 拓扑图



标准ACL实验拓扑图

#### 实验目的

学习如何配置一条标准ACL。

#### 实验步骤

1. 配置上面的网络。在两台路由器上加入一条静态路由，领导到任何网络的任何流量都从串行接口发出。这么做的原因是，尽管这不是一个路由实验，仍然需要路由的流量。把 .1 地址加到路由器 A 的串行接口， .2 地址加到路由器 B 的串行接口。

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 s0/1/0
RouterB(config)#ip route 0.0.0.0 0.0.0.0 s0/1/0
```

1. 在路由器A上配置一条标准ACL，放行 192.168.1.0/10 网络。默认情况下，其它所有网络都将被阻止。

```
RouterA(config)#access-list 1 permit 192.168.1.0 0.0.0.63
RouterA(config)#int Serial0/1/0
RouterA(config-if)#ip access-group 1 in
RouterA(config-if)#exit
RouterA(config)#exit
RouterA#
```

1. 从路由器 B 上测试该条ACL， 默认将使用 10.0.0.1 地址。

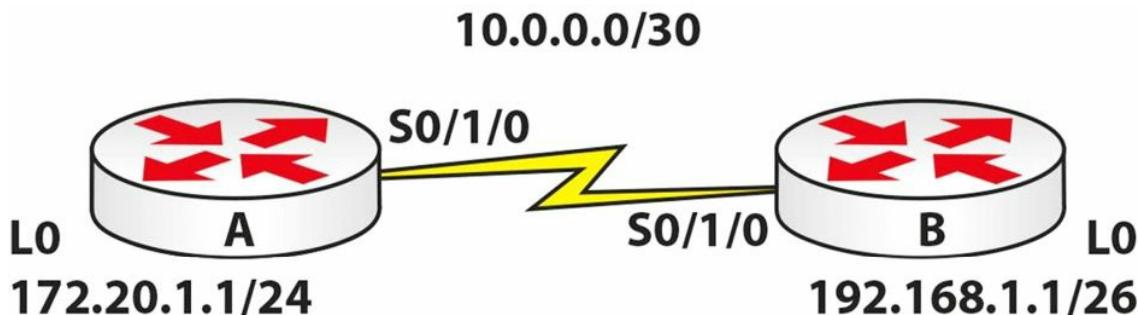
```
RouterB#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
UUUUU
Success rate is 0 percent (0/5)
```

1. 以源地址 192.168.1.1 来做另一个ping测试，这将没有问题。

```
RouterB#ping
Protocol [ip]:
Target IP address: 10.0.0.1
Repeat count [5]:Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 192.168.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/31/32 ms
```

## 扩展ACL实验

### 拓扑图



扩展ACI实验的拓扑图

### 实验目的

学习如何配置一条扩展ACL。

### 实验步骤

1. 配置上述网络。在路由器 B 上加入一条静态路由，令到前往所有网络的所有流量都从串行接口上发出。这么做是因为，尽管这不是一个路由实验，仍然需要路由流量。

```
RouterB(config)#ip route 0.0.0.0 0.0.0.0 s0/1/0
```

1. 在路由器 A 上配置一条扩展ACL。仅允许往环回接口上发起Telnet流量。

```

RouterA(config)#access-list 100 permit tcp any host 172.20.1.1 eq 23
RouterA(config)#int s0/1/0
RouterA(config-if)#ip access-group 100 in
RouterA(config-if)#line vty 0 15
RouterA(config-line)#password cisco
RouterA(config-line)#login
RouterA(config-line)#^Z
RouterA#

```

上面的那条ACL编号为 100，这就告诉路由器，它是一条扩展ACL。所要允许的是TCP。该条ACL允许来自任何网络的，目的地址为 172.20.1.1 的Telnet端口，端口号为 23。在执行 show run 命令时，就会看到，路由器实际上会将端口号替换为其对应的名称，就像下面演示的这样。

```
access-list 100 permit tcp any host 172.20.1.1 eq telnet
```

1. 现在，从路由器B上做一个Telnet测试。首先往路由器 A 的串行接口上Telnet，将会被阻止。接着测试环回接口。

```

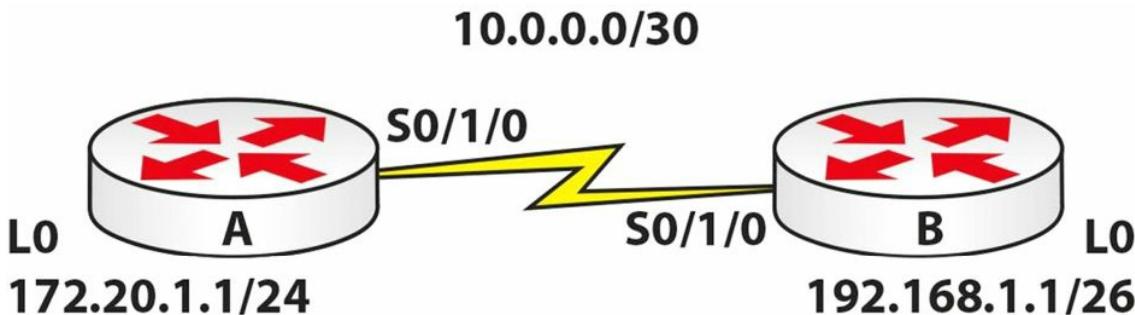
RouterB#telnet 10.0.0.1
Trying 10.0.0.1 ...
% Connection timed out; remote host not responding
RouterB#telnet 172.20.1.1
Trying 172.20.1.1 ...Open
User Access Verification -password won't show when you type it
Password:
RouterA> ^Hit Control+Shift+6 together and then let go and press the X key to quit.

```

**注意：** 我们会在其它实验中涉及ACLs，但你真的需要完全地掌握这些内容。为此，要尝试其它的TCP端口，比如 80、25 等等。另外，要试试那些UDP端口，比如 53。如没有将一台PC接上路由器，则是无法对这些其它端口进行测试的。

## 命名ACL实验

### 拓扑图



命名ACL实验拓扑图

### 实验目的

学习如何配置一条命名ACL。

### 实验步骤

- 配置上面的网络。在两台路由器上加入一条静态路由，领导到任何网络的任何流量都从串行接口发出。这么做的原因是，尽管这不是一个路由实验，仍然需要路由的流量。

```
RouterA(config)#ip route 0.0.0.0 0.0.0.0 s0/1/0
RouterB(config)#ip route 0.0.0.0 0.0.0.0 s0/1/0
```

- 在路由器 B 上加入一条扩展的命名ACL。只放行主机 172.20.1.1，阻止其它任何主机或网络。

```
RouterB(config)#ip access-list extended blockping
RouterB(config-ext-nacl)#permit icmp host 172.20.1.1 any
RouterB(config-ext-nacl)#exit
RouterB(config)#int s0/1/0
RouterB(config-if)#ip access-group blockping in
RouterB(config-if)#
```

- 现在分别从路由器 A 的串行接口和换回接口发出 ping 来测试该条ACL。

```
RouterA#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:
UUUUU
Success rate is 0 percent (0/5)
RouterA#ping
Protocol [ip]:
Target IP address: 192.168.1.1
Repeat count [5]:
Datagram size [100]:
Timeout in seconds [2]:
Extended commands [n]: y
Source address or interface: 172.20.1.1
Type of service [0]:
Set DF bit in IP header? [no]:
Validate reply data? [no]:
Data pattern [0xABCD]:
Loose, Strict, Record, Timestamp, Verbose[none]:
Sweep range of sizes [n]:
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
Packet sent with a source address of 172.20.1.1
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 31/34/47 ms
```

**注意：**你需要搞清楚各种服务，以及各种服务所用到的端口。否则，要配置ACL就会非常棘手。本条ACL相当简单，因此可以仅用一行完成。在有着路由协议运行时，需要放行它们。

要放行RIP，就要像这样指定。

```
access-list 101 permit udp any any eq rip
```

要放行OSPF，要像这样指定。

```
access-list 101 permit ospf any any
```

要放行EIGRP，要像这样指定。

```
access-list 101 permit eigrp any any
```



## 第10天 路由的一些概念

### Routing Concepts

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第10天的任务

- 阅读今天的课文
- 回顾昨天的课文
- 完成今天的实验
- 阅读ICND1记诵指南
- 在网站[subnetting.org](http://subnetting.org)上花15分钟

ICND1考试要求你对**基本路由** (basic routing) 及**数据包流经某个网络的过程**(packet flow accross a network)，有所掌握。我们也会对**各种路由协议背后的技术有所了解**(take a look at the technology behind routing protocols)。

今天将会学到以下知识。

- 路由基础知识, basic routing
- 各种有类和无类协议, classful and classless protocols
- 路由协议的分类, routing protocol classes

本模块对应了CCNA大纲要求的以下方面。

- 描述基本路由的一些概念
  - CEF
  - 包转发, packet forwarding
  - 寻获路由器的过程, router lookup process
- 区分不同路由和路由协议的方式
  - 链路状态对距离矢量, Link State vs. Distance Vector
  - 下一跳, next hop
  - IP路由表, IP routing table
  - 被动接口(它们的工作方式), passive interfaces (how they work)

## 路由基础知识

### Basic Routing

路由协议的角色，一是动态地学习其它网络，二是与其它设备交换路由信息，三就是连接上内部和/或外部网络。

务必要清楚，路由协议不会跨越网络发送数据包。它们是用来确定路由的最佳路径（their role is to determine the best path for routing）。受路由的那些协议（routed protocols）才真正发出数据，而一个最常见的受路由协议实例，就是IP。

不同路由协议采用不同方式来确定到某个网络或网络节点的最优路径。一些类型的路由协议，在静态环境或者说几乎没有变化的环境中运行最好，但却在这些环境发生变化后，需要很长时间进行收敛（converge）。另一些协议，则能够对网络中发生的变化迅速反应而能快速地进行收敛。

当网络中所有路由器有着同样的视图（view）并对那些最优路由达成一致时，就实现了网络收敛（network convergence）。在要很长时间才能实现收敛时，将会发生远端网络之间间歇性的丢包及连通性丢失。除了这些问题之外，慢速的收敛还会导致网络路由循环（network routing loops）及完全的网络中断（outright network outages）。所用到的路由协议算法确定了收敛情况。

因为这些路由协议有着不同特征，而在其各自的伸缩性（scalability）和性能上有所不同。一些路由协议适合于小型网络，而其它协议则既可用于小型、中型网络，又可在大型网络中使用。

## 包转发

### Packet Forwarding

包转发涉及两个过程。

- 确定最优路径, determining the best path
- 发出数据包（交换）， sending the packet(switching)

当路由器接收到一个发往其直接连接网络的数据包时，该路由器就检查其路由表并将该数据包转发到那个网络，如图10.1所示。

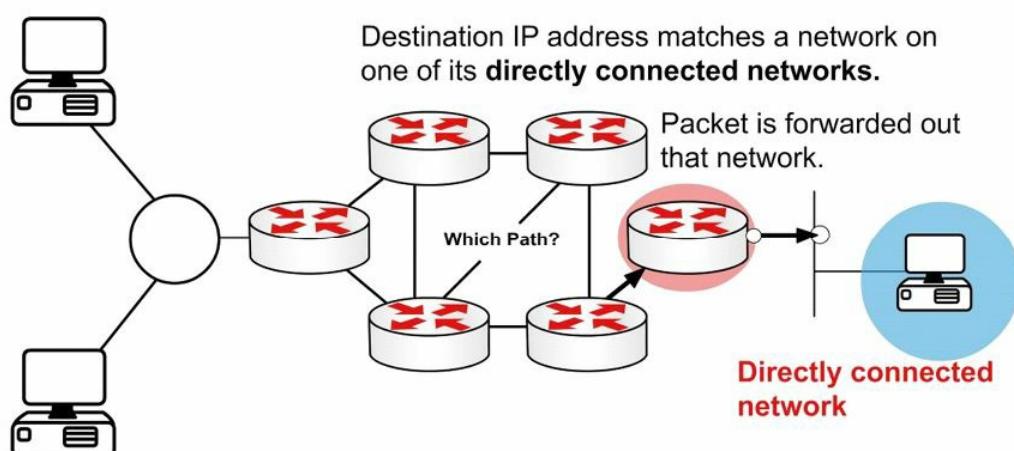


图10.1 -- 直连网络

如数据包的目的地是一个远端网络，就会检查路由表，如果有一条路由或默认路由，那么就转发数据包到下一跳路由器，如下图10.2所示。

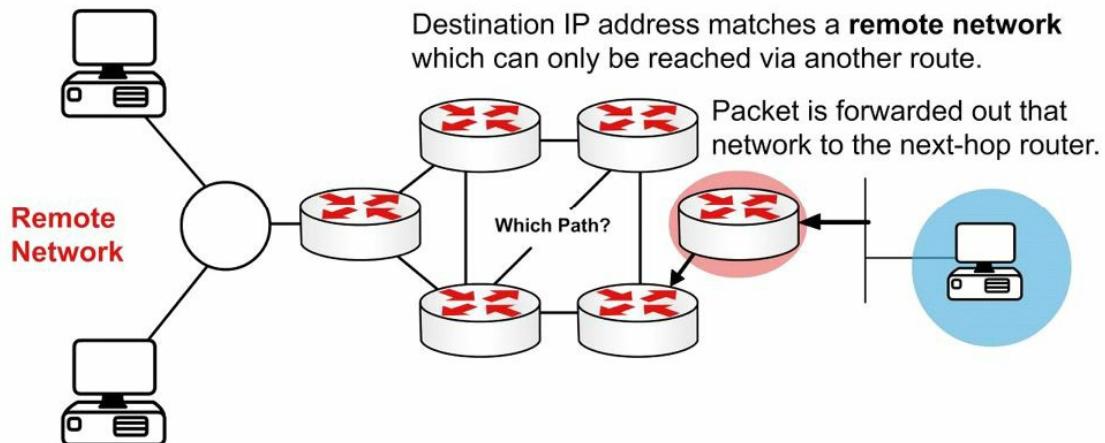


图10.2 -- 远端网络

如数据包以一个不在路由表中的网络为目的地，且又不存在默认路由，那么该数据包就不丢弃，如下图10.3 所示。

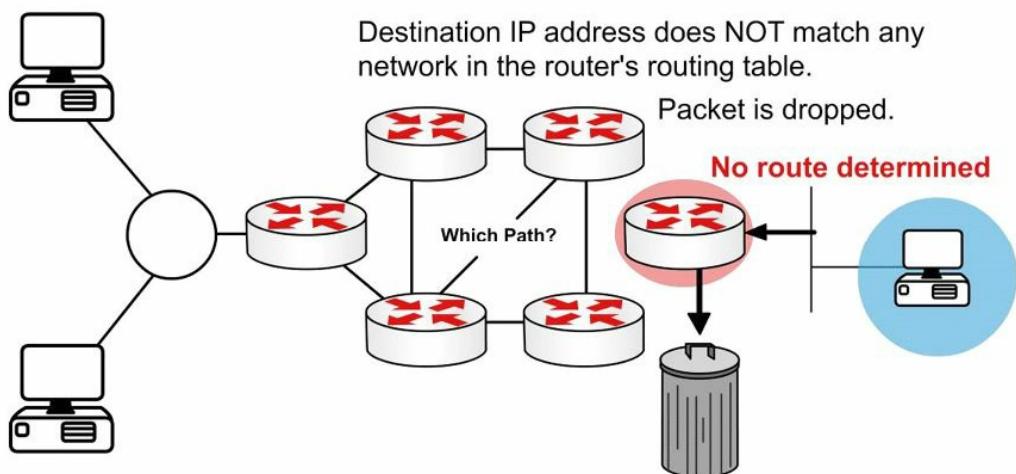


图10.3 -- 没有路由

**交换过程 (the switching process)** 允许路由器通过一个接口接收数据包，并从另一接口发出。同时路由器也会以外出链路的适当数据链路帧方式，对数据包进行封装。

可能会要求你对自一个网络接收，并以另一个网络为目的地的数据包所发生的事情进行解释。首先，路由器通过移除二层帧的头部和尾部，实现三层数据包的解封装；接着，路由器查看该IP数据包的目的IP地址，以找出路由表中的最佳路径；最后，路由器将该三层数据包封装为一个新的二层帧，并将该帧从离开接口转发出去，那么**封装方式就可能从以太网变为HDLC**。此过程在下图10.4中进行了演示。

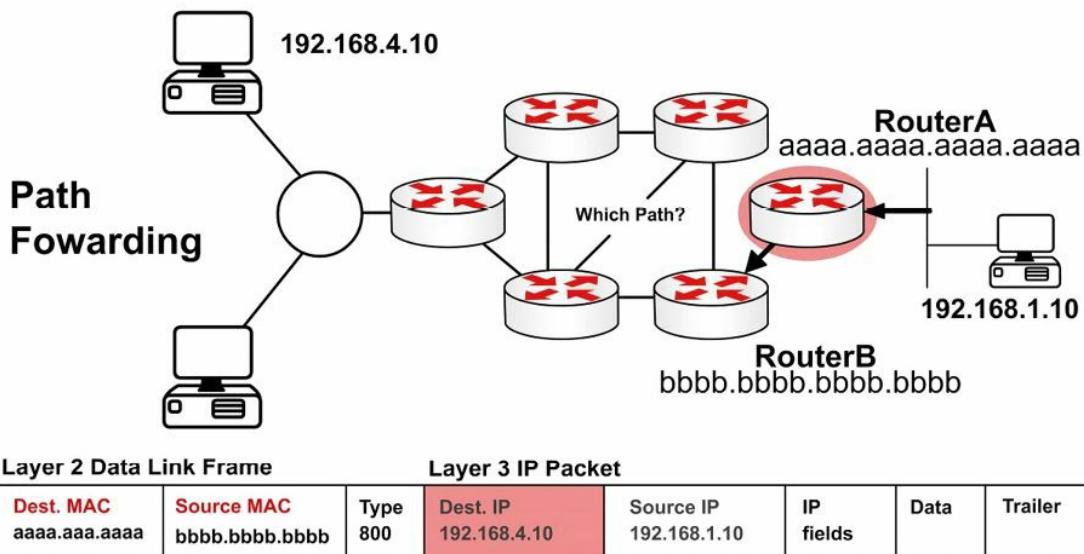


图10.4 -- 某数据包的三层地址

记住在较早的模块中曾提到，当数据包往其最终目的漫游时，源和目的IP地址绝不会变化。而MAC地址则会改变，以允许在那些中间设备之间进行传输。这在下图10.5中有演示。

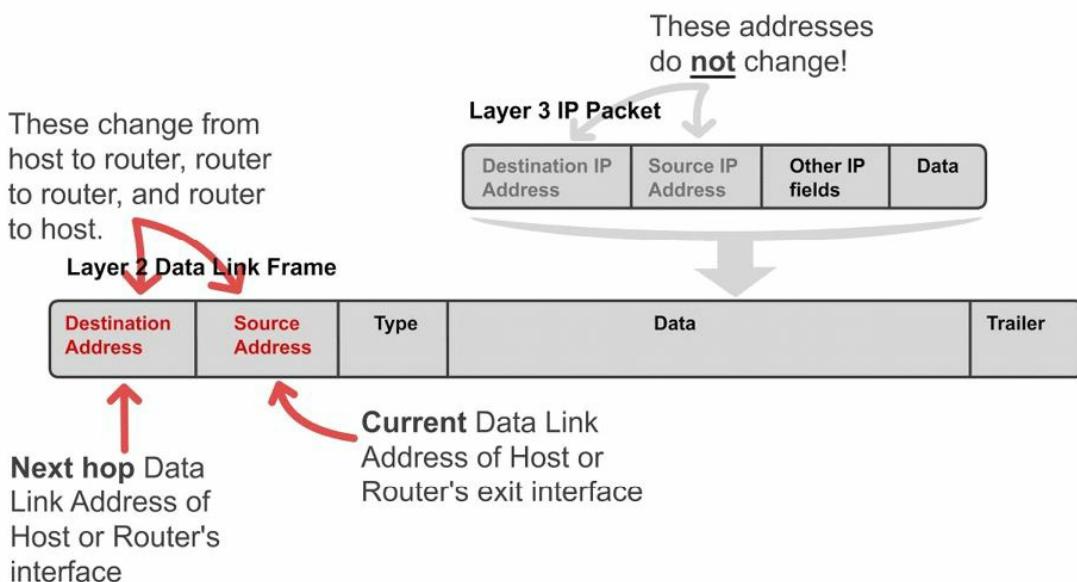


图10.5 -- 二层地址改变

图10.6展示了从主机 X 离开，前往主机 Y 的数据包。注意其下一跳MAC地址属于路由器 A（采用了代理ARP）；但其IP地址则是属于主机 Y。在帧到达路由器 B 时，以太网头部和尾部将换成WAN协议的头部和尾部。

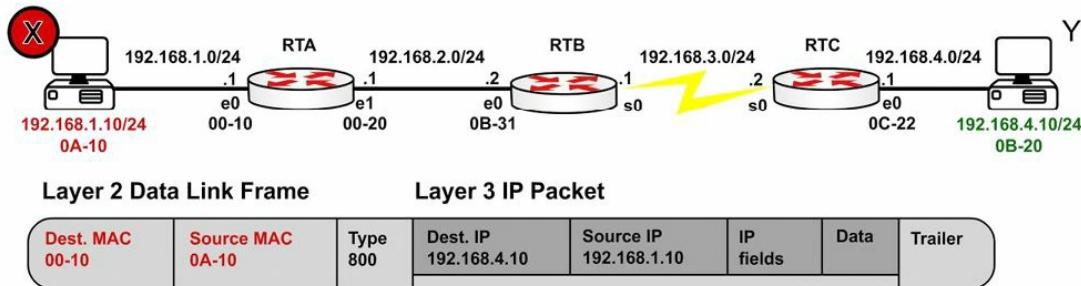


图10.6 -- 离开主机X的数据包

图10.7展示了离开路由器 A 前往路由器 B 的同样数据包。这里有有着一次路由查找，接着数据包就被从接口 E1 交换出去(there is a route lookup and then the packet is switched out of interface E1 )。类型 800 ( Type 800 ) 表明该数据包是一个IPv4数据包。

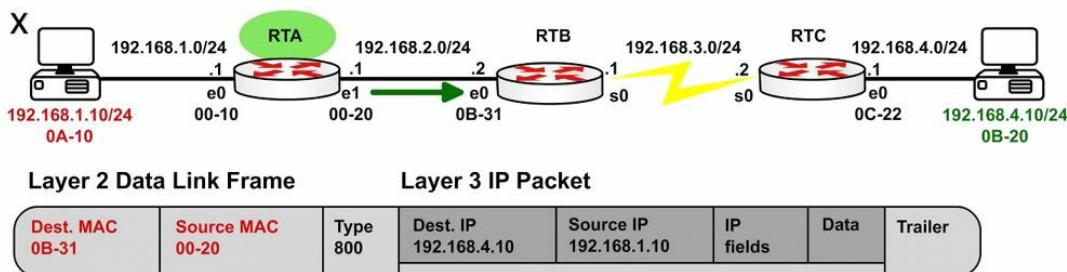


图10.7 -- 离开路由器A的数据包

图10.8展示了该帧最终到达路由器 C 并被转发给主机 Y。

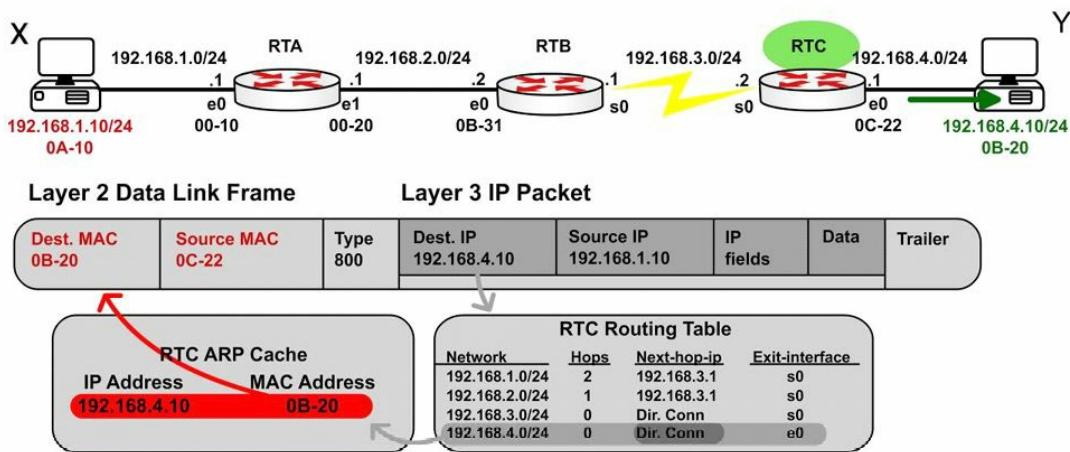


图10.8 -- 离开路由器 c 的数据包

## 互联网协议路由基础知识

### Internet Protocol Routing Fundamentals

正是某种路由协议，才令到路由器实现如何达到其它网络的动态学习。路由协议还令到路由器实现与其它路由器或主机，就学习到的网络信息进行交换。这些路由协议可用于连接内部园区网络（connecting interior/internal campus networks），也用于连接不同企业或路由域（connecting different enterprises or

**routing domains**）。因此，不光要掌握这些路由协议的复杂之处，还要牢固掌握何时在何种情况下要用这种路由协议，而不用另一种的原因。

## 平坦及层次化路由算法

### Flat and Hierarchical Routing Algorithms

路由协议算法要么以平坦路由系统运作，要么就以层次化路由系统运作（routing protocol algorithms operate using either a flat routing system or a hierarchical routing system）。层次化路由系统在路由器纳入到被称作域、区域或自治系统的逻辑分组中时，采用的是层次化方法（a hierarchical routing system uses a layered approach wherein routers are placed in logical groupings referred to as domains, areas, or autonomous systems）。这样做允许网络中的不同路由器完成各自特定任务，从而优化在这些层上完成的功能。层次化系统中的一些路由器可与其它域或区域的路由器通信，而其他路由器只能与同一域或区域中的路由器进行通信。这样做可以减少该域中路由器必须处理信息的数量，从而实现网络内的快速收敛。

平坦路由系统没有层次。在此类系统中，路由器一般都要连接到网络中的其它所有路由器，且每台路由器基本上都有着同样的功能。在甚小型网络中，此类算法可以工作得很好；但是，这些算法不是可伸缩的。此外，伴随网络增长，故障排除就变得更为棘手，因为比如原本只需努力解决确切的几个区域的问题，现在却不得不面对整个网络的问题。

由层次化路由系统所带来的主要优势，就是这类系统的可伸缩性。层次化路由系统还令到对网络改变十分容易，这和包含了核心、分布和接入层的传统层次化网络设计带来的优势十分相似。此外，层次化算法可用于在网络的一些区域减少路由更新流量，并减小路由表大小，同时仍然保证完整的网络连通性。

## IP分址和地址汇总

### IP Addressing and Address Summarisation

一个IP地址是分作两部分的。第一部分指明了网络地址，而第二部分指明的是主机地址。在设计某个网络时，就会用到某种IP分址方案，来将网络中的主机及设备进行唯一区分。该IP分址方案应是层次化的，且应建立在传统的逻辑层次化模型上。这样做就能实现该分址方案于网络中提供出一些指定点位，在这些点位完成有效的路由汇总。

汇总（summarisation）减少路由器所必须处理信息的数量，以此就可以实现网络的快速收敛。汇总还通过隐藏掉网络中某些区域的详细拓扑信息，从而令到因网络发生改变而受影响区域的大小受限。此概念在下图10.9中进行了演示。

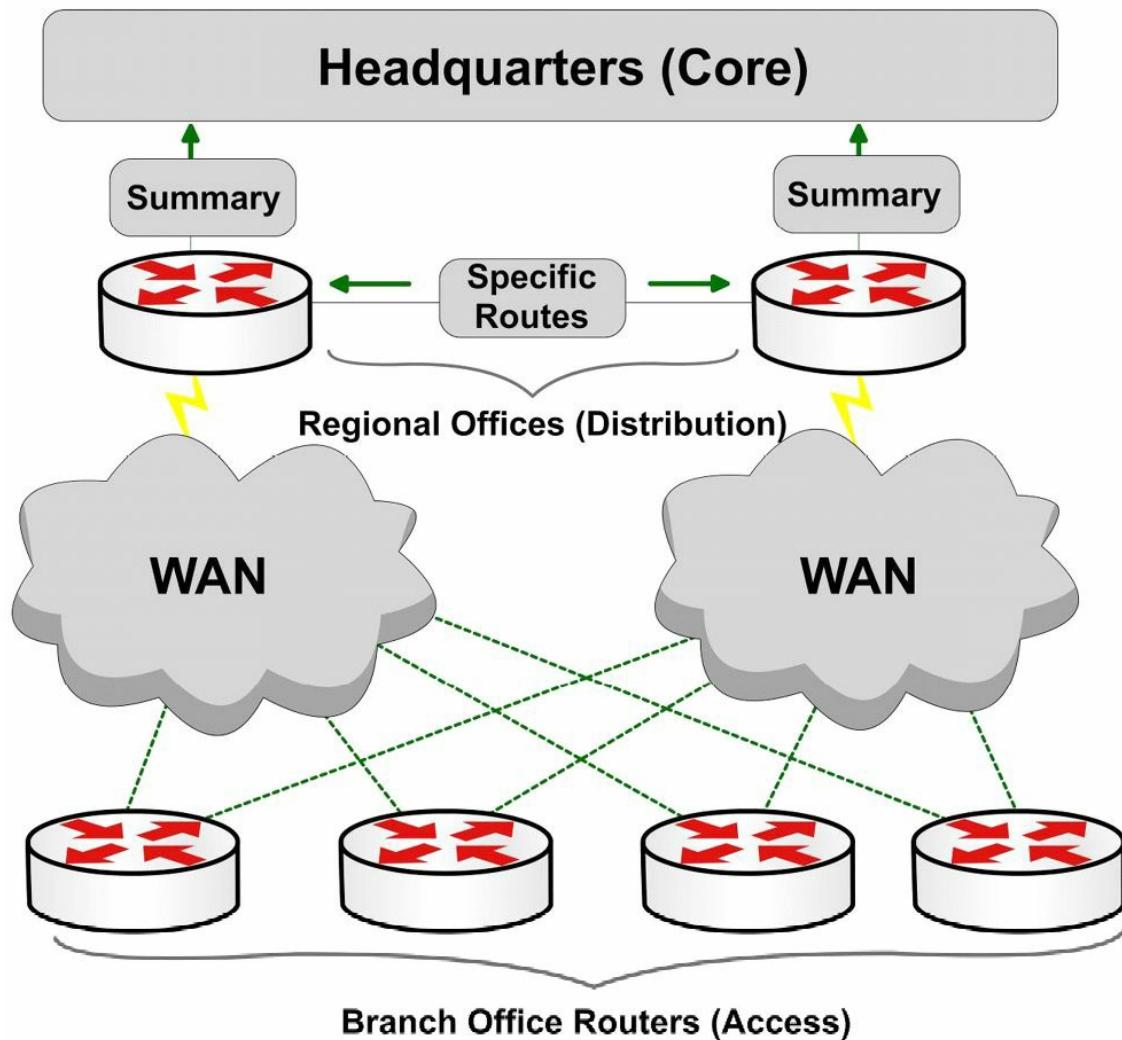


图10.9 -- 采用思科设计模型的路由汇总

通过图10.9可以看出，那些分支局（接入层）到地区局路由器（分布层）都是双线路接入（the branch offices(Access Layer) are dual-homed to the regional office routers(Distribution Layer)）。这些层都是采用思科设计模型（Cisco design models）定义出来的。采用某种层次化分址方案，就令到分布层路由器将仅一条的那些分支局子网的汇总路由，通告给核心层。这样做保护了核心层免受任何位处分布层和接入层之间路由器路由抖动的影响，因为除非某条汇总路由所继承自的那些更为具体前缀都从路由表中移除，该条汇总路由是不会抖动的（this protects the Core Layer from the effects of any route flapping between the Distribution and Access Layer routers, because a summary route will not flap until every last one of the more specific prefixes from which it is derived is removed from the routing table）。这又带来了该区域稳定性的提升。此外，核心层路由表大小得以极大地减小。

## 管理距离

### Administrative Distance

管理距离用于决定一个路由信息来源对另一个的可靠性（administrative distance is used to determine the reliability of one source of routing information from another）。一些路由信息来源被认为相较其它源更为可靠；那么，当自两种或更多不同路由协议得出两种或更多到同一目的的路径时，管理距离就可用于决定到某个目的网络或网络节点的最优或首选路径。

在思科IOS软件中，所有路由信息来源都分配了一个默认管理距离值。该默认值是一个 0 到 255 之间的整数，其中值 0 分配给最可靠的路由信息来源，值 255 分配给最不可靠的来源。任何分配了管理距离值 255 的路由，都被认为是不受信任的，且不会被放入到路由表中。

**管理距离是一个仅影响本地路由器的本地有意义值。**该值不会在路由域中传播 (this value is not propagated throughout the routing domain)。因此，对一台路由器上某个或某些路由来源默认管理距离的修改，仅影响那台路由器对路由信息来源的选用。表10.1展示了思科IOS软件中所用到的默认管理值（考试要求掌握这些值）。

**表10.1 -- 路由器管理距离**

**Router Administrative Distances(ADs)**

路由来源	管理距离 (AD)
连接的接口，Connected Interfaces	`0`
静态路由，Static Routes	`1`
增强内部网关路由协议汇总路由，Enhanced Interior Gateway Routing Protocol(EIGRP) Summary Routes	`5`
外部边界网关协议路由，External Border Gateway Protocol(eBGP) Routes	`20`
内部的增强内部网关路由协议路由，Internal Enhanced Interior Gateway Routing Protocol(EIGRP) Routes	`90`
开放最短路径优先的内部和外部路由，Open Shortest Path First(OSPF) Internal and External Routes	`110`
中间系统到中间系统的内部和外部路由，Intermediate System-to-Intermediate System(IS-IS) Internal and External Routes	`115`
路由信息协议路由，Routing Information Protocol(RIP) Routes	`120`
外部网关协议路由，Exterior Gateway Protocol(EGP) Routes	`140`
按需路由的路由，On-Demand Routing(ODR) Routes	`160`
外部的增强内部网关路由协议路由，External Enhanced Interior Gateway Routing Protocol(EIGRP) Routes	`170`
内部的边界网关协议路由，Internal Border Gateway Protocol(iBGP) Routes	`200`
不可达或未知路由，Unreachable or Unknown Routes	`255`

默认路由来源管理距离会显示在 `show ip protocols` 命令的输出中。下面的输出演示了这点。

```
R1#show ip protocols
Routing Protocol is "isis"
  Invalid after 0 seconds, hold down 0, flushed after 0
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Redistributing: isis
  Address Summarization:
    None
  Maximum path: 4
  Routing for Networks:
    Serial0/0
  Routing Information Sources:
    Gateway          Distance     Last Update
    10.0.0.2          115          00:06:53
  Distance: (default is 115 )
```

## 路由度量值

### Routing Metrics

各种路由协议算法都会用到度量值，度量值是一个关联到特定路由的数值（routing protocol algorithms use metrics, which are numerical values that are associated with specific values）。使用这些数值来从路由协议学习到的路径中，从最优先到最不优先的顺序，进行优先选用（these values are used to prioritise or prefer routes learned by the routing protocol, from the most preferred to the least preferred）。本质上具有较低路由度量值的路由，就是该路由协议的较高优先顺序的路由。具有最低度量值的路由，通常就是到目的网络代价最小，或者说最优的路由。该条路由将被放入到路由表，并被用于将数据包转发到目的网络。

不同路由算法用到不同变量来计算路由度量值。一些路由算法仅用到一个变量，而其它先进路由算法会用到多于一个变量来决定某条特定路由的度量值。多数情况下，由一种路由协议计算出的度量值，是不兼容于那些由另一种路由协议所使用的度量值的。不同路由协议的度量值都会基于下面变量的一种或几种。

- 带宽, bandwidth
- 开销, cost
- 延迟, delay
- 负载, Load
- 路径长度, path length
- 可靠性, reliability

### 带宽

带宽一词，指的是在给定时间内，从一点往另一点可以传输数据的数量。一些路由算法会用到带宽来决定何种链路类型较其它类型更为首选。比如，某种路由协议会首选 GigabitEthernet 而不是 FastEthernet，因为前者比起后者有着提升了的容量。

在思科IOS软件中，接口配置命令 `bandwidth` 可用于修改某个接口的默认带宽值，从而有效地操纵某种路由协议选择一个接口而不是另一个。比如，在用接口配置命令 `bandwidth 1000000` 将 FastEthernet 接口进行配置后，那么 FastEthernet 和 GigabitEthernet 二者在路由协议看起来就具有了相同的传输容量，而会分配到同样的度量值。其中一条链路仍然是 FastEthernet 链路，而另一条是 GigabitEthernet 的事实，与路由协议不相关。

从一名网络管理员的角度看，重要的是理解 `bandwidth` 命令不会影响接口的物理容量（因此该命令又是被成为一个道具命令(a cosmetic command)）。也就是说，在 FastEthernet 接口上配置了更高的带宽，并不意味着其就具备了支持 GigabitEthernet 速率的能力。**开放路径优先 (OSPF) 和增强内部网关路由协议 (EIGRP) 都在度量值计算中用到了带宽变量。**

## 成本

### Cost

在涉及路由算法时，成本指的是通信成本（the cost, as it pertains to routing algorithms, refers to communication cost）。比如，某公司选择按传输的数据、或按使用时间付费的私有链路，而不是公共链路，就会造成成本的使用。**中间系统到中间系统 (IS-IS) 路由协议支持一个可选的，度量链路使用成本的费用度量值**（an optional expense metric）。依据不同协议，配置成本会有所不同。

### 延迟

### Delay

延迟的类型有多种，所有的延迟又影响不同类型的流量。一般意义上的延迟，是指将一个数据包通过互联网络，从其源处移到目的处所需要的时间长度。在思科IOS软件中，接口延迟值以微秒（us）计算的。

通过接口配置命令 `delay` 来配置接口的延迟值。在配置接口延迟值时，重要的是记住**这样做并不会影响到流量**（又是一个道具命令）。比如，配置了一个 5000 的延迟值，并不意味着从该接口发出的流量将有一个额外的 5000us 延迟。下表10.2展示了思科IOS软件中常见接口的默认延迟值。

接口类型	延迟 (us)
`10Mbps Ethernet`	`1000`
`FastEthernet`	`100`
`GigabitEthernet`	`10`
`T1`串行线路	`20000`

EIGRP将接口延迟数值用作其度量值计算的部分。手动修改接口延迟值会造成EIGRP度量值的重新计算。

### 负载

### Load

负载对不同的人来说有不同的意思。例如，在一般计算术语中，负载是指某项计算资源，譬如CPU，当前的使用量。而在此处，负载是指某个特定路由器接口使用的比例（load, as it applies in this context, refers to the degree of use for a particular router interface）。接口上的负载是一个 255 的分数。比如，一个 255/255 的负载就表明该接口已完全饱和，而一个 128/255 的负载则表明该接口是 50% 饱和的。默认情况下，负载是按 5 分钟平均值计算的（真实世界中常使用接口配置命令 `load-interval 30` 将其修改为一个最小的 30s）。**接口负载值可用于EIGRP中的度量值计算。**

### 路径长度

### Path Length

路径长度度量值是自本地路由器到目的网络所经过路径的总长度。不同路由算法在表示该值时有着不同的形式。比如路由信息协议（Routing Information Protocol, RIP）对在本地路由器和目的网络之间的**路由器**进行计数（跳数，hops），并使用该跳数作为度量值，而边界网关协议（Border Gateway Protocol, BGP）则对在本地路由器和目的网络之间**所经过的自治系统**进行计数，并使用该自治系统数来选择最优路径。

### 可靠性

### Reliability

和负载一样，可靠性一词，也是依据其所在上下文的不同，有着不同的意义。在这里，除非另有说明，总是可以假定可靠性是指网络链路或接口的可靠性、可信任性。在思科IOS软件中，某条链路或某个接口的可靠性表示为一个 255 的分数。比如，一个 255/255 的可靠性值表明接口是 100% 可靠的。与接口负载类似，某接口的默认可靠性是以过去 5 分钟平均值进行计算的。

## 前缀匹配

### Prefix Matching

思科路由器在决定使用位于路由表中的何条路由，来将流量转发到某个目的网络或节点时，采用的是最长前缀匹配规则（the longest prefix match rule）。在决定采用何条路由表条目来将流量路由至计划的目的网络或节点时，更长，或者说更具体的条目优先于像汇总地址那样的，不那么具体的条目。

最长前缀或最具体路由将用于路由流量到目的网络或节点，此时就会忽视该最具体路由来源的管理距离，如有多条经由同样路由协议学习到的重叠前缀，也甚至会忽视分配给该最长前缀的路由协议度量值。表10.3演示了某台将数据包发往地址 1.1.1.1 的路由器上路由选择的顺序。该顺序是基于最长前缀匹配查找的（this order is based on the longest prefix match lookup）。

**表10.3 -- 匹配最长前缀**

路由表条目	用到的顺序
`1.1.1.1/32`	第一
`1.1.1.0/24`	第二
`1.1.0.0/16`	第三
`1.0.0.0/8`	第四
`0.0.0.0/0`	第五

**注意：**尽管在表10.3中默认路由是位列路由选择顺序最后的，但要记住一条默认路由并非总是出现在路由表中的。如路由表中没有默认路由，同时也没有到地址 1.1.1.1 的路由条目，那么路由器就会简单地丢弃到那个目的地的数据包。大多数情况下，路由器会发给源主机一条ICMP消息，告知其目的主机不可达。而一条默认路由就是用于将目的网络未在路由表中显式列出的数据包，导向默认路由。

## IP路由表的建立

### Building the IP Routing Table

如没有生成一张包含远端网络路由条目的路由表，或称之为路由信息库，路由器就不能将数据包转发到这些远端网络（without a populated routing table, or Routing Information Base(RIB), that contains entries for remote networks, routers will not be able to forward packets to those remote networks）。路由表可能包含了一些特定网络的条目或简单的一条默认路由。转发进程（the forwarding process）使用路由表中的信息将流量转发到目的网络或主机。路由表本身不会去实际转发流量。

思科路由器使用管理距离、路由协议度量值及前缀长度，来决定哪些路由要实际放入到路由表中，这就允许路由器建立其路由表。通过下面的一般步骤，建立起路由表。

1. 如路由表中当前不存在该路由条目，就将该条目加入到路由表。
2. 如某路由条目比起一条既有路由更为具体，就将其加入到路由表。应注意原较不具体的条目在路由表中仍有留存。

3. 如某路由条目与一条既有条目一样，但其是从一个更为首选的路由源处收到的，就用该新条目替换旧条目。
4. 如该路由条目与一条既有条目一样，又是从同一协议收到的，就做以下处理。
  - 如其比既有路由有着更高的度量值，就丢弃新路由；或
  - 如新路由的度量值更低，就替换既有路由；或
  - 新旧路由的度量值一样时，将两条路由用作负载均衡

默认情况下建立路由信息库，当路由器在决定哪些路由要放入路由表时，总会选用有着最低管理距离值的路由协议。比如，某台路由器收到经由**外部的EIGRP**、OSPF及内部BGP给出的 10.0.0.0/8 前缀时，OSPF的路由将被放入到路由表中。而在那条路由被移除，或是不再收到时，外部EIGRP路由将被放入路由表中。最后如果OSPF和外部EIGRP路由都不再出现时，内部BGP路由就被用到。

一旦路由已放入到路由表，默认情况下比起那些较不具体的路由，最为具体或有着最长匹配前缀的路由总是优先选用的。这在下面的实例中进行了演示，该实例展示了包含有 80.0.0.0/8 、 80.1.0.0/16 及 80.1.1.0/24 前缀路由条目的一个路由表。这三条路由前缀分别通过 EIGRP、OSPF及RIP路由协议接收到。

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route
Gateway of last resort is not set
R          80.1.1.0/24 [120/1] via 10.1.1.2, 00:00:04, Ethernet0/0.1
D          80.0.0.0/8 [90/281600] via 10.1.1.2, 00:02:02, Ethernet0/0.1
O E2      80.1.0.0/16 [110/20] via 10.1.1.2, 00:00:14, Ethernet0/0.1
```

从上面的输出看出，第一条路由是 80.1.1.0/24。该路由是经由RIP学习到的，因此有着默认的管理距离值120。第二条路由是 80.0.0.0/8。该路由是经由**内部的EIGRP**学习到的，因此有着默认管理距离值90。第三条路由是 80.1.0.0/16。该路由是通过OSPF学习到的，且是一条有着管理距离110的**外部的OSPF**路由。

**注意：**因为这些路由协议度量值各不相同，在有着来自不同协议的路由安装到路由表时，**这些度量值是在决定要使用的最佳路由时的非要素**。下面的部分将说明思科IOS软件是如何来建立路由表的。

基于该路由表的内容，如路由器收到一个目的为 80.1.1.1 的数据包，就会使用那条RIP路由，因为这是最具体的条目，尽管EIGRP和OSPF都有着更好的管理距离值而是更为优先的路由来源。`show ip route 80.1.1.1` 命令可用于检验这点。

```
R1#show ip route 80.1.1.1
Routing entry for 80.1.1.0/24
  Known via "rip", distance 120, metric 1
  Redistributing via rip
  Last update from 10.1.1.2 on Ethernet0/0.1, 00:00:15 ago
  Routing Descriptor Blocks:
    * 10.1.1.2, from 10.1.1.2, 00:00:15 ago, via Ethernet0/0.1
      Route metric is 1, traffic share count is 1
```

## 有类和无类协议

### Classful and Classless Protocols

有类协议无法使用VLSM（也就是RIPv1和IGRP，它们都已不在CCNA大纲中了）。这是因为它们不会去识别除了默认网络掩码外的其它任何东西。

```
Router#debug ip rip
RIP protocol debugging is on
01:26:59: RIP: sending v1 update to 255.255.255.255 via Loopback0
192.168.1.1
```

有类协议用到VLSM（也就是RIPv2和EIGRP）。

```
Router#debug ip rip
RIP protocol debugging is on
01:29:15: RIP: received v2 update from 172.16.1.2 on Serial0
01:29:15:192.168.2.0/24 via 0.0.0.0
```

## 被动接口

### Passive Interfaces

路由协议设计和配置的一个重要考虑，就是要限制不必要的对等传送（an important routing protocol design and configuration consideration is to limit unnecessary peerings），如下图10.10所示。这是通过使用被动接口实现的，被动接口可以阻止路由器在指定接口上形成路由邻接关系（routing adjacencies）。基于特定路由协议，此功能的使用会有所不同，但其做法通常有以下两类。

- 路由器不在被动接口上发出路由更新
- 路由器不在该接口上发送 Hello 数据包，这样做就不会形成邻居关系

被动接口通常能接收到路由更新或 Hello 数据包，但不允许发出任何种类的路由协议信息出去。

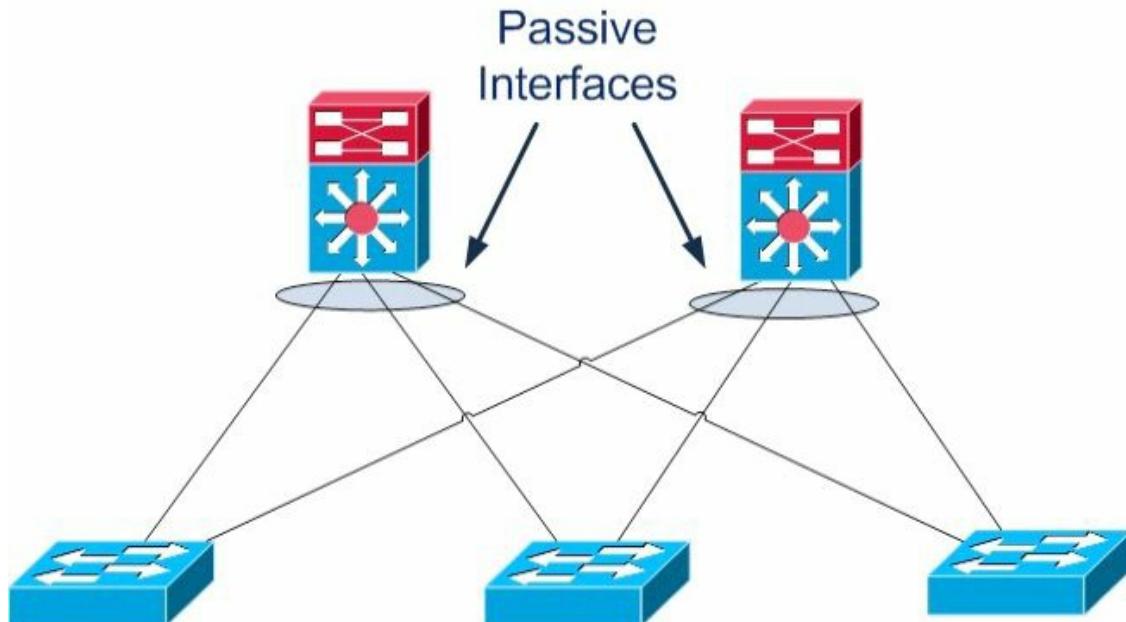


图10.10 -- 限制不必要的对等数据交换

被动接口的一个用例实例就是用于防止路由协议数据自分布层对等传送（peerings）到接入层，就如同上面的图10.10所示。当留有三层跨越这些性质不同的接入层交换机的对等传输时（也就是在跨越交换机区块的不同交换机上有着多台主机），就只会增加内存负载、路由协议更新开销及更多的复杂性。同样，如发生

了某条链路故障，流量会经由一台邻居接入交换机传输，而到达另一个VLAN成员处（by having Layer 3 switches across the different Access Layer switches(i.e., having multiple hosts on different switches across switch blocks) you are basically adding memory load, routing protocol update overhead, and more complexity. Also, if there is a link failure, the traffic may transit through a neighbouring Access Layer switch to get to another VLAN member）。

也就是说，你想要消除不必要的路由对等邻接（unnecessary routing peering adjacencies），那么就要将那些面向二层交换机的端口，配置为被动接口，以此来抑制路由更新通告（suppress routing updates advertisements）。如某台分布层交换机在这些接口之一上，一条都没有接收到来自一台潜在对等设备的路由更新，其就不必去处理这些更新，也就不会通过那个设备形成邻居邻接关系。达成此配置的命令通常就是路由进程配置模式（the Routing Process Configuration mode）中的 `passive-interface [interface number]` 命令。要获得更多有关思科设计模型（the Cisco design model）的信息，请阅读一份CCDA手册。

## 路由协议分类

### Routing Protocol Classes

路由协议有两大分类 -- **距离矢量**和**链路状态**(Distance Vector and Link State)。距离矢量路由协议在决定通过网络的最优路径（一条或多条）时，传统上使用一个一维矢量（a one-dimensional vector），而链路状态路由协议在决定通过网络的最优路径（一条或多条）时，使用最短路径优先（the Shortest Path First, SPF）。在深入探究路由协议的这两种类别的具体细节之前，我们先看看不同矢量，以及难以搞懂的SPF算法。

### 理解矢量

#### Understanding Vectors

一个一维矢量就是一个有方向的量。它就是一个在特定方向或路线上量（数字）。下图10.11演示了矢量这个概念。

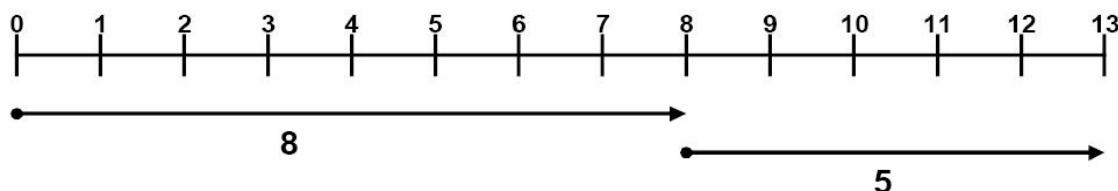


图10.11 -- 理解矢量

在图10.11中，第一条线路从 0 开始，到 9 结束，同时第二条线路从 8 开始，在 13 处结束。那么第一条线路的矢量就是 8，第二条的矢量就是 5。运用基础数学，就知道  $8+5=13$ 。矢量的开始点和结束点是无关的。而是与矢量长度及其经过的距离有关。

**注意：**矢量亦可朝相反的方向通过（也就是以负数表示的矢量）。

### 最短路径优先算法

#### The Shortest Path First Algorithms

最短路径优先算法通过路由器创建出到某个区域或网络骨干中的所有主机的最短路径树，该路由器处于树的根部，并由该路由器完成树的生成计算。为让SPF算法以正确的方式工作，区域中的所有路由器都要有同样的数据库信息。在OSPF中，数据库信息同步是通过数据库交换过程实现的（the SPF algorithm creates a shortest-path tree to all hosts in an area or in the network backbone with the router that is performing the

calculation at the root of that tree. In order for the SPF algorithm to work in the correct manner, all routers in the area should have the same database information. In OSPF, this is performed via the database exchange process)。

## 距离矢量路由协议

### Distance Vector Routing Protocols

距离矢量是一种用距离或跳数计数作为主要度量值，来决定最佳转发路径的路由协议。距离矢量路由协议又是主要建立在Bellman-Ford算法基础上。距离矢量路由协议发送其邻居路由器的完整路由表，以保持这些路由器有关网络状态的最新信息（Distance Vector routing protocols periodically send their neighbour routers copies of their entire routing tables to keep them up to date on the state of the network）。**在某个小型网络中这也许可以接受，而当网络增大时，通过网络发送的流量数量就会增长。**所有距离矢量路由协议都有着以下的特征。

- 计数到无限大，counting to infinity
- 水平分割，split horizon
- 反向投毒，poison reverse
- 保持计数器，hold-down timers

对计数到无穷的运用，如某个目的网络远于路由协议所允许的最大跳数，该网络就认为其是不可达的。该网络的路由条目因此就不会安装到IP路由表中。

水平分割特征指明路由信息再不能从其接收到的接口再发送出去。这样做就可以阻止路由信息再通告给学习到它的源路由器。尽管此特征是一种了不起的防止环回机制，但也有显著的不利之处，特别是在辐射状网络中。

反向投毒（或路由投毒）是水平分割的拓展。在与水平分割配合使用时，反向投毒可令到自某个接口上收到的网络，再从同样接口通告出去。但反向投毒造成路由器将某网络以“不可达”度量值，通告回发出该网络的路由器，那么收到这些条目的路由器就不会将这些条目再加入到其路由表中了。

保持计数器用于阻止那些先前通告的网络由于宕掉而被放回到路由表中（hold-down timers are used to prevent networks that were previously advertised as down from being placed back into the routing table）。在路由器收到一个某网络宕掉的更新时，就启动它的保持计数器。该计数器告诉路由器在接受任那个网络的状态变化之前，等待一段指定的时间。

在保持期间，路由器压制住该网络并阻止通告出无效信息（during the hold-down period, the router suppresses the network and prevents advertising false information）。就算路由器收到来自网络可达的其它路由器（它们可能没有收到网络宕掉的触发更新）的信息，也不会将数据包路由到该不可达网络。此机制设计用于阻止黑洞流量（the router also does not route to the unreachable network, even if it receives information from another router(that may not have received the triggered update)that the network is reachable. This mechanism is designed to prevent black-holing traffic）。

两个最常见的距离矢量协议，就是RIP和IGRP。**EIGRP是一个高级距离矢量协议**，距离矢量和链路状态两方面的特性在EIGRP中都有用到（也就是说，它是一个混合协议(hybrid protocol)）。

## 链路状态路由协议

### Link State Routing Protocols

链路状态路由协议属于层次化的路由协议，采用区域的概念（the concept of areas），在网络中对路由器进行分组。这样做令到链路状态协议比起距离矢量协议能够更好地伸缩并以一种效率更高的方式运行。运行链路状态路由协议的路由器，创建出一个包含了网络全部拓扑的数据库。这样做令到同一区域中的路由

器，都有着网络的同样视图（the same view of the network）。

由于网络中的路由器都有着网络的同样视图，用于在网络间转发数据包的路径就是最优的，且路由环回的可能性得以消除。因此，诸如水平分割和路由投毒这样的技巧对链路状态协议就不适用了，因为它们是用于距离矢量路由协议的。

链路状态路由协议通过发送链路状态通告（Link State Advertisement），或者说链路状态数据包（Link State Packets）给位于同一区域内其它路由器的方式运作。这些数据包包含了有关链路所连接的接口、链路度量值及链路其它变量的信息。随着路由器对这些信息的积累，它们就运行SPF算法并计算出到各台路由器及目的网络的最短（最佳）路径。使用接收到的链路状态信息，路由器建立其链路状态数据库（the Link State Database, LSDB）。在相邻两台路由器的LSDBs同步了时，就说它们形成了邻接关系。

与发送给其邻居的是它们的完整路由表的距离矢量路由协议不同，链路状态路由协议在探测到网络拓扑发生改变时，发送的是增量更新，这点令到链路状态路由协议在较大型的网络中效率更高。使用增量更新也令到链路状态路由协议对网络变化的响应更为迅速，因此比起距离矢量路由协议有着更短的收敛时间。表10.4列出了不同的内部网关协议及其所属类别。

**表10.4 -- IGP类别**

协议名称	有类/无类	协议类别
RIP(版本1)	有类	距离矢量
IGRP	有类	距离矢量
RIP (版本2)	无类	距离矢量
EIGRP	无类	高级距离矢量
IS-IS	无类	链路状态
OSPF	无类	链路状态

## 路由协议的各种目标

### The Objectives of Routing Protocols

这些路由算法尽管生来就有所不同，但都有着同样的基本目标。虽然一些算法好于其它一些，但所有路由协议都有其优势和不足。这些路由算法的设计，都有着下面这些目标和目的。

- 最优路由, optimal routing
- 稳定性, stability
- 易于使用, easy of use
- 灵活性, flexibility
- 快速收敛, rapid of convergence

## 最优路由

### Optimal Routing

所有路由协议的主要目标之一，就是选择通过网络从源子网或主机到目的子网或主机的最优路径。最优路由依据就是这些路由协议所使用的度量值。一种协议所认为的最优路由，并不一定也是从另一协议角度看的最优路由。比如，RIP可能认为一条仅有两跳长的路径是到某个目的网络的最优路径，尽管这些链路都

是 64Kbps，而诸如OSPF和EIGRP那样的先进协议则会到相同目的的最优路径是经过了 4 台路由器却有着 10Gbps 速率的链路。

## 稳定性

### Stability

网络的稳定与否，是这些路由算法的另一个主要目标。路由算法应足够稳定，以容许无法遇见的网络事件（to accommodate unforeseen network events），比如硬件故障甚至错误实现等。尽管这是一个所有路由算法的典型特征，但由于它们对这些事件应对的方式和所用时间，令到一些算法相对其它算法做得更好，因此在现今网络中用到更多。

## 易于使用

### Ease of Use

路由算法被设计得尽可能简单。除了要提供对复杂互联网络部署的支持能力，路由协议还应考虑运行其算法所需的计算资源问题。一些路由算法比起其它算法需要更多的硬件和软件资源（比如CPU和内存）来运行；但它们却能够提供比其它替代的简单算法更多的功能。

## 灵活性

### Flexibility

除了提供路由功能外，路由算法还应富于特性，从而令到这些算法支持在不同网络中遇到的不同需求。但需注意此能力是以诸如下面即将说到的收敛等其它特性为代价的。

## 快速收敛

### Rapid Convergence

快速收敛又是所有路由算法的另一主要目标。如早前指出的那样，当网络中的所有路由器都有着同样视图且对最优路由达成一致时，就出现了收敛。在需要长时间才能收敛时，远端网络之间就会出现间歇性的包丢失及失去连通性。除了这些问题外，慢速收敛还会导致路由环回和完全的网络中断。

## 路由故障避开机制

### Routing Problems Avoidance Mechanisms

距离矢量路由协议因其过于简单的“依据传言的路由”方法，而容易造成大问题（it is a known fact that Distance Vector routing protocols are prone to major problems as a result of their simplistic "routing by rumor" approach）。距离矢量和链路中台协议采用不同方法来防止路由故障。有下面这些最为重要的机制。

- **无效计数器**, invalidation timers: 在很长时间内都没有收到一些路由的更新时，这些计数器被用于将这些路由标记为不可达。
- **跳数限制**, hop count limit: 当一些路由的跳数，比预先定义的跳数限制还多时，此参数就将这些路由标记为不可达。RIP的跳数限制是15，而大型网络通常不会使用RIP。不可达路由不会作为最佳路由安装到路由表中。跳数限制防止网络中的环回更新，就想IP头部的TTL字段一样。
- **触发的更新**, triggered updates: 此特性允许有重要更新时对更新计数器进行旁路、忽视。比如，在有一个重要的路由更新必须要在网络中宣传时，就可以忽略RIP的 30 秒计数器。

- **保持计数器**, hold-down timers: 如某条特定路由的度量值持续变差, 那条路由的更新就会在一个延迟时期内不被接受了。
- **异步的更新**, asynchronous updates: 异步更新代表另一种防止网络上的路由器, 在同一时间其全部路由信息被冲掉的安全机制。在前面提到, OSPF每30分钟执行一次异步更新。异步更新机制为每台设备生成一个小小的延时, 因此这些设备不会准确地在同一时间信息全被冲掉。这样做可以改进带宽的使用以及处理能力。
- **路由投毒**, route poisoning: 此特性防止路由器通过已为无效的路由发送数据。距离矢量协议使用这个特性表明某条路由不再可达。路由投毒是通过将该路由的度量值设置为最大值完成的。
- **水平分割**, split horizon: 水平分割防止路由更新再从收到的接口上发送出去, 因为在那个区域中的路由器应该已经知道了那条特定路由了。
- **反向投毒**, poison reverse: 该机制是因被投毒路由而造成的水平分割的一个例外。

## 基于拓扑的思科快速转发交换

### Topology-Based(CEF, Cisco Express Forwarding) Switching

将某数据包预期的目的地址与IP路由表进行匹配, 需要使用一些路由器的CPU运算周期。企业路由器可能有着数十万的路由条目, 并能对同样数量的数据包与这些条目进行匹配。在尝试以尽可能高的效率来完成这个过程中, 思科构建出了各种不同的交换方法 (various switching methods)。第一种叫做进程交换 (process switching), 而它用到路由查找及已分级的最佳匹配方法 (uses the route lookup and best match already outlined)。此方式又在快速交换 (fast switching) 之上进行了改进。路由器生成的最近转发数据包IP地址清单, 连同IP地址匹配下的数据链路层地址会被复制下来。作为对快速转发的改进, 思科快速转发 (Cisco Express Forwarding, CEF) 技术得以构建。当下思科路由器的所有型号, 默认运行的都是 CEF。

## 思科快速转发

### Cisco Express Forwarding(CEF)

CEF运行于数据面 (the data plane), 是一种拓扑驱动的专有交换机制 (a topology-driven proprietary switching mechanism), 创建出捆绑到路由表 (也就是控制面, the control plane) 的转发表。开发CEF是为消除因基于数据流交换中用到的, 进程交换的首个数据包查找方法出现的性能问题 (CEF was developed to eliminate the performance penalty experienced due to the first-packet process-switched lookup method used by flow-based switching)。CEF通过允许为基于硬件的三层路由引擎用到的路由缓存, 在接收到某个传输流的任何数据包之前, 将所有三层交换所需的必要信息, 包含到硬件当中。照惯例保存在路由缓存中的信息, 现在是保存在CEF交换的两个数据结构中。这两个数据结构提供了高效率包转发的优化查找, 它们分别成为FIB (Forwarding Information Base, 转发信息库) 和邻居表。

**注意:** 重要的是记住就算有了CEF, 在路由表发生变化时, CEF转发表同样会更新。在新的CEF条目创建过程中, 数据包会在一个较慢的交换路径中, 使用比如进程交换的方式, 进行交换。所有当前的思科路由器型号及当前的IOS都使用CEF。

## 转发信息库

### Forwarding Information Base(FIB)

CEF使用一个FIB来做出基于IP目的地址前缀的交换决定 (CEF uses a FIB to make IP destination prefix-based switching decisions)。FIB在概念上与路由表或信息库是相似的。FIB维护着包含在IP路由表中的转发信息的一个镜像。也就是说, FIB包含了来自路由表中的所有IP前缀。

当网络中的路由或拓扑发生改变时，IP路由表就会被更新，同时这些变化在FIB中也会反映出来。FIB维护着建立在IP路由表中信息上的下一跳地址信息。因为在FIB条目和路由表条目之间有着一一对应关系，FIB就包含了所有已知路由，并消除了在诸如快速交换方式和最优交换（optimum switching）方式中于交换路径（switching paths）有关的路由缓存维护需求。

此外，因为FIB查找表中包含了所有存在于路由表中的已知路由，FIB就消除了路由缓存维护，以及快速交换和进程交换的转发场景。这样做令到CEF比典型的demand-caching方案要更为高效地交换流量。

## 邻接表

### The Adjacency Table

创建邻接关系表来包含所有直连的下一跳。邻接节点就是只有一跳的节点（也就是直接连接的）。在发现邻接关系后，就生成了该邻接关系表。一旦某个邻居成为邻接关系，将用于到达那个邻居的一个叫作MAC字串或MAC重写（a MAC string or a MAC rewrite）的数据链路层头部，就被创建出来并存入到邻接表中。在以太网段上，头部信息依次包含了目的MAC地址、源MAC地址以及以太网类型（EtherType）。

而一旦某条路由得到解析，就会指向到一个邻接的下一跳。如在邻接表中找到了某个邻接，那么一个指向该适当邻接的指针就在FIB条目中进行缓存（as soon as a route is resolved, it points to an adjacent next hop. If an adjacency is found in the adjacency table, a pointer to the appropriate adjacency is cached in the FIB element）。而如果存在到某个同样目的网络的多条路径，则指向每条邻接的所有指针就会被加入到load-sharing结构体中，这样做可以实现负载均衡。当多条前缀加入到FIB时，那些需要例外处理的前缀，会以特别邻接关系进行缓存。

## 加速的及分布式CEF

### Accelerated and Distributed CEF

默认下，所有基于CEF技术的思科Catalyst交换机都使用一个中心化三层交换引擎（a central Layer 3 switching engine），在那里由单独的处理器对交换机中所有端口上接收到的流量，做出全部的三层交换决定。尽管思科Catalyst交换机中用到的三层交换引擎提供了高性能，但在某些网络中，即便使用单独的三层交换引擎来完成所有三层交换，仍然不能提供足够的性能。为解决这个问题，思科 Catalyst 6500 系列交换机允许通过使用特别的转发硬件对CEF进行优化（to address this issue, Cisco Catalyst 6500 series switches allow for CEF optimisation through the use of specialised forwarding hardware）。CEF优化有两种实现方式，加速的CEF或分布式CEF。

加速的CEF允许让FIB的一个部分分布到 Catalyst 6500 交换机中的具备此功能的线路卡模块上去（Accelerated CEF allows a portion of the FIB to be distributed to capable line card modules in the Catalyst 6500 switch）。这样做令到转发决定在本地线路卡上使用本地存储的缩小的CEF表做出。假如有FIB条目在缓存中没有找到，就会向三层交换引擎发出需要更多FIB信息的请求。

分布式CEF指的是使用分布在安装于机架上的多块线路卡上的多个CEF表。在应用dCEF时，三层交换引擎（MSFC）维护着路由表并生成FIB，FIB将被所有线路卡动态完整下载，令到多个三层数据面（multiple Layer 3 data plane）同时运行。

总体上说，dCEF和aCEF都是用到多个三层交换引擎的技术，这样就实现了多个三层交换操作同时并行运作，从而提升整体系统性能。CEF技术提供以下好处。

- 性能改善，improved performance -- 比起快速交换路由缓存技术，CEF是较少CPU-密集的（CEF is less CPU-intensive than fast-switching route caching）。那么就有更多的CPU处理能力用在譬如QoS和加密等的三层业务上。
- 伸缩性，scalability -- 当dCEF模式开启时，CEF在诸如Catalyst 6500系列交换机等的高端平台的所有线路卡上，提供了全部的交换能力。

- 迅速恢复的能力, resilience -- CEF提供了大型动态网络中无例可循水平的数据交换一致性和稳定性。在动态网络中, 快速交换缓存条目由于路由变化而频繁地过期和作废。这些变动能够引起流量经由使用路由表的进程交换而不是使用路由缓存的快速交换 (CEF offers an unprecedented level of switching consistency and stability in large dynamic networks. In dynamic networks, fast-switching cache entries are frequently invalidated due to routing changes. These changes can cause traffic to be process-switched using the routing table rather than fast-switched using the route cache)。

## CEF的配置

### Configuring Cisco Express Forwarding

开启CEF只需简单的一条命令, 那就是全局配置命令 `ip cef [distributed]`。关键字 `[distributed]` 仅适用于像是 Catalyst 6500 系列、支持 dCEF 的高端交换机。下面的输出展示了如何在一台诸如 Catalyst 3750 系列交换机的低端平台上配置CEF。

```
VTP-Server-1(config)#ip cef
VTP-Server-1(config)#exit
```

下面的输出演示了在 Catalyst 6500 系列交换机上如何开启 dCEF。

```
VTP-Server-1(config)#ip cef distributed
VTP-Server-1(config)#exit
```

**注意:** 并没有用于配置或开启aCEF的显式命令。

## 路由问题的故障排除

### Troubleshooting Routing Issues

当在网络设备上配置路由时, 必须按照设计小心仔细地配置**静态或动态路由**(static or dynamic routing)。如有存有故障而无法通过网络发送/接收流量, 这时多半有着某种配置问题。在初次设置某台路由器时, 总会有一些类型的配置问题要你去排除。而如果某台路由器已经运行了一段时间, 而突然完全没有了流量(没有通信), 就要做一下情况分析, 看看路由协议有没有如预期那样发挥功能。

有时某些路由会间歇性地从路由表中消失又出现、消失又出现, 以致造成到特定目的网络的间歇性通或不通。这可能是由于某个确切网络区域存在某些通信故障, 而沿着该路径上的路由器在那个区域每次变得可用时都会宣告新的路由信息造成的。该过程就叫作“路由抖动(route flapping)”, 而使用一种叫作“路由惩罚(route dampening)”的特性, 可对这些特定抖动路由进行屏蔽 (be blocked), 以令到整个网络不受路由抖动的影响。

**注意:** 在使用静态路由时, 路由表一直不会变化, 所以对发生在不同网络区域内的故障, 也得不到任何信息。

在处理路由故障时, 标准方法就是依据路由表来检查沿路径的每条路由 (when troubleshooting routing issues the standard approach is to follow the routing table for every route along the path)。可能会要执行一下 `traceroute`, 来准确找出数据包去了哪里, 并看看路径上的那些路由器。采用这种方法, 就可以准确知道可能是哪台设备引起的该故障, 同时可以开始调查某些特定路由器上的路由表了。

在进行这样一个排错过程时, 一个常犯的错误就是仅在一个方向上调查该故障 (比如只从源到目的方向)。正确的做法是应在去和回两个方向进行排错, 因为可能会偶然遇到数据包在一个方向被阻止而从目的到源没有返回流量的情形。为保证一个最优的传输流, 沿路径处于两点之间的设备上的路由表中应在两

个方向上都有正确指向。

通常情况下都会用到第三方提供的连接，所以在想要对某个确切区域进行排错时，就要与服务提供商进行沟通，以共同解决问题。这就包括了分享路由表信息。

动态路由协议的采行，令到排错过程更为容易，因为可以检查由路由器发出和接收到的路由更新。而对路由更新的检查，可以通过抓包或内部的设备机制完成，同时将帮助我们看到路由表是在何时、如何生成的。有着一张拓扑图及其它列出了每个前缀在网络中所处位置的文档，将更好地帮助你对路由更新的理解，进而缩短排错的过程。在这样的一个排错过程中，一般的主张就是依网络的设计，决定某个特定数据包将会采取的路径，并调查一下到底这个数据包在该路径的何处，偏离了该路径。

要对网络设备进行监控，有着不同工具。而这些工具都用到同样的网络管理协议，那就是简单网络管理协议（Simple Network Management Protocol, SNMP），该协议设计从某台管理工作站对网络设备发起不同参数的查询（ICND2涵盖了SNMP）。除了检查标准的“健康度”参数（比如CPU、内存、磁盘空间等等）外，SNMP还会查询路由器的下面这些参数。

- 接口上数据包计数
- 使用到的带宽及通过量
- 设备接口上的CRC及其他类型的错误
- 路由表信息

其它可以用到工具就是标准的用于验证端到端连通性的 `ping` 和 `traceroute` 了。它们亦能展示一些可能有助于确定出网络中发生故障的点位的相关输出。

下面是在对几乎所有路由故障进行排错时所涉及的步骤。

- 检查路由是否开启
- 检查路由表是否有效
- 检查当前的路径选择

## 检查路由是否开启

### Verifying that Routing is Enabled

路由排错的第一步，就是检查路由协议是否开启及正确配置。这既可以通过检查当前运行配置（也就是 `show run` 命令），又可以使用结合了每种特定路由协议的 `show` 命令。这些路由协议的选项有下面这些。

```

Router#show ip ospf ?
<1-65535>          Process ID number
border-routers        Border and boundary router information
database              Database summary
flood-list            Link state flood list
interface             Interface information
max-metric            Max-metric origination information
mpls                  MPLS related information
neighbor              Neighbor list
request-list          Link state request list
retransmission-list   Link state retransmission list
rib                  Routing information base (RIB)
sham-links            Sham link information
statistics            Various OSPF Statistics
summary-address       Summary-address redistribution information
timers                OSPF timers information
traffic               Traffic related statistics
virtual-links         Virtual link information
|
<cr>

Router#show ip eigrp ?
<1-65535>          Autonomous System
accounting            IP-EIGRP accounting
interfaces            IP-EIGRP interfaces
neighbors             IP-EIGRP neighbors
topology              IP-EIGRP topology table
traffic               IP-EIGRP traffic statistics
vrf                  Select a VPN routing/forwarding instance

Router#show ip bgp ?
A.B.C.D               Network in the BGP routing table to display
A.B.C.D/nn             IP prefix <network>/<length>, e.g., 35.0.0.0/8
all                   All address families
cidr-only              Display only routes with non-natural netmasks
community              Display routes matching the communities
community-list         Display routes matching the community-list
dampening              Display detailed information about dampening
extcommunity-list      Display routes matching the extcommunity-list
filter-list            Display routes conforming to the filter-listinconsistent-as Display only ro
injected-paths         Display all injected paths
ipv4                  Address family
ipv6                  Address family
labels                 Display labels for IPv4 NLRI specific information
neighbors              Detailed information on TCP and BGP neighbor connections
nsap                  Address family
oer-paths              Display all oer controlled paths
paths                 Path information
peer-group             Display information on peer-groups
pending-prefixes       Display prefixes pending deletion
prefix-list            Display routes matching the prefix-list
quote-regexp           Display routes matching the AS path "regular expression"
regexp                 Display routes matching the AS path regular expression
replication            Display replication status of update-group(s)
rib-failure            Display bgp routes that failed to install in the routing table (RIB)
route-map              Display routes matching the route-map
summary                Summary of BGP neighbor status
template               Display peer-policy/peer-session templates
update-group           Display information on update-groups
vpnv4                 Address family
|
<cr>

```

## 检查路由表是否正确

### Verifying That the Routing Table Is Valid

在成功确定已开启路由进程后，下一步就要对各协议的路由表进行分析，看看那里列出的信息是否正确。一些需要着重注意的地方有下面这些。

- 验明经由正确的协议学习到正确的前缀
- 验明学到的前缀条数
- 验明这些路由的度量值及下一跳信息

依据路由协议的不同，还要对从设备向外通告的那些前缀的正确性进行检查。

## 检查路径选择的正确性

在检查了有关前缀在路由表中确有出现后，就应对这些前缀的属性值（译者注：其路由跳数及各条路由的度量值、下一跳等信息）及路径选择方式进行仔细分析。这些分析包括下面这些。

- 检查通告了该前缀的所有路由协议（还要包括静态路由）
- 对AD进行比较和修改，以令到优先选择某种指定的路由协议，而不是默认正确的
- 检查并调整这些协议的度量值

通过对网络中某台路由器的恰当配置，并在配置过程中对每一步都做好文档，以及对网络中两点自荐路径的持续监测，就能够对网络中流量是如何准确地流经那些设备，有扎实掌握。

## 第10天问题

1. What is a routing protocol?
2. \_\_\_\_\_ is used to determine the reliability of one source of routing information from another.
3. If a router learns a route from both EIGRP (internal) and OSPF, which one would it prefer?
4. What is the RIP AD?
5. What is the eBGP AD?
6. Name at least four routing metrics.
7. Once routes have been placed into the routing table, by default the most specific or longest match prefix will always be preferred over less specific routes. True or false?
8. \_\_\_\_\_ operates at the data plane and is a topology-driven proprietary switching mechanism that creates a forwarding table that is tied to the routing table (i.e., the control plane).
9. CEF uses a \_\_\_\_\_ to make IP destination prefix-based switching decisions.
10. Link State routing protocols are those that use distance or hop count as its primary metric for determining the best forwarding path. True or false?

## 第10天问题答案

1. A protocol that allows a router to learn dynamically how to reach other networks.
2. Administrative distance.
3. EIGRP.
4. 120.
5. 20.
6. Bandwidth, cost, delay, load, reliability, and hop count.
7. True.

- 8. CEF.
- 9. FIB.
- 10. False.

## 第10天的实验

### 路由概念实验

采用两台直连的路由器，并测试本模块中提到的那些基本命令。RIP已不在CCNA考试中了，但其对于一个简单的实验来说，是十分简单易用的。

- 给直连接口分配一个IPv4地址（10.10.10.1/24及10.10.10.2/24）
- 用 ping 测试直连的连通性
- 在两台路由器上都配置一个环回接口，并从两个不同范围为其分配上地址（11.11.11.1/32及12.12.12.2/32）
- 配置标准RIP并通告所有本地网络

```
R1:  
router rip  
version 2  
no auto  
network 10.10.10.0  
network 11.11.11.0  
  
R2:  
router rip  
version 2  
no auto  
network 10.10.10.0  
network 12.12.12.0
```

- 自R1向R2的环回接口进行 ping 操作，以测试连通性
- 执行一条 show ip route 命令，来检查经由RIP收到了那些路由
- 执行一条 show ip protocols 命令，来检查有配置了RIP且RIP在设备上是允许着的

## 第11天 静态路由

### Static Routing

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第11天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND1记诵指南
- 在[subnetting.org](http://subnetting.org)上用 15 分钟

选择作一名网络管理员，就是要在网络中用到动态路由协议或坚持使用静态路由，所谓静态路由，就是手动将网络的所有路由，加入到所有路由器上去。

经常会有人问我（原作者）哪个路由协议是“最好的”。是没有一种方式适合于每个网络的，因为就算某家特定机构的网络需求，也会随时间变化。配置静态路由需要时间和精力，但可以省下一些网络带宽和CPU运算周期。如要加入一条新路由，就必须在所有路由器上进行手动添加。此外，如有某台路由器宕掉，静态路由就没有办法处理这个事情，所以还会往宕掉的网络发送流量（有关可靠静态路由的部分，不再CCNA大纲的范围之内）。

今天要学到下面这些。

- 静态路由的配置
- 静态路由排错

本模块对应了以下CCNA大纲要求。

- 按照给出的特定路由要求，配置并验证一条静态或默认路由的路由配置
- 区分不同路由方式及各种路由协议
  - 静态路由对动态路由
  - 下一跳

如回头看一下第 10 天中的管理距离表，就会发现手动配置的网络比起路由协议，是首选的路由。这么做的理由就是，作为网络管理员，期望着比任何协议都要对网络有更好的了解，并比任何协议都清楚要达到什么目的。那么现在，就应该明白，在需要达到某种目的时，可以结合动态路由来使用静态路由。

## 静态路由配置

### Configuring Static Routes

配置一条静态路由（见下图11.1）需要以下这些命令。

- network address/prefix mask
- address **or** exit interface
- distance (**optional**)

这里是一个这些命令使用的实例。

```
RouterA(config)#ip route network prefix mask {address | interface} [distance]
```



图11.1 -- 静态路由示例网络

要加入上面网络的一条静态路由，就要在左边的路由器上写出下面这行配置。

```
Router(config)#ip route 192.168.1.0 255.255.255.0 172.16.1.2
```

对静态路由，需要指定在前往目的地址的路途上，路由器需要去往的下一跳IP地址，或者也可以指定一个出去的接口。通常不需要知道下一跳地址，因为那就是ISP，或者IP地址会随时变化（见下图11.2）。如果是这样，就要使用出去的接口。



图11.2 -- 不总是知道下一跳地址的情形

```
Router(config)#ip route 192.168.1.0 255.255.255.0 s0/0
```

上面的命令行告诉路由器将目的为 192.168.1.10 网络的流量，从串行接口发出。而下面的命令则是告诉路由器将所有网络的所有流量，都从串行接口发出。

```
Router(config)#ip route 0.0.0.0 0.0.0.0 s0/0
```

上面的路由实际上就是一条默认路由（a default route）。默认路由用于引导那些未在路由表中显式列出的目的网络的数据包。

### 静态IPv6路由的配置

#### Configuring Static IPv6 Routes

静态IPv6路由的配置，与静态IPv4路由的配置遵循同样的逻辑。在思科IOS软件中，全局配置命令 `ipv6 route [ipv6-prefix/prefix-length] [next-hop-address | interface] [distance <1-254> | multicast | tag | unicast]` 用于配置静态IPv6路由。当中的一些关键字是熟悉的，因为它们也适用于IPv4静态路由，而 `[multicast]` 关键字则是IPv6所独有的，用于配置一条IPv6静态多播路由(an IPv6 static Multicast route)。如用到此关键字，该路由就不会进到单薄路由表 (the Unicast routing table)，同时也绝不会用于转发单播流量。为确保该路由绝不会安装到单播路由信息库 (the Unicast RIB)，思科IOS软件将该条路由（静态多播路由）的管理距离设置为 255。

相反，`[unicast]` 关键字则是用于配置一条IPv6静态单播路由。如用到此关键字，该条路由就绝不会进入到多播路由表 (the Multicast routing table)，并仅被用于转发单播流量。而既没用到 `[multicast]` 关键字，也没用到 `[unicast]` 关键字时，默认情况下，该条路由会用于单播数据包的转发，也会用于多播数据包的转发。

以下的配置示例，演示了如何来配置 3 条静态IPv6路由。第一条路由，到子网 `3FFF:1234:ABCD:0001::/64`，会将流量从 `FastEthernet0/0` 转发出去。此路由仅用于单播流量的转发。第二条路由，到子网 `3FFF:1234:ABCD:0002::/64`，会将到那个子网的数据包从 `Serial0/0`，使用下一跳路由器的数据链路层地址，作为IPv6的下一跳地址转发出去。本条路由仅会用于多播流量。最后，同样配置了一条指向 `Serial0/1` 作为出口接口的默认路由。此默认路由将会通过 `Serial0/1`，使用下一跳路由器的本地链路地址作为IPv6下一跳地址，转发那些到未知IPv6目的地址的数据包。这些路由如下面所示。

```
R1(config)#ipv6 route 3FFF:1234:ABCD:0001::/64 Fa0/0 unicast
R1(config)#ipv6 route 3FFF:1234:ABCD:0002::/64 Se0/0 FE80::2222 multicast
R1(config)#ipv6 route ::/0 Serial0/1 FE80::3333
```

依此配置，命令 `show ipv6 route` 可用于验证在本地路由器上应用的静态路由配置，如下所示。

```
R1#show ipv6 route static
IPv6 Routing Table - 13 entries
Codes: C - Connected, L - Local, S - Static, R - RIP, B - BGP
U - Per-user static route
I1 - ISIS L1, I2 - ISIS L2, IA - ISIS inter area, IS - ISIS summary
O - OSPF intra, OI - OSPF inter, OE1 - OSPF ext 1, OE2 - OSPF ext 2
ON1 - OSPF NSSA ext 1, ON2 - OSPF NSSA ext 2
S ::/0 [1/0]
via FE80::3333, Serial0/1
S 3FFF:1234:ABCD:1::/64 [1/0]
via ::, FastEthernet0/0
S 3FFF:1234:ABCD:2::/64 [1/0]
via FE80::2222, Serial0/0
```

除了使用 `show ipv6 route` 命令外，命令 `show ipv6 static [prefix] [detail]` 也可一用来对所有或仅是某条特定静态路由的细节信息进行查看。下面输出演示了如何使用这个命令。

```
R1#show ipv6 static 3FFF:1234:ABCD:1::/64 detail
IPv6 static routes
Code: * - installed in RIB
* 3FFF:1234:ABCD:1::/64 via interface FastEthernet0/0, distance 1
```

## 静态路由排错

### Troubleshooting Static Routes

排错总会涉及到某个配置问题（如果不是接口宕掉的话）。如流量没有到达目的地，就可以使用命令 `traceroute` 测试该路由。

注意 -- 今天内容很少，所以请前往第12天吧，因为那将是个非常充实的主题。

## 第11天问题

1. Name the three parameters needed to configure a static route.
2. What is the command used to configure a static route?
3. What is the command used to configure a default static route?
4. What is the command used to configure an IPv6 static route?
5. What is the command used to view IPv6 static routes?

## 第11天答案

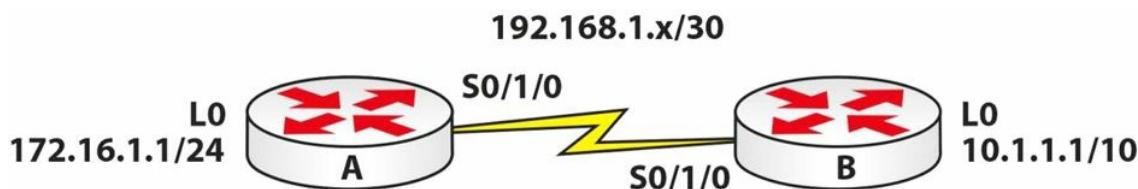
1. Network address, subnet mask (prefix length), and next-hop address or exit interface.
2. The `ip route` command.
3. The `ip route 0.0.0.0 0.0.0.0` command.
4. The `ipv6 route` command.
5. The `show ipv6 route static` command.

## 第11天实验

### 静态路由实验

#### Static Routes Lab

#### 拓扑图



#### 实验目的

学习如何以下一跳地址和出口接口方式，将静态路由指定给一台路由器。

#### 实验步骤

1. 按照上面的拓扑图分配IP地址。Router A 可以是 `192.168.1.1/30`，Router B 可以是 `.2`。
2. 通过串行链路进行 `ping` 操作，以确保该链路是工作的。
3. 在 Router A 上指定一条静态路由，将到 `10.1.1.0/10` 网络的所有流量，从串行接口发送出去。当然要使用你自己的串行端口编号；不要只是拷贝我的配置，你的接口有不同编号！

```

RouterA(config)#ip route 10.0.0.0 255.192.0.0 Serial0/1/0
RouterA(config)#exit
RouterA#ping 10.1.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 18/28/32 ms
RouterA#
RouterA#show ip route
Codes: C - Connected, S - Static, I - IGRP, R - RIP, M - Mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - Candidate default, U - Per-user static route, o - ODR
      P - Periodic downloaded static route
Gateway of last resort is not set
      10.0.0.0/10 is subnetted, 1 subnets
S          10.0.0.0 is directly connected, Serial0/1/0
      172.16.0.0/24 is subnetted, 1 subnets
C          172.16.1.0 is directly connected, Loopback0
      192.168.1.0/30 is subnetted, 1 subnets
C          192.168.1.0 is directly connected, Serial0/1/0
RouterA#
RouterA#show ip route 10.1.1.1
Routing entry for 10.0.0.0/10
Known via "static", distance 1, metric 0 (connected)
  Routing Descriptor Blocks:
    * directly connected, via Serial0/1/0
      Route metric is 0, traffic share count is 1
RouterA#

```

- 在 Router B 上配置一条静态路由，将到 172.16.1.0/24 网络的所有流量，发到下一跳地址 192.168.1.1。

```

RouterB(config)#ip route 172.16.1.0 255.255.255.0 192.168.1.1
RouterB(config)#exit
RouterB#ping 172.16.1.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 172.16.1.1, timeout is 2 seconds:
!!!!!
RouterB#show ip route 172.16.1.1
Routing entry for 172.16.1.0/24
Known via "static", distance 1, metric 0
  Routing Descriptor Blocks:
    * 192.168.1.1
      Route metric is 0, traffic share count is 1
RouterB#

```

# 第12天 OSPF基础知识

## OSPF Basics

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 今天的任务

- 阅读今天的理论课文
- 复习昨天的理论课文

先前版本的CCNA考试只要求对OSPF有基本了解。现今版本要求对 OSPFv2、v3 及多区域OSPF都要有更为深入的掌握。OSPF考点在ICND1和ICND2中都有，ICND2中增加了难度。

今天将会学到下面这些内容。

- 链路状态基础，Link State fundamentals
- OSPF组网类型，OSPF network types
- OSPF的配置，Configuring OSPF

本模块对应了以下CCNA大纲要求：

- OSPF（单区域）的配置和验证
  - 单区域的好处, benefit of single area
  - OSPFv2的配置, configure OSPFv2
  - 路由器ID, router ID
  - 被动接口, passive interface

## 开放最短路径优先

### Open Shortest Path First

开放最短路径优先，是一个开放标准的链路状态路由协议 (an open-standard Link State routing protocol)。所有链路状态路由协议都是对链路状态进行通告。当链路状态路由器开始在某条网络链路上运作时，逻辑网络的相关信息就被添加到路由器的本地链路状态数据库(Link State Database, LSDB)中。该本地路由器此时在其运作链路上，发出 Hello 数据包，以确定出是否有其它链路状态路由器也在其各自接口上运行着链路状态路由协议。OSPF直接运行在IP协议上，使用IP下编号为 89 号的协议。

## OSPF概述及基础知识

## OSPF Overview and Fundamentals

人们为OSPF撰写了多个请求评议（Requests for Comments, RFCs）。在本小节，将通过一些OSPF有关的最常见的几个RFCs，来了解一下OSPF的历史。OSPF工作组成立于 1987 年，自成立以后，该工作组发布了数众多的RFCs。下面列出了OSPF有关的一些最常见的RFCs。

- [RFC 1131](#) -- OSPF 规格, OSPF Specification
- [RFC 1584](#) -- OSPF 的多播扩展, Multicast Extensions to OSPF
- [RFC 1587](#) -- OSPF 的 NSSA 选项, the OSPF NSSA Option
- [RFC 1850](#) -- OSPF 版本 2 的管理信息库, OSPF Version 2 Management Information Base
- [RFC 2328](#) -- OSPF 版本 2
- [RFC 2740](#) -- OSPF 版本 3

[RFC 1131](#) 对OSPF的第一次迭代（the first iteration of OSPF）进行了说明，而应用在明确该协议是否工作的早期测试中。

[RFC 1584](#) 为OSPF提供了对IP多播流量的支持扩展。这通常被称为多播OSPF（Multicast OSPF, MOSPF）。但该标准不常用到，而最重要的是思科不支持该标准。

[RFC 1587](#) 对一种OSPF的次末梢区域（Not-So-Stubby Area, NSSA）的运作方式进行了说明。NSSA允许通过一台自治系统边界路由器（an Autonomous System Boundary Router, ASBR），采用一条NSSA的外部LSA，实现外部路由知识的注入（the injection of external routing knowledge）。在本模块的稍后会对不同的NSSAs进行说明。

[RFC 1850](#) 实现了使用简单网络管理协议（Simple Network Management Protocol, SNMP）对OSPF的网络管理。在网络管理系统中，SNMP用于监测接入网络设备中需要留心的一些情况。本标准的应用超出了CCNA考试要求范围，不会在本书中进行说明。

[RFC 2328](#) 详细陈述了OSPF版本 2 的最新更新，而OSPF版本 2 正是现今在用的默认版本。OSPF版本 2 最初是在 [RFC 1247](#) 中进行说明的，该RFC解决了OSPF版本 1 初次发布中发现的一系列问题，并对该协议进行了修正，实现了未来修改不致产生出向后兼容问题。正因为如此，OSPF版本 2 与版本 1 是不兼容的。

最后，[RFC 2740](#) 说明了为支持IPv6而对OSPF做出的修改（也就是版本 3）。应假定本模块中所有对OSPF一词的使用，都是指的OSPF版本 2。

## 链路状态基础

### Link State Fundamentals

当对某条特定链路（也就是接口）开启链路状态路由协议时，与那个网络有关的信息就被加入到本地LSDB中。该本地路由器此时就往其运作的各链路上发送 Hello 数据包，以确定有否其它**链路状态路由器**也在接口上运行着。**Hello 数据包**用于邻居发现，并在邻居路由器之间维护邻接关系。本模块稍后部分会详细说明这些消息。

在找到一台邻居路由器后，假定两台路由器在同一子网且位于同一区域，同时诸如认证方法及计时器等其它参数都是一致的（identical），那么本地路由器就尝试建立一个邻接关系（adjacency）。此邻接关系令到两台路由器将**摘要的LSDB信息**通告给对方。这种信息交换，交换的并非真实的详细数据库信息，而是数据的摘要。

各台路由器参照其本地LSDB，对收到的摘要信息做出评估，以确保其有着最新信息。如邻接关系的一侧认识到它需要一个更新，路由器就从邻接路由器请求新信息。而来自邻居路由器的更新就包含了LSDB中的具体数据。此交换过程持续到两台路由器都拥有同样的LSDB。OSPF用到不同类型的报文，以交换数据库信息，从而确保所有路由器都有着网络的统一视图。这些不同的数据包类型将在本模块稍后进行详细说明。

紧接着数据库的交换，SPF算法就运行起来，创建出到某个区域或网络主干中所有主机的最短路径树，SPF 算法将执行运算的路由器，作为该树的根（Following the database exchange, the SPF algorithm runs and creates a shortest path tree to all hosts in an area or in the network backbone, with the router that is performing the calculation at the root of that tree）。在第10天中，对SPF算法进行了简要介绍。

## OSPF基础

### OSPF Fundamentals

与EIGRP能够支持多个网络层协议不同，OSPF只能支持IP，也就是IPv4和IPv6。和EIGRP相同的是，OSPF支持VLSM、认证及在诸如以太网这样的多路访问（Multi-Access networks）网络上，于发送和接收更新时，利用IP多播技术（IP Multicast）。

OSPF是一种层次化的路由协议，将网络以逻辑方式，分为称作区域的众多子域。这种逻辑分段方法，用于限制链路状态通告在OSPF域中扩散的范围（OSPF is a hierarchical routing protocol that logically divides the network into subdomains referred to as areas. This logical segmentation is used to limit the scope of Link State Advertisements(LSAs) flooding throughout the OSPF domain）。LSAs是由运行OSPF的路由器发出的特殊类型数据包。在区域内和区域间用到不同类型的LSAs。通过限制一些类型的LSAs在区域间传播，OSPF的层次化实现有效地减少了OSPF网络中路由协议流量的数量。

**注意：** OSPF的这些LSAs会在第39天详细说明。

在多区域OSPF网络中，必须指定一个区域作为**骨干区域**，或者叫 Area 0。OSPF骨干就是此OSPF网络的逻辑中心。其它**非骨干区域**都必须物理连接到骨干。但因为在非骨干区域和骨干区域之间有着一条物理连接，并非总是可能或可行的，所以OSPF标准允许使用到骨干的虚拟连接。这些虚拟连接也就是常说的虚拟链路，但此概念是不包括在当前的CCNA大纲中的（In a multi-area OSPF network, one area must be designated as **the backbone area**, or Area 0. The OSPF backbone is **the logical centre** of the OSPF network. All other non-backbone areas must be connected physically to the backbone. However, because it is not always possible or feasible to have a physical connection between a non-backbone area and the backbone, the OSPF standard allows the use of virtual connections to the backbone. These virtual connections are known as virtual links, but this concept is not included in the current CCNA syllabus）。

位处各区域中的路由器，都存储着其所在区域的详细拓扑信息。而在各区域中，一台或多台的路由器，又被称为**区域边界路由器**（Area Border Routers, ABRs），区域边界路由器通过在不同区域之间通告汇总路由信息，而促进区域间的路由（facilitate inter-area routing by advertising summarised routing information between the different areas）。本功能实现OSPF网络中的以下几个目标。

- 在OSPF域层面减小LSAs的扩散范围，Reduces the scope of LSAs flooding throughout the OSPF domain
- 在区域之间隐藏详细拓扑信息，Hides detailed topology information between areas
- OSPF域中端到端连通性的实现，Allows for end-to-end connectivity within the OSPF domain
- 在OSPF域内部创建逻辑边界，Creates logical boundaries within the OSPF domain

**注意：** 尽管ICND1大纲仅涉及到单区域OSPF（single-area OSPF），但为把大部分理论纳入讨论背景，有必要说一下多区域OSPF（multi-area OSPF）。

OSPF骨干区域从ABRs接收到汇总路由信息。该路由信息被散布到OSPF网络中的所有其它非骨干区域。在网络拓扑发生变化时，变化信息就被散布到整个的OSPF域，令到所有区域中的所有路由器都有着网络的统一视图（The OSPF backbone area receives summarised routing information from the ABRs. The routing information is disseminated to all other non-backbone areas within the OSPF network. When a change to the network topology occurs, this information is disseminated throughout the entire OSPF domain, allowing all routers in all areas to have a consistent view of the network）。下图12.1演示的网络拓扑，就是一个多区域OSPF部署的示例。

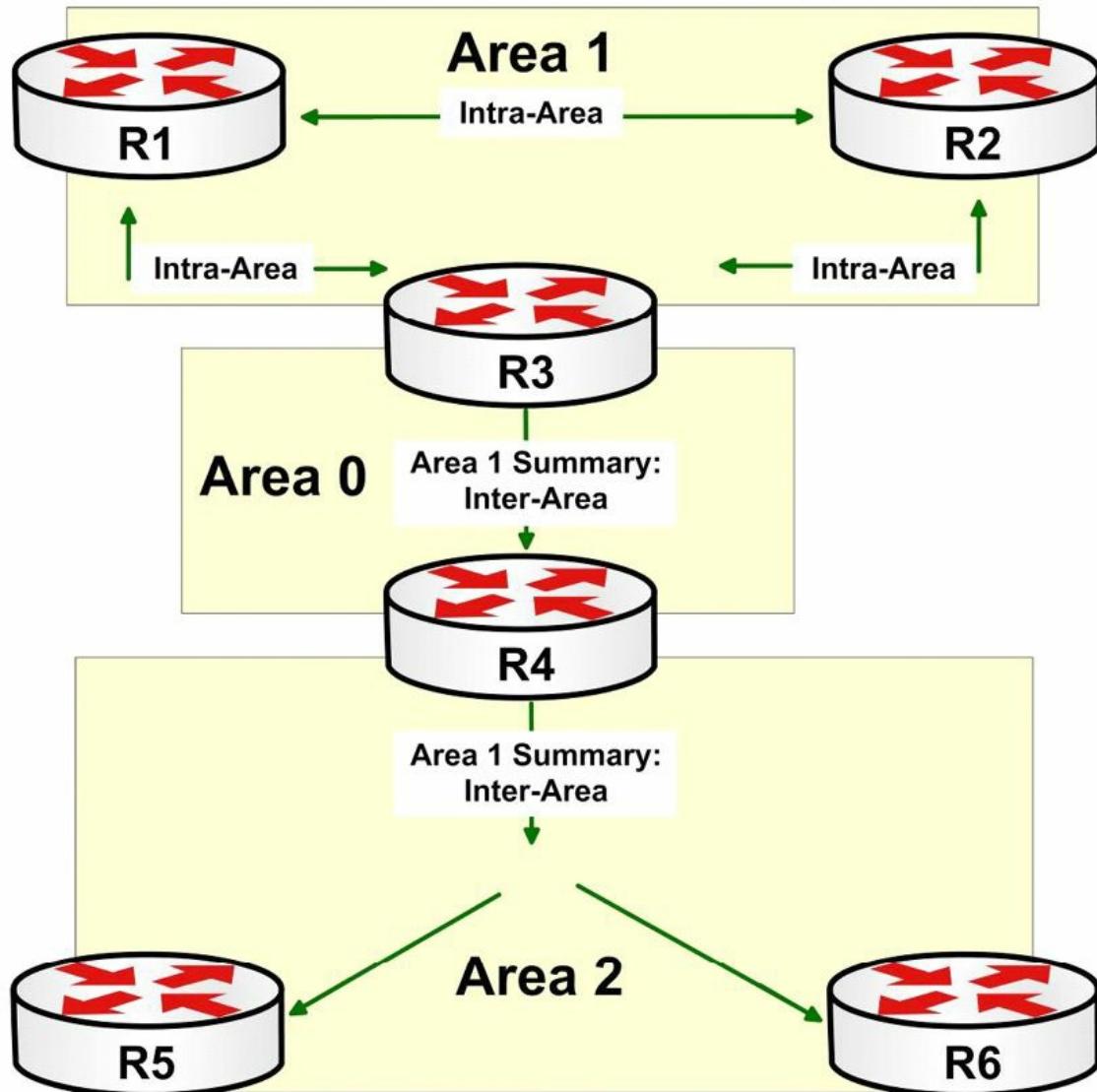


图12.1 -- 一个多区域OSPF网络

图12.1演示了一个基本的多区域OSPF网络。1、2号区域连接到0号区域，也就是OSPF骨干上。1号区域中，路由器R1、R2和R3交换着区域内（intra-area）路由信息，并维护着那个区域的详细拓扑。R3作为ABR，生成一条区域间汇总路由（an inter-area summary route），并将该路由通告给OSPF骨干。

R4，也就是2号区域的ABR，从0号区域接收到R3通告出的汇总信息，并将其扩散到其邻接区域。这样做就令到R5和R6知悉位处其本地区域外、但仍在OSPF域内的那些路由了。同样概念也适用于2号区域内的路由信息（R4, the ABR for Area 2, receives the summary information from Area 0 and floods it into its adjacent area. This allows routers R5 and R6 to know of the routes that reside outside of their local area but within the OSPF domain. The same concept would also be applicable to the routing information within Area 2）。

总的来讲，ABRs都维护着所有其各自连接区域的LSDB信息。而各个区域中的所有路由器，都有着属于其特定区域的详细拓扑信息。这些路由器交换着区域内的路由信息。ABRs则将所连接区域的汇总信息通告给其它OSPF区域，以实现域内各子域（区域）间的路由（In summation, the ABRs maintain LSDB information for all the areas in which they are connected. All routers within each area have detailed

topology information pertaining to that specific area. These routers exchange intra-area routing information. The ABRs advertise summary information from each of their connected areas to other OSPF areas, allowing inter-area routing within the domain)。

注意：本书后面会详细说明OSPF ABRs及其它OSPF路由器类型。

## 组网类型

### Network Types

对不同传输介质，OSPF采用不同默认组网类型，有下面这些：

- 非广播组网（在多点非广播多路复用传输介质上，也就是FR和ATM, 默认采用此种组网类型， Non-Broadcast, default on Multipoint Non-Broadcast Multi-Access(FR and ATM)）
- 点对点组网（在HDLC、PPP、FR及ATM的P2P子接口，以及ISDN介质上，，默认采用此种组网类型， Point-to-Point, default on HDLC, PPP, P2P subinterface on FR and ATM, and ISDN）
- 广播组网（在以太网和令牌环介质上，，默认采用此种组网类型， Broadcast, default on Ethernet and Token Ring）
- 点对多点组网（Point-to-Multipoint）
- 环回组网（默认在环回接口上采用此种组网类型， Loopback, default on Loopback interfaces）

**非广播网络**是指那些没有原生的广播或多播流量支持的网络类型。非广播类型网络的最常见实例就是帧中继网络。非广播类型网络**需要额外配置，以实现广播和多播支持**。在这种网络上，OSPF选举出一台指定路由器(a Designate Router, DR), 及/或一台备用指定路由器 (a Backup Designated Router, BDR)。在本书后面会对这两台路由器进行说明。

思科IOS软件中，非广播类型网络上开启了OSPF的路由器， 默认每 30 秒发出 Hello 数据包。

若 4 个 Hello 间隔，也就是 120 秒中都没有收到 Hello 数据包，那么该邻居路由器就被认为是“死了”。下面的输出演示了在一个帧中继串行接口上 show ip ospf interface 命令的输出。

```
R2#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
    Internet Address 150.1.1.2/24, Area 0
    Process ID 2, Router ID 2.2.2.2, Network Type NON_BROADCAST, Cost: 64
    Transmit Delay is 1 sec, State DR, Priority 1
    Designated Router (ID) 2.2.2.2, Interface address 150.1.1.2
    Backup Designated Router (ID) 1.1.1.1, Interface address 150.1.1.1
    Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
        oob-resync timeout 120
        Hello due in 00:00:00
    Supports Link-local Signaling (LLS)
    Index 2/2, flood queue length 0
    Next 0x0(0)/0x0(0)
    Last flood scan length is 2, maximum is 2
    Last flood scan time is 0 msec, maximum is 0 msec
    Neighbor Count is 1, Adjacent neighbor count is 1
        Adjacent with neighbor 1.1.1.1 (Backup Designated Router)
    Suppress Hello for 0 neighbor(s)
```

一条点对点连接，简单来说就是一条两个端点之间的连接。P2P连接的实例，包括采用HDLC及PPP封装的物理WAN接口，以及FR和ATM的点对点子接口。**OSPF点对点组网类型中，不会选举出DR和BDR**。在P2P类型网络上，OSPF每 10 秒发出 Hello 数据包。在这些网络上，“死亡”间隔是 Hello 间隔的 4 倍，也就是 40 秒 (A Point-to-Point(P2P) connection is simply a connection between two endpoints only.)

Examples of P2P connections include physical WAN interfaces using HDLC and PPP encapsulation, and Frame Relay(FR) and Asynchronous Transfer Mode(ATM) Point-to-Point subinterfaces. No DR or BDR is

elected on OSPF Point-to-Point network types. By default, OSPF sends Hello packets out every 10 seconds on P2P network types. The "dead" interval on these network types is four times the Hello interval, which is 40 seconds)。下面的输出演示了在一条P2P链路上的 `show ip ospf interface` 命令的输出。

```
R2#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 150.1.1.2/24, Area 0
  Process ID 2, Router ID 2.2.2.2, 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 00:00:03
  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 1.1.1.1
  Suppress Hello for 0 neighbor(s)
```

广播类型网络，是指那些原生支持广播和多播流量的网络，最常见例子就是以太网。就如同在非广播网络中一样，OSPF也会在广播网络上选举一台DR及/或BDR。默认情况下，OSPF每隔 10 秒发出 Hello 数据包，而如在 4 倍Hello间隔中没有收到 Hello 数据包，就宣告邻居“死亡”。下面的输出演示了在一个 FastEthernet 接口上 `show ip ospf interface` 命令的输出。

```
R2#show ip ospf interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  Internet Address 192.168.1.2/24, Area 0
  Process ID 2, Router ID 2.2.2.2, Network Type BROADCAST, Cost: 64
  Transmit Delay is 1 sec, State BDR, Priority 1
  Designated Router (ID) 192.168.1.3, Interface address 192.168.1.3
  Backup Designated Router (ID) 2.2.2.2, Interface address 192.168.1.2
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:04
  Supports Link-local Signaling (LLS)
  Index 1/1, 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 192.168.1.3 (Designated Router)
  Suppress Hello for 0 neighbor(s)
```

点对多点是一种**非默认OSPF组网类型**。也就是说，此种组网类型必须使用接口配置命令 `ip ospf network point-to-point-multipoint [non-broadcast]` 手动进行配置。默认情况下，该命令默认应用于一个广播型点对多点类型网络的。此默认组网类型允许OSPF采用多播数据包来动态地发现其邻居路由器。此外在**广播型点对多点网络类型上，不进行DR/BDR选举**（**Point-to-Multipoint is a non-default OSPF network type.** In other words, this network type must be configured manually using the `ip ospf network point-to-multipoint [non-broadcast]` interface configuration command. By default, this command defaults to a **Broadcast Point-to-Multipoint network type**. This default network type allows OSPF to use Multicast packets to discover its neighbour routers. In addition, there is no DR/BDR election held on Broadcast Point-to-Multipoint network types）。

关键字 [non-broadcast] 将点对多点网络配置为**非广播点对多点网络**。这样做就要求静态的**OSPF邻居配置**，因为此时OSPF不会使用多播来动态地发现其邻居路由器。此外，这种网络类型不要求为指定网段进行DR及/或BDR选举。此种组网的主要用途，即允许分配邻居开销到邻居路由器，而非使用指定给接口的开销，作为那些自所有邻居处收到的路由的开销（The [non-broadcast] keyword configures the Point-to-Multipoint network type as a **Non-Broadcast Point-to-Multipoint network**. This requires **static OSPF neighbour configuration**, as OSPF will not use Multicast to discover dynamically its neighbour routers. Additionally, this network type does not require the election of a DR and/or a BDR router for the designated segment. The primary use of this network type is to allow neighbor costs to be assigned to neighbors instead of using the interface-assigned cost for routes received from all neighbors）。

点对多点组网类型，典型地用于部分全通辐射状**非广播多路访问网络**。不过此种组网类型也可指定给诸如广播多路访问网络（比如以太网）等的其它类型网络（The Point-to-Multipoint network type is typically used in **partial-mesh hub-and-spoke Non-Broadcast Multi-Access(NBMA)** networks. However, this network type can also be specified for other networks, such as Broadcast Multi-Access networks(e.g., Ethernet)）。默认情况下，在点对多点网络上，OSPF每 30 秒发出一个 Hello 数据包。默认死亡间隔是 Hello 间隔的 4 倍，也就是 120 秒。

下面的输出演示了在一个经手动配置为点对多点网络的帧中继串行接口上的 show ip ospf interface 命令的输出。

```
R2#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 150.1.1.2/24, Area 0
  Process ID 2, Router ID 2.2.2.2, Network Type POINT_TO_MULTIPOINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_MULTIPOINT
  Timer intervals configured, Hello 30, Dead 120, Wait 120, Retransmit 5
    oob-resync timeout 120
    Hello due in 00:00:04
  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 2
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 1.1.1.1
  Suppress Hello for 0 neighbor(s)
```

OSPF要求链路上两台路由器组网类型一致（一致的意思是两台路由器要么都进行选举要么都不进行选举）的主要原因在于计时器的数值。就像上面各个输出中演示的那样，不同组网类型采用了不同 Hello 数据包发送及死亡计时器间隔。为成功建立一个OSPF邻接关系，在两台路由器上这些数值必须匹配。

思科IOS软件允许通过使用接口配置命令 ip ospf hello-interval <1-65535> 及 ip ospf dead-interval [<1-65535>|minimal]，对默认OSPF Hello 数据包及死亡计时器进行修改。 ip ospf hello-interval <1-65535> 命令用于指定 Hello 间隔的秒数。在执行该命令后，软件会自动将死亡间隔配置为所配置的 Hello 包间隔的 4 倍。比如，假定某台路由器做了如下配置。

```
R2(config)#interface Serial0/0
R2(config-if)#ip ospf hello-interval 1
R2(config-if)#exit
```

通过在上面的 R2 上将 Hello 数据包间隔设置为 1，思科IOS软件就会自动的将默认死亡计时器调整为 Hello 间隔的 4 倍，就是 4 秒。下面的输出对此进行了演示。

```
R2#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 10.0.2.4/24, Area 2
  Process ID 4, Router ID 4.4.4.4, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 1, Dead 4, Wait 4, Retransmit 5
    oob-resync timeout 40
    Hello due in 00:00:00
...
[Truncated Output]
```

## 配置OSPF

### OSPF Configuration

本节对OSPF配置基础进行说明。

### 在思科IOS软件中开启OSPF

#### Enabling OSPF in Cisco IOS Software

在思科IOS软件中，通过使用全局配置命令 `router ospf [process id]` 开启OSPF。关键字 `[process id]` 是本地有效的(locally significant)，邻接关系的建立无需网络中所有路由器的进程号一致。运用本地有效的进程号，允许在同一台路由器上配置多个OSPF实例。

OSPF进程号是一个 1 与 65535 之间的整数。每个OSPF进程都维护着其独立链路状态数据库（LSDB）；但是，所有路由都放进的是同一IP路由表。也就是说，对配置在路由器上的各个单独OSPF进程，并没有各自唯一的IP路由表。

在思科IOS软件早期版本中，如路由器上没有至少一个的接口配置了有效IP地址且处于 `up/up` 状态，就无法开启OSPF。此限制在当前版本思科IOS软件中去除了。假如路由器没有接口配置了有效IP地址且处于 `up/up` 状态，那么思科IOS将创建出一个接近数据库（a Proximity Database, PDB）并允许创建出进程。但是，要记住除非选定路由器ID，该进程就是非活动的进程，而**路由器ID的选定**，可通过下面两种方式完成。

- 在某个接口上配置一个有效IP地址，并将该接口开启
- 使用命令 `router-id` 为该路由器手动配置一个ID（见下）

作为一个例子，看看下面的所有接口都关闭的路由器。

```
R3#show ip interface brief
Interface      IP-Address      OK?      Method      Status          Protocol
FastEthernet0/0  unassigned      YES       manual     administratively down      down
Serial0/0        unassigned      YES       NVRAM      administratively down      down
Serial0/1        unassigned      YES       unset      administratively down      down
```

接着，使用全局配置命令 `router ospf [process id]` 在该路由器上开启了OSPF，如下面输出所示。

```
R3(config)#router ospf 1
R3(config-router)#exit
```

基于此配置，思科IOS软件分配给该进程一个默认 `0.0.0.0` 的路由器ID，如下面 `show ip protocols` 命令的输出所示。

```
R3#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 0.0.0.0
  Number of areas in this router is 0. 0 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    Reference bandwidth unit is 100 mbps
    Routing Information Sources:
      Gateway      Distance      Last Update
      Distance: (default is 110)
```

但是，命令 `show ip ospf [process id]` 揭示出该进程实际上并不是活动的，且表明需要配置一个路由器 ID，其输出如下面所示。

```
R3#show ip ospf 1
%OSPF: Router process 1 is not running, please configure a router-id
```

## 开启接口或网络的OSPF路由

### Enabling OSPF Routing for Interfaces or Networks

在开启OSPF后，就可以执行两个操作，来为路由器上一个或更多的网络或接口开启OSPF路由。这两个操作如下。

- 使用路由器配置命令(router configuration command) `[network] [wildcard] area [area id]`
- 使用接口配置命令 `ip ospf [process id] area [area id]`

与EIGRP不同，OSPF强制使用反掩码且必须配置反掩码；但与在EIGRP中的情况一样，反掩码提供了同样的功能，也就是匹配指定范围中的接口（unlike EIGRP, the wildcard is mandatory in OSPF and must be configured; however, as is the case with EIGRP, it serves the same function in that it matches interfaces within the range specified）。比如，语句 `network 10.0.0.0 0.255.255.255 area 0`，就会对 `10.0.0.1/30`、`10.5.5.1/24`，甚至 `10.10.10.1/25` 这样的IP地址和子网掩码组合的接口，开启OSPF路由。基于该OSPF网络配置，这些接口都会被分配到0号区域。

**注意：** OSPF反掩码可以与传统子网掩码同样格式敲入，比如 `network 10.0.0.0 255.0.0.0 area 0`。在这种情况下，思科IOS软件就会将子网掩码翻转，将得到的反掩码输入到运行配置（the running configuration）。另外要记住OSPF也支持使用全1s和全0s反掩码，来对某个指定接口开启OSPF。这样的配置在某个特定接口上开启OSPF，但路由器通告配置在该接口自身的实际子网掩码（this configuration enables OSPF on a particular interface but the router advertises the actual subnet mask configured on the interface itself）。

在执行了 `network [network] [wildcast] area [area id]` 命令后，路由器就在与指定网络和反掩码组合匹配的那些接口上发出 Hello 数据包，来尝试发现邻居路由器。接着在OSPF数据库交换期间，将连接子网通告给一台或更多的邻居路由器，最终，它们将所有子网信息都被加入到这些OSPF路由器的OSPF链路状态数据库（OSPF LSDB）中。

在命令 `network [network] [wildcard] area [area id]` 之后，路由器又对最具体条目做出匹配，以决定将接口要分配给的区域。作为实例，想想下面这些OSPF网络语句。

- 第一条网络配置语句： `network 10.0.0.0 0.255.255.255 area 0`
- 第二条： `network 10.1.0.0 0.0.255.255 area 1`
- 第三条： `network 10.1.1.0 0.0.0.255 area 2`

- 第四条: `network 10.1.1.1 0.0.0.0 area 3`
- 第五条: `network 0.0.0.0 0.0.0.0 area 4`

按照路由器上的此种配置，同时路由器上又配置了如下表12.1中展示的这些环回接口。

接口	IP地址/掩码
`Loopback 0`	`10.0.0.1/32`
`Loopback 1`	`10.0.1.1/32`
`Loopback 2`	`10.1.0.1/32`
`Loopback 3`	`10.1.1.1/32`
`Loopback 4`	`10.2.0.1/32`

就像前面指出的那样，在执行了 `network [network] [wildcard] area [area id]` 命令后，路由器匹配最具体的网络条目（最小的网络），来决定该接口应分配到的区域。对于在路由器上的网络配置语句及已配置的接口，命令 `show ip ospf interface brief` 会显示出这些接口都分配给了以下OSPF区域。

```
R1#show ip ospf interface brief
Interface      PID   Area     IP Address/Mask      Cost    State      Nbrs F/C
Lo4           1     0        10.2.0.1/32          1       LOOP      0/0
Lo1           1     0        10.0.1.1/32          1       LOOP      0/0
Lo0           1     0        10.0.0.1/32          1       LOOP      0/0
Lo2           1     1        10.1.0.1/32          1       LOOP      0/0
Lo3           1     3        10.1.1.1/32          1       LOOP      0/0
```

**注意：**在运行配置 (the running configuration) 中，无需考虑网络语句敲入顺序，路由器的 `show running-config` 输出中最具体的网络条目，始终列在前面。

**接口配置命令** `ip ospf [process id] area [area id]` 令到无需使用**路由器配置命令** `network [network] [wildcard] area [area id]`。该命令对某个指定接口开启OSPF路由，同时将该接口分配给指定OSPF区域。这两个命令完成同样的基本功能，且可互换使用。

此外，比如有两台路由器是背靠背连接 (connected back-to-back)，一台使用接口配置命令 `ip ospf [process id] area [area id]` 进行了配置，而其邻居路由器使用路由器配置命令 `network [network] [wildcard] area [area id]` 进行了配置，假设两个区域IDs相同，那么两台路由器将成功建立OSPF邻接关系。

## OSPF区域

### OSPF Areas

OSPF区域号既可以配置为一个 0 到 4294967295 之间的整数，也可使用点分十进制表示法（也就是采用IP地址格式）。与OSPF进程号不同，**为建立邻接关系，OSPF区域号必须匹配**。最常见OSPF区域配置类型为使用一个整数来指定OSPF区域。确保对支持的两种区域配置方式都要熟悉。

## OSPF路由器ID

### OSPF Router ID

为令到OSPF在某个网络上运行起来，所有路由器都必须有个唯一身份编号 (a unique identifying number)，且在OSPF环境下要用到路由器ID。

在决定OSPF路由器ID时，思科IOS选用所配置环回接口中最高的IP地址。如未曾配置环回接口，软件就会使用所有配置的物理接口中最高的IP地址，来作为OSPF路由器ID。思科IOS软件同样允许管理员使用**路由器配置命令 router-id [address]**，来手动指定路由器ID。

环回接口极为有用，特别是在测试当中，因为它们无需硬件且是逻辑的，因此绝不会宕掉。

在下面的路由器上，给 `Loopback0` 配置了IP地址 `1.1.1.1/32`，给 `F0/0` 配置了 `2.2.2.2/24`。接着在路由器上给所有接口配置了OSPF。

```
Router(config-if)#router ospf 1
Router(config-router)#net 0.0.0 255.255.255.255 area 0
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#show ip protocols
Routing Protocol is "ospf 1"
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 1.1.1.1
    Number of areas in this router is 1. 1 normal 0 stub 0 nssa
    Maximum path: 4
    Routing for Networks:
        0.0.0.0 255.255.255.255 area 0
    Routing Information Sources:
        Gateway      Distance      Last Update
        1.1.1.1          110      00:00:14
    Distance: (default is 110)
```

但又想要将路由器ID硬编码（hard code）为 `10.10.10.1`。那么可通过再配置一个使用该IP地址的环回接口，或简单地将这个IP地址加在OSPF路由器ID处。**为令到改变生效，必须重启路由器或在路由器上清除该IP OSPF进程**（清除现有数据库）。

```
Router#conf t
Enter configuration commands, one per line.
End with CNTL/Z.
Router(config)#router ospf 1
Router(config-router)#router-id 10.10.10.1
Router(config-router)#Reload or use "clear ip ospf process" command, for this to take effect
Router(config-router)#end
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
Router#show ip prot
Routing Protocol is "ospf 1"
    Outgoing update filter list for all interfaces is not set
    Incoming update filter list for all interfaces is not set
    Router ID 10.10.10.1
    Number of areas in this router is 1. 1 normal 0 stub 0 nssa
    Maximum path: 4
    Routing for Networks:
        0.0.0.0 255.255.255.255 area 0
    Routing Information Sources:
        Gateway      Distance      Last Update
        1.1.1.1          110      00:03:15
    Distance: (default is 110)
```

到第 39 天，**DR和BDR选举时，就将看到这个路由器ID有着特别的重要性。**

## OSPF被动接口

### OSPF Passive Interfaces

被动接口可被描述成在其上没有路由更新发出的接口。在思科IOS软件中，通过使用**路由器配置命令** `passive-interface [name]`，将某接口配置为被动接口。如路由器上有多个接口需要配置为被动接口，就应使用 `passive-interface default` **这个路由器配置命令**。此命令将路由器上那些位处所配置网络范围内的所有接口，都配置为被动模式。而那些需要允许在其上形成邻接关系或邻居关系的接口，就应使用路由器配置命令 `no passive-interface [name]` 对其进行配置。

被动接口配置在OSPF和EIGRP中的工作方式是一样的，也就是一旦某接口被标记为被动接口，经由该接口形成的所有邻居关系都会被拆除，同时 **再也不会通过该接口发送或接收 Hello 数据包了**。不过，根据路由器上所配置的网络配置语句，该接口仍然会继续受通告。

```
Router(config)#router ospf 10
Router(config-router)#passive-interface f0/0
Router#show ip ospf int f0/0
FastEthernet0/0 is up, line protocol is up
  Internet address is 192.168.1.1/24, Area 0
  Process ID 10, Router ID 172.16.1.1, Network Type BROADCAST, Cost: 1
  Transmit Delay is 1 sec, State WAITING, Priority 1
  No designated router on this network
  No backup designated router on this network
  Timer intervals configured,Hello 10, Dead 40, Wait 40,Retransmit 5
    No Hellos (Passive interface)
```

## 第12天问题

1. What protocol does OSPF use?
2. How does OSPF determine whether other Link State routers are operating on the interfaces as well?
3. When a \_\_\_\_\_ routing protocol is enabled for a particular link, information associated with that network is added to the local Link State Database (LSDB).
4. OSPF utilises IP Multicast when sending and receiving updates on Multi-Access networks, such as Ethernet. True or false?
5. OSPF is a hierarchical routing protocol that logically divides the network into subdomains referred to as \_\_\_\_\_.
6. Name at least 4 OSPF network types.
7. Name the command used to enter OSPF configuration mode.
8. When determining the OSPF router ID, Cisco IOS selects the lowest IP address of the configured Loopback interfaces. True or false?
9. What command can you use to assign an interface to OSPF Area 2 (interface level command)?
10. \_\_\_\_\_ can be described as interfaces over which no routing updates are sent.

## 第12天答案

1. IP number 89.
2. By sending Hello packets.
3. Link State.
4. True.
5. Areas.
6. Non-Broadcast, Point-to-Point, Broadcast, Point-to-Multipoint, Point-to-Multipoint Non-Broadcast, and Loopback.

7. The `router ospf <id>` command.
8. False.
9. The `ip ospf <id> area 2`
10. Passive.

## 第12天实验

### OSPF基础实验

沿用第 10 天的实验场景（两台直接连接的路由器，各自其上的环回接口），但取代配置RIP及对物理和环回接口进行通告的是，使用OSPF 0 号区域实现（but instead of configuring RIP and advertising the physical and Loopback interfaces, do this using OSPF Area 0）。

- 分配一个IPv4地址给直接连接的接口（`10.10.10.1/24` 及 `10.10.10.2/24`）
- 运用 `ping` 操作，测试直接连通性
- 分别在两台路由器上配置一个环回接口，并自两个不同范围为其分配上地址（`11.11.11.1/32` 及 `12.12.12.2/32`）
- 配置上标准OSPF 1 号进程，并在 0 号区域中通告所有本地网络。同时为两台设备配置一个路由器 ID。

R1 :

```
router ospf 1
router-id 1.1.1.1
network 10.10.10.0 0.0.0.255 area 0
network 11.11.11.1 0.0.0.0 area 0
```

R2 :

```
router ospf 1
router-id 2.2.2.2
network 10.10.10.0 0.0.0.255 area 0
network 12.12.12.2 0.0.0.0 area 0
```

- 自 R1 向 R2 的环回接口执行ping操作，以测试连通性
- 执行一条 `show ip route` 命令，来验证有通过OSPF接收到路由
- 执行一条 `show ip protocols` 命令，来验证有配置OSPF且在设备上是活动的
- 坚持特定于OSPF的接口参数：`show ip ospf interface` 及 `show ip ospf interface brief`
- 在两台路由器上（直接连接接口）修改OSPF的Hello包和死亡计时器：`ip ospf hello` 及 `ip ospf dead`
- 执行一下 `show ip ospf 1` 命令，看看路由进程参数
- 重复该实验，但这次使用 `ip ospf 1 area 0 interface specific` 命令，而不是在router OSPF 下的 `network` 命令，对各个网络进行通告。

## 第13天 OSPF版本3

### OSPF version 3

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

## 第13天任务

- 阅读今天的理论课文
- 回顾昨天的理论课文

今天我们要着眼于OSPFv3, 这里将学习要下面的知识。

- OSPF基础

本模块对应了以下CCNA大纲要求。

- 配置OSPFv3
- 路由器ID
- 被动接口

## OSPF第3版

### OSPF Version 3

OSPFv3 定义在 RFC 2740 中，而其功能与 OSPFv2 相同，不过 OSPFv3 显式地是为IPv6路由协议设计 (OSPFv3 is defined in RFC 2740 and is the counterpart of OSPFv2, but it is designed explicitly for the IPv6 routed protocol)。该版本号取自此种OSPF数据包中的版本字段，该字段已被更新到数字 3。OSPFv3 规格主要是基于 OSPFv2，但因为加入对IPv6的支持，而包含了一些额外功能增强。

OSPFv2 和 OSPFv3 能在同一台路由器上运行。也就是说，同一台物理路由器可同时路由IPv4和IPv6流量，因为每个地址家族都有不同的SPF进程；这就是说，同样SPF算法对 OSPFv2 和 OSPFv3 分别有一个单独实例。OSPFv2 和 OSPFv3 有以下共同点。

- OSPFv3 继续使用着为 OSPFv2 所用到的数据包。包括数据库说明数据包 (Database Description, DBD)，链路状态请求数据包 (Link State Requests, LSRs)，链路状态更新数据包 (Link State Updates, LSUs)，以及链路状态通告数据包 (Link State Advertisements, LSAs)
- OSPFv3 中的动态邻居发现机制及邻接关系形成过程 (OSPF所经历的从初始、尝试建立邻接关系到邻接关系完整建立的过程)，仍然和 OSPFv2 中一样
- 在不同通信技术方面，OSPFv3 仍然保持对RFC的遵循 (OSPFv3 still remains RFC-compliant on different technologies)。比如，若在某条PPP链路上开启 OSPFv3，那么组网类型仍然被指定为点对点

(Point-to-Point)。同样，如在FR上开启 OSPFv3，默认组网类型仍然是非广播类型 (Non-Broadcast)。此外，在思科IOS软件中，默认组网类型仍可通过使用不同的、特定于接口的命令，手动进行改变。

- OSPFv2 和 OSPFv3 使用同样的LSA散布及老化机制 (the same LSA flooding and aging mechanism) .
- 与 OSPFv2 类似，OSPFv3 的路由器ID (rid) 仍然需要使用一个 32 位的IPv4地址。当在某台运行着双栈 (dual-stack, 也就是同时有IPv4和IPv6) 的路由器上开启 OSPFv3 时，那么与在 OSPFv2 中为思科 IOS路由器所用到的同样RID选定过程，也用于确定OSPFv3中要用到的路由器ID。但是，在一台没有接口运行着IPv4的路由器上开启 OSPFv3 时，就强制性要求使用路由器配置命令 `router-id` 来手动配置 OSPFv3 的路由器ID。
- OSPFv3 链路ID表明，这些链路并非IPv6专用，同时这些链路ID跟 OSPFv2 中一样，仍然基于一个 32 位IPv4地址。

在 OSPFv2 与 OSPFv3 有着这些相同点的同时，重要的是掌握那些你必须熟悉的存在的明显不同点。包括下面这些。

- 以与EIGRP类似的方式，OSPFv3 是在链路上运行的 (in a manner similar to EIGRP, OSPFv3 runs over a link)。这就打消了 OSPFv3 中执行网络声明语句的需求。取而代之的是，通过使用接口配置命令 `ipv6 router ospf [process id] area [area id]`，来将该链路配置为某个OSPF进程的组成部分。但是，与 OSPFv2 类似，OSPF进程号仍然是通过在全局配置模式中，使用全局配置命令 `ipv6 router ospf [process id]` 进行指定。
- OSPFv3 使用本地链路地址 (Link-local address) 来区分 OSPFv3 邻接关系。与EIGRPv6类似，OSPFv3 路由的下一跳地址将反映邻接的或邻居路由器的本地链路地址。
- OSPFv3 引入了两种新的OSPF LSA类型。分别是链路LSA (the Link LSA)，被定义为LSA类型 0x0008 (LSA Type 0x0008，或LSA Type 8)，以及区域内前缀LSA (the Intra-Area-Prefix LSA)，被定义为LSA类型 0x0029 (LSA Type 0x0029，或LSA Type 29)。链路LSA提供了路由器的本地链路地址，及加诸路由器上的所有IPv6前缀。每条链路都有一个链路LSA。可能有多个带有不同链路状态IDs的区域内前缀LSAs。因此，区域LSA散布范围就既可能是与应用自网络LSA的所经过网络的相关前缀网络，也可能是参考自路由器LSA的某台路由器或末梢区域相关前缀 (There can be multiple Intra-Area-Prefix LSAs with different Link-State IDs. The Area flooding scope can therefore be an associated prefix with the transit network referencing a Network LSA or it can be an associated prefix with a router or Stub referencing a Router LSA)。
- OSPFv2 与 OSPFv3 所用到的传输方式是不同的。OSPFv3 报文是用 (封装成) IPv6数据包发出的。
- OSPFv3 使用两个标准IPv6多播地址。多播地址 FF02::5 与 OSPFv2 中用到的所有SPF路由器 (AllSPFRouters) 地址 224.0.0.5 等价，同时多播地址 FF02::6 就是所有DR路由器 (AllDRRouters) 地址，且与OSPFv2中用到的 224.0.0.6 组地址等价。(这将在ICND2部分讲到)。
- OSPFv3 利用到IPv6内建的 IPSec 的能力，并将AH和ESP扩展头部用着一种的认证机制，而不是想在 OSPFv2 中可配置的为数众多的认证机制 (OSPFv3 leverages the built-in capabilities of IPSec and uses the AH and ESP extension headers as an authentication mechanism instead of the numerous authentication mechanisms configurable in OSPFv2)。因此，在 OSPFv3 的OSPF数据包中，那些认证和AuthType字段就被移除了。
- 最终的最后一个明显区别就是，OSPFv3 Hello 数据包现在不包含任何地址信息，而是包含了一个接口ID，该接口ID是发出 Hello 数据包路由器分配的，用于对链路做其接口的唯一区分。此接口ID成为网络LSA (the Network LSA) 的链路状态ID (Link State ID)，判断该路由器是否应成为该链路上的指定路由器 (This interface ID becomes the Network LSA's Link State ID, should the router become the Designated Router on the link)。

## 思科IOS软件的OSPFv2和OSPFv3配置差异

### Cisco IOS Software OSPFv2 and OSPFv3 Configuration Differences

在思科IOS软件中，配置OSPFv2与OSPFv3时有着一些配置差异。但应注意到，这些区别与其它路由协议的IPv4和IPv6版本的差异相比，并不那么显著。

在思科IOS软件中，通过使用全局配置命令 `ipv6 router ospf [process id]`，来开启OSPFv6。和OSPFv2中的情况一样，OSPF进程ID是对路由器本地有效的，并不要求其在邻接路由器上为建立邻接关系保持一致。

**译者总结：**邻居路由器要形成邻接关系，要求：1. 区域号一致；2. 认证一直；3. Hello包、死亡间隔时间直一致；不要求：进程号一致。Hello数据包用于动态邻居发现和形成邻接关系，因此Hello数据包包含上述要求的参数，不包含不要求的参数。只有形成了邻接关系，才能开始发送和接受LSAs。

与EIGRPv6（将在ICND2中涵盖）所要求的一样，OSPFv3的路由器ID也必须予以手动指定，或配置成一个带有IPv4地址的运行接口（比如一个环回接口）。与EIGRPv6类似，在启用OSPFv3时，是没有网络命令的（网络宣告，network statement）。取而代之的是，OSPF的启用，是基于各个接口的，且在同一接口上可开启多个OSPFv3实例（similar to EIGRPv6, there are no network commands used when enabling OSPFv3. Instead OSPFv3 is enabled on a per-interface basis and multiple instances may be enabled on the same interface）。

最后，当在诸如FR及ATM这样的NBMA网络上配置OSPFv3时，是在指定接口下，使用接口配置命令 `ipv6 ospf neighbor [link local address]`，来指定邻居声明语句（the neighbor statements）。而在OSPFv2中，这些语句会是在路由器配置模式中配置的。

**注意：**当在NBMA传输技术上配置OSPFv3时，应该使用本地链路地址来创建出静态FR地图声明语句（static Frame Relay map statements）。这是因为正是使用本地链路地址，而不是全球单播地址，建立邻接关系。比如，为给一个FR部署创建一幅静态FR地图语句并指定一台OSPF邻居路由器，就要在该路由器上应用下面的配置（在ICND2部分将对FR进行讲解）。

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#exit
R1(config)#interface Serial0/0
R1(config-if)#frame-relay map ipv6 FE80::205:5EFF:FE6E:5C80 111 broadcast
R1(config-if)#ipv6 ospf neighbor FE80::205:5EFF:FE6E:5C80
R1(config-if)#exit
```

## 思科IOS软件中OSPFv3的配置和验证

### Configuring and Verifying OSPFv3 in Cisco IOS Software

接着上一部分，上部分强调了OSPFv2和OSPFv3之间配置差异，那么这部分就要过一遍那些在思科IOS软件中开启和验证OSPFv3功能及路由的步骤。在思科IOS软件中，需要依序采行下面这些步骤，来开启OSPFv3路由。

1. 使用全局配置命令 `ipv6 unicast-routing`，来全局性地开启IPv6路由。在思科IOS软件中，IPv6路由默认是关闭的。
2. 使用全局配置命令 `ipv6 router ospf [process ID]`，配置一或多个的OSPFv3进程。
3. 如路由器上没有配置IPv4地址的运行接口，就要使用路由器配置命令（router configuration command）`router-id [IPv4 Address]`，手动配置OSPFv3路由器ID（Router ID, RID）。
4. 在需要的接口上（on the desired interfaces），使用接口配置命令 `ipv6 address` 及 `ipv6 enable`，对这些接口开启IPv6。
5. 使用接口配置命令 `ipv6 ospf [process ID] area [area ID]`，在接口下开启一或更多的OSPFv3进程。

第一个基础多区域OSPFv3配置示例，建立在下图13.1所演示的拓扑之上。

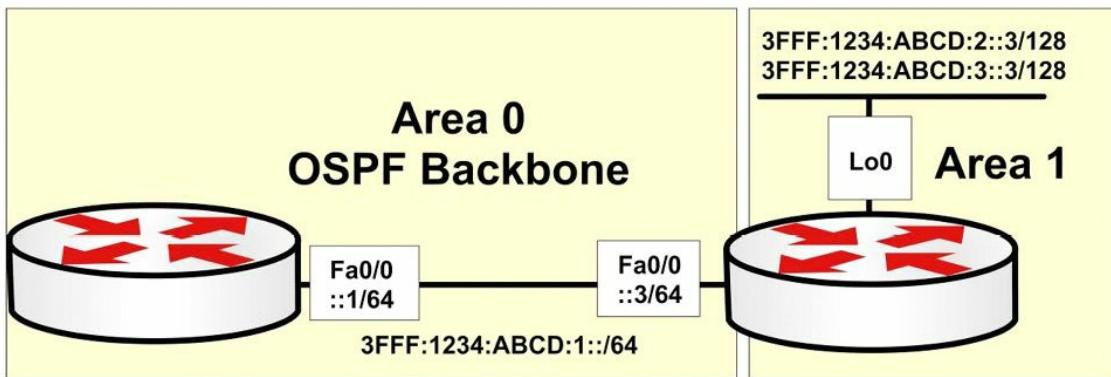


图13.1 -- 在思科IOS软件中配置基本多区域OSPFv3

依之间所讲到的顺序配置步骤，照下面这样，在路由器 R1 上就会配置上OSPFv3。

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router ospf 1
R1(config-rtr)#router-id 1.1.1.1
R1(config-rtr)#exit
R1(config)#interface FastEthernet0/0
R1(config-if)#ipv6 address 3fff:1234:abcd:1::1/64
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 ospf 1 Area 0
R1(config-if)#exit
```

而按照同样顺序的步骤，像下面这样在路由器 R3 上配置好OSPFv3路由。

```
R3(config)#ipv6 unicast-routing
R3(config)#ipv6 router ospf 3
R3(config-rtr)#router-id 3.3.3.3
R3(config-rtr)#exit
R3(config)#interface FastEthernet0/0
R3(config-if)#ipv6 address 3fff:1234:abcd:1::3/64
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 ospf 3 Area 0
R3(config-if)#exit
R3(config)#interface Loopback0
R3(config-if)#ipv6 address 3fff:1234:abcd:2::3/128
R3(config-if)#ipv6 address 3fff:1234:abcd:3::3/128
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 ospf 3 Area 1
R3(config-if)#exit
```

依据上述两台路由器上OSPFv3的配置，就可以使用命令 `show ipv6 ospf neighbor`，来检查OSPFv3的邻接状态，在 R1 上如下所示。

```
R1#show ipv6 ospf neighbor
Neighbor      ID Pri      State          Dead Time     Interface ID      Interface
3.3.3.3        1   FULL/BDR    00:00:36       4           FastEthernet0/0
```

通过将 `[detail]` 关键字追加到本命令的后面，还可以查看详细的邻居信息。

```
R1#show ipv6 ospf neighbor detail
Neighbor 3.3.3.3
  In the area 0 via interface FastEthernet0/0
  Neighbor: interface-id 4, link-local address FE80::213:19FF:FE86:A20
  Neighbor priority is 1, State is FULL, 6 state changes
  DR is 1.1.1.1 BDR is 3.3.3.3
  Options is 0x000013 in Hello (V6-Bit E-Bit R-bit )
  Options is 0x000013 in DBD (V6-Bit E-Bit R-bit )
  Dead timer due in 00:00:39
  Neighbor is up for 00:06:40
  Index 1/1/1, retransmission queue length 0, number of retransmission 0
  First 0x0(0)/0x0(0)/0x0(0) Next 0x0(0)/0x0(0)/0x0(0)
  Last retransmission scan length is 0, maximum is 0
  Last retransmission scan time is 0 msec, maximum is 0 msec
```

在上面的输出中，注意真实的邻居地址是本地链路地址，而不是所配置的全局IPv6单播地址。

## 第13天问题

1. Both OSPFv2 and OSPFv3 can run on the same router. True or false?
2. OSPFv2 and OSPFv3 use different LSA flooding and aging mechanisms. True or false?
3. Which is the equivalent of `224.0.0.5` in the IPv6 world?
4. As is required for EIGRPv6, the router ID for OSPFv3 must be either specified manually or configured as an operational interface with an IPv4 address. True or false?
5. Which command would you use to enable the OSPFv3 routing protocol?
6. Which command would you use to specify an OSPFv3 neighbour over an NBMA interface?
7. Which command would you use to see the OSPFv3 LSDB?
8. A significant difference between OSPFv2 and OSPFv3 is that the OSPFv3 Hello packet now contains no address information at all but includes an interface ID, which the originating router has assigned to uniquely identify its interface to the link. True or false?

## 第13天答案

1. True.
2. False.
3. `FF02::5`.
4. True.
5. The `ipv6 router ospf <id>`
6. The `ipv6 ospf neighbor`
7. The `show ipv6 ospf database`
8. True.

## 第13天实验

### OSPFv3基础实验

重复第 12 天的实验场景（两台路由器直连，各自又有环回接口），但以配置IPv6地址并在设备间使用 OSPFv3 对这些地址进行通告，取代配置IPv4的OSPF。

- 给直连接口分配上IPv6地址（`2001:100::1/64` 及 `2001:100::2/64`）

- 用 `ping` 测试直接连通性
- 在两台路由器上分别配置一个环回接口，并从两个不同范围分配地址（`2002::1/128` 及 `2002::2/128`）
- 配置标准的OSPFv3 1 号进程并将所有本地网络在 0 号区域进行通告。同时为各设备配置一个路由器 ID。

R1:

```
ipv6 router ospf 1
router-id 1.1.1.1
int fa0/0(或特定接口编号)
ipv6 ospf 1 area 0
int lo0(或特定接口编号)
ipv6 ospf 1 area 0
```

R2:

```
ipv6 router ospf 1
router-id 2.2.2.2
int fa0/0(或特定接口编号)
ipv6 ospf 1 area 0
int lo0(或特定接口编号)
ipv6 ospf 1 area 0
```

- 自 R1 向 R2 的IPv6环回接口发出 `ping` 操作，以测试连通性
- 执行一个 `show ipv6 route` 命令，来验证有通过OSPFv3接收到路由
- 执行一个 `show ipv6 protocols` 命令，来验证有配置OSPFv3且在设备上是活动的
- 执行命令 `show ipv6 ospf interface` 及 `show ipv6 ospf interface brief`，检查接口特定于OSPF的那些参数
- 在两台路由器上（直连接口）修改 Hello 包和死亡计时器: `ipv6 ospf hello` 及 `ipv6 ospf dead`
- 执行一下 `show ipv6 ospf 1` 命令，来查看路由进程参数

## 第14天 DHCP及DNS

### DHCP and DNS

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第14天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND1记诵指南
- 花15分钟在[subnetting.org](http://subnetting.org)上

主机使用动态主机配置协议（Dynamic Host Configuration Protocol, DHCP），紧接着加电启动后，收集到包括了IP地址、子网掩码及默认网关等初始配置信息。因为所有主机都需要一个IP地址，以在IP网络中进行通信，而DHCP就减轻了手动为每台主机配置一个IP地址的管理性负担。

域名系统（Domain Name System, DNS）将主机名称映射到IP地址，使得你可[www.in60days.com](http://www.in60days.com)输入到 web 浏览器中，而无需输入寄存该站点的服务器IP地址。

今天将学到以下内容。

- DHCP操作, DHCP operations
- 配置DHCP, configuring DHCP
- DHCP故障排除, troubleshooting DHCP issues
- DNS操作, DNS operations
- 配置DNS, configuring DNS
- DNS故障排除, troubleshooting DNS issues

本课对应了以下CCNA大纲要求。

- 配置和验证DNS（IOS路由器）
  - 将路由器接口配置为使用DHCP, configure router interfaces to use DHCP
  - DHCP选项, DHCP options
  - 排除的地址, excluded addresses
  - 租期, lease time

## DHCP功能

## DHCP Functionality

### DHCP操作

#### DHCP Operations

DHCP通过在网络上给主机自动分配IP信息，简化了网络管理任务。分配的信息可以包括IP地址、子网掩码及默认网关，且通常实在主机启动时。

在主机第一次启动时，如其已被配置为采用DHCP（大多数主机都是这样的），它就会发出一个询问分配IP信息的广播报文。该广播将为DHCP服务器收听到，同时该信息会被中继。

Farai指出 -- "这是假定主机和DHCP服务器实在同一子网的情形，而如它们不在同一子网，就看下面的 `ip helper-address` 命令。"

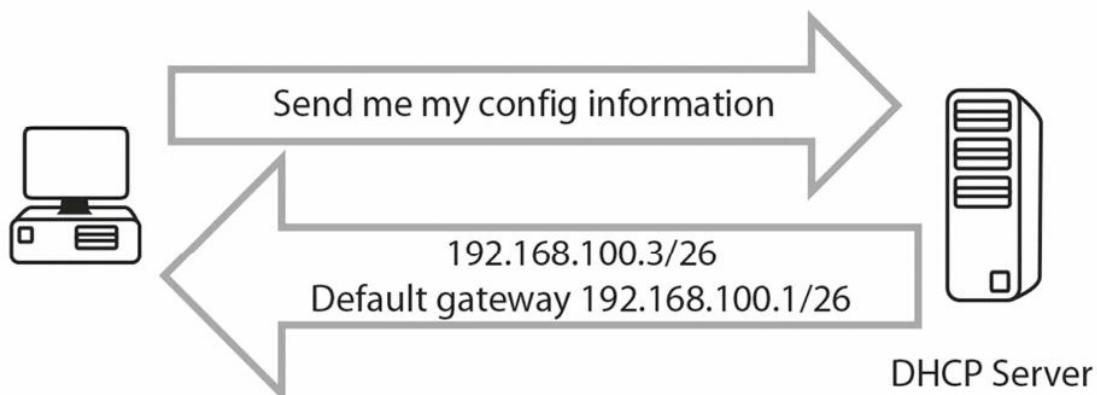


图14.1 -- 主机请求IP配置信息

DHCP具体使用UDP端口 67 和 68，来在网络上通信，同时，尽管在需要时路由器也可实现DHCP功能，但通常都会使用具体服务器作为DHCP服务器。在需要时，路由器同样可以配置为从DHCP服务器取得其接口IP地址，但很少这样做。配置这个特性的命令如下。

```
Router(config-if)#ip address dhcp
```

客户端的DHCP状态如下：

- 初始化，initialising
- 选择，selecting
- 请求，requesting
- 绑定，bound
- 更新，renewing
- 重绑定，rebinding

DHCP服务器可被配置为在一个名为租期的特定时期，赋予某台主机一个IP地址。租期可以是几个小时或几天。对于那些不能在网络上分配给主机的IP地址，可以也应该予以保留。这些保留的IP地址，将是已被路由器接口或服务器所使用的地址。如未能保留这些地址，就会看到网络上的重复IP地址告警，因为DHCP服务器已将配置给路由器或服务器的地址，分配给了主机。

下面的图14.2中，可以看到完整的DHCP请求和分配过程。

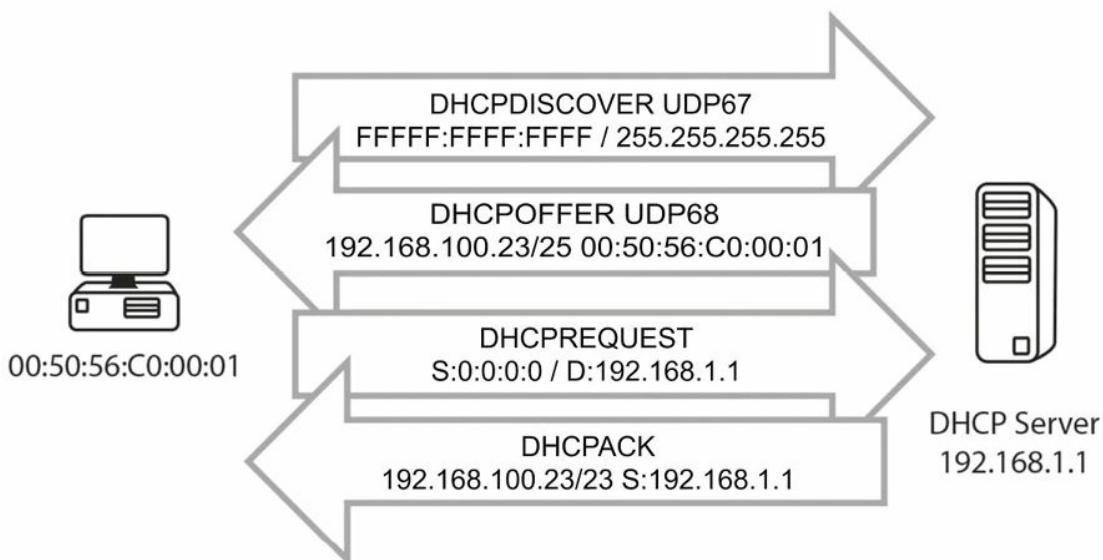


图14.2 -- DHCP 请求和分配过程

1. **DHCP发现数据包** (DHCP Discover packet) 当某台设备启动后，同时其被配置为通过DHCP取得一个地址时，就会发出一个自UDP端口 68 (UDP port 68 , bootpc ) 到UDP端口 67 (UDP port 67 , bootps ) 的广播数据包。该数据包将到达网络上的所有设备，包括任何位处网络上的可能的DHCP服务器。

**DHCP提议数据包** (DHCP Offer packet) ，本地网络上的DHCP服务器看到由客户端发出的广播发现报文 (the broadcasted Discover message) ，就用UDP源端口 bootps 67 及目的端口 bootpc 68，同样以广播地址的形式，发回一个响应 (就是DHCP提议数据包) 。之所以同样以广播地址形式，是因为客户端此时仍然没有IP地址，而无法接收单播数据包。

2. **DHCP请求数据包** (DHCP Request packet) ，一旦客户端工作站收到由DHCP服务器做出的提议 (an offer made by the DHCP server) ，它就会发出一个广播 (用于告知所有DHCP服务器，它已接受了来自某台服务器的提议) DHCP请求报文到某台特定的DHCP服务器，并再度使用UDP源端口 bootpc 68 及目的端口 bootps 67 。客户端可能会收到来自多台DHCP服务器的提议，但它只需单独一个IP地址，所以它必需选择一台DHCP服务器 (基于服务器标识) ，而选择通常都是按照“先到，先服务”原则完成的 (on a "first-come, first-served" basis) 。

3. **DHCP确认数据包** (DHCP ACK packet) ，选中的那台DHCP服务器发出另一个广播报文，来确认给那台特定客户端的地址分配，再度用到UDP源端口 bootps 67 及目的端口 bootpc 68 。

## DHCP的预订

### DHCP Reservations

DHCP服务器可被配置为以几种不同方式提供IP地址，包括下面这些。

- 动态分配, Dynamic allocation
- 自动分配, Automatic allocation
- 静态分配, Static allocation

### 动态分配

#### Dynamic allocation

通过DHCP指派地址的一个十分常用方法，就是采用动态分配过程，在此过程中，DHCP服务器配置为有着一个大的IP地址池，且根据客户端的请求，而为其分配地址池中的一个IP地址。在设备租期超时或设备离开网络时，该特定IP地址就被交还给DHCP服务器，之后就可被分配给另一客户端。

### 自动分配

#### automatic allocation

采用DHCP服务器分配IP地址的另一方式，叫做自动分配，该方式跟动态分配极为相似，但采用此种方式，DHCP服务器尝试维护一个所有过往分配地址清单，而如有某台“旧有”客户端请求一个IP地址，该客户端就会分配到一个跟以前一样的IP地址（也就是说，其曾于此前请求过一个IP地址）。自动分配是一种较为低效的分配IP地址方式，但如有着一个极大的可用IP地址池，这就是一种总能确保某网络中的客户端在每次开机时，获得同样IP地址的巧妙方法。

### 静态分配

#### Static allocation

DHCP服务器的IP地址静态分配，是指定义出一些期望在网络上出现的MAC地址，并手动为这些MAC地址都分配上一个唯一IP地址，因此就管理性地建立起一张 MAC-to-IP 关联表。这通常在服务器环境中用到，因为服务器必须使用可预期的IP地址，以可供访问。

## DHCP范围

### DHCP Scopes

打算配置一台DHCP服务器的网络管理员，作为配置过程的一部分，也需要配置DHCP范围。范围就是网络某个特殊部分的一组IP地址（A scope is a grouping of IP addresses for a particular section of the network）。而每个子网通常有着自己的范围。

范围也可以是可供DHCP服务器分配的一个连续地址池（a contiguous pool of addresses）。大多数DHCP服务器都提供了从地址池中排除一些地址的功能，以避免将这些地址动态地分配给客户端。这些排除的地址，就通常是那些手动分配给网络中服务器（及网络设备）的IP地址。

在定义的DHCP范围内部，可以配置诸如下面的一些参数。

- IP地址范围, IP address range
- 子网掩码, subnet mask
- 租约持续时间, lease duration
- 默认网关, default gateway
- DNS服务器, DNS server
- WINS服务器, WINS server

依据所使用的DHCP服务器，也可以使用不同参数，创建出不同的范围，而这通常与不同子网有关。

## DHCP租期

### DHCP Leases

DHCP所提供的主要优势之一，就是租借IP地址的能力，也就是说IP地址的分配是临时的。通常，当客户端离开网络时，其所分配到的特定IP地址将变成可用，并由DHCP服务器分配给其它设备。

DHCP租期关乎每次DHCP分配，限定允许用户使用一个分配到的IP地址多长时间。通常是在DHCP范围内对该参数进行管理性配置。每当有客户端重启后，它都必须再次从DHCP服务器请求一个IP地址。而DHCP服务器又通常被配置为给那台特定主机再度分配同样的地址并扩展租期。

工作站也能手动释放其IP地址，比如在以下情况下。

- 设备无限期关机，the device is turned off indefinitely
- 设备移至另一子网（比如，从有线网络移到无线网络），the device moves to another subnet(e.g, to a wireless network from a wired network)

租借过程有几个相关的计时器，因此可以肯定在所有网络设备上总是会有一个更新过的IP地址。下面是两个重要的DHCP计时器。

- 续借 (  $T_1$  ) 计时器** (renewal( $T_1$ ) timer, 默认是租期的一半)：在工作站取得一个IP地址后，此计时器就开始计时，当到达租期的 50% 时，DHCP客户端将向来源DHCP服务器重申租约。
- 重新绑定 (  $T_2$  ) 计时器** (rebinding( $T_2$ ) timer, 默认是租期的 87.5% )：这第二个计时器用在DHCP服务器未有在续借计时器超时后，进行回应或确认的情形。该计时器指出，如租期已过  $7/8$ ，那么客户端将尝试找到（发出一个DHCP请求）另一能够提供DHCP地址的DHCP服务器。

有了租借过程及上述有关计时器，就可以肯定总是会及时拥有一个IP地址，且连带不会有任何停止时间，同时自动地有着一种构建于DHCP过程中的冗余机制。

图14.3中展示了  $T_1$  及  $T_2$  计时器与租期的关系。

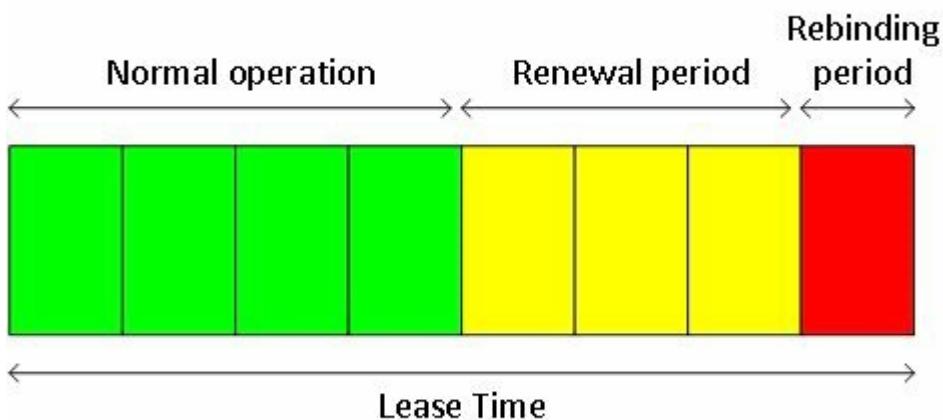


图14.3 -- DHCP 租期计时器

## DHCP选项

### DHCP Options

在DHCP中有一个特殊字段，可用于帮助扩展一些自动配置过程的性能。可在此字段中放入在DHCP RFC中给出的许多不同配置选项。

**注意：** BOOTP选项曾被称作“厂商扩展”。

DHCP提供了 256 选项值，其中仅 254 个是可用的，因为 0 是垫底选项，而 255 是最后选项（0 is the pad option and 255 is the end option）。许多DHCP选项都是通常所了解的经常使用到的参数，包括下面这些。

- 子网掩码，subnet mask
- 域名服务器，domain name server
- 域名，domain name

这些年来，已加入一些额外的DHCP选项，尤其是VoIP用途的那些选项，比如下面这些。

- 选项 129 : 呼叫服务器IP地址

- 选项 135：话机相关应用的HTTP代理服务器

所有这些选项都是直接在DHCP服务器上配置，但不是所有DHCP服务器都提供了设置DHCP选项的能力。如网络管理员要用到这些特性，就应该采用一种企业级别的DHCP服务器。在将小型路由器作为家庭办公环境的DHCP服务器是，就可能不会有这些功能上的益处。

## 配置DHCP

### Configuring DHCP

## 思科路由器上的DHCP服务器

### DHCP Servers on Cisco Routers

第一步就是在路由器上开启DHCP服务。这是通过使用 `service dhcp` 命令完成的，如下面所示（as exemplified below）。

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#service dhcp
```

下一步就是创建一个DHCP池，该DHCP池定义出将分配给客户端的IP地址池。在本例中，名为 `SUBNET_A` 的池将提供来自范围 `192.168.1.0/24` 的IP地址。

```
Router(config)#ip dhcp pool SUBNET_A
Router(dhcp-config)#network 192.168.1.0 255.255.255.0
Router(dhcp-config)#default-router 192.168.1.1
Router(dhcp-config)#dns-server 8.8.8.8
Router(dhcp-config)#domain-name Network+
Router(dhcp-config)#lease 30
```

该DHCP池配置模式（the DHCP Pool Configuration mode）同时也是配置其它DHCP选项的地方。在上面的配置输出中，配置了以下这些参数。

- 默认网关： `192.168.1.1` (指派到将该路由器作为DHCP服务器所服务网络中的路由器接口地址)
- DNS服务器： `8.8.8.8`
- 域名： `Network+`
- 租期： `30 天`

在需要时，也可以配置一些从 `192.168.1.0/24` 范围中排除的地址。我们就说要排除路由器接口IP地址（`192.168.1.1`）及 `192.168.1.250` 到 `192.168.1.255` 地址范围，从该范围就可手动为网络中的服务器分配地址。这是通过下面的配置完成的。

```
Router(config)#ip dhcp excluded-address 192.168.1.1
Router(config)#ip dhcp excluded-address 192.168.1.250 192.168.1.255
```

可使用下面的命令来查看当前由该路由器DHCP服务器所服务的客户端。

```
Router#show ip dhcp binding
Bindings from all pools not associated with VRF:
IP address      Client-ID/ Lease expiration    Type      Hardware address/
192.168.1.2      Mar 02 2014 12:07 AM        Automatic  0063.6973.636f.2d63
```

在上面的输出中，由该DHCP服务器服务的是单独一台客户端，同时分到到DHCP范围的第一个非排除IP地址：192.168.1.2。还可以看到租期超时日期及设备MAC地址。

## 思科路由器上的DHCP客户端

### DHCP Clients on Cisco Routers

除了DHCP服务器功能，思科路由器同样允许将其接口配置为DHCP客户端。这就是说接口将使用标准DHCP过程，请求到一个地址，而在特定子网上的任何服务器，都能分配该IP地址。

将一个路由器接口配置为DHCP客户端的命令如下。

```
Router(config)#int FastEthernet0/0
Router(config-if)#ip address dhcp
```

一旦某台DHCP服务器分配了一个IP地址，在路由器控制台上就可以看到下面的通知消息（该消息包含了地址和掩码）。

```
*Mar 1 00:29:15.779: %DHCP-6-ADDRESS_ASSIGN: Interface FastEthernet0/0 assigned DHCP address 10.10.10.1
```

使用命令 `show ip interface brief`，就可以观察到该DHCP分配方式。

```
Router#show ip interface brief
Interface      IP-Address  OK? Method   Status          Protocol
FastEthernet0/0 10.10.10.2  YES  DHCP     up            up
FastEthernet0/1 unassigned  YES  unset    administratively down  down
```

## DHCP数据包分析

### DHCP Packet Analysis

为实际掌握在本模块中介绍的这些知识点，将生成一些上述示例中涉及到设备的流量捕获。在配置好DHCP服务器及客户端工作站启动起来后，就会发生4步的DHCP过程，可在下面的截屏中观察到。

Time	Source	Destination	Protocol	Length	Info
191.391000 0.0.0.0	255.255.255.255	DHCP	618	DHCP Discover - Transaction ID 0x166f	
191.421000 c2:00:27:bc:00:00	Broadcast	ARP	60	Who has 192.168.1.2? Tell 192.168.1.1	
193.398000 192.168.1.1	255.255.255.255	DHCP	342	DHCP Offer - Transaction ID 0x166f	
193.418000 0.0.0.0	255.255.255.255	DHCP	618	DHCP Request - Transaction ID 0x166f	
193.438000 192.168.1.1	255.255.255.255	DHCP	342	DHCP ACK - Transaction ID 0x166f	
193.448000 c2:02:27:bc:00:00	Broadcast	ARP	60	Gratuitous ARP for 192.168.1.2 (Reply)	

图14.4 -- DHCP 4步过程

下面可以观察到DHCP发现数据包所包含的部分。

```

Frame 48: 618 bytes on wire (4944 bits), 618 bytes captured (4944 bits) on interface 0
Ethernet II, Src: c2:02:27:bc:00:00 (c2:02:27:bc:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x0000166f
  Seconds elapsed: 0
  Bootp flags: 0x8000 (Broadcast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: c2:02:27:bc:00:00 (c2:02:27:bc:00:00)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
  Option: (57) Maximum DHCP Message size
  Option: (61) Client identifier
  Option: (12) Host Name
  Option: (55) Parameter Request List
  Option: (255) End
  Padding

```

图14.5 -- DHCP发现数据包

正如你在截屏中看到的，该数据包（DHCP Discover packet）是由客户端发出，将其广播到网络上（目的地址是 255.255.255.255）。同时还看到其报文类型为“Boot Request (1)”。

下一个数据包就是DHCP提议数据包（DHCP Offer packet），如下面所示。

```

Frame 50: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on interface 0
Ethernet II, Src: c2:00:27:bc:00:00 (c2:00:27:bc:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 192.168.1.1 (192.168.1.1), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootps (67), Dst Port: bootpc (68)
Bootstrap Protocol
  Message type: Boot Reply (2)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x0000166f
  Seconds elapsed: 0
  Bootp flags: 0x8000 (Broadcast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 192.168.1.2 (192.168.1.2)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: c2:02:27:bc:00:00 (c2:02:27:bc:00:00)
  Client hardware address padding: 000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
  Option: (54) DHCP Server Identifier
  Option: (51) IP Address Lease Time
  Option: (58) Renewal Time Value
  Option: (59) Rebinding Time Value
  Option: (1) Subnet Mask
  Option: (3) Router
  Option: (6) Domain Name Server
  Option: (15) Domain Name
  Option: (255) End
  Padding

```

图14.6 -- DHCP提议数据包

该数据包是由服务器（源IP：192.168.1.1）发出到广播地址（目的地址：255.255.255.255），同时包含了提议的IP地址（192.168.1.2）。同时也可看到报文类型为“Boot Reply(2)”。

第三个数据包是DHCP请求数据包（DHCP Request packet）。

```

Frame 51: 618 bytes on wire (4944 bits), 618 bytes captured (4944 bits) on interface 0
Ethernet II, Src: c2:02:27:bc:00:00 (c2:02:27:bc:00:00), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
Internet Protocol Version 4, Src: 0.0.0.0 (0.0.0.0), Dst: 255.255.255.255 (255.255.255.255)
User Datagram Protocol, Src Port: bootpc (68), Dst Port: bootps (67)
Bootstrap Protocol
  Message type: Boot Request (1)
  Hardware type: Ethernet
  Hardware address length: 6
  Hops: 0
  Transaction ID: 0x0000166f
  Seconds elapsed: 0
  Bootp flags: 0x8000 (Broadcast)
  Client IP address: 0.0.0.0 (0.0.0.0)
  Your (client) IP address: 0.0.0.0 (0.0.0.0)
  Next server IP address: 0.0.0.0 (0.0.0.0)
  Relay agent IP address: 0.0.0.0 (0.0.0.0)
  Client MAC address: c2:02:27:bc:00:00 (c2:02:27:bc:00:00)
  Client hardware address padding: 00000000000000000000000000000000
  Server host name not given
  Boot file name not given
  Magic cookie: DHCP
  Option: (53) DHCP Message Type
  Option: (57) Maximum DHCP Message size
  Option: (61) Client identifier
  Option: (54) DHCP Server Identifier
  Option: (50) Requested IP Address
    Length: 4
    Requested IP Address: 192.168.1.2 (192.168.1.2)
  Option: (51) IP Address Lease Time
  Option: (12) Host Name
  Option: (55) Parameter Request List
  Option: (255) End
  Padding

```

图14.7 -- DHCP请求数据包

DHCP请求数据包是由客户端发出到广播地址。可以看到报文类型是 Boot Request(1)。该数据包与最初的DHCP发现数据包类似，但包含了一个非常重要的字段，就是 50 选项: 被请求的IP地址 ( 192.168.1.2 )

( a very important field, which is Option 50: Requested IP Address(192.168.1.2) )。这就是在DHCP提议数据包中由DHCP服务器所提供的同一IP地址，而该客户端对其进行了确认和接受。

DHCP分配过程的最后数据包就是由服务器发出的DCHP确认数据包了(the DHCP ACK packet)。

```

    □ Option: (53) DHCP Message Type
      Length: 1
      DHCP: ACK (5)
    □ Option: (54) DHCP Server Identifier
      Length: 4
      DHCP Server Identifier: 192.168.1.1 (192.168.1.1)
    □ Option: (51) IP Address Lease Time
      Length: 4
      IP Address Lease Time: (2592000s) 30 days
    □ Option: (58) Renewal Time value
      Length: 4
      Renewal Time Value: (1296000s) 15 days
    □ Option: (59) Rebinding Time value
      Length: 4
      Rebinding Time Value: (2268000s) 26 days, 6 hours
    □ Option: (1) Subnet Mask
      Length: 4
      Subnet Mask: 255.255.255.0 (255.255.255.0)
    □ Option: (3) Router
      Length: 4
      Router: 192.168.1.1 (192.168.1.1)
    □ Option: (6) Domain Name Server
      Length: 4
      Domain Name Server: 8.8.8.8 (8.8.8.8)
    □ Option: (15) Domain Name
      Length: 8
      Domain Name: Network+
    □ Option: (255) End
      Option End: 255
      Padding

```

图14.8 -- DHCP确认选项数据包

该数据包发自DHCP服务器并被广播到网络上；其同样包含了在上面的截屏中所看到的一些额外字段。

- DHCP服务器标识：该DHCP服务器的IP地址（192.168.1.1）
- 路由器上配置的所有选项。
  - 租期：30天（以及派生出的早前讨论的过续租时间和重新绑定时间值）
  - 子网掩码：255.255.255.0
  - 默认网关（路由器）：192.168.1.1
  - DNS服务器：8.8.8.8
  - 域名：Network+

## DHCP故障排除

### Troubleshooting DHCP Issues

跟NAT一样，DHCP故障基本上总是因为错误配置造成的（开玩笑说就是第8层问题，意思是人为疏忽，jokingly referred to as Layer 8 issue, meaning somebody messed up）。

命令 `service dhcp` 默认是开启的，但有些时候其已被网络管理员因为某些原因关闭了。（作者就曾遇到过有管理员在他们的路由器上敲入 `no ip routing` 命令后因为紧急的路由故障打电话给思科 -- 真的！）

如在另一子网上使用一台服务器来管理DHCP配置，就要允许路由器放行DHCP数据包。在地址分配过程中，DHCP用到广播报文（而路由器是不会转发广播报文的），那么就需要将DHCP服务器的IP地址加入到路由器，以令到路由器将该广播报文作为单播数据包进行转发。命令 `ip helper-address` 就可以实现这点。这是另一个考试喜欢的问题哦。

同样可以使用下面的 `debug` 命令作为排错过程中的部分。

```
debug ip dhcp server events  
debug ip dhcp server packet
```

## DNS操作

### DNS Operations

DNS将主机名映射到IP地址（而不是反过来）。这就允许你在web浏览器中浏览一个网址，而无需输入服务器IP地址。

在主机或路由器想要将一个域名解析到IP地址（或反过来将IP地址解析到域名时），DNS用到UDP 53号端口。而在两台DNS服务器之间打算同步或分享它们的数据库时，就使用TCP 53号端口。

## 配置DNS

### Configuring DNS

如想要容许路由器找到web上的某台DNS服务器，就使用命令 `ip name-server 1.1.1.1`，或是服务器相应的地址。

也可以将某个主机名设置到路由器上的一个IP地址表中来节省时间，或是令到更易于记住要 `ping` 的或是连接到的哪台设备，如下面的输出所示。

```
Router(config)#ip host R2 192.168.1.2  
Router(config)#ip host R3 192.168.1.3  
Router(config)#exit  
Router#ping R2  
Router#pinging 192.168.1.2  
!!!!
```

## DNS故障排除

### Troubleshooting DNS Issues

路由器配置默认将会有个 `ip domain-lookup` 命令。如此命令已被关闭，则DNS将不工作。某些时候路由器管理员会因避免在输入错误命令时，等待路由器执行数秒DNS查询，而关闭该命令。可通过下面的命令关闭DNS查询。

```
Router(config)#no ip domain-lookup
```

访问控制清单（access control lists, ACL）常常拦阻DNS，那么这是另一个故障原因。使用命令 `debug domain`，可在路由器上对DNS进行调试。

## 第14天问题

1. DHCP simplifies network administrative tasks by automatically assigning \_\_\_\_\_ to hosts on a network.
2. DHCP uses UDP ports \_\_\_\_\_ and \_\_\_\_\_ .
3. What are the six DHCP states for clients?
4. Which command will prevent IP addresses 192.168.1.1 to 192.168.1.10 from being used in the pool?
5. Which command will set a DHCP lease of 7 days, 7 hours, and 7 minutes?
6. Which command will enable the router to forward a DHCP Broadcast as a Unicast?
7. DNS uses UDP port \_\_\_\_\_ .
8. Which command will set a DNS server address of 192.168.1.1 on your router?
9. If the \_\_\_\_\_ - \_\_\_\_\_ command has been disabled on your router, then DNS won't work.
10. Which command will debug DNS packets on your router?

## 第14天问题答案

1. IP information (IP addresses).
2. 67 and 68.
3. Initialising, Selecting, Requesting, Bound, Renewing, and Rebinding.
4. The ip dhcp excluded-address 192.168.1.1 192.168.1.10
5. The lease 7 7 7 command under DHCP Pool Configuration mode.
6. The ip helper-address command.
7. 53.
8. The ip name-server 192.168.1.1 command.
9. ip domain-lookup .
10. The debug domain command.

## 第14天实验

### 路由器上的DHCP实验

#### 拓扑

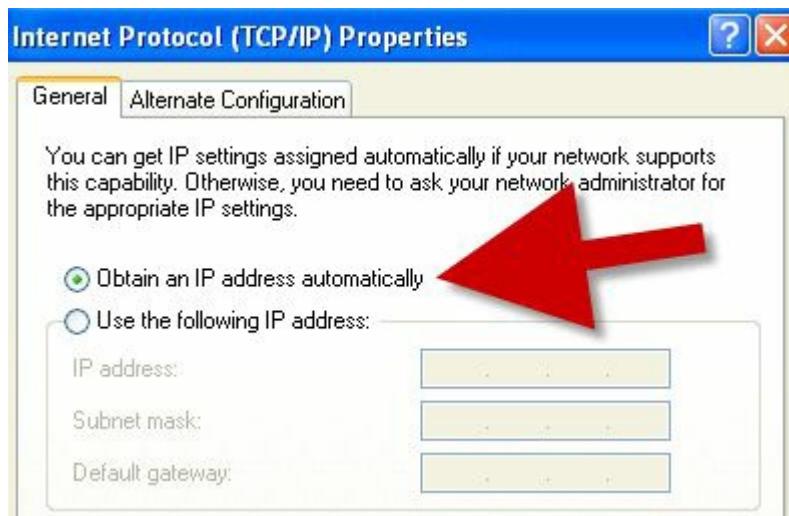


#### 实验目的

学习可如何将路由器用作DHCP服务器。

#### 实验步骤

1. 如你使用着家用电脑或笔记本电脑，就将网络适配器设置为自动获取IP地址。在Packet Tracer中也可这样设置。让后使用交叉线将PC连接到路由器的以太网端口。



1. 将IP地址 172.16.1.1 255.255.0.0 加入到路由器接口。如忘记了这个怎么配置，就请看看前面的实验。要确保 no shut 该接口。
2. 配置DHCP地址池。接着为地址配置一个 3 天 3 小时 5 分的租期。最后将 1 到 10 的地址排除在分配给主机的地址之外。假设这些地址已为其它服务器或接口使用。

```
Router#conf t
Router(config)#ip dhcp pool 60days
Router(dhcp-config)#network 172.16.0.0 255.255.0.0
Router1(dhcp-config)#lease 3 3 5      -- command won't work on Packet Tracer
Router1(dhcp-config)#exit
Router(config)#ip dhcp excluded-address 172.16.1.1 172.16.1.10
Router(config)#
```

1. 执行一个 ipconfig /all 命令，查看是否有IP地址分配到PC。如旧地址仍在使用，就需要执行一下 ipconfig /renew 命令。

```
PC>ipconfig /all
Physical Address.....: 0001.C7DD.CB19
IP Address.....: 172.16.0.1
Subnet Mask.....: 255.255.0.0
Default Gateway.....: 0.0.0.0
DNS Servers.....: 0.0.0.0
```

1. 如想要的话，可回到DHCP地址池配置模式（DHCP Pool Configuration mode），加入一个默认网关及 DNS服务器地址，它们也将在主机PC上得到设置。

```
Router(config)#ip dhcp pool 60days
Router(dhcp-config)#default-router 172.16.1.2
Router(dhcp-config)#dns-server 172.16.1.3
PC>ipconfig /renew
IP Address.....: 172.16.0.1
Subnet Mask.....: 255.255.0.0
Default Gateway.....: 172.16.1.2
DNS Server.....: 172.16.1.3
```

## 路由器上的DNS实验

### DNS on a Router lab

在一台有着某种到互联网连通性的路由器上完成此实验。确保该路由器可以 ping 通比如Google公司的 DNS服务器 8.8.8.8 这样的公网IP地址。将该地址配置为一个名字服务器。

```
ip name-server 8.8.8.8
```

接着尝试解析一些公网网站名字，比如通过 ping www.cisco.com 。

请访问[www.in60days.com](http://www.in60days.com), 观看我是怎么完成这个实验的。

# 第15天 一二层排错

## Layer 1 and Layer 2 Troubleshooting

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

# 第15天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND1记诵指南
- 在[subnetting.org](https://subnetting.org)花15分钟

先前数课中已涵盖ICND1排错的许多要求，尤其是关于ACLs及IP分址方面。许多可能的故障都发生在一二层，一二层故障及其原因，是今天这课的重点。

LAN交换是一种用在局域网中的包交换形式。LAN交换是在数据链路层的硬件中完成的。正因为其是基于硬件的，使用到被称为介质访问控制地址（Media Access Control addresses, MAC地址）的硬件地址。**LAN交换机使用MAC地址来转发帧。**

今天将学习以下内容。

- 物理层排错
- VLAN、VTP及中继概述
- VLANs排错
- 运用 show vlan 命令

本模块对应了下面的CCNA大纲要求。

- 一层故障的排错及处理
  - 组帧，framing
  - 循环冗余校验，CRC
  - 畸形帧，runts
  - 巨大帧，giants
  - 丢掉的数据包，dropped packets
  - 晚发冲突，late collision
  - 输入/输出错误，Input/Output errors
- VLAN故障的排错和处理
  - 验证VLANs已配置, verify that VLANs are configured
  - 验证端口成员关系是正确的, verify that port membership is correct

- 验证配置了IP地址, verify that the IP address is configured
- 思科交换机上中继问题的排错和处理
  - 验证中继状态是正确的, verify that the trunk states are correct
  - 验证封装类型是正确配置的, verify that encapsulation is configured correctly
  - 验证那些VLANs是被放行的, verify that VLANs are allowed

## 物理层上的排错

### Troubleshooting at the Physical Layer

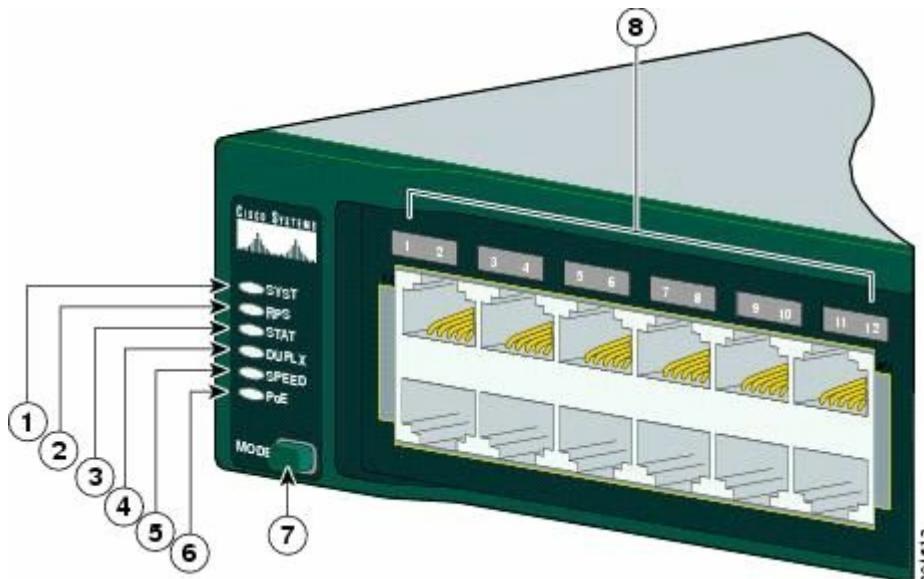
思科IOS交换机支持好几个可用于一层，或至少怀疑是一层故障排错的命令。但是，除了对这些软件命令工具要熟悉外，对可用于链路状态排错，或示出错误情形的物理指示器（也就是那些LEDs）的掌握，也是重要的。

### 使用发光二极管（LEDs）的链路状态排错

#### Troubleshooting Link Status Using Light Emitting Diodes(LEDs)

如可物理接触到交换机，那么LEDs就会是一项有用的排错工具。不同类型的思科Catalyst交换机，提供了不同的LED能力。掌握这些LEDs的意义，是Catalyst交换机链路状态及系统排错所不可或缺的部分。思科Catalyst交换机有一些可用于判断链路状态及其它一些诸如系统状态等变量的前面板LEDs。

经由Google "Catalyst 2960 Switch Hardware Installation Guide"，来查看Catalyst 2960型号交换机的思科文档。该安装和配置手册包含了数百页的注记、建议和技术信息。通读一下该文档是值得的，但不要期望从该文档得到CCNA考试大纲的内容（CCNA考试大纲内容在这本书才有）。



`1`	系统LED	`5`	速率LED
`2`	冗余电源 (redundant power supply, RPS) LED	`6`	PoE LED
`3`	状态LED	`7`	模式按钮
`4`	双工LED	`8`	端口LEDs

图15.1 -- 思科2960交换机LEDs，图片版权归思科系统公司

PoE LED只有在Catalyst 2960交换机型号上才能找到。

### 系统LED

#### System LED

系统LED表明系统通电了的（或是未通电）且正常发挥功能。

下表15.1列出了系统LED颜色及其所表明的状态。

表15.1 -- 系统LED

系统LED颜色	系统状态
不亮	系统未通电
绿色	系统运行正常
琥珀色 (amber)	系统已通电，但未有正确发挥功能

### 冗余电源LED

#### RPS LED

冗余电源LED只在那些有着冗余电源的交换机上才有。下表15.2列出了RPS LED的颜色和其意义。

表15.2 -- 冗余电源LEDs

RPS LED颜色	状态
绿色	连接了RPS，且RPS在需要时就可提供后备电力
绿色闪烁 (Blinking Green)	连接了RPS，但因为其正为另一设备提供电力（冗余已被分配给一台相邻设备）而不可用
琥珀色	RPS处于待机模式或故障状态（in standby mode or in a fault condition）。按下RPS上的Standby/Active按钮，此时该LED应变成绿色。如未变成绿色，则该RPS风扇可能损坏。请联系思科系统公司。
琥珀色闪烁	交换机内部电源失效，且正由RPS给交换机供电（冗余电源已分配给该设备）

### 端口LEDs及其模式

#### Port LEDs and Modes

端口LEDs提供了一组端口或单个端口的信息，如下表15.3所示。

表15.3 -- 端口LEDs的模式

所选模式LED	端口模式	说明
1 -- 系统		
2 -- RPS		RPS状态
3 -- 状态	端口状态	端口状态（默认模式）
4 -- 双工	端口双工情况	双工模式：全双工或半双工
5 -- 速率	端口速率	端口运行速率：10, 100或1000Mbps
6 -- PoE	PoE端口供电	PoE状态
7 -- 模式		循环显示端口状态、双工模式及速率LEDs
8 -- 端口		依不同模式有不同含义

不停按下模式按钮（the Mode button）可在不同模式之间循环，直到需要的模式设置。这会改变端口LED颜色的意义，如下表15.4所示。

表15.4 -- 模式设置

端口模式	LED颜色	系统状态
状态	不亮	未插入网线或管理性关闭
	绿色	有链路且链路无问题
	绿色闪烁	活动的：端口在发送或接收数据
	绿色 琥珀色交替闪烁	链路故障（link fault）：出现可影响连通性的错误帧，以及过多的冲突、循环冗余校验（CRC），同时将对以太网的alignment及jabber问题进行检测（ <a href="#">以太网错误描述</a> , <a href="#">以太网错误</a> ）
	琥珀色	端口被生成树协议（Spanning Tree Protocol, STP）阻塞而未转发数据。注意：在某端口重新配置后，端口LED将保持琥珀色30秒，因为STP会检查网络拓扑有没有可能的环回。
	琥珀色闪烁	端口被STP阻塞同时没有发送或接收数据。
	双工	不亮 端口以半双工方式运行。 绿色 端口以全双工方式运行。
速率	10/100及10/100/1000Mbps端口	
	不亮	端口以10Mbps速率运行。
	绿色	端口以100Mbps速率运行。
	绿色闪烁	端口以1000Mbps运行。
	SPF(小封装可插拔, small form-factor pluggable, SPF)端口	
	不亮	端口以10Mbps速率运行。
	绿色	端口以100Mbps速率运行。
PoE	绿色闪烁	端口以1000Mbps速率运行。
	不亮	PoE关闭。如被供电设备从交流电源取得电力，那么就算被供电设备是连接到交换机的，PoE端口LED也会不亮。
	绿色	PoE开启。端口LED只在该交换机端口供电时才亮起绿色。
	绿色和琥珀色交替亮起	因为向被供电设备提供电力会超出交换机电源功率，而将PoE禁用了。Catalyst 2960-24PC-L、2960 48PST-L、2960-48PST-S及2960-24PC-S可以提供最高370W的电力。而Catalyst 2960-24LT-L和2960-24LC-S交换机只能提供最高124W的电力。

	琥珀色闪烁	<p>PoE因为故障而关闭。</p> <p>注意：在做网线不合规及加电的设备连接到PoE端口(non-compliant cabling or powered devices are connected to a PoE port)时，都会导致PoE故障。在将思科认证的IP电话、无线接入点或符合IEEE 802.3af规范的设备连接到PoE端口时，只能使用标准规范的做网线方式。必须将导致PoE故障的网线或设备从网络上移除。</p> <p>(Only standard-compliant cabling can be used to connect Cisco prestandard IP phones, wireless access points, or IEEE 802.3af-compliant devices to PoE ports. You must remove the cable or device that cause the PoE fault from the network.)</p>
	琥珀色	端口的PoE已被关闭。默认PoE是开启的。

除了要掌握这些不同颜色的意义外，重要的是掌握修复这些故障所需的做法。比如，假设正在对一台 Catalyst 6500 交换机进行排错，并注意到管理引擎（或交换模块）的状态LEDs是红色或不亮。在此情况，就可能是该模块脱离了其插槽，或是因为某个新模块没有正确插入到其机架上。那么建议做法就是重新插好该模块。而有些时候，还需要重启整个系统。

在某条链路或某个端口LED颜色不是绿色时，就往往表明某种失效或其它故障，而重要的是记住**一条链路发出绿光也并不总意味着网线是完全没有问题的**。比如，只有一根线坏掉或是一个关闭端口，就可能导致一侧显示线路绿色光而另一侧不显示绿色光。这可能是因为网线出现了物理压力而引起该网线具备临界级别的功能。在这种情况下，就可以使用CLI来完成额外的派错。

## 线缆故障排错

### Troubleshooting Cable Issues

在对线路故障进行排错时（一层排错），因为可以直接受查看及检查网线，所以通常都是非常容易找到问题的。但是，有些时候线路问题可以是看不见的，所以就不得不完成一个系统性的排错过程，以确保问题确实是在一层当中。一个一般性建议就是在进行复杂步骤之前，先适当地对所有网线进行测试（however, sometimes cabling problems can be invisible, so you will have to engage in a systematic troubleshooting process to make sure the problem is really localised at Layer 1. A general recommendation is to properly test all cabling before engaging in a complex infrastructure implementation）。下面是一些常见的线路问题。

- 有插入网线但没有连接
- 有插入网线且有得到连接，但那条连接的吞吐量极低
- 所有都工作正常，但突然没有了连接，接着又恢复正常，接着又无连接（也就是抖动，flapping）
- 间歇的连通性，看起来工作正常，但信号一次又一次地丢失

一些针对这些问题的建议测试有下面这些。

- 检查交换机链路灯是否亮起
- 检查链路灯有没有间隙地开启和关闭
- 检查网线压得对不对
- 检查网线未有物理损坏
- 检查网线不是过长（这会导致信号恶化）
- 检查网线连接头没有问题（可能需要另外的连接头）
- 检查电线针脚顺序是正确的（如是铜线的话）

如要确认遇到的不是网线故障，最简单的做法就是换一根好的网线，在进行同样的测试。这很容易办到同时可马上解决问题，而无需在排错过程中耗费过多的时间和资源。

**注意：**就算是全新的网线有时也有问题，所以不要假定一根新网线就会如预期那样起作用。

## 模块故障的排错

### Troubleshooting Module Issues

大多数企业网络中用到的路由器和交换机都提供了铜质端口连接性，也提供了将不同类型的发送接收器进行板上组装的专门接口。这些发送接收器通常用于光纤连接，不过也有铜质的发送和接收器（copper-compatible transceivers）。

光纤连接可在很长距离上运作，同时这些特定端口通常都是模块化的，需要一个兼容的SFP（小型可插入发送和接收器），如图15.2中给出的那样。



图15.2 -- SFP 模块

尽管SFP模块看起来都一样，但依据所使用的连接类型，根据下面这些参数，来选用适当的SFP模块。

- 介质类型：光纤还是铜缆, optical fibre or copper
- 光纤类别：单模还是多模光纤 (single-mode or multi-mode fibre)
- 带宽, bandwidth
- 波长, wavelength
- 光纤规格, core size
- 模带宽, modal bandwidth
- 运行距离, operating distance

**注意：**在为网络采购收发器时，应总要对设备端口、模块类型及所使用的光纤进行检查。

任何时候都可将收发器插入到网络设备（比如交换机、路由器、防火墙等），或从其上拔下，而无需重启设备。在没有连接时，不会在SFP模块上看到活动，而这就是在可接触到设备时最容易排错的故障了。

此外，插入光纤将激活那个端口，但又因为各种不同故障而致使连通性受到影响（比如性能恶化或是间歇的连通性），或是没有连通性。此时，有着下面几种可供采行的方法。

- 依据收发器的类型，检查使用的线缆类型是正确的（多模还是单模）
- 检查线缆是完好的，要使用那些专门的光纤测试工具

- 检查所使用的收发器是正确的类型
- 检查收发器没有硬件故障（换另一个收发器并进行测试）
- 依据所使用的收发器和线缆类别，检查设备端口有配置上正确的参数

为令到连接停机时间最低，就应检测那些插入了SFP模块的端口，以观察出现在统计信息中的可能错误。而这可通过标准监测工具完成，最常用的就是SNMP。

## 使用命令行接口来对链路故障进行排错

### Using the Command Line Interface to Troubleshoot Link Issues

思科IOS Catalyst交换机上，可使用好几个命令行接口命令来对一层故障进行排错。常用的命令包括 `show interfaces`、`show controllers` 以及 `show interface [name] counters errors` 命令。除了要知道这些命令，还要能解读这些命令的输出或所提供的信息。

`show interfaces` 命令是一个提供过剩信息的强大工具，提供包括以下这些信息。

- 交换机端口的管理状态
- 端口允许状态
- 介质类型（对于特定交换机及端口，for select switches and ports）
- 端口输入及输出数据包数目
- 端口缓存失效数及端口错误数
- 端口输入及输出错误
- 端口输入及输出队列丢失情况

下面是在一个GigabitEthernet交换端口上的 `show interfaces` 命令的输出。

```
Catalyst-3750-1#show interfaces GigabitEthernet3/0/1
GigabitEthernet0/1 is up, line protocol is down (notconnect)
Hardware is GigabitEthernet, address is 000f.2303.2db1 (bia 000f.2303.2db1)
MTU 1500 bytes, BW 10000 Kbit, DLY 1000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, Loopback not set
Keepalive not set
Auto-duplex, Auto-speed, link type is auto, media type is unknown
input flow-control is off, output flow-control is desired
ARP type: ARPA, ARP Timeout 04:00:00
Last input never, output never, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
    0 packets input, 0 bytes, 0 no buffer
    Received 0 broadcasts (0 multicasts)
    0 runts, 0 giants, 0 throttles
    0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
    0 watchdog, 0 multicast, 0 pause input
    0 input packets with dribble condition detected
    0 packets output, 0 bytes, 0 underruns
    0 output errors, 0 collisions, 1 interface resets
    0 babbles, 0 late collision, 0 deferred
    0 lost carrier, 0 no carrier, 0 PAUSE output
    0 output buffer failures, 0 output buffers swapped out
```

多数思科Catalyst交换机端口默认都是 `notconnect` 状态，如同该命令打印输出的第一行所示。但如果网线从该端口拔出或未有正确连接，端口状态也会转换成该状态。在连接的网线有问题或是网线另一端没有插入到活动端口或设备（比如某台工作站插入交换机的端口是关闭的）时，将同样显示为 `notconnect`。

**注意：** 在对GigabitEthernet端口排错时，若两端使用了不正确千兆接口转换器（Gigabit Interface Converters, GBICs），也会导致 `notconnect` 端口状态。

输出的第一部分是该命令打印出的第一行（也就是 `[interface] is up`），表示特定接口的物理层状态。输出的第二部分（也就是 `line protocol is down`）表明该接口的数据链路层状态。而如该部分指示 `up`，就意味着该接口可发送和接收保持活动信号。**记住交换机端口可能在物理层是起来的，却在数据链路层是宕掉的**，比如，当端口是一个SPAN(Switch)目的端口时，或者本地端口连接到一台CatOS交换机的一个关闭的端口时，都会这样（*if this indicates an "up", then it means that the interface can send and receive keepalives. Keep in mind that it is possible for the switch port to indicate that the Physical Layer is up while the Data Link Layer is down, for example, such as when the port is a SPAN destination port(for sniffer traffic) or if the local port is connected to a CatOS(older switch operating system) switch with its port disabled*）。

输入队列（the Input queue）表明因为超出最大队列尺寸而丢弃帧的实际数量。其中的“`flushes`”列对 Catalyst 6000系列交换机上的SPD（selective packet discard, 选择性数据包丢弃, [Selective Packet Discard, Understanding Selective Packet Discard](#)）丢弃数据包进行计数。SPD在CPU超负荷时将低优先级的数据包丢弃，从而为高优先级数据包节省下处理能力。`show interfaces` 命令输出中的 `flushes` 计数器随SPD部分而增长，SPD对路由器IP处理队列运用一种选择性数据包丢弃策略（a selective packet drop policy）。因此SPD仅用在进程交换流量上（*applies only to process-switched traffic*）。

总的输出丢弃数量（the total output drops）表示由于输出队列充满而丢弃的数据包数量。输出丢弃经常在正将来多个高带宽入站链路（比如几条千兆以太网链路）的流量，交换到单个的出站低带宽链路（比如一条快速以太网）时见到。输出丢弃的增长，是因为入站和出站带宽不匹配而造成的超出流量将该接口击败造成的（*this is often seen when traffic from multiple inbound high-bandwidth links(e.g., GigabitEthernet links) is being switched to a single outbound lower-bandwidth(e.g., a FastEthernet link). The output drops increment because the interface is overwhelmed by the excess traffic due to the speed mismatch between the inbound and outbound bandwidths*）。

`show interfaces` 的输出中的一些其它接口相关的可分析的，同时在一、二层排错中非常有用的术语，有下面这些。

- **帧数目**（frame number）：该字段给出了接收到的带有不正确的CRC及大小不是整数个字节的数据包数目。这通常是由不正常功能的以太网设备（硬件错误）而导致的冲突造成的。
- **循环冗余校验**（CRC）：该字段表示由发送设备生成的CRC与接收设备计算出的校验和不一致。这通常表示LAN上的传输错误、冲突或是系统传输不良数据。
- **畸形帧**（runts）：此字段表示由于比最小数据包大小还小而丢弃的包数量。在以太网段上，比64字节还小的包都被看作畸形帧。
- **巨大帧**（giants）：此字段表示由于比最大数据包大小还大而丢弃的包数量。在以太网段上，比起1518字节还大的数据包被看作巨大帧。
- **晚发冲突**(late collisions): 晚发冲突通常在网线过长或网络中有过多中继器时。冲突数目反应了因为以太网冲突而导致的重传报文数目。而这通常是由于对LAN的过度扩展造成的。
- **输入错误**（input errors）：该字段提供所有畸形帧、巨大帧、CRC错误帧、超出帧（overruns）及忽略数据包的总数。
- **输出错误**（output errors）：该字段提供了阻止数据报最后从接口发出的错误总数（*this field provides the total sum of all errors that prevented the final transmission of datagrams out of the interface*）。

除了 `show interfaces` 命令，命令 `show interfaces [name] counters errors` 也可以用来查看接口错误及促进一层的排错。下面就是命令 `show interface [name] counters errors` 打印出的输出。

```
Catalyst-3750-1#show interfaces GigabitEthernet3/0/1 counters errors
Port      Align-Err   FCS-Err   Xmit-Err   Rcv-Err UnderSize
Gi3/0/1      0          0          0          0          0
Port      Single-Col Multi-Col Late-Col Excess-Col Carri-Sen Runts
Gi3/0/1      0          0          0          0          0          0
Port      Giants
Gi3/0/1      0
```

接下来的部分对命令 `show interfaces [name] counters errors` 输出中的一些错误字段，以及这些字段所表示的故障或问题，进行讲述。

`Align-Err` 字段反应了接收到的没有以偶数个字节结束，同时有着错误CRC帧的数目。这些错误通常是由不匹配的双工不匹配或物理问题造成，比如线路问题、坏端口或坏网卡造成的。在网线头一次插入端口时，一些这类错误就会发生。此外，如有集线器连接到端口，集线器上其它设备之间的冲突也会造成这些错误。

`FCS-Err` 字段反应了有帧校验序列（Frame Check Sequence, FCS）错误的大小有效(valid-sized)、没有组帧错误的帧数目。这通常是因为物理故障，诸如网线做得不好、坏端口或者坏网卡造成的。此外，该字段下的非零值，可能表明存在双工不匹配。

`Xmit-Err` 字段中的非零值是内部发送（Tx）缓冲器充满的表征。当有来自多个入站高带宽链路（比如多条GigabitEthernet链路）的流量正转发到单一的出站低带宽链路（比如一条FastEthernet链路）时，通常会见到这种情形。

字段 `Rcv-Err` 表示收到帧错误的总和。该计数器在接口收到诸如畸形帧、巨大帧或FCS错误帧时增长。

`UnderSize` 字段在交换机接收到长度小于 64 字节的帧时增长。这通常是由故障发送设备造成的。

不同的 `collisions` 字段表示接口上的冲突。接口上的冲突通常发生在半双工以太网上，而这在现代网络中几乎是不存在的。因此，这些计数器对于全双工链路不应增长。如果这些计数器下出现了非零数值，那么通常表明存在全双工不匹配故障。当探测到全双工不匹配时，交换机会在控制台或日志中打印出类似于下面的消息。

```
%CDP-4-DUPLEX_MISMATCH: duplex mismatch discovered on FastEthernet0/1 (not full duplex), with R2 FastE
```

如同将在生成树协议（Spanning Tree Protocol, STP）章节中介绍的那样，全双工不匹配可能导致在某端口连接到另一交换机时，交换网络中的STP循环。这些不匹配可通过手动配置交换机端口的速率和双工方式予以解决。

当以太网控制器每次想要在半双工连接上发送数据时，`Carri-Sen`（载波侦听，carrier sense）都会增长。控制器侦听线路并确保在传输前线路是空闲的。该字段下的非零值表示接口运行于半双工模式。这对半双工来说是正常的。

因为双工不匹配或其它物理层问题，比如坏网线、坏端口以及所连接设备上的坏网卡，也可能导致 `Runts` 字段下可以看到非零值。畸形帧是指所接收到的有着错误的CRC、小于最小IEEE 802.3帧大小，也就是以太网的64字节的那些帧。

最后，当接收到的帧超过IEEE 802.3最大帧大小，非巨大以太网（non-jumbo Ethernet, [Jumbo frame](#), [Linux\\_Jumbo\\_frame](#)）的 1518 字节时，并有着坏的FCS时，`Giants` 计数器就会增长。对于那些连接到某台工作站的端口或接口，该字段下的非零数值典型地是由所连接设备上的坏网卡导致的。不过，对于那些连接到另一交换机（比如通过中继链路）的端口或接口，如采用的是802.1Q封装方式，则该字段将会包含一个非零数值。**在802.1Q下，其打标记机制（the tagging mechanism）对帧进行了修改，因为中继设备插入了一个 4 字节的标记，并随后再度计算了FCS。**

对已有最大以太网帧大小的帧插入 4 字节后，就构成了一个 1522 字节的帧，那么接收设备就会将其看着是一个幼小巨大帧（a baby giant frame）。因此，尽管交换机仍将处理这些帧，该计数器将增长并包含一个非零值。为解决这个问题，802.3委员会建立一个名为802.3ac的小组来将以太网最大大小扩展到 1522 字节；这样以来，采行802.1Q中继时就不常见到该字段下的非零值了。

类似与 `show interfaces` 及 `show interfaces <name> counters errors` 命令所提供的信息，命令 `show controllers ethernet-controller <interface>` 也可以用来现实流量计数及错误计数信息。`show controllers ethernet-controllers <interface>` 命令的输出如下所示。

```
Catalyst-3750-1#show controllers ethernet-controller GigabitEthernet3/0/1
Transmit GigabitEthernet3/0/1    Receive
4069327795 Bytes            3301740741 Bytes
  559424024 Unicast frames      376047608 Unicast frames
  27784795 Multicast frames     1141946 Multicast frames
  7281524 Broadcast frames      1281591 Broadcast frames
    0 Too old frames          429934641 Unicast bytes
    0 Deferred frames         226764843 Multicast bytes
    0 MTU exceeded frames     137921433 Broadcast bytes
    0 1 collision frames       0 Alignment errors
    0 2 collision frames       0 FCS errors
    0 3 collision frames       0 Oversize frames
    0 4 collision frames       0 Undersize frames
    0 5 collision frames       0 Collision fragments
    0 6 collision frames
    0 7 collision frames      257477 Minimum size frames
    0 8 collision frames      259422986 65 to 127 byte frames
    0 9 collision frames      51377167 128 to 255 byte frames
    0 10 collision frames     41117556 256 to 511 byte frames
    0 11 collision frames     2342527 512 to 1023 byte frames
    0 12 collision frames     5843545 1024 to 1518 byte frames
    0 13 collision frames       0 Overrun frames
    0 14 collision frames       0 Pause frames
    0 15 collision frames
    0 Excessive collisions      0 Symbol error frames
    0 Late collisions          0 Invalid frames, too large
    0 VLAN discard frames      18109887 Valid frames, too large
    0 Excess defer frames        0 Invalid frames, too small
  264522 64 byte frames        0 Valid frames, too small
  99898057 127 byte frames
  76457337 255 byte frames      0 Too old frames
  4927192 511 byte frames      0 Valid oversize frames
  21176897 1023 byte frames      0 System FCS error frames
  127643707 1518 byte frames      0 RxPortFifoFull drop frames
  264122631 Too large frames
    0 Good (1 coll) frames
    0 Good (>1 coll) frames
```

**注意：**根据该命令执行所在平台的不同，上面的输出会略有不同。比如，Catalyst 3650系列交换机还包含了一个 `Discarded frames` 字段，该字段显示因资源不可用而导致的放弃传输尝试的帧总数（a `Discarded frames` field, which shows the total number of frames whose transmission attempt is abandoned due to insufficient resources）。该字段中出现了较大的数值就典型地表明存在网络壅塞故障（a network congestion issue）。在上面的输出中，应探究一下 `RxPortFifoFull drop` 帧字段，该字段表示因为入口队列充满而丢弃的接口所接收到的帧总数（the `RxPortFifoFull drop` frame field, which indicates the total number of frames received on an interface that are dropped because the ingress queue is full）。

## 端口配置排错

### Troubleshooting Port Configuration

各台网络设备都可以不同方式进行配置。多数类型的错误配置都产生网络中的问题，包括下面这些。

- 极低的流量吞吐，poor throughput
- 没有连通性，lack of connectivity

某台设备可以连接到网络，有着网络信号，同时可以与Internet及其它设备通信，却有着以持续的、易于重现方式的低通信性能。这种情况可能在正常运行中，包括与网络其它部分进行文件传输或其它类型的通信中出现。

在重大配置问题下，该故障可能以完全的连通性缺失，包括特定设备端口上连接信号灯不亮的形式出现 (with major configuration issues, the issue might manifest as lack of connectivity, including no link lights on the specific device ports)。有时连接灯亮起但仍然没有任何类型的连通性。这显示在网线上有信号，也就是说没有网线问题，而是在端口上的端口问题故障或其它问题。这就要对设备的配置进行问题调查。

配置端口时有几项不同的设置，包括下面这些。

- 速率，speed
- 双工，duplex
- 封装/VLAN，encapsulation/VLAN

大多数的这些参数都必须在链路两侧保持一致，要么通过手动配置，或是通过开启端口自动配置。如能探测到，自动配置方式将在链路上发送协商数据包，来探测另一端的各种能力，并就两端设备的最佳参数达成一致，以建立最优传输。问题在于有的时候自动配置并不会按照需求选择出最佳参数，所以就要对此进行检查并针对各种特定情形对端口进行手动配置。

如正对各个端口进行手动配置，那么要注意的第一个参数，就是接口速率。在**链路两端上的接口速率必须一致**。如在一侧配置不正确，链路就可以不会运作。另一个有关的设置就是**端口双工方式**，这可以配置为半双工或全双工。**可以在一侧配置为半双工、另一侧配置为全双工**，这样做尽管链路也会运行，但通过流量将极大地受到影响，因为两侧都期望以不同方式处理通信。同时这样做也会导致对那条链路上的传输造成影响的冲突。为令到流量能够尽可能有效地进行发送，请确保两侧都使用同样的双工设置。

而如在企业级环境下运营 (if you are operating in an enterprise-level environment)，就可能需要使用不同的VLANs对流量进行分段。所有交换机在这方面都必须准确配置，如此一来所有交换机端口都分配到正确的VLAN中。而如直接将配置到使用不同VLAN IDs的端口直接连接，那么就算物理层上显示没有故障，二层上的通讯仍将遭到破坏。

通过对上述的端口配置选项进行检查，以及确保链路两端的所有参数都保持同步，那么就可以肯定所连接设备的连通性及流量吞吐都将是最优的。

## VLANs及中继排错

### Troubleshooting VLANs and Trunking

先前的小节中，我们谈到可用用于物理层故障排除的三个命令行命令的使用。本节将给出一些用于对VLAN内连通性故障进行鉴别及排错的常见方法 (the use of three CLI commands that can be used for troubleshooting Physical Layer issues. this section describes some common approaches to identifying and troubleshooting intra-VLAN connectivity issues)。VLAN内部连通性故障的一些相对来讲更为常见的原因，有下面这些。

- 双工不匹配，duplex mismatches
- 坏的网卡或网线，bad NIC or cable
- 堵塞，congestion
- 硬件故障，hardware issues

- 软件故障, software issues
- 资源过度预订, resource oversubscription, [Cisco MDS交换机端口组速率模式介绍](#)
- 配置问题, configuration issues

**双工不匹配**可导致甚低网络性能及连通性。尽管已有对自动协商的改进，同时采行自动协商被认为是有效做法，双工不匹配仍有可能发生。比如，在网卡设置为100/Full，而交换机端口是自动协商时，网卡将保持其100/Full设置，但交换机端口将被设置为100/Half。而与此相反，也会出现双工不匹配的问题。也就是网卡设置自动协商，交换机端口设置为100/Full。此时，网卡将自动协商为100/Half，而交换机端口保持其静态的100/Full配置，导致双工不匹配。

因此好的做法就是在那些可以手动设置的地方，对10/100以太网连接的速率和双工手动进行设置，以避免自动协商带来的双工不匹配问题。**双工不匹配可能不仅会对直接连接到交换机上的用户造成影响，还可能对有着不匹配双工设置的交换机间连接上通过的网络流量造成影响。**使用命令 `show interface` 就可以查看到端口的接口速率和双工设置。

**注意：**因为Catalyst交换机仅支持1Gbps链路全双工，所以对于GigabitEthernet连接，双工问题并不常见。

思科IOS软件中的多个计数器都可用来鉴别潜在**坏网卡或网线问题**。通过对不同的 `show` 命令中的一些计数器的检查，来识别网卡或网线问题。比如，假设交换机端口计数器显示带有无效CRC或FCS错误的帧数持续增长，就有极大可能是因为工作站或机器的坏网卡，以及坏的网线。

网络壅塞同样可能引起交换网络中的间隙性连通故障。VLAN超载的第一个表象就是某端口上的接收或发送缓冲过度预订(oversubscribed)。此外，端口上过多的帧丢弃也是网络壅塞的指标。而网络壅塞的常见原因，就是对主干连接的聚合带宽需求估计不足。那么，**壅塞问题就可以通过配置以太网信道或往现有以太网信道中加入更多的端口，得到解决。**同时网络壅塞又是连通性故障的常见原因，同时重要的是要知道**交换机本身可能经历壅塞问题，而交换机本身的壅塞问题有可能会对网络性能产生类似的影响**(a common cause of network congestion is due to underestimating aggregate bandwidth requirements for backbone connections. In such cases, congestion issues can be resolved by configuring EtherChannels or by adding additional ports to existing EtherChannels. While network congestion is a common cause of connectivity issues, it is also important to know that the switch itself can experience congestion issues, which can have a similar impact on network performance)。

**交换机内部壅塞**，有限的交换机带宽可能导致壅塞问题，由此造成的壅塞可能对网络性能造成极为严重的影响。在LAN交换中，带宽是指交换机内部交换线路 ([the switch fabric](#)) 的传输能力。因此，如果交换线路的传输能力是 5Gbps，而要尝试将 7Gbps 的流量通过交换机传输，结果就是数据包丢失及差强人意的网络性能了。在那些所有端口的聚合传输容量可能超出总的骨干容量的超出预订平台上，这是一个常见的问题 (this is a common issue in oversubscribed platforms, where the aggregate capacity of all ports can exceed the total backplane capacity)。

在交换LAN中，**硬件故障**也可能引起连通性问题。这类问题的实例包括交换机的坏端口或坏模块。尽管在可能的情况下可以通过查看诸如LEDs等物理指示器来对这类故障进行排错，某些时候这类问题是难于排错及诊断的。在怀疑存在潜在的硬件故障的多数情况下，都应需求技术支持中心 (Technical Assistance Centre, TAC) 的支持。

相比上面这些问题，软件缺陷 (software bugs) 就更难于分辨出来，因为软件缺陷引起难于对其进行排错的偏差 (deviation)。在怀疑有软件缺陷导致连通性问题时，应该就发现的问题，联系技术支持中心 (TAC)。此外，如在控制台或日志中有打印出错误消息，就也可以使用思科提供的一些在线工具，来采取一些替代方法 (implement a workaround) 或是得到某个已经解决该问题并得到验证的软件版本的建议。

与其它硬件设备相比，交换机有着受限的资源，比如物理内存。在这些资源过度使用时，就会导致严重的性能问题。而像是高CPU使用率这样的问题，就可能对交换机及网络性能造成极为严重的影响。

最后，如同其它技术一样，不正确的配置同样会直接或间接地造成连通性问题。比如，根桥放置粗劣就会导致慢速用户连通性。而将一台不当配置的交换机直接加入到生产网络，则会导致一些或全部用户的网络连接完全中断（the poor placement of the Root Bridge may result in slow connectivity for users. Directly integrating or adding an incorrectly configured switch into the production network could result in an outright outage for some or all users）。下面的小节对一些常见的VLAN相关故障、其可能的原因，以及为排除这些故障可采取的做法，进行了讲解。

## 动态VLAN通告排错

### Troubleshooting Dynamic VLAN Advertisements

思科Catalyst交换机使用VLAN中继协议（VLAN Trunk Protocol, VTP）来在交换域中传播VLAN信息（propagate VLAN information dynamically throughout the switched domain）。VTP是一个思科专有的二层报文发送协议，用于管理位处同一VTP域中的交换机上VLANs的添加、删除及重命名。

某台交换机在加入到VTP域时无法动态接收任何VLAN信息的原因有好几个。下面是一些常见的原因。

- 二层中继配置错误，Layer 2 trunking misconfiguration
- 不正确的VTP配置，incorrect VTP configuration
- 配置修订号，configuration revision number
- 物理层故障，Physical Layer issues
- 软件或硬件故障或缺陷，software or hardware issues or bugs
- 交换机性能问题，switch performance issues

为令到交换机采用VTP交换VLAN信息，交换机间必须建立中继链路。思科IOS交换机支持ISL和802.1Q两种中继机制。尽管一些交换机默认采用ISL这种思科专有中继机制，不过当前思科IOS Catalyst交换机默认都采用802.1Q了。在提供交换机间中继时，手动指定中继封装协议被认为是好的做法。这是通过在将链路配置为中继端口时，使用接口配置命令 `switchport trunk encapsulation [isl|dot1q]` 完成的。

可用于对中继连通性问题进行排错的命令有好几个。可使用 `show interfaces` 命令来检查基本的端口运行及管理性状态（you can use the `show interfaces` command to verify basic port operational and administrative status）。此外，可通过在 `show interfaces` 命令后追加 `trunk` 或 `errors` 关键字来进行额外排错和检查。而命令 `show interfaces [name] counters trunk` 则可用于查看中继端口上传输和接收到的帧数目。

该命令的输出还包括了封装错误数，而封装错误数可用于检查802.1Q和ISL，以及中继封装不匹配数目，如下面的输出所示。

```
Cat-3550-1#show interfaces FastEthernet0/12 counters trunk
Port      TrunkFramesTx    TrunkFramesRx    WrongEncap
Fa0/12        1696            32257             0
```

参考上面的输出，可以反复执行该命令，以确保Tx及Rx栏是持续增长的，并以此完成更多的排错。比如，假设该交换机没有发出任何帧，则该接口就可能并未配置为中继接口，或者其是宕掉的或关闭的（or it might be down or disabled）。而如果Rx栏没有增长，则可能是远端交换机未有正确配置。

用于对可能的二层错误配置进行排错的另一个命令，就是 `show interfaces [name] trunk`。该命令的输出包含了中继封装协议及模式、802.1Q的原生VLAN、允许通过中继链路VLANs、VTP域中活动的VLANs，以及被修剪掉的VLANs（the output of `show interfaces [name] trunk` includes the trunking encapsulation protocol and mode, the native VLAN for 802.1Q, the VLANs that are allowed to traverse the trunk, the VLANs that are active in the VTP domain, and the VLANs that are pruned）。一个VLAN传播的常见问题，就是上游交换机已通过使用接口配置命令 `switchport trunk allowed vlan`，被配置为对某些VLANs进行过滤。命令 `show interfaces [name] trunk` 的输出如下所示。

```
Cat-3550-1#show interfaces trunk
Port      Mode       Encapsulation  Status      Native vlan
Fa0/12   desirable    n-802.1q     trunking    1
Fa0/13   desirable    n-802.1q     trunking    1
Fa0/14   desirable    n-isl       trunking    1
Fa0/15   desirable    n-isl       trunking    1
Port      Vlans allowed on trunk
Fa0/12   1-4094
Fa0/13   1-4094
Fa0/14   1-4094
Fa0/15   1-4094
Port      Vlans allowed and active in management domain
Fa0/12   1-4
Fa0/13   1-4
Fa0/14   1-4
Fa0/15   1-4
Port      Vlans in spanning tree forwarding state and not pruned
Fa0/12   1-4
Fa0/13   none
Fa0/14   none
Fa0/15   none
```

另一个常见中继错误配置故障就是原生VLAN不匹配。在配置802.1Q中继链路时，中继链路两端的原生VLAN必须匹配；否则该链路便不会工作。如存在原生VLAN不匹配，STP就会将该端口置为端口VLAN ID不一致状态（a port VLAN ID(PVID) inconsistent state），且不会在该链路上进行转发。在此情况下，将有类似于下面的消息在控制台或日志中打印出来。

```
*Mar 1 03:16:43.935: %SPANTREE-2-RECV_PVID_ERR: Received BPDU with inconsistent peer vlan id 1 on FastEthernet0/11
*Mar 1 03:16:43.935: %SPANTREE-2-BLOCK_PVID_PEER: Blocking FastEthernet0/11 on VLAN0001. Inconsistent
*Mar 1 03:16:43.935: %SPANTREE-2-BLOCK_PVID_LOCAL: Blocking FastEthernet0/11 on VLAN0002. Inconsistent
*Mar 1 03:16:43.935: %SPANTREE-2-RECV_PVID_ERR: Received BPDU with inconsistent peer vlan id 1 on FastEthernet0/12
*Mar 1 03:16:43.935: %SPANTREE-2-BLOCK_PVID_PEER: Blocking FastEthernet0/12 on VLAN0001. Inconsistent
*Mar 1 03:16:43.939: %SPANTREE-2-BLOCK_PVID_LOCAL: Blocking FastEthernet0/12 on VLAN0002. Inconsistent
```

尽管STP排错将在本书后面进行讲解，该不一致状态仍可通过使用 `show spanning-tree` 命令进行查证，如下所示。

```
Cat-3550-1#show spanning-tree interface FastEthernet0/11
Vlan          Role     Sts      Cost      Prio.Nbr      Type
-----  -----
VLAN0001      Desg    BKN*    19        128.11      P2p *PVID_Inc
VLAN0002      Desg    BKN*    19        128.11      P2p *PVID_Inc
```

如已经查明该中继链路确实是正确配置，及两台交换机间是可运作的，接下来就应对VTP配置参数进行检查了。这些参数包括VTP域名、正确的VTP模式及VTP口令，如对该VTP域配置了某个参数，就要使用相应的 `show vtp status` 及 `show vtp password` 命令。`show vtp status` 命令的输出如下所示。

```
Cat-3550-1#show vtp status
VTP Version : running VTP2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 8
VTP Operating Mode : Server
VTP Domain Name : TSHOOT
VTP Pruning Mode : Enabled
VTP V2 Mode : Enabled
VTP Traps Generation : Disabled
MD5 digest : 0x26 0x99 0xB7 0x93 0xBE 0xDA 0x76 0x9C
...
[Truncated Output]
```

在应用 `show vtp status` 命令时，要确保交换机使用同一版本的VTP。默认情况下，Catalyst交换机允许VTP版本 1。而运行VTP版本 1 的交换机是不能加入到VTP版本 2 的域中的。而如某交换机不兼容VTP版本 2，那么就要使用全局配置命令 `vtp version`，将所有VTP版本 2 的交换机配置为运行版本 1。

**注意：**如在服务器上修改了VTP版本，那么此改变将自动传播到VTP域中的客户端交换机。

VTP客户端/服务器 (client/server) 或服务器/服务器(server/server)设备上的VTP传播是开启的。而如果在某台交换机上VTP是关闭的（也就是透明模式），那么该交换机将不会经由VTP动态地接收VLAN信息。不过，要留意**VTP版本2的透明模式交换机，将在其中继端口转发出接收到的VTP通告，而充当VTP中继**。就算VTP版本不一样，该过程也会照常进行。域中交换机上的VTP域名称也应保持一致。

最后，`show vtp status` 命令的输出也包含了用于认证目的的MD5散列值。该散列值是从VTP域名称和密码生成的，域中所有交换机上的该散列值应是一致的。而如在这些交换机上的域名称和密码不同，则计算出的MD5也会不同。而如域名称或密码不同，那么 `show vtp status` 命令就会示出一条MD5摘要校验和不匹配 (an MD5 digest checksum mismatch) 消息，如下面的输出所示。

```
Cat-3550-1#show vtp status
VTP Version : running VTP2
Configuration Revision : 0
Maximum VLANs supported locally : 1005
Number of existing VLANs : 8
VTP Operating Mode : Server
VTP Domain Name : TSHOOT
VTP Pruning Mode : Enabled
VTP V2 Mode : Enabled
VTP Traps Generation : Disabled
MD5 Digest : 0x26 0x99 0xB7 0x93 0xBE 0xDA 0x76 0x9C
*** MD5 digest checksum mismatch on trunk: Fa0/11 ***
*** MD5 digest checksum mismatch on trunk: Fa0/12 ***
...
[Truncated Output]
```

最后，在应用VTP时，**配置修订号可能会造成严重破坏**。VTP域中的交换机使用配置修订号来保持对域中最新信息的跟踪 (the configuration revision number can wreak havoc when using VTP. Switches use the configuration revision number to keep track of the most recent information in the VTP domain)。域中所有交换机都将其前一次从一条VTP通告中收听到的配置修订号存储起来，同时在每次接收到新信息时该号码都被增加。而在任何交换机接收到带有高于其自身配置修订号的通告报文时，都将覆写任何存储的VLAN信息，并将其自身存储的VLAN信息与所接收到的通告报文中的信息进行同步。

因此，如想知道为何加入到VTP域中的交换机没有接收任何的VLAN信息，那么可能是该交换机已有一个较高的配置修订号，而导致所有其它交换机覆写了它们的本地VLAN信息，并将本地VLAN信息替换了接收自新交换机的通告报文中的VLAN信息。为避免此种情形，在**将新的交换机加入到某VTP域之前，总是要确**

**保其配置修订号被设置为0。**这可以通过在该交换机上修改VTP模式或修改VTP域名称完成。配置修订号在命令 `show vtp status` 的输出中有包含。

## VLAN内部端到端连接丢失

### Troubleshooting Loss of End-to-End Intra-VLAN Connectivity

某个VLAN中端到端连通性丢失有好几个原因。而最常见的原因包括下面这些。

- 物理层故障, Physical Layer issues
- VTP修剪, VTP pruning
- VLAN中继链路过滤, VLAN trunk filtering
- 新的交换机, new switches
- 交换机性能问题, switch performance issues
- 网络壅塞, network congestion
- 软件或硬件问题或缺陷, software or hardware issues or bugs

**注意:** 为简明扼要地讲解, 这里只会对中继、VTP修剪、以及往域内新加入交换机三个方面进行说明。软件或硬件问题或缺陷及交换机性能问题在本书中已有说明。而物理层排错在本模块早前已经进行了讲解。

**VTP修剪在没有本地端口属于某些VLANs时, 将那些VLANs从本地交换机的VLAN数据库中移除。**VTP修剪通过消除不必要的广播、多播及通过网络泛洪的那些未知流量, 而提升中继链路效率 (VTP pruning increases the efficiency of trunks by eliminating unnecessary Broadcast, Multicast, and unknown traffic from being flooded across the network)。

尽管VTP修剪是一项值得部署的特性, 但不正确的配置或是部署可能导致端到端连通性的丢失。应仅在客户端/服务器环境中开启 (in client/server environments)。在包含透明模式交换机的网络中应用修剪, 就可能造成连通性丢失。如网络中有一或多台的交换机处于VTP透明模式, 就应该要么对整个VTP域全局关闭修剪, 否则就要通过在适当的接口下, 使用接口配置命令 `switchport trunk pruning vlan`, 以确保到上游的透明模式交换机中继链路上的VLANs都无资格修剪 (也就是它们在这些链路上不被修剪, ensure that all VLANs on the trunk link(s) to the upstream transparent mode switch(es) are pruning ineligible, i.e., they are not pruned, using the `switchport trunk pruning vlan` interface configuration command under the applicable interfaces)。

### 对放行的VLANs及中继状态进行检查

#### Verify Allowed VLANs and Trunk Status

除了VTP修剪外, 交换机中继链路上对VLANs的不正确过滤, 也可能导致端到端VLAN连通性的丢失。**默认允许所有VLANs通过所有中继链路;**但是思科IOS软件允许管理员通过使用接口配置命令 `switchport trunk allowed vlan`, 在指定中继链路上选择性地移除 (或加入) VLANs。可以使用命令 `show interfaces [name] trunk` 及 `show interfaces [name] switchport`, 来查看中继链路上被修剪和限制的VLANs。作为检查某个中继端口上放行VLANs最容易的方式, 命令 `show interfaces [name] trunk` 的输出如下所示。

```
Cat-3550-1#show interfaces trunk
Port      Mode      Encapsulation      Status      Native vlan
Fa0/1    on        802.1q            trunking    1
Fa0/2    on        802.1q            trunking    1
Port      Vlans allowed on trunk
Fa0/1    1,10,20,30,40,50
Fa0/2    1-99,201-4094
Port      Vlans allowed and active in management domain
Fa0/1    1,10,20,30,40,50
Fa0/2    1,10,20,30,40,50,60,70,80,90,254
Port      Vlans in spanning tree forwarding state and not pruned
Fa0/1    1,10,20,30,40,50
Fa0/2    1,40,50,60,70,80,90,254
```

同样要检查中继链路上通告的正确VLANs。在中继链路上放行的不适当VLANs可能引起功能缺失或安全问题。也想要确保中继链路两端有着同样的放行VLANs (Improper VLANs allowed on the link can lead to a lack of functionality or security issues. Also, you want to make sure that the same VLANs are allowed on both ends of a trunk)。

**注意：**在将另外的需要在某条中继链路上放行的VLAN(s)加入进去时，应非常谨慎地不要忘记关键字 add。比如，在已经配置了 switchport trunk allowed vlan 10, 20，而打算同样放行VLAN 30时，就需要输入命令 switchport trunk allowed vlan add 30。而如只是简单地配置 switchport trunk allowed vlan 30，那么先前所允许的VLANs 10和20就会从中继链路上移除，这将导致VLANs 10和20的通信中断。

由命令 show interfaces trunk 命令所提供的另一重要信息，就是中继端口状态。中继端口状态信息确认该中继链路是否形成，同时必须要在链路两端对此进行检查。如该端口未处于“中继”模式，此时最重要的就是必须对端口中继运作模式 (the mode of operation, on/auto等) 进行检查，看看在该模式下是否能与链路另一端形成中继状态。

### 检查封装类型

#### Verify Encapsulation Type

解决中继故障的另一重要步骤，就是查明中继链路两端所配置的正确封装类型。大多数思科交换机都允许 ISL 及 802.1Q 封装类型。而尽管大多数现代网络都是设计使用 dot1Q，仍然可能存在一些网络优先使用 ISL 的情形。封装类型是通过使用接口配置命令 switchport trunk encapsulation <type> 配置的。而可用于查看封装类型的命令如下。

- show interfaces trunk
- show interfaces <number> switchport

在某个已被静态配置为 802.1Q 中继链路端口上的 show interfaces [name] switchport 命令的输出如下所示。

```

Cat-3550-2#show interfaces FastEthernet0/7 switchport
Name: Fa0/7
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Administrative Native VLAN tagging: enabled
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk native VLAN tagging: enabled
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk associations: none
Administrative private-vlan trunk mappings: none
Operational private-vlan: none
Trunking VLANs Enabled: 3,5,7
Pruning VLANs Enabled: 2-8
Capture Mode Disabled
Capture VLANs Allowed: ALL
Protected: false
Unknown unicast blocked: disabled
Unknown multicast blocked: disabled
Appliance trust: none

```

如之前小节所说到的，网络中一台新交换机的加入，可能导致管理域中VLAN信息的丢失（the integration of a new switch into the network can result in a loss of VLAN information in the management domain）。而此VLAN信息丢失又可能导致同一VLAN中设备间连通性的丢失。所以在将新交换机加入到LAN之前，一定要确保其配置修订号被重置到0。

## 使用“show vlan”命令

### Using the "show vlan" Command

除了在前面几个小节中介绍的那些命令外，还有一些用于对VLAN配置进行检查和排错的有用思科IOS软件的命令。最常用到的VLAN检查和排错命令之一就是 `show vlan` 命令了。该命令显示管理域内所有VLANs的参数，如下面的输出所示。

```

Cat-3550-1#show vlan
VLAN Name          Status    Ports
----- -----
1    default        active    Fa0/11, Fa0/12,
                           Fa0/13, Fa0/14,
                           Fa0/20, Fa0/21,
                           Fa0/22, Fa0/23,
                           Fa0/24
150   VLAN_150      active    Fa0/2, Fa0/3, Fa0/4,
                           Fa0/5, Fa0/6, Fa0/7,
                           Fa0/8, Fa0/9, Fa0/10
160   VLAN_160      active    Fa0/15, Fa0/16,
                           Fa0/17, Fa0/18,
                           Fa0/19
170   VLAN_170      active    Gi0/1, Gi0/2
1002  fddi-default  active
1003  token-ring-default  active
1004  fdnet-default   active
1005  trnet-default   active
VLAN Type SAID     MTU      Parent RingNo BridgeNo Stp  BrdgMode
----- -----
1    enet  100001   1500     -        -      -      -      -
150   enet  100150   1500     -        -      -      -      -
160   enet  100160   1500     -        -      -      -      -
170   enet  100170   1500     -        -      -      -      -
1002  fddi  101002   1500     -        -      -      -      -
1003  tr    101003   1500     -        -      -      -      -
1004  fdnet 101004   1500     -        -      ieee  -
1005  trnet 101005   1500     -        -      ibm   -
Trans1 Trans2
-----
0    0
0    0
0    0
0    0
0    0
0    0
0    0
0    0
Remote SPAN VLANS
-----
Primary Secondary Type    Ports
----- -----

```

该命令打印出所有可用的VLANs，以及所分配到每个单独VLANs的那些端口。该命令的输出所包含的端口仅是接入端口，且不管这些端口是否开启或宕掉，都会显示出来。该命令输出不包括中继链路，因为这些输出属于所有所有VLANs。 show vlan 命令还提供了RSPAN(Remote Switch Port Analyser, 远程交换机端口分析器) VLANs，以及交换机上私有VLAN (Private VLAN, P VLAN, 这是一个CCNP考点) 的信息。 show vlan 命令还可以带上一些额外关键字来使用，以提供更具体的信息。下面的输出显示了可与该命令一起使用的所支持的附加关键字。

```

Cat-3550-1#show vlan ?
brief      VTP all VLAN status in brief
id         VTP VLAN status by VLAN id
ifindex    SNMP ifIndex
name       VTP VLAN status by VLAN name
private-vlan Private VLAN information
remote-span Remote SPAN VLANs
summary    VLAN summary information
|          Output modifiers<cr>

```

`brief` 字段打印所有活动VLANs的简要信息。此命令的输出与上面的相同，唯一的区别就是省掉了后两个部分。`id` 字段提供了和 `show vlan` 一样的信息，但如下面的输出所示，只包含特定VLAN的信息。

```
Switch-1#show vlan id 150
VLAN Name                               Status    Ports
----- -----
150  VLAN_150                           active   Fa0/1, Fa0/2, Fa0/3,
                                            Fa0/4, Fa0/5, Fa0/6,
                                            Fa0/7, Fa0/8, Fa0/9,
                                            Fa0/10
VLAN Type     SAID      MTU      Parent RingNo BridgeNo Stp  BrdgMode
----- -----
150  enet    100150    1500     -       -       -       -
Trans1 Trans2
----- -----
0      0
0      0
Remote SPAN VLAN
----- -----
Disabled
Primary Secondary Type          Ports
----- -----
```

VLAN与属于该VLAN的接入端口一样，再度包含在了输出中。中继端口因为是属于所有VLANs，而不包含在输出中。额外信息包括了VLAN MTU、RSPAN配置（如适用），以及PVLAN配置参数（如适用）。

`name` 字段允许指定VLAN名称而不是其ID。该命令打印与 `show vlan id <number>` 命令同样的信息。`ifindex` 字段现实VLAN的SNMP IfIndex(如适用)，而 `private-vlan` 及 `remote-span` 字段打印相应的PVLAN及RSPAN配置信息。最后，`summary` 字段打印管理域（the management domain）中活动的VLANs数目一个汇总信息。活动VLANs包括了标准和扩展VLANs。

带上或不带参数的 `show vlan` 命令，在排错过程的以下方面，都是最为有用的命令。

- 确认设备上所配置的VLANs
- 判定端口成员关系

另一个有用的VLAN排错命令，就是 `show vtp counters`。该命令打印有关VTP数据包统计的信息。以下是在某台配置为VTP服务器的交换机上，`show vtp counters` 的输出。

```
Cat-3550-1#show vtp counters
VTP statistics:
Summary advertisements received      : 15
Subset advertisements received       : 10
Request advertisements received     : 2
Summary advertisements transmitted  : 19
Subset advertisements transmitted   : 12
Request advertisements transmitted  : 0
Number of config revision errors   : 0
Number of config digest errors     : 0
Number of V1 summary errors        : 0
VTP pruning statistics:
Trunk   Join Transmitted   Join Received   Summary advts received
                                                from non-pruning-
                                                capable device
----- -----
Fa0/11      0            1                  0
Fa0/12      0            1                  0
```

`show vtp counters` 命令打印输出的前六行，提供了三种类型VTP数据包的统计信息：通告请求（advertisement requests）、汇总通告（summary advertisements）以及子集通告（subset advertisements）。随后小节将对这些不同报文进行讲解。

**VTP通告请求**是对配置信息的请求。这些报文是由VTP客户端发出给VTP服务器，用以请求其没有的VLAN及VTP信息。在交换机重置、VTP域名称改变，或交换机接收到一条带有比其自身更高的配置修订号的VTP汇总通告帧时，客户端交换机便发出一条VTP通告请求报文。VTP服务器应仅显示接收计数器增长，而所有VTP客户端都应只显示发送计数器增长。

**VTP汇总通告**是由服务器默认每隔5分钟发出的。这些报文类型用于告知邻接交换机当前VTP域名称、配置修订号及VLAN配置状态，及包括时间戳、MD5散列值及子网数目通告等其它VTP信息（VTP summary advertisements are used to tell an adjacent switch of the current VTP domain name, the configuration revision number and the status of the VLAN configuration, as well as other VTP information, which includes the time stamp, the MD5 hash, and the number of subnet advertisements to follow）。而如果服务器上的这些计数器在增长，那么在VTP域中就有不知一台交换机充当或配置为VTP服务器。

**VTP子集通告**是由VTP服务器在某个VLAN配置改变时，比如有VLAN被加入、中止、改变、删除或其它VLAN指定参数（比如VLAN的MTU等）发生变化时所发出的。在VTP汇总通告之后，会有一或更多的子集通告发出。而一条子集通告包含了一个VLAN信息清单。而如果涉及多个VLANs，就需要多于一条的子集通告，以实现对所有VLANs的通告。

字段 `Number of config revision errors` 显示了交换机因其接收到带有相同配置修订号，却有着不同MD5散列值的数据包，而无法接受的通告数目。在同一VTP域中有两台以上的服务器交换机上的VTP信息同时发生变动，且中间交换机(an intermediate switch)于同一时间接收到来自这些服务器的通告时，这是常见会发生的。此概念在下图15.3中进行了演示，该图演示了一个基本的交换网络。

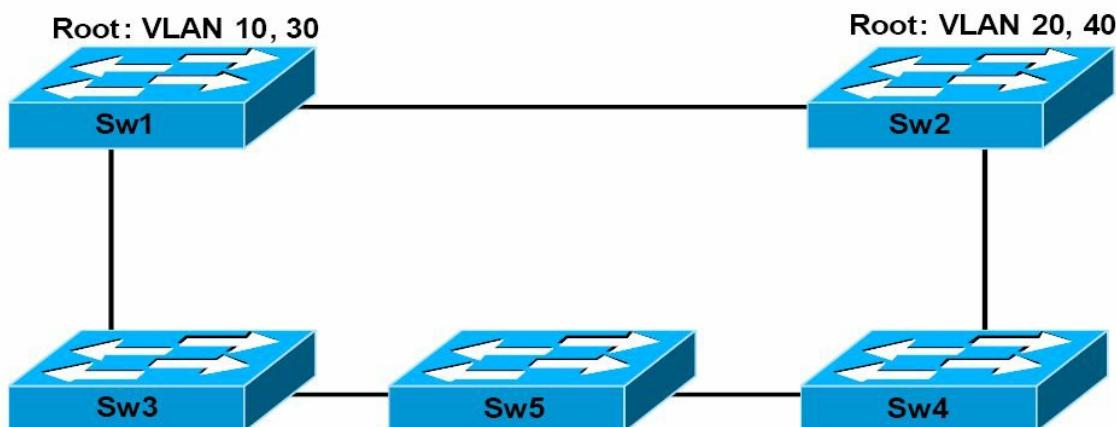


图15.3 -- 配置修订号错误的排错

图15.3演示了一个结合了冗余和负载均衡（incorporates redundancy and load sharing）的基本网络。应假设Sw1和Sw2都是配置成的VTP服务器，而Sw3是配置成的VTP客户端。Sw1是VLANs 10及30的根交换机，同时Sw2是VLANs 20及40的根交换机。假设在Sw1和Sw2上同时应用了改变，将VLAN 50加入到了Sw1，VLAN 60加入到了Sw2。在数据库的改变之后两台交换机都发出一条通告。

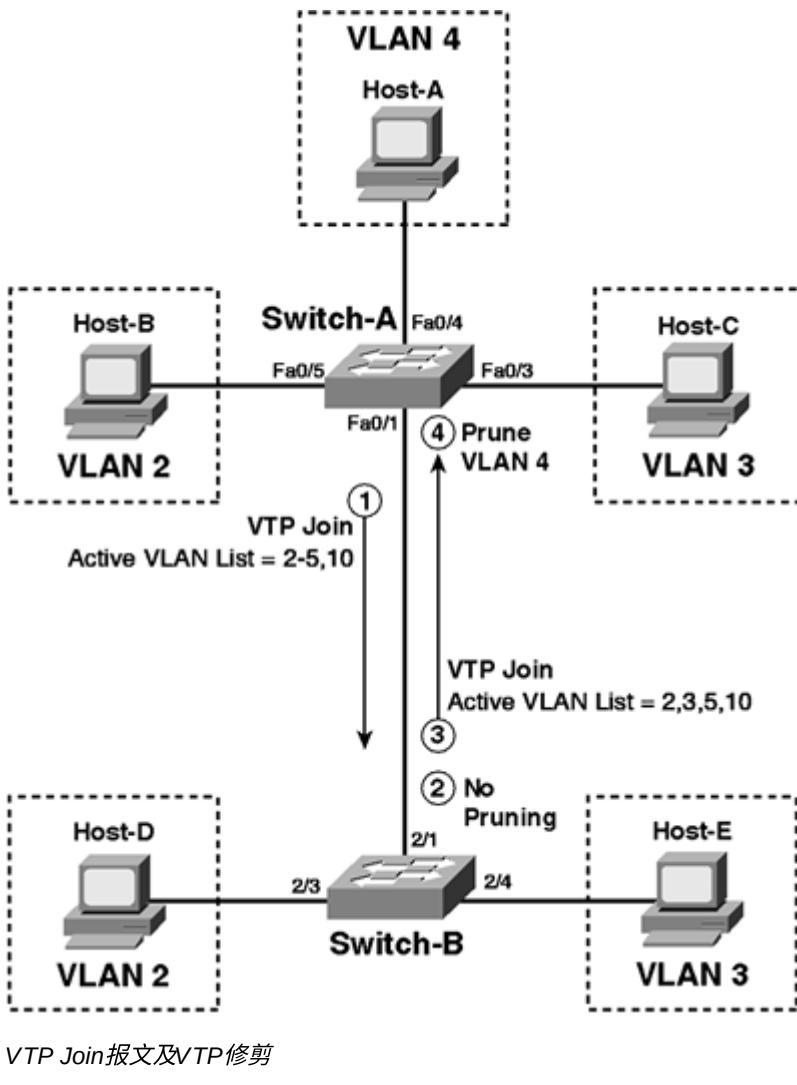
变动在域中传播，覆写其它接收到该信息交换机先前的数据库。假设Sw5同时接收到来自邻居的同样信息，同时接收到的两条通告都包含了同样配置修订号。那么在此情况下，该交换机就无法接受这两条通告，因为它们有着同样配置修订号，但却有着不同的MD5散列值。

当这种情况发生时，交换机就将 `Number of config revision errors counter` 字段加一，同时不更新其VLANs数据库。而这种情况可能导致一个或多个VLANs中连通性的丢失，因为在该交换机上的VLAN信息没有得到更新。为解决此问题并确保该交换机上的本地数据库保持更新，就要在其中一台服务器交换机上配

置一个虚构的VLAN (a dummy vlan) , 这样就导致对所有交换机本地数据库的覆写, 从而允许Sw5也更新其数据库。切记这并不是一种常见现象 (this is not a common occurrence) ; 但还是可能发生, 因此, 这里将这么多也是有必要的。

在交换机接收到一条带有与其计算出的MD5散列值不一致的MD5散列值的通告时, `Number of config digest errors counter` 字段就会增长。这是在交换机上配置了不同VTP密码的结果。可使用 `show vtp password` 命令检查所配置的VTP密码是正确的。同样重要的是记住在密码一致时, 硬件或软件的问题或缺陷也会造成VTP数据包的数据错误, 从而也会导致这样的错误出现。

最后, 字段 `VTP pruning statistics` 将只在VTP域的VLAN修剪开启时, 才会包含非零值。**修剪是在服务器上开启的, 同时该配置在该VTP域中得以传播**。在某VTP域的修剪开启时, 服务器将接收来自客户端的Join报文 (the VTP Join messages) (pruning is enabled on servers and this configuration is propagated throughout the VTP domain. Servers will receive joins from clients when pruning has been enabled for the VTP domain, [VTP pruning, InformIT](#)) 。



## 第15天问题

1. What is the colour of the system LED under normal system operations?
2. What is the colour of the RPS LED during a fault condition?

3. You can cycle through modes by pressing the Mode button until you reach the mode setting you require. This changes the status of the port LED colours. True or false?
4. What port speed is represented by a blinking green LED?
5. If you want to be sure that you are not dealing with a cabling issue, one of the simplest things to do is to \_\_\_\_\_ the cable and run the same tests again.
6. Which command is generally used to troubleshoot Layer 1 issues (besides show interfaces)?
7. The \_\_\_\_\_ status is reflected when the connected cable is faulty or when the other end of the cable is not connected to an active port or device (e.g., if a workstation connected to the switch port is powered off).
8. What are runts?
9. The \_\_\_\_\_ command can also be used to view interface errors and facilitate Layer 1 troubleshooting.
10. Which command prints a brief status of all active VLANs?

## 第15天答案

1. Green.
2. Amber.
3. True.
4. 1000Mbps.
5. Replace.
6. The `show controllers` command.
7. `notconnect`.
8. Packets that are smaller than the minimum packet size (less than 64 bytes on Ethernet).
9. `show interfaces [name] counters errors`.
10. The `show vlan brief` command.

## 第15天实验

### 一层排错实验

在真实设备上对本模块中提到的一层排错相关命令进行测试。

- 如同模块中所讲解的那样，检查不同场景下交换机系统及端口LED状态
- 执行一下`show interface`命令，并对本模块中所说明的有关信息进行查证
- 对`show controllers`及`show interface counters errors`进行同样的执行

### 二层排错使用

在真实设备上对本模块中提到的二层排错相关命令进行测试。

- 在交换机之间配置VTP，并将一些VLANs从VTP服务器通告到VTP客户端（查看第三天的VTP实验）
- 在两台交换机之间配置一条中继链路，并生成一些流量（ping操作）
- 测试`show vlan`命令
- 测试`show interface counters trunk`命令
- 测试`show interface switchport`命令
- 测试`show interface trunk`命令
- 测试`show VTP status`命令
- 测试`show VTP counter`命令



# 第31天 生成树协议

## Spanning Tree Protocol

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

# 第31天任务

- 阅读今天的课文
- 完成今天的实验
- 阅读ICND2记诵指南
- 在[subneting.org](https://subneting.org)上花15分钟

生成树协议 (Spanning Tree Protocol, STP) 的作用，是在具备冗余的交换网络拓扑中，允许存在多条物理链路的同时，通过建立一个无循环逻辑拓扑，阻止网络上循环的发生 (the role of Spanning Tree Protocol(STP) is to prevent loops from occurring on your network by creating a loop-free logical topology, while allowing physical links in redundant switched network topologies)。随着网络中所用到交换机数量的急剧增加，以及传播VLAN信息的主要目的下，围绕网络数据帧无尽循环问题开始出现。

先前CCNA考试仅要求对STP有基本理解。但当前版本则希望对此方面有极好的掌握。

今天将学习以下内容。

- STP的需求，the need of STP
- STP桥ID，STP Bridge ID
- STP根桥选举，STP Root Bridge election
- STP开销及优先级，STP cost and priority
- STP根及指定端口，STP Root and Designated Ports
- STP增强，STP enhancements
- STP排错，Troubleshooting STP

本课对应了以下CCNA大纲要求。

- PVSTP运作的配置和验证，configure and verify PVSTP operation
  - 对根桥选举进行描述，describe root bridge election
  - 生成树的模式，spanning tree mode

## STP的需求

### The Need for STP

STP是在IEEE 802.1D标准中定义的。为维护起一个无循环逻辑拓扑，交换机每两秒传递桥协议数据单元（Bridge Protocol Data Units, BPDUs）。BPDUs是一些在生成树拓扑中用到、用于传递有关端口、地址、优先级及开销等信息的数据报文。**BPDUs被打上VLAN ID标签。**

下图31.1显示了网络中循环是如何能创建出来的。因为各台交换机都学到VLAN 20，同时这些交换机也将其能达到VLAN 20的情况，通告给其它交换机。很快，所有交换机都认为其是VLAN 20流量的源，且造成了一个循环，因此所有以VLAN 20为目的地的帧将自一台交换机往另一台不停传递。

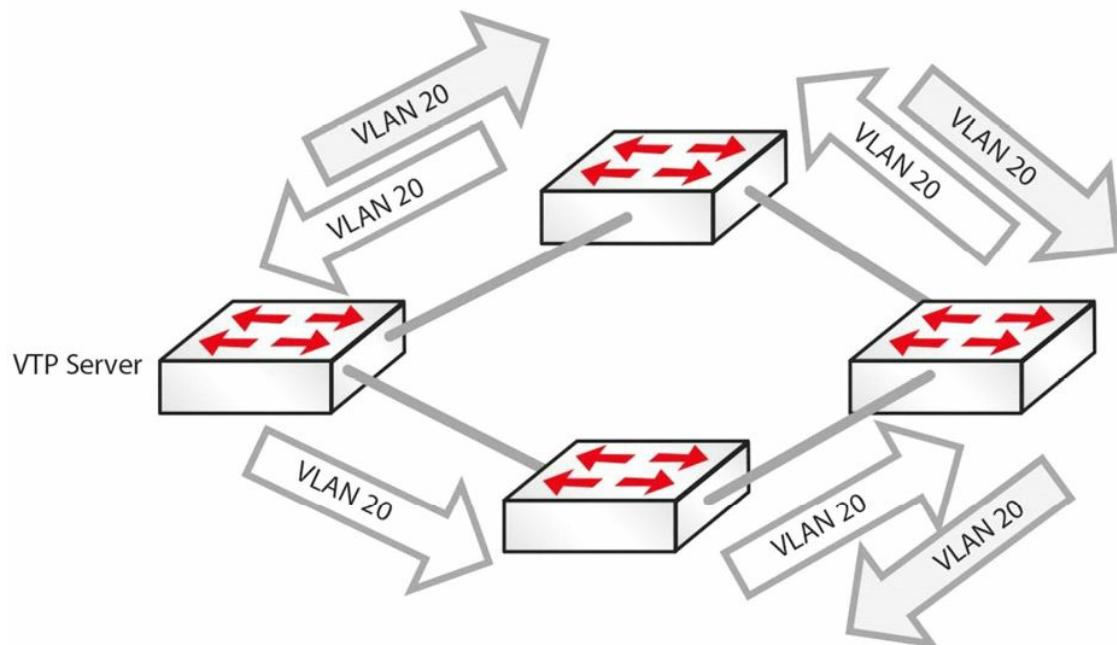


图31.1 -- 循环是如何建立的

STP运行着一种算法，用于根据所考虑的特定VLAN，决定出哪些端口保持开放或活动，以及哪些端口需要对特定VLAN关闭。

**位处生成树域中的所有交换机，都使用BPDUs来沟通和交换报文。** STP利用BPDUs的交换，来确定网络拓扑，而网络拓扑则是由以下三个变量决定的。

- 与各台交换机相关联的唯一MAC地址（交换机识别符），the unique MAC address(switch identifier) that is associated with each switch
- 各个交换机端口到根桥的路径开销，the path cost to the Root Bridge associated with each switch port
- 各个交换机端口的端口识别符（该端口的MAC地址），the port identifier(MAC address of the port) associated with each switch port

BPDUs都是每两秒发出的，此特性允许实现快速的网络循环探测及拓扑信息交换。BPDUs的两个类型分别是**配置BPDUs及拓扑变化通知BPDUs**（Configuration BPDUs and Topology Change Notification BPDUs）；这里只会对配置BPDUs进行说明。

## IEEE 802.1D的配置BPDUs

### IEEE 802.1D Configuration BPDUs

配置BPDUs是由LAN交换机发出，用于生成树拓扑进行通信和计算。在交换机端口初始化后，该端口被置为阻塞状态，同时一个BPDU被发送给交换机中的所有端口。**默认情况下，直到其与其它交换机进行配置BPDUs的交换为止，所有交换机最初都假定其为生成树的根。**在某端口仍将其自身配置BPDUs视为最具吸引力 (the most attractive) 时，其就会持续发送配置BPDUs。这些交换机基于以下4个因素 (以列出顺序)，确定出最佳配置BPDU (the best Configuration BPDU)。

1. 有着最低的根桥ID的, lowest Root Bridge ID
2. 有着到根桥最低根路径开销的, lowest Root path cost to Root Bridge
3. 有着最低发送者桥ID的, lowest sender Bridge ID
4. 有着最低发送者端口ID的, lowest sender Port ID

配置BPDU交换的完成，导致以下动作。

- 选举出整个生成树域的根桥, a Root Switch is elected for the entire Spanning Tree domain
- 选举出生成树域中所有非根交换机上的根端口, a Root Port is elected on every Non-Root Switch in the Spanning Tree domain
- 选举出所有LAN网段中的指定交换机, a Designated Switch is elected for every LAN segment
- 选举出所有网段的指定交换机的指定端口(根交换机上的所有活动端口也都是指定端口), a Designated Port is elected on the Designated Switch for every segment(all active ports on the Root Switch are also designated)
- 通过阻塞冗余路径，网络中的循环得以消除，loops in the network are eliminated by blocking redundant paths

**注意：**随着逐步深入本模块内容，这些特性将会一一介绍。

一旦所有交换机端口都处于转发或阻塞状态，生成树网络 (the Spanning Tree network) 就完成了收敛，此时配置BPDUs就由根桥以默认每两秒的间隔发出。这就是**配置BPDUs的发端**。配置BPDUs通过根桥上的指定端口，转发到下游邻居交换机 (this is referred to as the origination of Configuration BPDUs. The Configuration BPDUs are forwarded to downstream neighboring switches via the Designated Port on the Root Bridge)。

当非根桥 (a Non-Root Bridge) 在其提供了到根桥最优路径的根端口上，接收到一个配置BPDU时，就会通过其指定端口，发送出一个该BPDU的更新版本。这就是**BPDU的传播** (when a Non-Root Bridge receives a Configuration BPDU on its Root Port, which is the port that provides the best path to the Root Bridge, it sends an updated version of the BPDU via its Designated Port(s). This is referred to as the propagation of BPDUs)。

**指定端口**则是**指定交换机**上，在转发来自那个LAN网段的数据包到根桥时，有着最低路径开销的端口 (**the Designated Port** is a port on **the Designated Switch** that has the lowest cost when forwarding packets from that LAN segment to the Root Bridge)。

一旦生成树网络得以收敛，便总是会有自根桥传输给STP域内其它交换机的一个配置BPDU在传送。而要记住在生成树网络完成收敛后的配置BPDUs数据流的最简单方法，就是记住以下4条规则。

1. 配置BPDUs是从根桥发出且通过指定端口发送的, a Configuration BPDU originates on the Root Bridge and is sent via the Designated Port
2. 配置BPDUs是由非根桥的根端口上接收的, a Configuration BPDU is received by a Non-Root Bridge on a Root Port

3. 配置BPDU是由非根桥的指定端口上传送的，a Configuration BPDU is transmitted by a Non-Root Bridge on a Designated Port
4. 在所有单个LAN区段上，都只有一个指定端口（在某台指定交换机上），there is only one Designated Port (on a Designated Switch) on any single LAN segment

下图31.2演示了该STP域中的配置BPDU数据流，对上面列出的4条简单规则进行了说明。

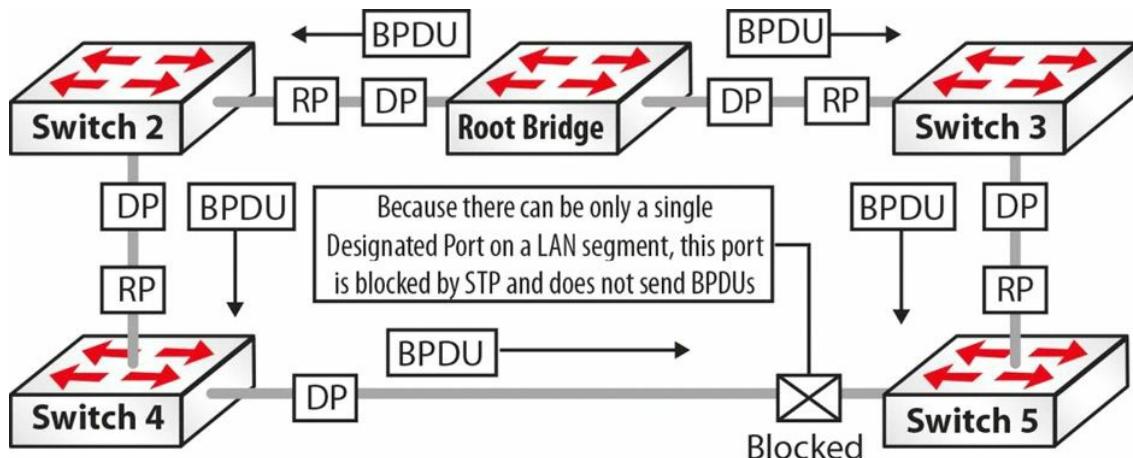


图31.2 -- STP 域中的配置BPDU数据流

1. 参考图31.2, 该配置BPDU源自根桥，且是通过根桥上的指定端口发送出来，前往那些非根桥交换机，也就是Switch 2和Switch 3。
2. 非根桥Switch 2和Switch 3在其有着到根桥最优路径的根端口上，接收到配置BPDU。
3. Switch 2和Switch 3对接收到的配置BPDU进行修改（更新），让后在其指定端口上转发出去。Switch 2就是该LAN网段上其自身及Switch 4的指定交换机，Switch 3是该LAN网段上其自身及Switch 5的指定交换机。而存在于指定交换机上的指定端口，则是在转发来自该LAN区段数据包到根交换机时，有着最低路径开销的端口。
4. 在Switch 4和Switch 5之间的LAN网段上，Switch 4被选举为指定交换机，同时指定端口也处于其上。因为在一个网段上只能有一台指定交换机，所以Switch 4和Switch 5之间网段上，Switch 5的端口，就被阻塞掉了。该端口将不会转发任何BPDUs。

## 生成树端口的各种状态

### Spanning Tree Port States

生成树算法 (Spanning Tree Algorithm, STA) 定义了STP控制下端口在进入到活动的转发状态之前，需要经历的几种状态。802.1D标准中端口状态有下面这些。

- 阻塞中 -- 仅接收BPDUs (为期20s)， blocking -- BPDUs received only (20 seconds)
- 倾听中 -- 有BPDUs发出和接收 (为期15s)， listening -- BPDUs sent and received (15 seconds)
- 学习中 -- 桥接表被建立起来 (为期15s)， learning -- bridging table is built (15 seconds)
- 转发中 -- 发送/接收数据，forwarding -- sending/receiving data
- 关闭 -- 管理性关闭，disabled -- administratively down

端口按以下方式在这些状态间依序移动。

1. 从初始化状态到阻塞中状态
2. 从阻塞中状态到侦听中状态或关闭状态
3. 从侦听状态到学习状态或关闭状态
4. 从学习状态到转发或关闭状态
5. 从转发状态到关闭状态

在该过程中用到**STP计时器**来控制收敛。

- Hello计时器 -- 2s (每个配置BPDU直接的时间)
- 转发延迟计数器 -- 15s (侦听/控制学习状态的为期) , Forward Delay -- 15 seconds (controls durations of Listening/Learning states)
- 最大存活时间 -- 20s (控制阻塞状态的为期) , Max Age -- 20 seconds (controls the duration of the Blocking state)

默认收敛时间是30到50秒。

## 生成树阻塞状态

### Spanning Tree Blocking State

处于阻塞状态的交换机端口，完成以下动作。

- 丢弃在该端口上接收到的来自所连接网段的数据帧，discards frames received on the port from the attached segment
- 丢弃交换自另一端口的数据帧，discards frames switched from another port
- 不将工作站地址放入到其地址数据库中，does not incorporate station location into its address database
- 接收BPDUs并将这些BPDUs引导给系统模块，receives BPDUs and directs them to the system module
- 不传送自系统模块接收到的BPDUs，does not transmit BPDUs received from the system module
- 接收网络管理报文，并对这些报文进行响应，receives and responds to network management messages

## 生成树侦听状态

### Spanning Tree Listening State

侦听状态是端口在阻塞状态之后所进入的第一个过渡状态。在STP确定端口应参与到帧转发时，该端口就进入此状态。处于侦听状态的交换机端口完成以下动作。

- 丢弃接收自所连接网段的帧，discards frames received from the attached segment
- 丢弃转发自另一端口的帧，discards frames switched from another port
- 不将工作站地址加入到其地址数据库，does not incorporate station location into its address database
- 接收BPDUs并将这些BPDUs引导给系统模块，receives BPDUs and directs them to the system module

- 接收、处理并传送接收自系统模块的BPDUs（在这一点上，与阻塞状态有所不同），receives, processes, and transmits BPDUs received from the system module
- 对网络管理报文进行接收和响应，receives and responds to network management messages

## 生成树学习状态

### Spanning Tree Learning State

学习状态是端口所进入的第二个过渡状态。此状态在侦听状态之后，且在端口进入转发状态之前到来。在此状态中，端口学习MAC地址并将学习到的MAC地址装入到其转发表中。处于学习状态的交换机端口完成以下动作。

- 丢弃接收自所连接网段的帧，discards frames received from the attached segment
- 丢弃转发自另一端口的帧，discards frames switched from another port
- 将工作站地址包含（安装）到其地址数据库（这一点与侦听状态有所不同），incorporates(installs) station location into its address database
- 接收BPDUs并将这些BPDUs引导给系统模块，receives BPDUs and directs them to the system module
- 接收、处理并传送接收自系统模块的BPDUs，receives, processes, and transmits BPDUs received from the system module
- 对网络管理报文进行接收和响应，receives and responds to network management messages

## 生成树转发状态

### Spanning Tree Forwarding State

转发状态是端口在学习状态之后所进入的第三个过渡状态。处于转发状态的端口对帧进行转发。处于转发状态的交换机端口完成以下动作。

- 转发接收自所连接网段的数据帧
- 转发交换自另一端口的数据帧（以上两点与学习状态不同，标志着开始转发数据）
- 将站点地址信息加入（安装）到其地址数据库
- 接收BPDUs并将这些BPDUs导向给系统模块
- 处理接收自系统模块的BPDUs
- 接收网络管理报文并对其进行响应

## 生成树关闭状态

### Spanning Tree Disabled State

关闭状态不是端口正常STP进展的部分。而是端口被网络管理员进行管理性关闭，或因为某种错误条件而被系统所关闭时，就被认为处于关闭状态。关闭的端口完成以下动作。

- 丢弃接收自所连接网段的数据帧
- 丢弃转发自另一端口的数据帧

- 不将工作站地址加入其地址数据库
- 接收BPDUs但不将这些BPDUs导向给系统模块
- 不接收来自系统模块的BPDUs
- 对网络管理报文进行接收和响应

## 生成树桥ID

### Spanning Tree Bridge ID

位于某个生成树域中的交换机，都有一个用于对其进行唯一性区分的桥ID（Bridge ID, BID）。BID还用于协助完成STP根桥(an STP Root Bridge)的选举，STP根桥将在稍后讲到。BID是由一个6字节的MAC地址及2字节的桥优先级（a 2-byte Bridge Priority）构成的8字节字段。下图31.3演示了BID。



图31.3 -- 桥ID格式

**桥优先级**是该交换机相对于其它交换机的优先级。桥优先级取值范围是0到65535。思科Catalyst交换机的默认值为32768。

```

Switch2#show spanning-tree vlan 2

VLAN0002
  Spanning tree enabled protocol ieee
  Root ID Priority    32768
    Address      0009.7c87.9081
    Cost        19
    Port        1 (FastEthernet0/1)
    Hello Time   2 sec Max Age 20 sec Forward Delay 15 sec
  Bridge ID Priority 32770 (priority 32768 sys-id-ext 2)
    Address      0008.21a9.4f80
    Hello Time   2 sec Max Age 20 sec Forward Delay 15 sec
    Aging Time   300

  Interface  Port ID          Designated          Port ID
  Name       Prior.Nbr     Cost   Sts Cost      Bridge ID      Prior.Nbr
  -----      -----          -----          -----
  Fa0/1      128.1           19    FWD  0  32768  0009.7c87.9081  128.13
  Fa0/2      128.2           19    FWD 19 32770  0008.21a9.4f80  128.2

```

上面输出中的MAC地址是得自交换机背板或管理引擎的硬件地址（the hardware address derived from the switch backplane or supervisor engine，又名为基底MAC地址，the base MAC address）。**在802.1D标准中，每个VLAN都需要一个唯一BID。**

大多数思科Catalyst交换机都有一个可用作VLANs的BIDs的、1024个MAC地址的地址池。这些MAC地址被顺序分配，也就是该范围中的第一个MAC地址分配给VLAN 1，第二个给VLAN 2，第三个给VLAN 3，以致第四个第五个等等。这样就提供了支持标准范围VLANs的支持能力，但要支持扩展范围的VLANs，就需要更多的MAC地址。该问题在802.1t（802.1D的技术和编辑修正）标准中得以解决（this issue was resolved in the 802.1t(Technical and Editorial corrections for 802.1D) standard）。

## 生成树根桥选举

### Spanning Tree Root Bridge Election

默认情况下，紧接着初始化之后，所有交换机最初都假定它们是生成树的根，直到它们与其他交换机交换BPDUs为止。在交换机交换BPDUs时，就举行一次选举，而网络中有着最低桥ID的交换机就被选举为STP根桥(the STP Root Bridge)。如有两台或更多交换机有着相同的优先级，则选取有着最低顺序MAC地址的交换机作为根桥。下图31.4对此概念进行了演示。

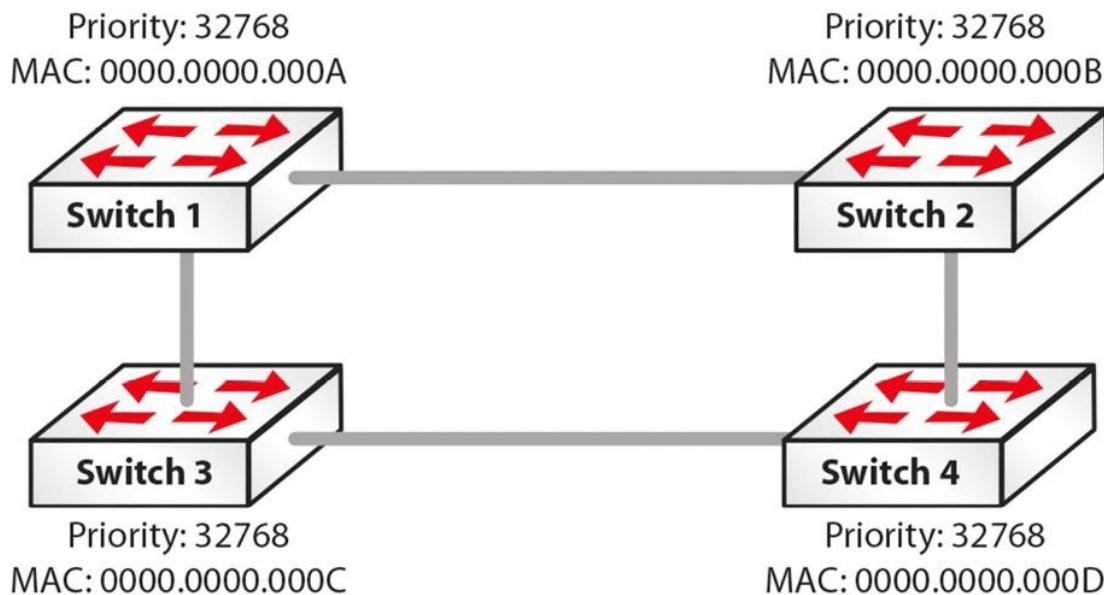


图31.4 -- STP根桥的选举

在图31.4中，四台交换机--Switch 1、Switch 2、Switch 3及Switch 4, 处于同一STP域中。默认所有交换机都有着桥优先级32768。为确定哪台交换机将成为根桥，并由此打破不分胜负的局面，STP将基于最低顺序MAC地址选择出根桥交换机 (in order to determine which switch will become the Root Bridge, and thus break the tie, STP will select the switch based on the lowest-order MAC address)。那么基于此标准，并参考图31.4给出的信息，Switch 1将被选举为根桥。

一旦选定，根桥就成为生成树网络的逻辑中心。这并不是说根桥位处该网络的物理中心。确保不要做出那样的错误假设。

**注意：**重要的是记住在STP根桥选举期间，是没有流量在该相同STP域上转发的。

**思科IOS软件允许管理员对根桥选举施加影响。**此外，管理员也可以配置一台备份根桥 (administrator can also configure a backup Root Bridge)。备份根桥是一台管理员优先选择、在当前根桥失效或从网络中移除时成为根桥的交换机。

**为生成树域配置一台备份根桥交换机，始终是好的做法。**这样做允许在根桥失效时，网络具有确定性。最常见的做法就是在根桥上配置最高的优先级（也就是优先级为最低数值），并将第二高的优先级配置在当前根桥失效时作为根桥的备份交换机上。下图31.5对此进行了演示。

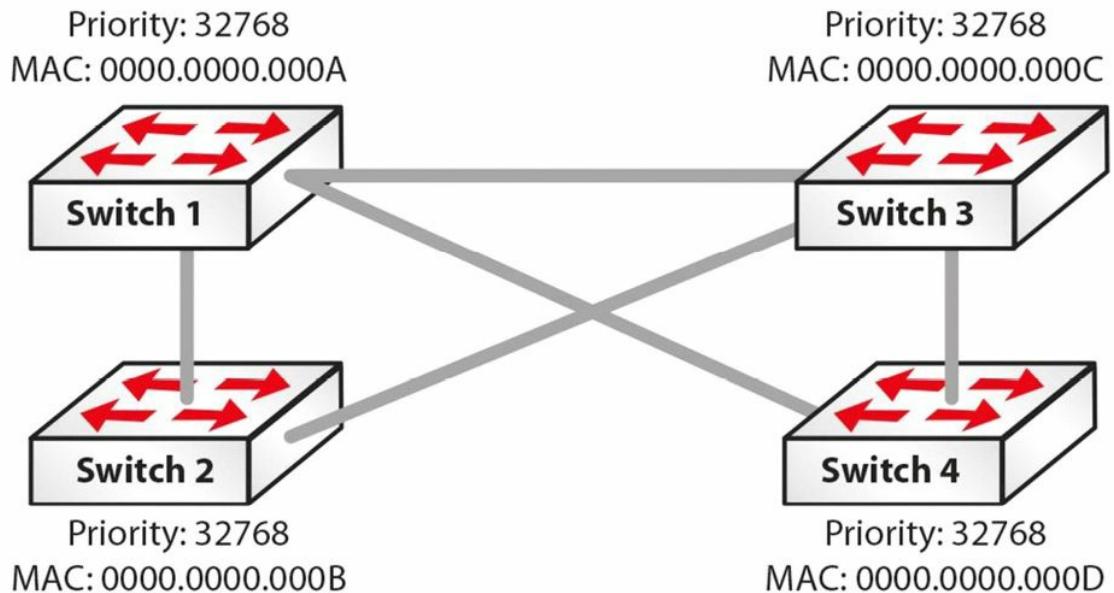


图31.5 -- STP根桥选举 (续)

基于图31.5中的配置，最有可能被选举作为根桥的交换机是Switch 1。这是因为尽管所有优先级都一样，但该交换机有着最低顺序的MAC地址。而假如Switch 1失效，STP就会选举Switch 2作为根桥，因为它有着第二低的MAC地址。但是这将导致一个次优的网络拓扑 (however, this would result in a suboptimal network topology)。

为解决此问题，管理员可手动修改Switch 1上的优先级到可能的最低值（0），以及Switch 2的优先级到可能的第二低优先级值（4096）。这样做将确保在根桥（Switch 1）失效时，Switch 2被选举为根桥。因为管理员知道网络拓扑并了解哪台交换机将承担根桥功能，那么就建立了一个具有确定性、更容易排错的网络。

**根ID (the Root ID) 承载于BPDUs中，包含了根桥的桥优先级及MAC地址。**

**考试技巧：**如要强制某台交换机成为根桥，可执行下面的命令（同时参见下图31.6）。

- 可以手动设置优先级

```
Switch(config)#spanning-tree vlan 2 priority ?
<0-61440>  bridge priority in increments of 4096
```

- 或者使用宏命令 primary 或 secondary 将其设置为根桥

```
Switch(config)#spanning-tree vlan 2 root ?
primary      Configure this switch as primary root for this spanning tree
secondary    Configure switch as secondary root
```

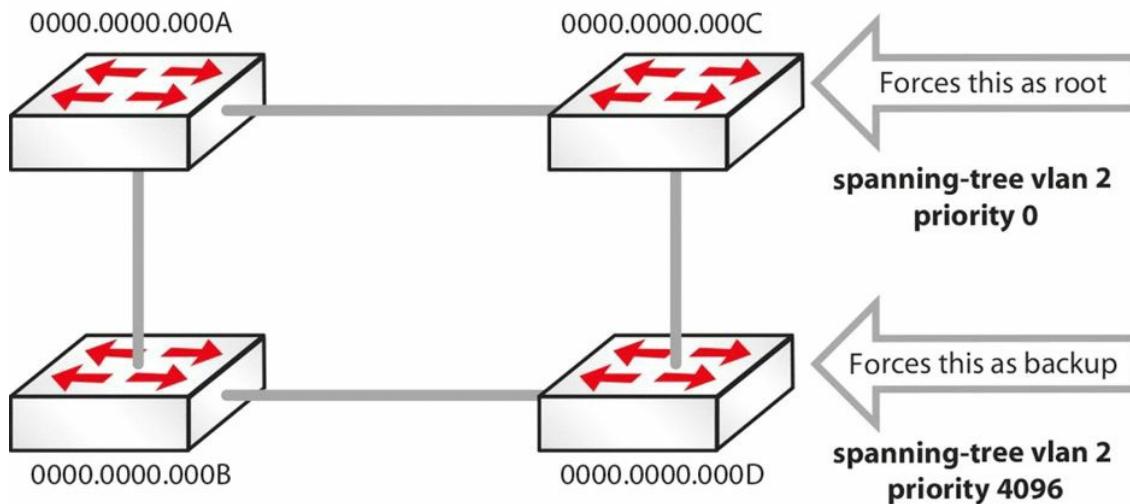


图31.6 -- 强制某台交换机成为根桥

```

SwitchC#show spanning-tree vlan 5
VLAN0005
Spanning tree enabled protocol ieee
Root ID  Priority  0
Address  0000.0000.000c
This bridge is the root
Bridge ID  Priority  0 (priority 0 sys-id-ext 5)
SwitchD#show spanning-tree vlan 5
VLAN0005
Spanning tree enabled protocol ieee
Root ID  Priority  4096
Address  0000.0000.000d
Bridge ID  Priority  4096 (priority 8192 sys-id-ext 5)
SwitchD#show spanning-tree vlan 5
VLAN0005
Spanning tree enabled protocol ieee
Root ID  Priority  4096
Address  0000.0000.000d
Bridge ID  Priority  4096 (priority 8192 sys-id-ext 5)

```

注意到VLAN编号通常会被加到优先级数字上，如下面的输出展示的那样。

```

SwitchA#show spanning-tree vlan 5
Bridge ID  Priority 32773 (priority 32768 sys-id-ext 5)
Address 0013.c3e8.2500
Hello Time 2 sec Max Age 20 sec Forward Delay 15 sec
Aging Time 300
Interface  Role   Sts    Cost    Prio.Nbr  Type
-----  ----  -----  -----  -----
Fa0/15    Desg   FWD    19      128.15    P2p
Fa0/18    Desg   FWD    19      128.18    P2

```

## 生成树开销及优先级

### Spanning Tree Cost and Priority

STP使用开销及优先级数值来确定到根桥的最优路径。这些数值此时用在根端口（the Root Port）的选举中，根端口选举将在接着的小节中讲到。掌握开销及优先级数值的计算，对于理解为何生成树选举一个端口而不是另一个，是十分重要的。

生成树算法的一项关键功能，就是尝试提供出网络中的各台交换机自根桥的最短路径。而该最短路径一旦选定，就被用于转发数据，而将其它冗余链路置为阻塞状态。生成树算法用到两个数值来确定哪个端口将被置为转发状态（也就是到根桥的最优路径），以及哪些端口将被置为阻塞状态。这两个数值就是端口开销和端口优先级。二者都将在下面的小节讲到。

## 生成树端口开销

### Spanning Tree Port Cost

802.1D规格分配16位（短整数）基于端口带宽的默认端口开销值给每个端口。因为管理员同时有着手动分配端口开销值（1和65535之间）的能力，所以该16位值就只用在那些未有具体配置了端口开销的端口。下表31.1列出了在应用短整数方式计算端口开销时各种类型端口的默认值。

表 31.1 -- 默认STP端口开销值

带宽	默认端口开销
4Mbps	250
10Mbps	100
16Mbps	62
100Mbps	19
1Gbps	4
10Gbps	2

在思科IOS Catalyst交换机中，可通过执行 `show spanning-tree interface [name]` 查看默认端口开销值，如下面的输出中演示的那样，该输出展示了一个FastEthernet接口的默认短整数端口开销。

```
VTP-Server#show spanning-tree interface FastEthernet0/2
Vlan      Role     Sts    Cost    Prio.Nbr    Type
-----  -----  -----  -----  -----  -----
VLAN0050  Desg    FWD    19      128.2      P2p
```

下面的输出显示了同样的长整数端口开销分配（the following output shows the same for long port cost assignment）。

```
VTP-Server#show spanning-tree interface FastEthernet0/2
Vlan      Role     Sts    Cost    Prio.Nbr    Type
-----  -----  -----  -----  -----  -----
VLAN0050  Desg    FWD    200000  128.2      P2p
```

重要的是记住带有更低的（数值）开销的端口是更为首选的端口；端口开销越低，那个特定端口被选举为根端口的可能性就越高（the lower the port cost, the higher the probability of that particular port being elected the Root Port）。**端口开销全局重要，并影响整个生成树网络。**该数值被配置在生成树域中的所有非根交换机上（on all Non-Root Switches in the Spanning Tree domain）。

## 生成树的根端口及指定端口

### Spanning Tree Root and Designated Ports

STP选举出两种类型用于转发BPDUs的端口：指向根桥的根端口，以及指向根端口另一边的指定端口

(STP elects two types of ports that are used to forward BPDUs: the Root Port, which points towards the Root Bridge, and the Designated Port, which points away from the Root Bridge)。掌握这两种端口类型的作用及其选举过程，十分重要。

## 生成树根端口选举

### Spanning Tree Root Port Election

生成树算法定义了三种端口类型：**根端口**、**指定端口**及**非指定端口**。这些端口类型是有生成树算法选举出来，并被置为相应状态（比如转发中或阻塞中状态）。在生成树选举过程中，如存在悬而不决的情况，就会用到以下数值作为打破僵局方式。

1. 最低的根桥ID, lowest Root Bridge ID
2. 到根桥的最低根路径开销, lowest Root path cost to Root Bridge
3. 最低的发送方桥ID, lowest sender Bridge ID
4. 最低的发送方端口ID, lowest sender Port ID

**注意：**为掌握生成树选举及指定出在任何给定情形下不同端口类型，那么重要的是记住这些打破平局的标准了。这些标准不仅要对其进行测试，还要为真实世界中设计、部署及支持互联网络而牢固掌握这个知识点。

生成树根端口是在该设备将数据包转发到根桥时，提供出最优路径，或最低开销的端口。也就是说，根端口是接收到该交换机的最优BPDU的端口，而这又表明了在路径开销上其是到根桥的最短路径。根端口是基于根桥路径开销选举出的。

根桥路径开销又是基于连接到根桥的所有链路的累积开销（路径开销）计算出的。路径开销是各个端口贡献给根桥开销的数值（the path cost is the value that each port contributes to the Root Bridge path cost）。因为此概念通常是十分令人困惑，在下图31.7中对其进行了演示。

**注意：**图31.7中除了一条链路外，其它链路都是GigabitEthernet链路。应假定用于端口开销计算的方法是传统的802.1D方法。因此，默认GigabitEthernet的端口开销就是4，同时FastEthernet是19。

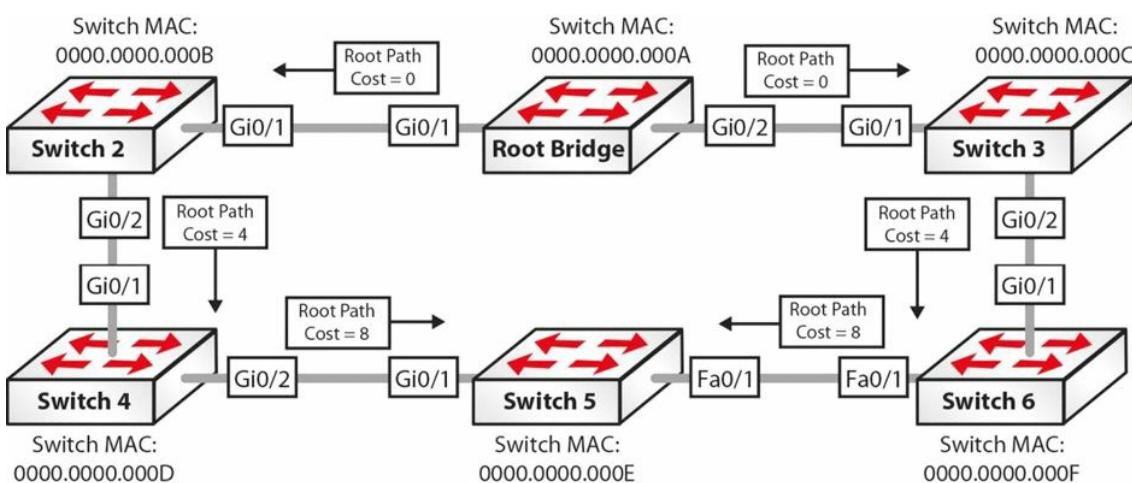


图31.7 -- 生成树根端口选举

**注意：**下面的解释对网络中交换机间的BPDU数据流进行了说明。与其它信息一起，这些BPDU包含了根桥路径开销信息，而根桥路径开销在接收交换机上的入站端口处被增加（along with other information, these BPDU contain the Root Bridge path cost information, which is incremented by the ingress port on the receiving switch）。

1. 根桥发出一个带有根桥路径开销值0的BPDU，因为其端口直接位于该根桥上。此BPDU发送给Switch 2 和Switch 3。
2. 当Switch 2和Switch 3接收到来自根桥的BPDU时，它们便基于各自入站借口加上其自己的路径开销。因为Switch 2和Switch 3都是通过GigabitEthernet连接与根桥相连，所以它们将从根桥接收到的路径开销值（0）与它们的GigabitEthernet路径开销值（4）相加。Switch 2及Switch 3经由GigabitEthernet0/1 到根桥的根桥路径开销也就是 $0+4=4$ 。
3. Switch 2和Switch 3将新的BPDUs送出至其各自的邻居，也就是Switch 4和Switch 6。这些BPDUs包含了新的累积值（4）作为根桥路径开销。
4. 当Switch 4和Switch 6接收到分别来自Switch 2和Switch 3的BPDUs时，它们根据入站借口对接收到的根桥路径开销予以增长。因为使用的是GigabitEthernet，从Switch 2和Switch 3接收到的值被加上4。那么在Switch 4和Switch 6上经由其各自GigabitEthernet0/1接口的根桥路径开销就是 $0+4+4=8$ 。
5. Switch 5接收到两个BPDUs：一个来自Switch 4，另一个来自Switch 6。接收自Switch 4的BPDU有着根桥路径开销 $0+4+4+4=12$ 。接收自Switch 6的BPDU有着根桥路径开销 $0+4+4+19=27$ 。因为包含于接收自Switch 4的BPDU中的根桥路径开销值好于接收自Switch 6的，Switch 5将选举GigabitEthernet0/1 作为根端口（the Root Port）。

**注意：**交换机2、3、4、6都将选举其各自的GigabitEthernet端口作为根端口。



### 更多解释

#### Further Explanation

为更为细致的进行解释（To explain further），并有助于掌握根端口选举过程，假定上图31.7中所有端口都是GigabitEthernet端口。这就意味着在上面的第5步中，Switch 5将接收到两个带有相同根桥ID的BPDUs，且两个都有着 $0+4+4+4=12$ 的根路径开销值。为了选举出根端口，STP将进入到下面所列出的打破僵局标准的下一选项（前两个选项已经用到，就被移除了）。

1. 最低发送方桥ID，lowest sender Bridge ID
2. 最低发送方端口ID，lowest sender Port ID

基于第三个选举标准，Switch 5将优先使用来自Switch 4的BPDU，因为Switch 4的BID（0000.0000.000D）低于Switch 6的BID（0000.0000.000F）。Switch 5选出端口GigabitEthernet0/1作为根端口。

## 生成树指定端口的选举

#### Spanning Tree Designated Port Election

与根端口不同，指定端口是指向与STP根相反方向的端口。该端口是指定设备（交换机）连接LAN的端口。指定端口同时也是在将来自LAN的数据包转发给根桥时有着最低路径开销的端口。

**注意：**一些人将指定端口当作是指定交换机。这两个术语是可以互换的，且指的是同一个东西。也就是说，这是用于将来自某个特定LAN网段的帧，转发到根桥的交换机，或端口。

**指定端口的主要目的是阻止循环。**在超过一台的交换机连接到同一网段时，所有交换机都将尝试对在那个网段上接收到的某个帧进行转发。这样的默认行为可能导致该帧的多个拷贝被多台交换机同时转发--从而造成网络循环。为避免这种默认行为，**STP在所有网段上都选举出一个指定端口。**\*这是因为根桥路径开销将

始终为0。\*STA的指定端口选举过程在下图31.8中进行了演示。

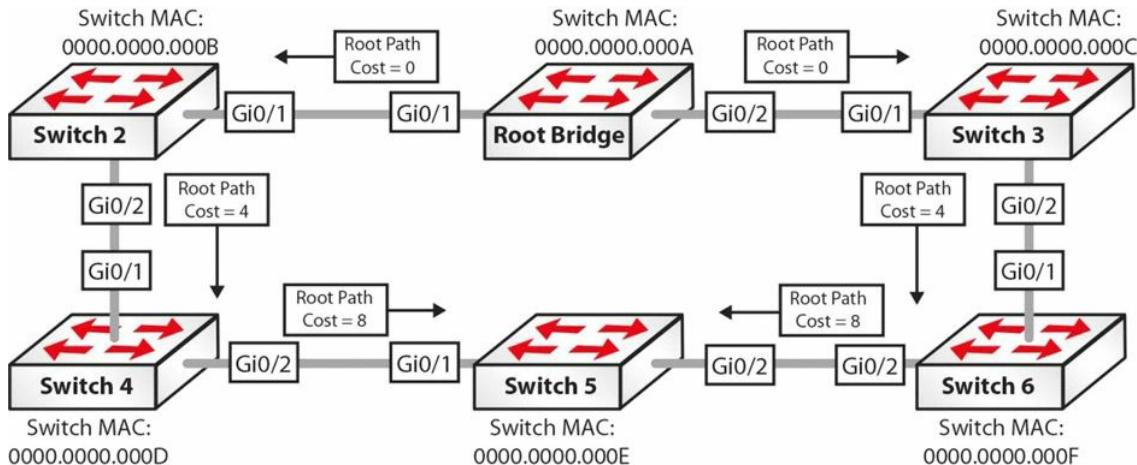


图31.8 -- 生成树指定端口选举

1. 在根桥和Switch 2之间的网段上，根桥的GigabitEthernet0/1被选举为指定端口，因为该端口有着较低的根桥路径开销0。
2. 在根桥和Switch 3之间的网段，根桥的GigabitEthernet0/2端口被选举作为指定端口，因为其有着较低的根桥路径开销0。
3. 在Switch 2和Switch 4之间的网段，Switch 2上的GigabitEthernet0/2被选举为指定端口，因为Switch 2有着最低的根桥路径开销4。
4. 在Switch 3和Switch 6之间的网段，Switch 3上的GigabitEthernet0/2端口被选举为指定端口，因为Switch 3有着最低的根桥路径开销4。
5. 在Switch 4和Switch 5之间的网段，Switch 4上的GigabitEthernet0/2端口被选举为指定端口，因为Switch 4有着最低的根桥路径开销8。
6. 在Switch 5和Switch 6之间的网段，Switch 6上的GigabitEthernet0/2被选举为指定端口，因为Switch 6有着最低的根桥路径开销8。

非指定端口 (the Non-Designated Port) 实际上不是一种生成树端口类型。而是其作为一个术语，只是简单地表示某个不作为某LAN网段上指定端口的端口。非指定端口将始终被STP置为阻塞状态。基于根端口及指定端口的计算，下图31.9中展示了用于根端口和指定端口选举示例的交换网络的最终生成树拓扑 (Based on the calculation of Root and Designated Ports, the resultant Spanning Tree Topology for the switched network that was used in the Root Port and Designated Port election examples is shown in Figure 31.9 below)。

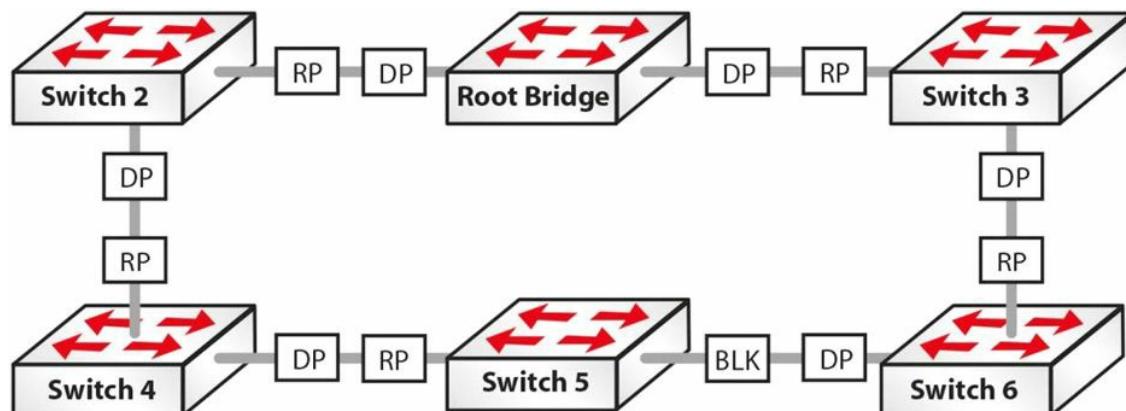


图31.9 -- 已收敛的生成树网络

## 思科生成树增强

### Cisco Spanning Tree Enhancements

如早前指出的那样，STP对其所在环境做出以下两点假设。

- 所有链路都是双向的，而能够发送和接收桥协议数据单元。
- 所有交换机都能正常地接收、处理及发出BPDUs

在现实世界的网络中，这两个假设并不总是正确。在这种情况下，STP就可能无法阻止网络中循环的形成  
(in situations where that is the case, STP may not be able to prevent loops from being formed within the network)。正是由于存在这种可能，且为提升基本的802.1D STA性能，思科引入了一些对IEEE 802.1D标准的增强，将在下面进行说明。

## 端口快速

### Port Fast

端口快速是一项典型地对连接了一台主机的端口或接口开启的特性。当该端口上的链路起来时，交换机将跳过STA的第一阶段并直接过渡到转发状态。与通常的看法相反，端口快速特性并不在选定的端口上关闭生成树。这是因为就算带有端口快速特性，该端口仍能发送并接收BPDUs。

这在该端口所连接的诸如某台工作站的网卡这样的，没有发送或响应BPDUs的网络设备时不是问题。但如该端口所连接的设备确实在发出BPDUs，比如另一台交换机，这可能造成交换循环。这是因为该端口跳过了侦听及学习阶段而立即进入到转发状态 (this may result in a switching loop. This is because the port skips the Listening and Learning states and proceeds immediately to the Forwarding state)。端口快速简单地令到该端口相较经历所有STA步骤，快得多地开始转发以太网帧。

## BPDU守护

### BPDU Guard

BPDU守护特性用于保护生成树域免受外部影响。BPDU默认是关闭的，但建议在所有开启了端口快速特性的端口上予以开启。在配置了BPDU守护特性的端口接收到一个BPDU时，就立即转变成错误关闭状态 (the errdisable state)。

在那些关闭了生成树的端口上，这样做阻止了错误信息注入到生成树域中去。BPDU守护的运行，结合端口快速特性，在下面及后续的图31.10、31.11及31.12中，进行了演示。

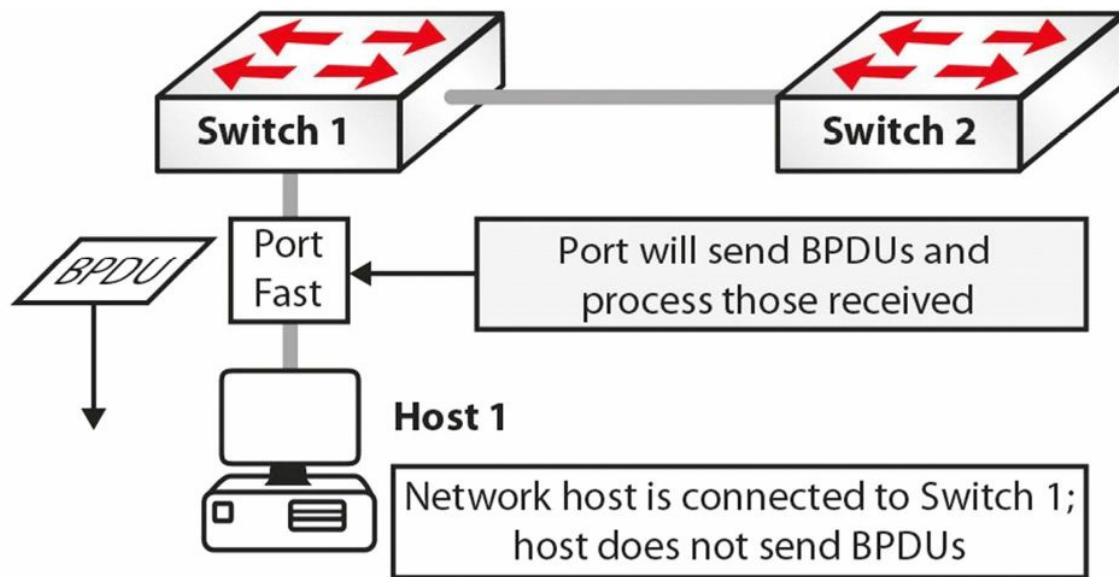


图31.10 -- 掌握BPDU守护

图31.10中，Switch 1到Host 1的连接上开启了端口快速。那么在初始化后，该端口便过渡到转发状态，这就消除了该端口在没有省略掉STA而要走完侦听及学习状态所要花掉的30秒。因为该网络主机是一台工作站，其不在那个端口上发送BPDUs。

要么因为偶然，或是由于一些其它恶意目的，Host 1从Switch 1上断开连接。使用同一端口，Switch 3被连接到Switch 1。Switch 3同时也连接到Switch 2。因为端口快速在连接Switch 1到Switch 3的端口上开启，此端口就从初始化变成转发状态，从而省略掉了一般STP初始化过程。此端口将接收并处理所有由Switch 3发送的BPDUs，如下图31.11所示。

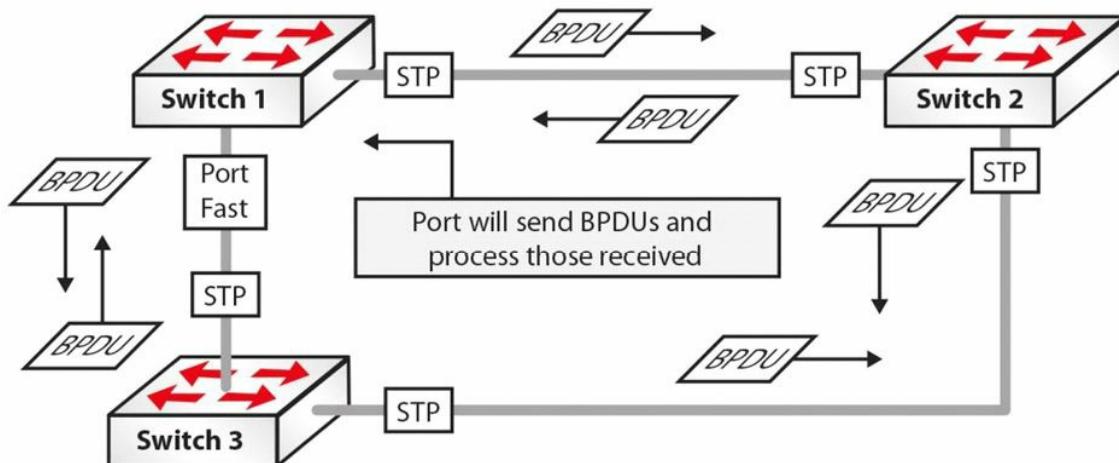


图31.11 掌握BPDU守护 (续)

基于上面所演示的端口状态，可很快看出一个循环将在此网络中如何建立起来。为阻止此情形的发生，就应在所有的那些开启了端口快速的端口上，开启BPDU守护。这在下面的图31.12中进行了演示。

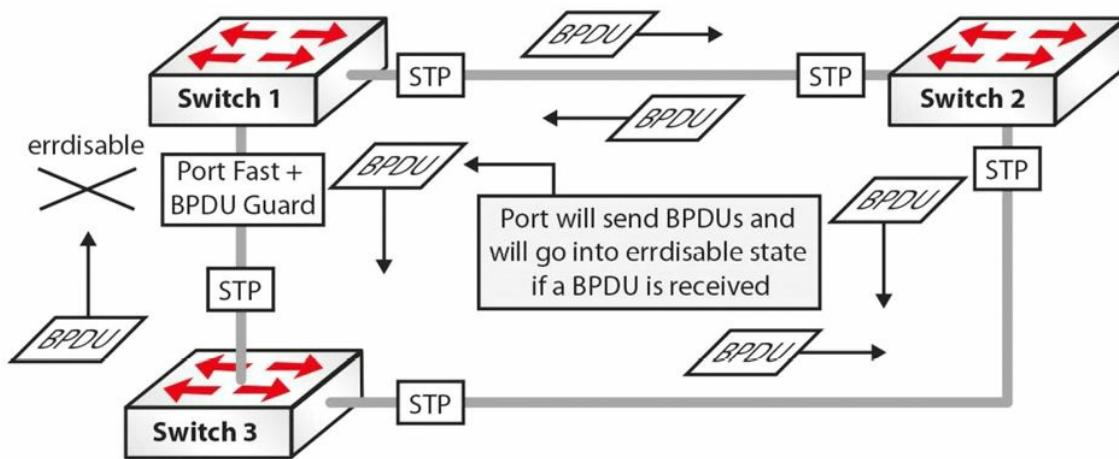


图31.12 -- 掌握BPDU守护 (续)

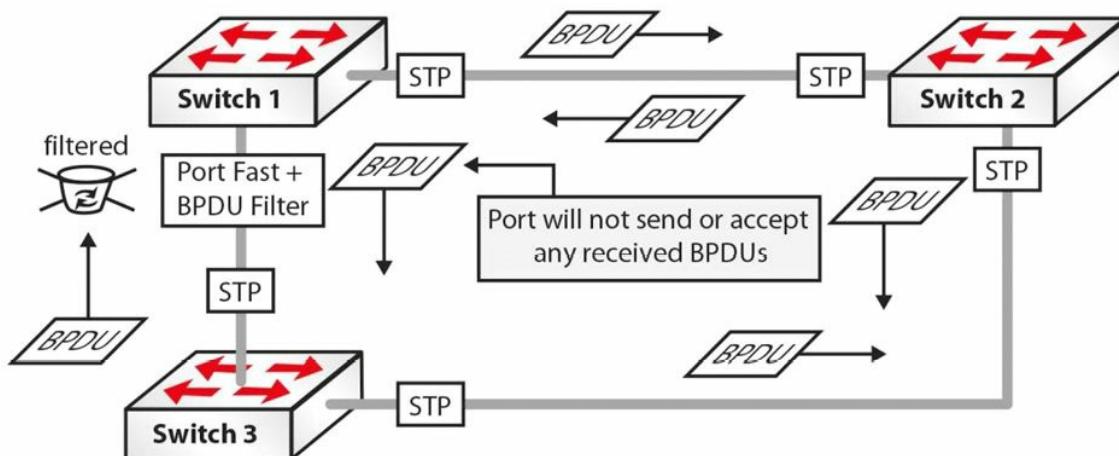
在端口快速端口上带有BPDU守护下，在Switch 1接收到来自Switch 3的一个BPDU时，就立即将该端口转变成错误关闭状态（immediately transitions the port into the errdisabled state）。结果就是STP计算不受该冗余链路的影响，且该网络不会有任何循环。

## BPDUs过滤器

### BPDUs Filter

BPDUs过滤器与BPDU守护两个特性常常混淆或甚至被想成是同一个特性。但它们是不同的，而掌握它们之间的区别就很重要。在某个端口上开启了端口快速时，该端口将发出BPDUs且将接受及处理收到的BPDUs。BPDU守护特性阻止该端口接收任何的BPDUs，但不阻止其发送BPDUs。如有接收到任何BPDUs，该端口就将成为错误关闭端口（if any BPDUs received, the port will be errdisabled）。

而BPDU过滤器特性有着两方面的功能（the BPDU Filter feature has dual functionality）。当在接口级别配置上BPDU过滤器时，它将有效地在选定端口上，通过阻止这些端口发送或接收所有BPDUs，而关闭这些端口的STP。而在全局配置了BPDU过滤器，并与全局端口快速配合使用时，它会将任何接收到BPDUs的端口，还原成端口快速模式。下图31.13对此进行了演示。



## 循环守护

### Loop Guard

循环守护特性用于防止生成树网络中循环的形成。循环守护对根端口及阻塞端口进行探测，并确保它们继续接收BPDUs。当交换机在阻塞端口上接收到BPDUs，该信息就被忽视，因为来自根桥的最佳BPDUs仍通过根端口，正在接收着。

如该交换机链路是运行的，又没有接收到BPDUs（因为该链路是单向链路，due to a unidirectional link），该交换机就假设将该链路开启是安全的，那么该端口就转换到转发状态并开始对接收到的BPDUs进行中继。如有某台交换机连接到该链路的另一端，这将有效地建立起一个生成树循环。下图31.14对此概念进行了演示。

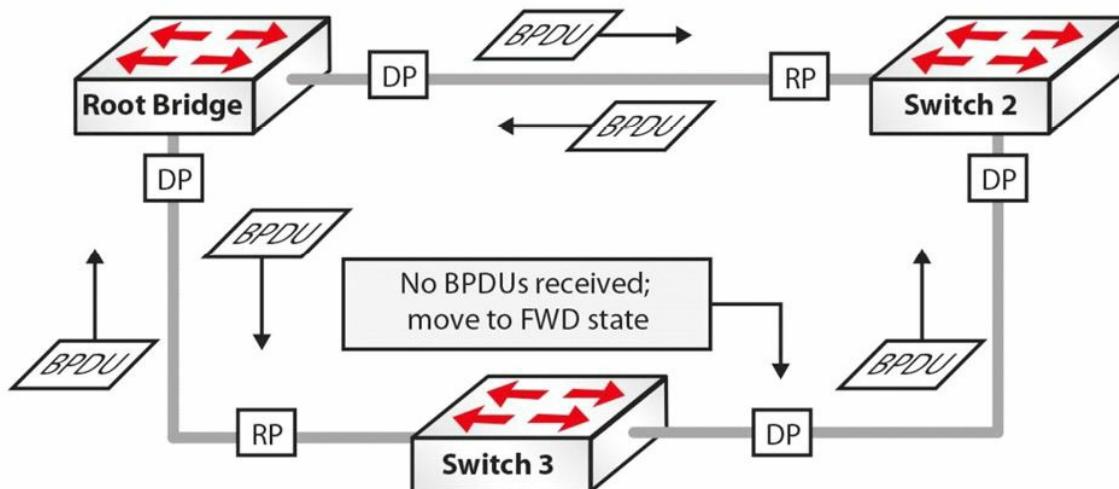


图31.14 -- 掌握循环守护

图31.14中，该生成树网络已完成收敛，从而所有端口都处于阻塞或转发状态。但是，因为一条单向链路，Switch 3上的阻塞端口停止了接收来自Switch 2上的指定端口的BPDUs。Switch 3假定该端口可被转换成转发状态，并开始此转换。该交换机此时就将接收到的BPDUs中继出那个端口，从而导致网络循环。

在循环守护开启时，Switch 3保持对所有非指定端口的追踪。在端口持续接收到BPDUs时，该端口就是好的；但如该端口停止接收到BPDUs，就被转移到循环不一致状态（a loop-inconsistent state）。也就是说，在循环守护开启时，STP端口状态机（the STP port state machine）被修改为在缺少BPDUs时，阻止该端口从非指定端口角色转变成指定端口角色（in other words, when Loop Guard is enabled, the STP port state machine is modified to prevent the port from transitioning from the Non-Designated Port role to the Designated Port role in the absence of BPDUs）。在应用循环守护时，应知道以下这些应用准则。

- 不能在开启了根守护（Root Guard）的交换机上开启循环守护，Loop Guard cannot be enabled on a switch that also has Root Guard enabled
- 循环守护不影响上行快速（Uplink Fast）或骨干快速（Backbone Fast）的运行，Loop Guard does not affect Uplink Fast or Backbone Fast operation
- 循环守护只是必须在点对点链路上开启，Loop Guard must be enabled on Point-to-Point links only
- 循环守护的运行不受生成树计时器的影响，Loop Guard operation is not affected by the Spanning Tree timers
- 循环守护无法真正探测出一条单向链路，Loop Guard cannot actually detect a unidirectional link
- 循环守护无法在端口快速或动态VLAN端口上开启，Loop Guard cannot be enabled on Port Fast or Dynamic VLAN ports

## 根守护

### Root Guard

**根守护特性阻止指定端口成为根端口。**如在某个根守护特性开启的端口上接收到一个优良BPDU (a superior BPDU)，根守护将该端口移入根不一致状态 (a root-inconsistent state) ,从而维持当前根桥状态 (thus maintaining the current Root Bridge status quo) 。下图31.15对此概念进行了演示。

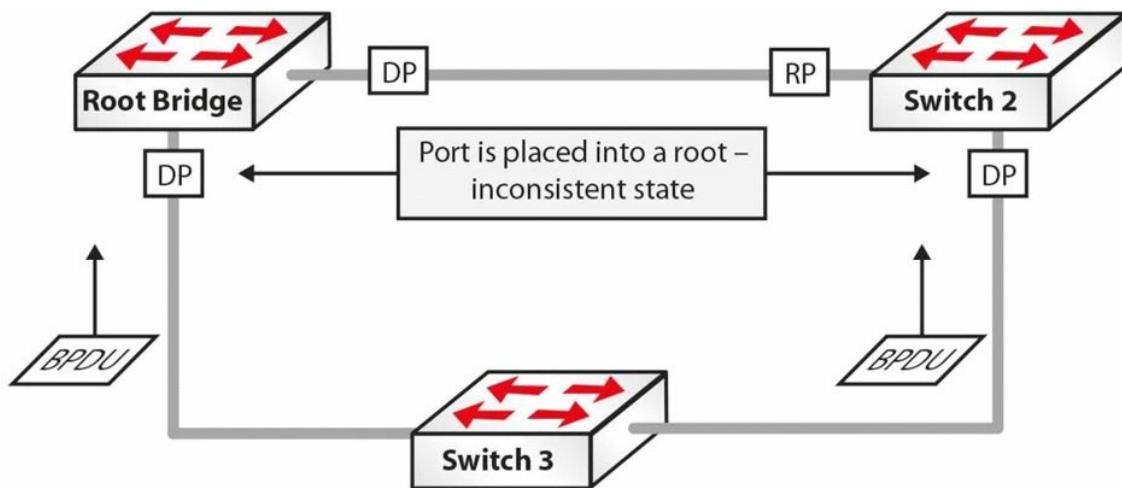


图31.15 -- 掌握根守护

图31.15中，Switch 3被加入到当前STP网络，并发出比当前根桥更优质的BPDUs。在通常情况下，STP将重新计算整个拓扑，同时Switch 3将会被选举为根桥。但因为当前根桥及Switch 2上的指定端口上开启了根守护特性，在接收到来自Switch 3的优良BPDUs时，两台交换机都会将这些指定端口置为根不一致状态。这样做保护了生成树拓扑。

**根守护阻止某个端口成为根端口，因此确保该端口始终是指定端口。**与其它可同时在全局基础上开启的思科STP增强不同，根守护必须手动在所有根桥不应出现的端口上开启 (unlike other STP enhancements, which can also be enabled on a global basis, Root Guard must be manually enabled on all ports where the Root Bridge should not appear) 。因为这点，在LAN中STP的设计和部署时确保拓扑的确定性就很重要 (because of this, it is important to ensure a deterministic topology when designing and implementing STP in the LAN) 。根守护令到网络管理员可以强制指定网络中的根桥 (Root Guard enables an administrator to enforce the Root Bridge placement in the network) ，确保不会有客户设备因疏忽或其它原因而成为生成树的根，所以根守护常用在ISP网络面向客户设备的边界 (so it is usually used on the network edge of the ISP towards the customer's equipment) 。

## 上行快速

### Uplink Fast

**上行快速特性提升了在主要链路失效（根端口的直接失效）时，更快的到冗余链路的切换** (the Uplink Fast feature provides faster failover to a redundant link when the primary link fails(i.e., direct failure of the Root Port)) 。该特性的主要目的是在出现上行链路失效时，提升STP的收敛时间。该特性在带有到分布层冗余链路的接入层交换机上用的最多；这也是其名称的由来。

在接入层交换机有着到分布层的双宿主时，其中一条链路被被STP置为阻塞状态以防止环回 (when Access Layer switches are dual-homed to the Distribution Layer, one of the links is placed into a Blocking state by STP to prevent loops) 。在到分布层的主链路失效时，处于阻塞状态的端口就必须在开始转发流量之前，转换到侦听和学习状态。这导致在交换机能够转发以其它网段为目的的帧之前，有一个30秒的延迟。上行快速的运作，在下图31.16中进行演示。

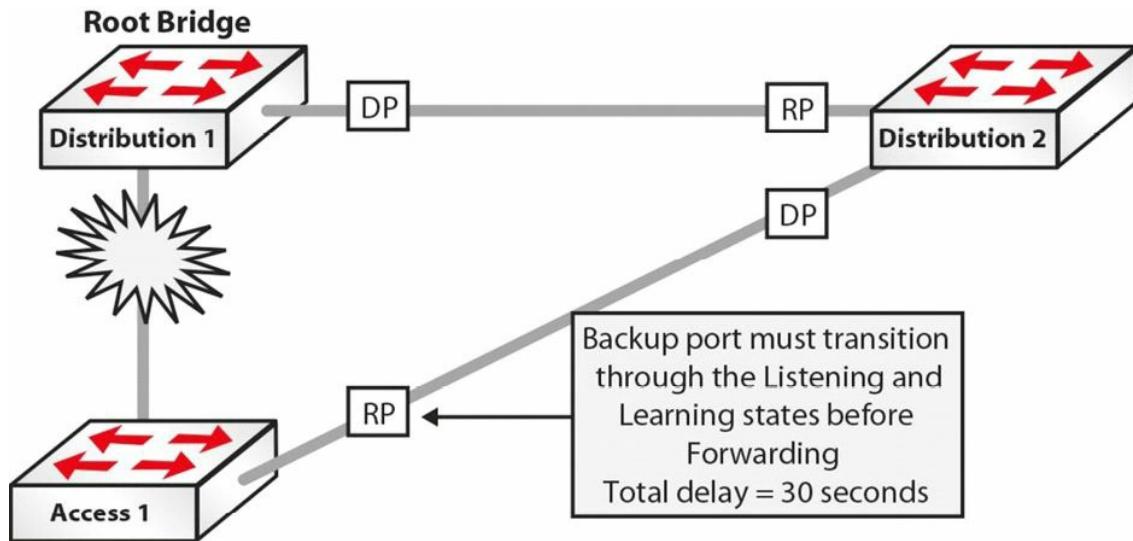


图31.16 -- 掌握上行快速

图31.16中，在Access 1和Distribution 1之间的链路上出现了失效，Distribution 1是STP根桥，此失效意味着STP会将Access 1和Distribution 2之间的链路移入转发状态（也就是“阻塞中”>“侦听中”>“学习中”>“转发中”，Blocking > Listening > Learning > Forwarding）。侦听和学习阶段各耗时15秒，所以该端口只需在总共30秒过去之后，便开始转发数据帧。而在上行快速开启时，到分布层的后备端口被立即置为转发状态，从而带来无网络宕机时间的结果。下图31.17对此概念进行了演示。

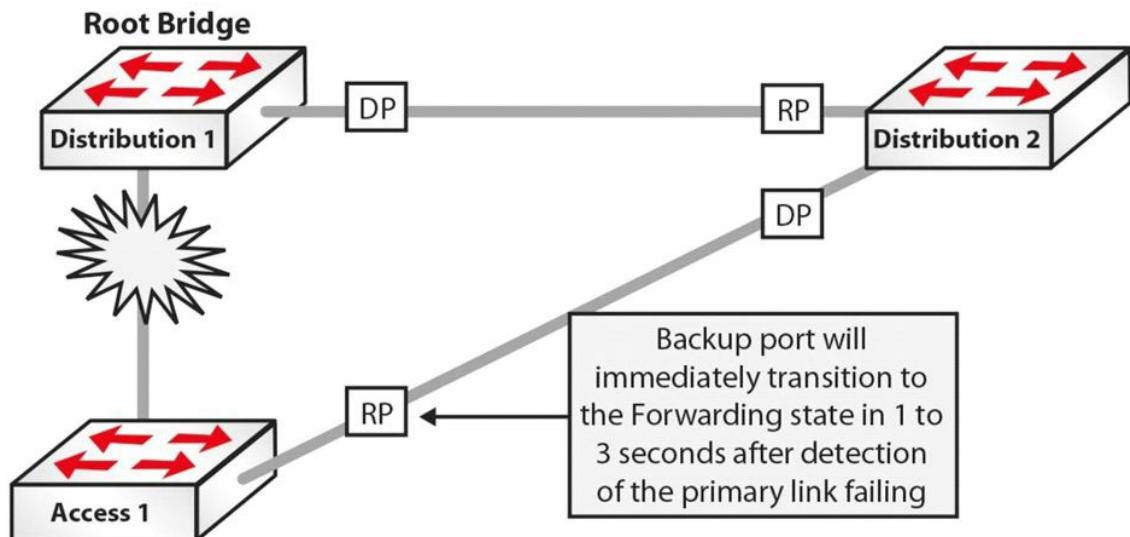


图31.17 -- 掌握上行快速 (续)

## 骨干快速

### Backbone Fast

骨干快速特性提供了STP域中一条非直连链路出现失效时的快速切换。在交换机从其指定桥（在其根端口上）接收到一个较差BPDU时，快速切换便发生了。一个较差BPDU表明指定桥失去了其到根桥的连接，所以该交换机知悉存在上游失效而无需等待计时器超时就改变根端口。下图31.18中对此进行了演示。

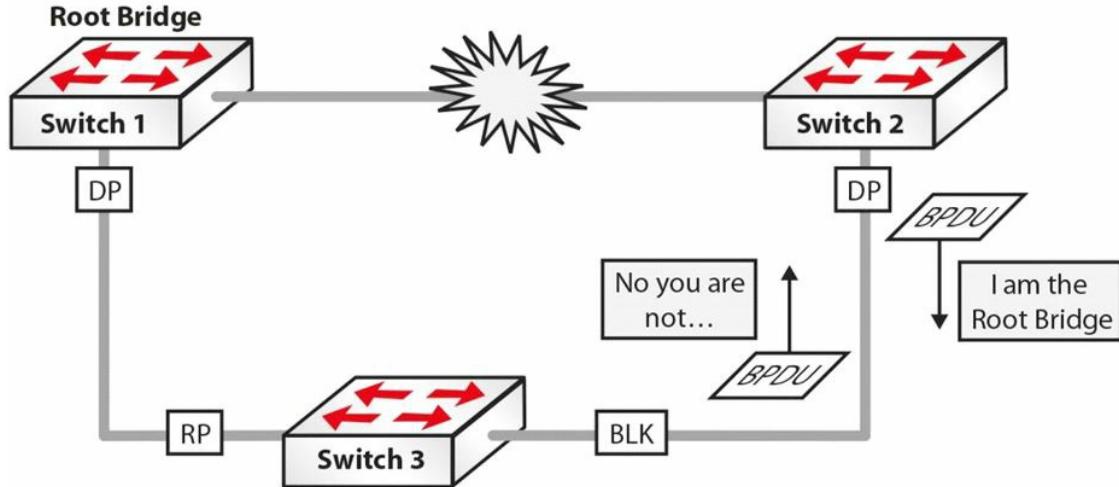


图31.18 -- 掌握骨干快速

图31.18中，Switch 1和Switch 2之间的链路挂掉了。Switch 2探测到这个问题并发出BPDUs表明它是根桥。在来自Switch 1的BPDUs信息仍然保存着的Switch 3上，接收到较差的BPDUs。

Switch 3将忽略这些较差BPDUs，直到最大存活值（the Max Age value）超时。在此期间，Switch 2继续将BPDUs发送给Switch 3。在最大存活时间超时后，Switch 3会将来自根桥、存储的BPDU信息老化排除，并转换到侦听状态，接着将把从根桥接收到的BPDU发送出去，发送给Switch 2。

因为此BPDU好于Switch 2自己的，Switch 2将停止发送BPDUs，同时Switch 2和Switch 3之间的端口经历侦听及学习状态的转换，并最终进入到转发状态。STP过程的此默认运行方式将意味着Switch 2将至少在50秒内无法转发数据帧。

骨干快速特性包含了一种允许在接收到一个较差的BPDU时，立即检查某个端口上存储的BPDU信息是否仍然有效的机制。此特性通过一种叫做RLQ PDU 的新协议数据单元及根链路请求实现的（this is implemented with a new PDU and the Root Link Query(RLQ)，which is referred to as the RLQ PDU）。

紧接着较差BPDU的接收，该交换机将在除接收该较差BPDU的端口外的所有非指定端口上，发出一个RLQ PDU。如该交换机是根桥或失去了到根桥的连接，就将对对该RLQ进行响应。否则，该RLQ将向上游传播（otherwise, the RLQ will be propagated upstream）。如该交换机在其根端口上接收到一个RLQ响应，那么到根桥的连通性仍然是完整的。如该响应实在非根端口上接收到的，就意味着到根桥的连通性已丢失，同时在交换机上的本地交换生成树必须重新计算且最大存活时间计数器被置为超时，如此就能重新找到一个新的根端口（if the response is received on a Non-Root Port, it means that connectivity to the Root Bridge is lost, and the local switch Spanning Tree must be recalculated on the switch and the Max Age timer expired so that a new Root Port can be found）。此概念在下图31.19中进行了演示。



图31.19 -- 掌握骨干快速 (续)

参考图31.19，紧接着较差BPDU的接收，Switch 3在除了该BPDU所接收到的端口之外的所有非指定端口上，发出一条RLQ请求。根桥通过一条从其指定端口发出的RLQ回应，对Switch 3的RLQ请求进行响应。因为是在Switch 3的根端口上接收到的该响应，该响应被认为是一条肯定响应（a positive response）。但如果该响应是在非根端口上接收到的，那么该响应就被认为是否定的且该交换机将需要再度完成整个的生成树计算。

基于Switch 3上接收到的肯定响应，就可以老化排除连接到Switch 2的端口而无需等待最大存活时间计数器过期（based on the positive response received on Switch 3, it can age out the port connected to Switch 2 without waiting for the Max Age timer to expire）。但是该端口仍必须经过侦听及学习状态。而通过立即将最大存活时间计数器进行老化清楚，骨干快速将收敛时间从50秒（20秒的最大存活时间 + 30秒的侦听和学习时间）减少到30秒（侦听和学习状态的时间）。

RLQs的类型有两种：RLQ请求和RLQ响应。**RLQ请求典型地在根端口上发出，用以检查到根桥的连通性。**  
**所有RLQ响应都是在指定端口上发出的。**因为RLQ请求包含了发送该RLQ响应的根桥BID，如到根桥路径中其它交换机仍能到达该RLQ响应中所指定的根桥，其就会响应给发出RLQ请求的交换机（because the RLQ request contains the BID of the Root Bridge that sent it, if another switch in the path to the Root Bridge can still reach the Root Bridge specified in the RLQ response, it will respond back to the sending switch）。如路径上的交换机已不能到达RLQ响应中的根桥，该交换机就简单地通过其根端口，往根桥转发该查询。

**注意：**RLQ PDU有着与普通BPDU同样的包格式，唯一区别在于RLQ PDU包含了两个用于请求和回应的思科SNAP(子网接入协议，[Subnetwork Access Protocol](#))地址。

## STP排错

### Troubleshooting STP

大多数二层故障都跟域中某种循环有关，而这又引起与其相关的多种问题，包括网络停机。在进行交换机配置的工作及将某台设备插入或拔出时，应确保没有在操作过程中建立循环。为缓和这类问题，就通常应在这些交换机上配置生成树协议，以避免出现在网络中的某处偶然创建出循环的情形（to mitigate against such problems, you should usually configure Spanning Tree Protocol on switches in order to avoid situations that might occur if you happen to accidentally create a loop somewhere in the network）。

网络中的所有交换机都是靠MAC地址进行通信的。在数据包进入时，就对MAC地址进行分析，从而基于二层头部中的目的MAC地址，确定出那个数据包的去向。网络中的所有设备都有着其自己的MAC地址，所以所有数据包在其走向上都是具体的。**不幸的是，像是广播及多播数据包前往交换机的所有端口。**如一个广播帧到达某个交换机端口，它将那个广播拷贝到可能连接到那台交换机的每台其它设备。此过程在网络中有着循环时，通常能是个问题。

应记住MAC地址数据包内部没有超时机制。在TCP/IP中（in the case of TCP/IP），IP协议在其头部有一个名为TTL（存活时间，Time to Live）的功能，该功能就是通过路由器的跳数，而不是事实上的时间单位。所以如果IP数据包碰巧处于循环中而通过多台路由器，它们将最终超时而被从网络中移除。但是，交换机并未提供那种机制。二层数据帧理论上可以永久循环，因为没有将其超时的机制，意味着如创建出一个循环，那个循环就会一直在那里，直到手动将其从网络中移除。

如正将一台工作站插入到网络时，某个广播帧到达该工作站，那么该广播数据帧将在那个点终结而不会是网络问题。但是，如在交换机侧端口进行了不当配置，或两端都插入了交换机而未开启STP，这将导致二层域内的广播风暴。广播风暴的发生，是因为广播数据包被转发到了所有其它端口，因此广播数据包保持继续存在并进入到同一网线上的另一交换机，引起二层循环。广播风暴能够引起高的资源使用甚至网络宕机。

如在这样的配置不当的网络上开启STP，交换机将识别到循环的出现，并会阻塞确定端口以避免广播风暴。而所有交换机中的其它端口则继续正常运作，所以网络不受影响。如未有配置STP，那么唯一可做的就是拔掉引起问题的网线，或者在还能对交换机进行操作的时候，将其管理性关闭。

STP故障通常有以下三类（STP issues usually fall within the following three categories）。

- 不正确的根桥, incorrect Root Bridge

- 不正确的根端口, incorrect Root Port
- 不正确的指定端口, incorrect Designated Port

## 不正确的根桥

优先级和基础MAC地址决定根桥是否是正确的 (priority and base MAC addresss decide whether the Root Bridge is incorrect)。可以执行 `show spanning-tree vlan <vlan#>` 命令查看MAC地址及交换机优先级。而运用 `spanning-tree vlan <vlan#> priority <priority>` 命令修复此问题。

## 不正确的根端口

根端口提供了自该交换机到根桥最快的路径，同时开销是跨越整个路径的累积 (the Root Port provides the fastest path from the switch to the Root Bridge, and the cost is cumulative across the entire path)。如怀疑存在正确的根端口，就可执行 `show spanning-tree vlan <vlan#>` 命令。如根端口是不正确的，可执行 `spanning-tree cost <cost>` 命令对其进行修复。

## 不正确的指定端口

指定端口是将某个网络区段连接到网络其它部分最低开销的端口 (the Designated Port is the lowest cost port connecting a network segment to the rest of the network)。如怀疑存在指定端口问题，就可以执行 `show spanning-tree vlan <vlan#>` 及 `spanning-tree cost <cost>` 命令。

而可对相关事件进行调试的一个有用的STP排错命令，就是 `Switch#debug spanning-tree events`。

# 第31天问题

1. How often do switches send Bridge Protocol Data Units ( BPDUs)?
2. Name the STP port states in the correct order.
3. What is the default Cisco Bridge ID?
4. Which command will show you the Root Bridge and priority for a VLAN?
5. What is the STP port cost for a 100Mbps link?
6. When a port that is configured with the \_\_\_\_\_ feature receives a BPDU, it immediately transitions to the errdisable state.
7. The \_\_\_\_\_ feature effectively disables STP on the selected ports by preventing them from sending or receiving any BPDUs.
8. Which two commands will force the switch to become the Root Bridge for a VLAN?
9. Contrary to popular belief, the Port Fast feature does not disable Spanning Tree on the selected port. This is because even with the Port Fast feature, the port can still send and receive BPDUs. True or false?
10. The Backbone Fast feature provides fast failover when a direct link failure occurs. True or false?

# 第31天答案

1. Every two seconds.
2. Blocking, Listening, Learning, Forwarding, and Disabled.
3. 32768.
4. The `show spanning-tree vlan x` command.
5. 19.

6. BPDU Guard.
7. BPDU Filter.
8. The `spanning-tree vlan [number] priority [number]` and `spanning-tree vlan [number] root [primary|secondary]` commands.
9. True.
10. False.

## 第31天实验

### 生成树根选举实验

#### 实验拓扑



#### 实验目的

学习如何对哪台交换机成为生成树根桥施加影响。

#### 实验步骤

1. 设置各台交换机的主机名并将其用交叉线连接起来。此时可以检查它们之间的接口是否被设置到“trunk”中继。

```
Switch#show interface trunk
```

1. 在将一侧设置为中继链路之前，可能看不到中继链路变成活动的。

```
SwitchB#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SwitchB(config)#int FastEthernet0/1
SwitchB(config-if)#switchport mode trunk
SwitchB(config-if)#^Z
SwitchB#sh int trunk
Port      Mode          Encapsulation      Status        Native vlan
Fa0/1    on           802.1q            trunking     1
Port      Vlans allowed on trunk
Fa0/1    1-1005
Port      Vlans allowed and active in management domain
Fa0/1    1
```

1. 将看到另一交换机是留作自动模式的。

```
SwitchA#show int trunk
Port      Mode          Encapsulation      Status        Native vlan
Fa0/1    auto         n-802.1q          trunking     1
Port      Vlans allowed on trunk
Fa0/1    1-1005
Port      Vlans allowed and active in management domain
Fa0/1    1
```

1. 在每台交换机上创建出两个VLANs。

```

SwitchA#conf t
Enter configuration commands, one per line. End with CNTL/Z.
SwitchA(config)#vlan 2
SwitchA(config-vlan)#vlan 3
SwitchA(config-vlan)#^Z
SwitchA#
%SYS-5-CONFIG_I: Configured from console by console
SwitchA#show vlan brief
VLAN Name          Status      Ports
-----  -----
1    default        active     Fa0/2, Fa0/3, Fa0/4,
                           Fa0/5, Fa0/6, Fa0/7,
                           Fa0/8, Fa0/9, Fa0/10,
                           Fa0/11, Fa0/12, Fa0/13,
                           Fa0/14, Fa0/15, Fa0/16,
                           Fa0/17, Fa0/18, Fa0/19,
                           Fa0/20, Fa0/21, Fa0/22,
                           Fa0/23, Fa0/24
2    VLAN0002       active
3    VLAN0003       active
1002 fddi-default   active
1003 token-ring-default   active

```

同时也在交换机B上创建出VLANs（拷贝上面的命令）。

1. 确定哪台交换机是VLANs 2和3的根桥。

```

SwitchB#show spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID    Priority    32770
              Address  0001.972A.7A23
              This bridge is the root
              Hello Time 2 sec
              Max Age   20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32770 (priority 32768 sys-id-ext 2)
              Address  0001.972A.7A23
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 20
  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/1        Desg FWD 19        128.1    P2p

```

可以看到，Switch B是根。在交换机A上完成同样的命令，并对VLAN 3进行检查。优先级是32768加上VLAN编号，这里就是2.最低MAC地址将确定出根桥。

```

SwitchB#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID    Priority    32771
              Address  0001.972A.7A23
              This bridge is the root
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
  Bridge ID  Priority    32771 (priority 32768 sys-id-ext 3)
              Address  0001.972A.7A23
              Hello Time 2 sec  Max Age 20 sec  Forward Delay 15 sec
              Aging Time 20
  Interface      Role Sts Cost      Prio.Nbr Type
  -----  -----
  Fa0/1        Desg FWD 19        128.1    P2p

```

这里Switch A的MAC地址较高，这就是为何其不会成为根桥的原因： 0010.1123.D245

- 将另一个交换机设置为VLANs 2和3的根桥。对VLAN 2使用命令 `spanning-tree vlan 2 priority 4096`，以及对VLAN 3的 `spanning-tree vlan 3 root primary` 命令。

```

SwitchA(config)#spanning-tree vlan 2 priority 4096
SwitchA(config)#spanning-tree vlan 3 root primary
SwitchA#show spanning-tree vlan 2
VLAN0002
  Spanning tree enabled protocol ieee
  Root ID      Priority    4098
                Address      0010.1123.D245
                This bridge is the root
                Hello Time    2 sec    Max Age 20 sec  Forward Delay 15 sec
  Bridge ID    Priority    4098 (priority 4096 sys-id-ext 2)
                Address      0010.1123.D245
                Hello Time    2 sec    Max Age 20 sec  Forward Delay 15 sec
                Aging Time   20
  Interface     Role Sts Cost      Prio.Nbr Type
  -----        --- --  ---      -----
  Fa0/1         Desg FWD   19       128.1    P2p
SwitchA#show spanning-tree vlan 3
VLAN0003
  Spanning tree enabled protocol ieee
  Root ID      Priority    24579
                Address      0010.1123.D245
                This bridge is the root
                Hello Time    2 sec    Max Age 20 sec  Forward Delay 15 sec
  Bridge ID    Priority    24579 (priority 24576 sys-id-ext 3)
                Address      0010.1123.D245
                Hello Time    2 sec    Max Age 20 sec  Forward Delay 15 sec
                Aging Time   20
  Interface     Role Sts Cost      Prio.Nbr Type
  -----        --- --  ---      -----
  Fa0/1         Desg FWD   19       128.1    P2p
SwitchA#

```

注意：尽管Switch B有较低的桥ID， Switch A还是被强制作为根桥。

# 第32天 快速生成树协议

## Rapid Spanning Tree Protocol

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

# 第32天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2记诵指南
- 在网站 [subnetting.org/](http://subnetting.org/) 花15分钟

IEEE 802.1D标准是在连通性从失去到恢复需要一分钟左右，就被认为性能已经可观的时期设计出来的。在 IEEE 802.1D STP下，恢复大约需要50秒，这其中包括20秒的最大老化计时器（the Max Age timer）超时，以及额外的给端口从阻塞状态过渡到转发状态的30秒。

随着计算机技术的进化，网络变得更为重要，更为快速的网络收敛显然是人们所需要的。思科通过开发一些包括骨干快速（Backbone Fast）及上行快速（Uplink Fast）等专有的STP增强，来满足此需求。

今天你将学到以下知识。

- RSTP的需求, the need for RSTP
- 配置RSTP, RSTP configuration

本课对应了以下CCNA大纲要求。

- 认识增强的交换技术, identify enhanced switching technologies
  - RSTP
  - PVSTP

## RSTP的需求

### the Need for RSTP

随着技术的持续演化，以及在同一物理平台上路由及交换的融合，在诸如OSPF及EIGRP这样的可以在更短时间内提供出替代路径的路由协议面前，交换网络的延迟就变得明显起来。802.1W标准就被设计出来解决此问题。

IEEE 802.1W标准，或者是快速生成树协议（Rapid Spanning Tree Protocol, RSTP），显著地缩短了在某条链路失效时，STP用于收敛的时间。在RSTP下，网络从故障切换到一条替代路径或链路可在亚秒级别完成（with RSTP, network failover to an alternate path or link can occur in a subsecond timeframe）。

RSTP是802.1D的一个扩展，执行与上行快速及骨干快速类似的功能。**RSTP比传统的STP执行得更好，且无需额外配置。此外，RSTP向后兼容最初的IEEE 802.1D STP标准。**其通过使用一种如下面的截屏中所示的修改的BPDU，实现的向后兼容。

```

Time: 152.515423 Source: Cisco_06:41:01 Destination: PVST+
Source: Cisco_f5:5b:01 Destination: PVST+
Source: Cisco_f5:5b:01 Destination: Spanning+Tree

Frame 15 (64 bytes on wire, 64 bytes captured)
- IEEE 802.3 Ethernet
- Logical-Link Control
- Spanning Tree Protocol
  Protocol Identifier: spanning Tree Protocol (0x000)
  Protocol Version Identifier: Rapid Spanning Tree (0x01)
  BPDU Type: Rapid/Multiple Spanning Tree (0x02)
  BPDU flags: 0x0e (Port Role: designated, Proposal)
  Root Identifier: 36768 / 00:0d:bd:06:41:00
  Root Path Cost: 0
  Bridge Identifier: 36768 / 00:0d:bd:06:41:00
  Port identifier: 0x8001
  Message Age: 0
  Max Age: 20
  Hello Time: 2
  Forward Delay: 15
  Version 1 Length: 0

```

图 32.1 -- 修改的BPDU

RSTP的各种端口状态可如下这样与STP端口状态对应起来。

- 关闭 -- 丢弃，Disabled -- Discarding
- 阻塞 -- 丢弃，Blocking -- Discarding
- 倾听 -- 丢弃，Listening -- Discarding
- 学习 -- 学习，Learning -- Learning
- 转发 -- 转发，Forwarding -- Forwarding

RSTP包含了以下的端口角色。

- 根端口（转发状态），Root(Forwarding state)
- 指定端口（转发状态），Designated(Forwarding state)
- 可变端口（阻塞状态），Alternate(Blocking state)
- 备份端口（阻塞状态），Backup(Blocking state)

对于考试，掌握上面这些着重号标记的内容是非常重要的，尤其是哪些端口状态转发流量（一旦网络完成收敛）。图32.2及32.3分别演示了一个RSTP可变端口及一个RSTP备份端口。

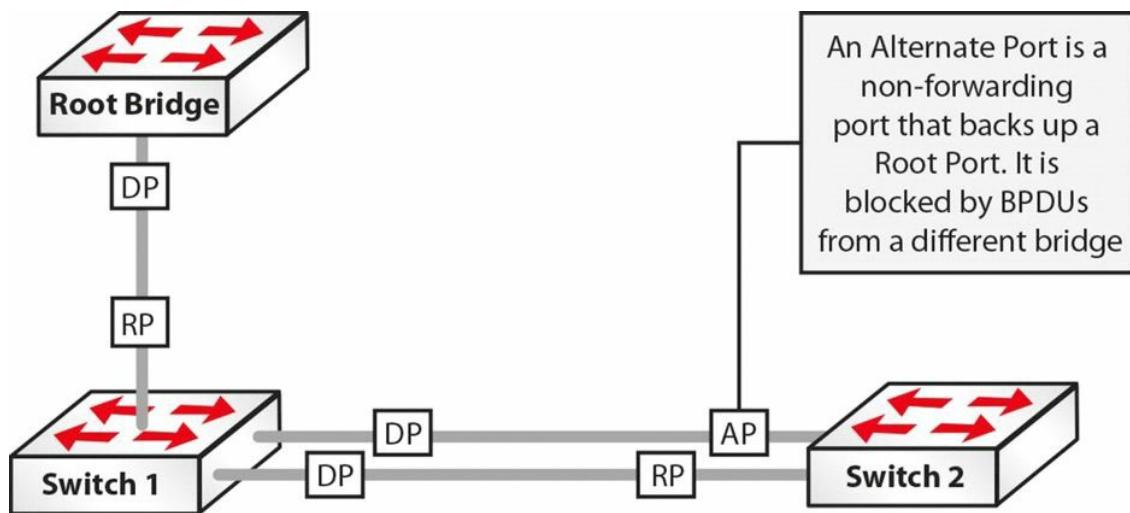


图 32.2 -- RSTP 可变端口

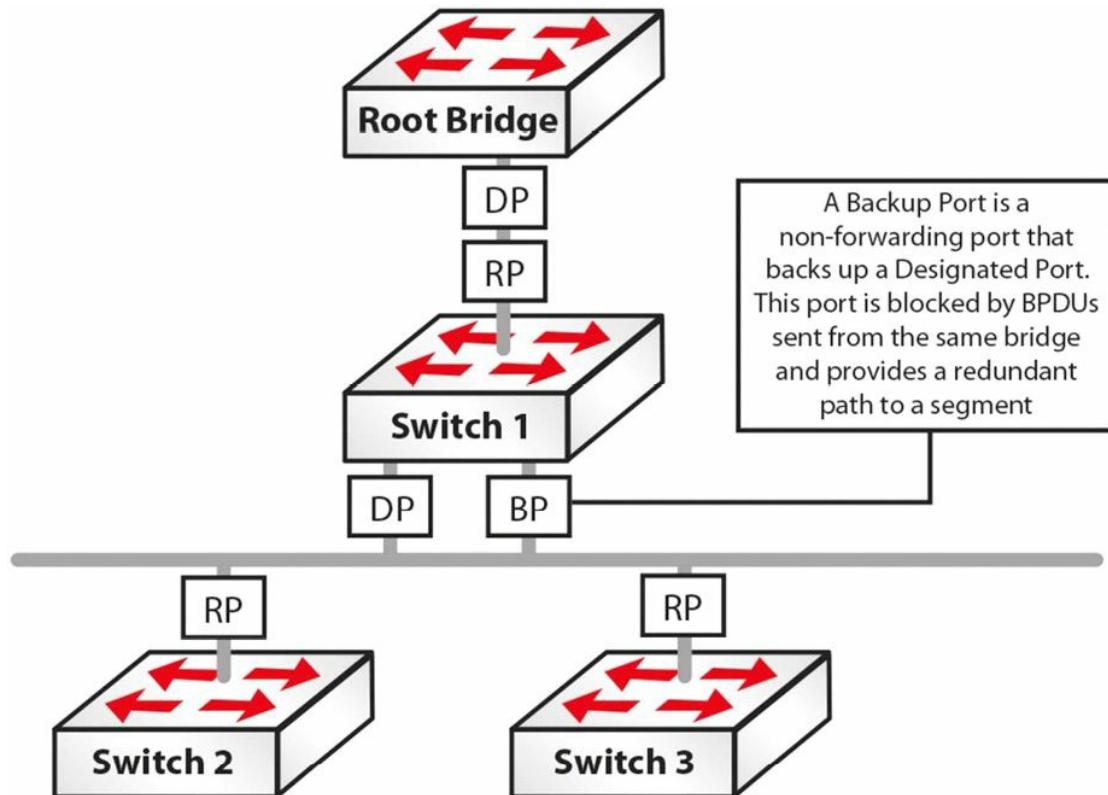


图 32.3 -- RSTP 备份端口

## 带有PVST+的RSTP

### RSTP with PVST+

加强版的基于各VLAN的生成树允许每个VLAN都有一个单独的STP实例（Per VLAN Spanning Tree Plus(PVST+) allows for an individual STP instance per VLAN）。传统或普通的PVST+模式在出现某条链路失效时，在网络收敛中，依赖较旧的802.1D STP的使用。

### RPVST+

快速的基于各VLAN的生成树加强版，允许与PVST+一起使用802.1W（Rapid Per VLAN Spanning Tree Plus(RPVST+) allows for the use of 802.1W with PVST+）。这就允许在每个VLAN都有一个单独的RSTP实例的同时，提供比起802.1D STP所能提供的更为快速的收敛。默认情况下，在某台思科交换机上开启RSTP时，也就在该交换机上开启了R-PVST+。

这里有一些可用来记住IEEE STP规格字母命名的记忆窍门。

- 802.1D (“经典的”生成树) -- It's dog-gone slow
- 802.1W(快速生成树) -- Imagine Elmer Fudd saying "rapid" as "wapid"
- 802.1S (多生成树) -- You add the letter "s" to nouns to make them plural(multiple) but this is a CCNP SWITCH subject

## RSTP的配置

### Configuring RSTP

RSTP的配置只需一个命令！

```
Switch(config)#spanning-tree mode rapid-pvst
Switch#show spanning-tree summary
Switch is in rapid-pvst mode
Root bridge for: VLAN0050, VLAN0060, VLAN0070
```

## 第32天问题

### Day 32 Questions

1. RSTP is not backward compatible with the original IEEE 802.1D STP standard. True or false?
2. What are the RSTP port states?
3. What are the four RSTP port roles?
4. Which command enables RSTP?
5. By default, when RSTP is enabled on a Cisco switch, R-PVST+ is enabled on the switch. True or false?

## 第32天问题答案

### Day 32 Answers

1. False.
2. Discarding, Learning, and Forwarding.
3. Root, Designated, Alernate, and Backup.
4. The `spanning-tree mode rapid-pvst` command.
5. True.

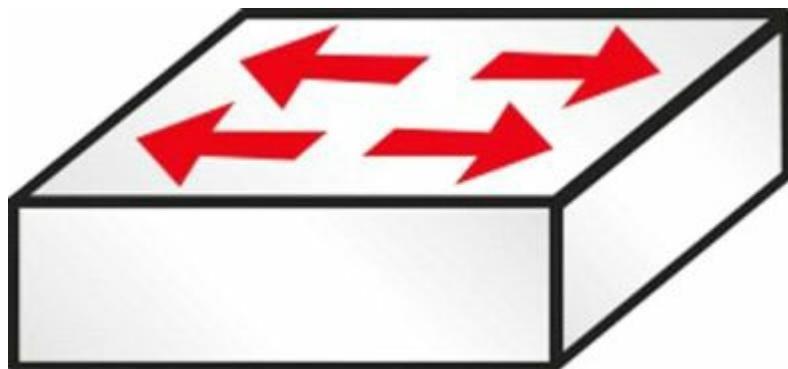
## 第32天实验

### Day 32 Lab

## RSTP实验

## RSTP Lab

### 拓扑图



### 实验目的

学习RSTP的配置命令。

### 实验步骤

1. 检查交换机上的生成树模式。

```
SwitchA#show spanning-tree summary  
Switch is in pvst mode  
Root bridge for: VLAN0002 VLAN0003
```

1. 将模式改为RSTP并再度检查。

```
SwitchA(config)#spanning-tree mode rapid-pvst  
SwitchA#show spanning-tree summary  
Switch is in rapid-pvst mode  
Root bridge for: VLAN0002 VLAN0003
```

1. 用RSTP模式来重复第31天的实验。
2. 你可以预先预测出那些端口将是根/指定/阻塞端口吗 (can you predict which ports will be Root/Designated/Blocking beforehand) ?

# 第33天 以太网通道及链路聚合协议

## EtherChannels and Link Aggregation Protocols

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第33天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2记诵指南
- 在网站[subnetting.org](http://subnetting.org)上花15分钟

思科IOS软件允许管理员将交换机上的多条物理链路（multiple physical links），结合成为一条单一的逻辑链路。这样做提供了一种负载分配以及链路冗余的理想方案，且可同时为二层及三层子系统所使用  
(provides an ideal solution for load sharing, as well as link redundancy, and can be used by both Layer 2 and Layer 3 subsystems)。

今天将学习以下内容。

- 掌握各种以太网通道, Understanding EtherChannels
- 端口聚合协议概述, Port Aggregation Protocol(PAgP) overview
- PAgP的端口模式, PAgP port modes
- PAgP 以太网通道协议的数据包转发, PAgP EtherChannel Protocol packet forwarding
- 链路聚合控制协议概述, Link Aggregation Control Protocol(LACP) overview
- 各种LACP端口模式, LACP port modes
- 不同以太网通道负载分配方法, EtherChannel load-distribution methods
- 不同二层以太网通道的配置和验证, Configuring and verifying Layer 2 EtherChannels

本课对应了以下ICND2大纲要求。

- 不同以太网通道技术, EtherChannels

## 掌握各种以太网通道

### Understanding EtherChannels

以太网通道是由一些物理的、单独的FastEthernet、GigabitEthernet或Ten-GigabitEthernet(10Gbps)链路绑定在一起，所构成的一条单一逻辑链路（links that are bundled together into a single logical link），如下面的图33.1所示。由FastEthernet链路所构成的以太网通道叫做FastEtherChannel(FEC)；由GigabitEthernet链路所构成的通道被称为GigabitEtherChannel(GEC)；最后，由Ten-GigabitEthernet链路所构成的以太网通道则被称为是Ten-GigabitEtherChannel(10GEC)。

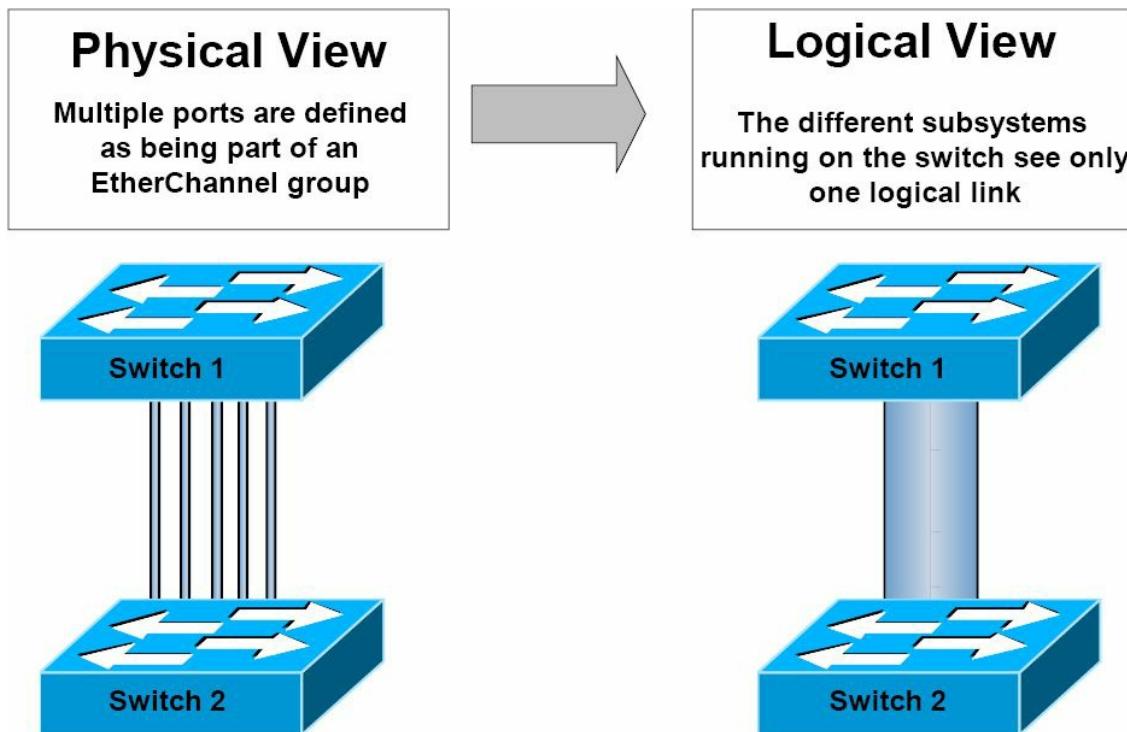


图33.1 -- 以太网通道的物理和逻辑视图

每个以太网通道最多可由8个端口构成。以太网通道中的物理链路必须有着相似特性(physical links in an EtherChannel must share similar characteristics)，诸如是定义在同一个VLAN中、或有着同样的速率以及双工设置。当在思科Catalyst交换机上配置以太网通道时，重要的是记住在不同Catalyst交换机型号之间，所支持的以太网通道数目会有所不同。

比如在Catalyst 3750系列交换机上，支持的数目是1到48个；在Catalyst 4500系列交换机上，是1到64个；而在旗舰的Catalyst 6500系列交换机，有效的以太网通道配置数目则是依据软件版本 (the software release)。对早于12.1(3a) E3的版本，有效数值是1到256；对于12.1(3a) E3、12.1(3a) E4以及12.1(4)E1，有效数值是1到64。而对于12.1(5c)EX及以后的版本，支持最大64的数量，范围从1到256。

**注意：** 并不要求知道不同IOS版本中所支持的以太网通道数量。

用于自动创建一个以太网通道组（an EtherChannel group）的链路聚合协议有两个：端口聚合协议（Port Aggregation Protocol, PAgP）及链路聚合控制协议(Link Aggregation Control Protocol, LACP)。PAgP是一个思科专有协议，同时LACP则是IEEE 802.3ad用于从几条物理链路建立逻辑链路规格的一部分。本模块中将详细对这两个协议进行讲述。

## 端口聚合协议概述

### Port Aggregation Protocol Overview

端口聚合协议 (Port Aggregation Protocol, PAgP) 是一个实现以太网通道自动建立的思科专有链路聚合协议 (a Cisco proprietary link aggregation protocol that enables the automatic creation of EtherChannels)。默认下, PAgP数据包在可作为以太网通道的端口之间发送 (PAgP packets are sent between EtherChannel-capable ports), 就以太网通道的形成进行协商。这些数据包被发送到目的多播 MAC地址 01-00-0c-cc-cc-cc (the destination Multicast MAC address 01-00-0c-cc-cc-cc), 而该多播MAC地址也是CDP、UDLD、VTP以及DTP所用到同一多播地址。下图33.2显示了在线路上所见到的一个PAgP数据帧中所包含的字段。

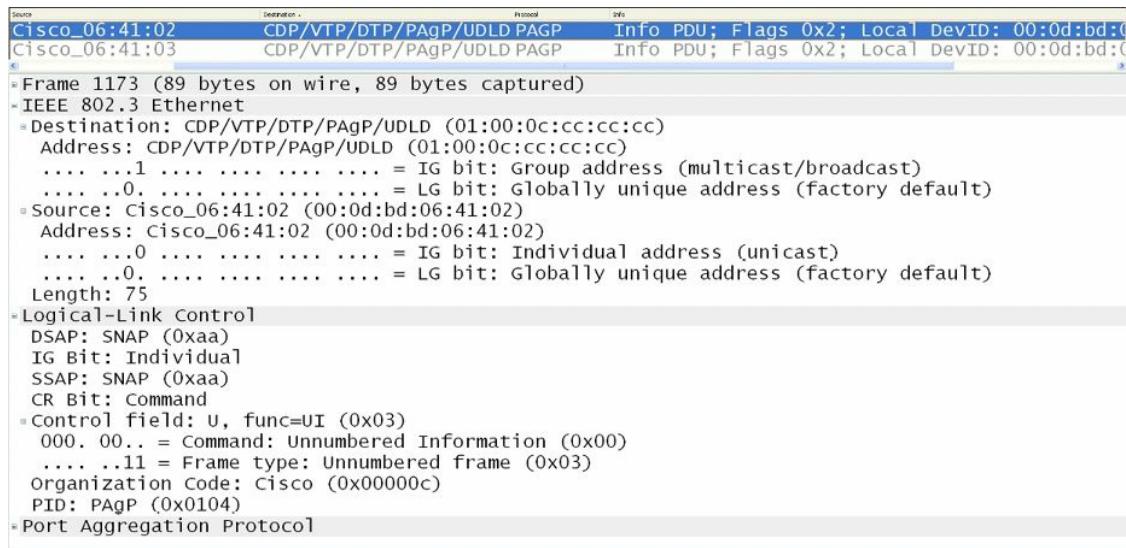


图 33.2 -- PAgP 以太网头部

尽管对PAgP数据包格式的深入探讨超出了CCNA考试要求范围, 下图33.3还是对一个典型的PAgP数据包所包含的字段进行了展示。PAgP数据所包含的一些字段与CCNA考试有关, 在本模块的跟进中将详细说明这些字段。

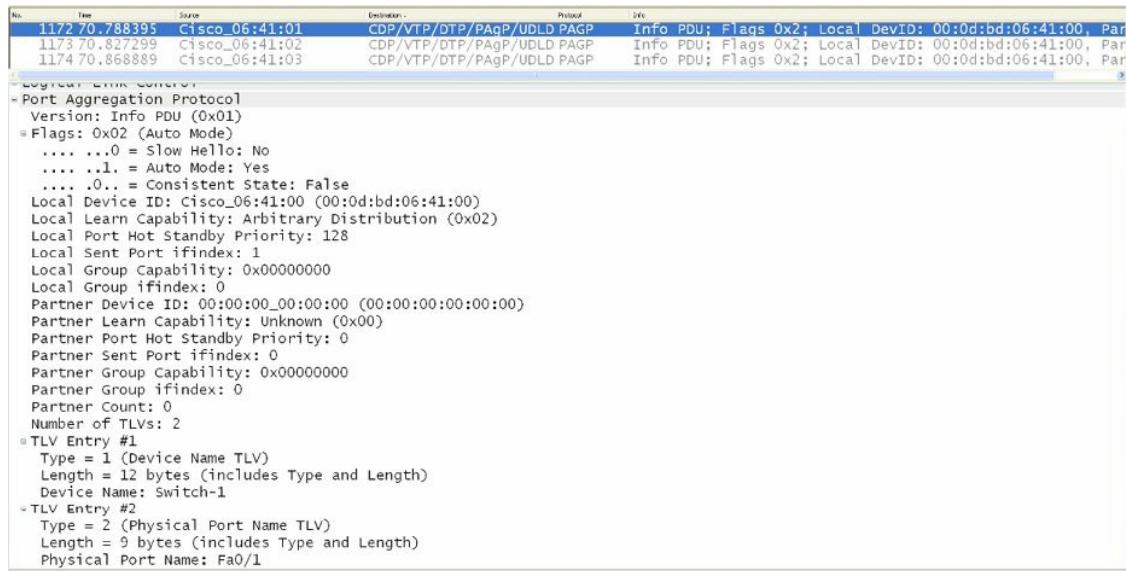


图 33.3 -- 端口聚合协议数据帧

## 各种PAgP端口模式

### PAgP Port Modes

PAgP支持不同端口模式，而这些端口模式则决定在两台支持PAgP的交换机(two PAgP-capable switches)之间将是否形成一个以太网通道。在深入到这两种PAgP端口模式之前，一种特别的模式需要专门关注。该模式（就是“on”模式）有时被误当作一种PAgP模式。事实上，其并不是一种PAgP的端口模式。

该 **on** 模式强制将某个端口无条件地置于某个通道当中。该通道将只在另一个交换机端口连接上、且被配置为 **on** 模式时建立起来。在此模式开启后，就不会有该通道的协商被本地以太网通道协议所执行。也就是说，这样做将切实关闭以太网通道协商并强制该端口到该通道 (when this mode is enabled, there is no negotiation of the channel performed by the local EtherChannel protocol. In other words, this effectively disables EtherChannel negotiation and forces the port to the channel)。该模式的运作与中继链路上的 `switchport nonegotiate` 类似。而重要的是记住配置为 **on** 模式的交换机接口不会对PAgP数据包进行交换。

采用PAgP的交换机以太网通道可被配置为以这两种模式运行：**自动** (`auto`) 或**我要** (`desirable`)。这两种PAgP模式的运作，在下面的小节进行说明。

## 自动模式

### Auto Mode

自动模式(`auto mode`)是一种仅在该端口接收到一个PAgP数据包时，才与另一PAgP端口进行协商的PAgP端口模式。在此模式开启后，该（这些）端口绝不会发起PAgP通信，而会在与邻居交换机建立一个以太网通道之前，被动地侦听任何接收到的PAgP数据包 (when this mode is enabled, the port(s) will never initiate PAgP communications but will instead listen passively for any received PAgP packets before creating an EtherChannel with the neighbouring switch)。

## 我要模式

### Desirable Mode

我要模式 (`desirable mode`) 是一种导致某端口发起与另一PAgP端口就通道建立而进行PAgP协商的PAgP端口模式 (desirable mode is a PAgP mode that causes the port to initiate PAgP negotiation for a channel with another PAgP port)。也就是说，在此模式下，该端口主动尝试与运行了PAgP的另一交换机建立一个以太网通道。

总的来说，要记住配置成 **on** 模式的交换机接口，不交换PAgP数据包，但它们会与那些配置为 **auto** 或 **desirable** 模式的伙伴接口进行PAgP数据包的交换 (but they do exchange PAgP packets with partner interfaces configured in the auto or desirable modes)。表33.1展示了不同的PAgP组合及其在建立一个以太网通道时所使用的结果。

表 33.1 -- 采用不同PAgP模式的以太网通道形成

交换机一PAgP模式	交换机二PAgP模式	以太网通道结果
Auto	Auto	不会形成以太网通道
Auto	Desirable	形成以太网通道
Desirable	Auto	形成以太网通道
Desirable	Desirable	形成以太网通道

## PAgP以太网通道协议数据包的转发

### PAgP EtherChannel Protocol Packet Forwarding

尽管PAgP允许以太网通道中的所有链路用于转发和接收用户流量，但应熟知一些关于在转发来自其它协议的流量时的限制。DTP及CDP透过以太网通道中的所有物理接口发送和接收（协议）数据包。而PAgP仅在那些起来（`up`）并开启了`auto`或`desirable`模式的接口上发送并接收PAgP协议数据单元（while PAgP allows for all links within the EtherChannel to be used to forward and receive user traffic, there are some restrictions that you should be familiar with regarding the forwarding of traffic from other protocols. DTP and CDP send and receive packets over all the physical interfaces in the EtherChannel. PAgP sends and receives PAgP Protocol Data Units only from interfaces that are up and have PAgP enabled for auto or desirable modes）。

在以太网通道捆绑（an EtherChannel bundle）被配置成一个中继端口时，该中继就在编号最低的VLAN上发送和接收PAgP数据帧。生成树协议总是选择以太网通道捆绑中的第一个可运作端口（when an EtherChannel bundle is configured as a trunk port, the trunk sends and receives PAgP frames on the lowest numbered VLAN. Spanning Tree Protocol(STP) always chooses the first operational port in an EtherChannel bundle）。命令`show pagp [channel number] neighbor`同样可用于验证将会用于STP数据包发送和接收的端口，确定出以太网通道捆绑中STP将使用的端口，如下面的输出所示。

```
Switch-1#show pagp neighbor
Flags: S - Device is sending Slow hello.   C - Device is in Consistent state.
          A - Device is in Auto mode.           P - Device learns on physical port.

Channel group 1 neighbors
      Partner    Partner        Partner        Partner Group
Port     Name      Device ID     Port     Age   Flags Cap.
Fa0/1   Switch-2  0014.a9e5.d640 Fa0/1   2s    SC    10001
Fa0/2   Switch-2  0014.a9e5.d640 Fa0/2   1s    SC    10001
Fa0/3   Switch-2  0014.a9e5.d640 Fa0/3   15s   SC    10001
```

根据上面的输出，STP将在端口`FastEthernet0/1`上发出其协议数据包，因为该端口是第一个可运作接口。而如那个端口失效，STP将在`FastEthernet0/2`上发出其协议数据包。而由PAgP所使用的默认端口则可由`show EtherChannel summary`命令进行查看，如下面的输出所示。

```
Switch-1#show EtherChannel summary
Flags: D - down
I - stand-alone
H - Hot-standby (LACP only)
R - Layer3
u - unsuitable for bundling
U - in use
d - default port
P - in port-channel
s - suspended
S - Layer2
f - failed to allocate aggregator
Number of channel-groups in use: 1
Number of aggregators: 1
Group  Port-channel  Protocol    Ports
-----+-----+-----+
1      Po1(SU)       PAgP        Fa0/1(Pd)  Fa0/2(P)  Fa0/3(P)
```

当在以太网通道上配置诸如`Loop Guard`这样的附加STP特性时，非常重要的是记住就算该通道捆绑中的其它端口是可运作的，在`Loop Guard`阻塞以太网通道捆绑的第一个端口时，就不会有BPDUs通过该通道得以发送了。这是因为PAgP将强制令到作为以太网通道端口组中的所有端口在`Loop Guard`配置上一致（when configuring additional STP features such as Loop Guard on an EtherChannel, it is very important to

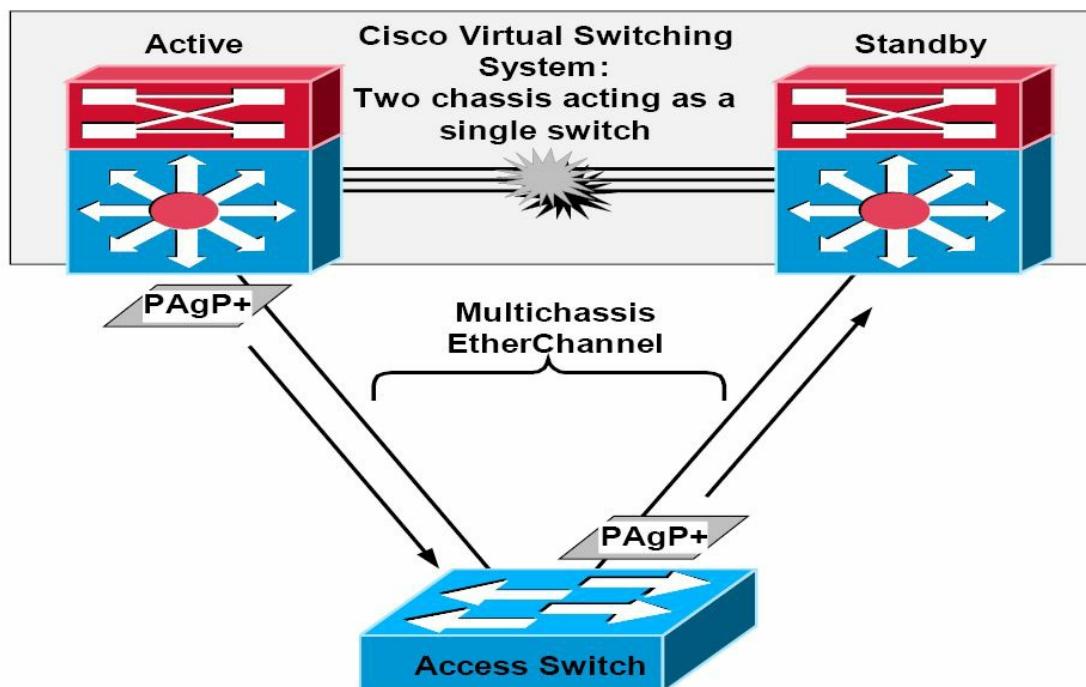
remember that if Loop Guard blocks the first port, no BPDUs will be sent over the channel, even if other ports in the channel bundle are operational. This is because PAgP will enforce uniform Loop Guard configuration on all of the ports that are part of the EtherChannel group)。

### 真实场景应用

#### Real-World Implementation

在生产网络中，可能会用到思科虚拟交换系统（the Cisco Virtual Switching System, VSS），该系统是由两台物理Catalyst 6500系列交换机所构成的一台单一逻辑交换机。在VSS中，一台交换机被选为活动交换机（the active switch），同时另一交换机就被选为了备用交换机（the standby switch）。这两台交换机就是通过以太网通道连接在一起，从而允许它们之间控制数据包的发送和接收。

接入交换机通过采用多机以太网通道（Multichassis EtherChannel, MEC）与VSS连接起来。而一个MEC就是一个对两台物理的Catalyst 6500交换机进行跨越而端接至一台逻辑虚拟交换机系统的以太网通道。增强的端口聚合协议（Enhanced PAgP, PAgP+）可用于允许Catalyst 6500交换机在其相互之间的以太网通道失效，导致两台交换机都假定其自身是活动角色（双活动），从而切实影响到交换网络中流量转发时，经由MEC进行通信（an MEC is simply an EtherChannel that spans the two physical Catalyst 6500 switches but terminates to the single logical VSS. Enhanced PAgP(PAgP+) can be used to allow the Catalyst 6500 switches to communicate via the MEC in the event that the EtherChannel between them fails, which would result in both switches assuming the active role(dual active), effectively affecting forwarding of traffic within the switched network）。这在下面的图表中进行了演示。



尽管VSS超出了CCNA考试要求范围，了解只有PAgP才能用于承载VSS控制数据包是有益处的。因此，如要在在一个VSS环境，或者要在一个最终会部署上VSS的环境中部署一些以太网通道，就会打算考虑运行PAgP而不是LACP，因为LACP是一个开放标准，不支持专有的VSS数据帧。本书中不会更为深入地涉及VSS。

## 链路聚合控制协议概述

#### Link Aggregation Control Protocol Overview

链路聚合控制协议（Link Aggregation Control Protocol, LACP）是IEEE 802.3ad规格的组成部分，用于从多条物理链路建立起一条逻辑链路。因为LACP与PAgP是不兼容的，所以链路的两端需要运行LACP以令到以太网通道组自动形成（Because LACP and PAgP are incompatible, both ends of the link need to run LACP in order to automate the formation of EtherChannel groups）。

与PAgP的情形一样，在配置LACP以太网通道时，所有LAN端口都必须是同样速率，且都必须被配置成二层或三层LAN端口。而当某端口通道中的一条链路失效时，那么先前由该链路所承载的流量就由该端口通道中剩下的链路进行交换。此外，在对某个端口通道中的活动绑定端口的编号进行修改后，流量模式将反应出该端口通道重新平衡之后的状态。

LACP通过在端口之间交换LACP数据包，实现对端口通道自动创建的支持。其对端口组别具备的各项能力进行动态学习，并通知给其它端口。而一旦LACP正确地识别出这些匹配的以太网链路，其就推进将这些链路编组为一个GigabitEthernet端口通道。与PAgP要求端口有着相同速率及双工设置不同，**LACP要求端口只能是全双工，因为半双工是不支持的**。某个LACP以太网通道中的那些半双工端口，被置为暂停状态（Half-duplex ports in an LACP EtherChannel are placed into the suspended state）。

默认情况下，一条链路上的所有入口广播及多播数据包在该端口通道的其它链路上的返回都被阻止（by default, all inbound Broadcast and Multicast packets on one link in a port channel are blocked from returning on any other link of the port channel）。LACP数据包被发送到IEEE 802.3慢速协议多播组地址（the IEEE 802.3 Slow Protocols Multicast group address）01-80-c2-00-00-02。LACP数据帧以EtherType数值0x8809进行编码。下图33.4演示了一个以太网数据帧中的这些字段。

Idx	Time	Source	Destination	Protocol	Info
1	0.000000	Cisco_06:41:01	Slow-Protocols	LACP	Link Aggregation Control
2	1.480355	Cisco_06:41:02	PVST+	STP	Conf. Root = 32768/00:0d:
3	1.480859	Cisco_06:41:02	Spanning-tree-(for-br	STP	Conf. Root = 32768/00:0d:
4	1.481722	Cisco_06:41:02	PVST+	STP	Conf. Root = 32768/00:0d:
5	1.482544	Cisco_06:41:02	PVST+	STP	Conf. Root = 32768/00:0d:

```

Frame 1 (124 bytes on wire, 124 bytes captured)
Ethernet II, Src: Cisco_06:41:01 (00:0d:bd:06:41:01), Dst: slow-Protocols (01:80:c2:00:00:02)
  Destination: Slow-Protocols (01:80:c2:00:00:02)
    Address: Slow-Protocols (01:80:c2:00:00:02)
      ....1 .... .... .... = IG bit: Group address (multicast/broadcast)
      ....0. .... .... .... = LG bit: Globally unique address (factory default)
  Source: Cisco_06:41:01 (00:0d:bd:06:41:01)
    Address: Cisco_06:41:01 (00:0d:bd:06:41:01)
      ....0 .... .... .... = IG bit: Individual address (unicast)
      ....0. .... .... .... = LG bit: Globally unique address (factory default)
  Type: Slow Protocols (0x8809)
  Link Aggregation Control Protocol

```

图 33.4 -- IEEE 802.3 LACP 数据帧

## LACP的端口模式

### LACP Port Modes

LACP通过在端口之间交换LACP数据包，实现对端口通道自动建立的支持。而LACP又是通过动态地掌握端口组的各项能力并将其通告给其它端口完成的端口间数据交换。一旦LACP正确地识别出那些匹配的以太网链路，就推进这些链路编组为一个端口通道。而一旦LACP模式得以配置，其仅会在某单个接口被分配到指定通道组时被改变。LACP支持两种模式，**主动**（active）及**被动**（passive）模式。后续小节将对这两种模式的运作进行说明。

### LACP主动模式

### LACP Active Mode

LACP主动模式将一个交换机端口置为经由发送LACP数据包，对远端端口发起协商的主动协商状态（an active negotiating state in which the switch port initiates negotiations with remote ports by sending LACP packets）。主动模式与PAgP的 desirable 模式等价。也就是说，在此模式下，交换机端口主动尝试与另一台同样运行LACP的交换机建立以太网通道。

### LACP被动模式

#### LACP Passive Mode

当交换机端口被配置为被动模式时，其只在接收到其它LACP数据包时，才就建立LACP通道进行协商。在被动模式下，该端口对其所接收到的LACP数据包进行响应，而并不发起LACP数据包协商。该设置减少了LACP数据包传输。在此模式下，该端口通道组将该接口附加到以太网通道捆绑。此模式与PAgP所用到的 auto 模式类似。

重要的是记住**主动和被动模式只在非PAgP接口上是有效的**（the active and passive modes are valid on non-PAgP interfaces only）。但是，如有着一个PAgP以太网通道，并打算将其转换到LACP，那么思科 IOS软件允许随时对协议进行改变。而其间唯一的限制，就是此改变导致全部现有以太网通道重置为新协议的默认通道模式。下表33.2展示了不同的LACP组合及它们在两台交换机之间建立一个以太网通道中应用的结果。

表 33.2 -- 使用不同LACP模式的以太网通道形成

Table 33.2 -- EtherChannel Formation Using Different LACP Modes

交换机一的LACP模式	交换机二的LACP模式	以太网通道结果
被动模式	被动模式	没有以太网通道形成
被动模式	主动模式	形成以太网通道
主动模式	主动模式	形成以太网通道
主动模式	被动模式	形成以太网通道

## 以太网通道的负载分配方式

#### EtherChannel Load-Distribution Methods

对于PAgP及LACP以太网通道，Catalyst交换机使用到一种利用数据包头部的一些关键字段，生成一个随后匹配到以太网通道组中的某条物理链路的散列值的多态算法。也就是说，交换机通过将由帧中的地址所形成的二进制模式，减少到从以太网通道中多条链路选出一条的一个数值，从而实现流量负载在这些链路上的分配（a polymorphic algorithm that utilises key fields from the header of the packet to generate a hash, which is then matched to a physical link in an EtherChannel group. In other words, the switch distributes the traffic load across the links in an EtherChannel by reducing part of the binary pattern formed from the addresses in the frame to a numerical value that selects one of the links in the EtherChannel）。

此操作可在MAC地址或IP地址上完成，并可仅基于源或目的地址，或同时基于源或目的地址。尽管对以太网通道负载分配中所用到的该散列值的实际计算的深入探讨，是超出CCNA考试要求范围的，但知道管理员可以指定头部中的哪些字段，作为确定某个数据包的传输物理链路所用到的算法的输入，是重要的（while delving into detail on the actual computation of the hash used in EtherChannel load distribution is beyond

the scope of the CCNA exam requirements, it is important to know that the administrator can define which fields in the header can be used as input to the algorithm used to determine the physical link transport to the packet)。

负载分配方式通过全局配置命令 `port-channel load-balance [method]` 进行配置。在任何时间，都只能使用一种单一方式。下表33.3列出并解释了在配置以太网通道负载分配时，思科IOS Catalyst交换机中可用的不同方式。

表 33.3 -- 以太网通道负载分配（负载均衡）的可选项

*Table 33.3 -- EtherChannel Load-Distribution(Load-Balancing) Options*

方式	说明
dst-ip	进行基于目的IP地址的负载分配, performs load distribution based on the destination IP address
dst-mac	进行基于目的MAC地址的负载分配, performs load distribution based on the destination MAC address
dst-port	进行基于目的第4层端口的负载分配, performs load distribution based on the destination Layer 4 port
src-dst-ip	进行基于源和目的IP地址的负载分配, performs load distribution based on the source and destination IP address
src-dst-port	进行基于源和目的第4层端口的负载分配, performs load distribution based on the source and destination Layer 4 port
src-ip	进行基于源IP地址的负载分配, performs load distribution based on the source IP address
src-mac	进行基于源MAC地址的负载分配, performs load distribution based on the source MAC address
src-port	进行基于源第4层端口的负载分配, performs load distribution based on the source Layer 4 port

## 以太网通道配置准则

### EtherChannel Configuration Guidelines

以下小节列出并说明了配置二层PAgP以太网通道所需要的步骤。但在深入到这些配置步骤之前，有必要熟悉下面这些配置二层以太网通道时的限制。

- 每个以太网通道可以有最多8个兼容配置的以太网接口。而LACP则允许一个以太网通道组中多于8个的端口。不过这些额外端口都是热备份（hot-standby）端口。
- 以太网通道中的所有接口都必须以相同的速率及双工模式运行。记住，与PAgP不同，LACP并不支持半双工端口。
- 确保以太网通道中的所有接口都是开启的。在某些情况下，如这些接口没有开启，那么该逻辑端口通道接口（the logical port channel interface）就不会被自动创建。
- 在初次配置一个以太网通道组时，重要的是记住这些端口与所加入的第一个组端口参数集一致（when first configuring an EtherChannel group, it is important to remember that ports follow the parameters set for the first group port added）。
- 如有为某个以太网通道中的某个成员端口配置交换机端口分析器（Switch Port Analyzer, SPAN），那么该端口将会从该以太网通道组中移除。

- 将以太网通道中的所有端口都指派到同一个VLAN，或将它们配置成中继端口，是必要的。而如果这些参数不同，该通道就不会形成。
- 记住有着不同STP路径开销（由某位管理员所修改的）的那些类似接口，仍可用于组成一个以太网通道。
- 在开始通道配置之前，建议首先关闭所有成员接口（it is recommended to shut down all member interfaces prior to beginning channelling configuration）。

## 配置并验证二层以太网通道

### Configuring and Verifying Layer 2 EtherChannels

该部分内容通过无条件地强制所选接口建立一个以太网通道，对二层以太网通道的配置进行了说明（this section describes the configuration of Layer 2 EtherChannels by unconditionally forcing the selected interfaces to establish an EtherChannel）。

- 第一个配置步骤是通过全局配置命令 `interface [name]` 或 `interface range [range]`，进入那些所需要的以太网通道接口的接口配置模式；
- 配置的第二步是通过接口配置命令 `switchport`，将这些接口配置为二层交换机接口；
- 第三个配置步骤是通过接口配置命令 `switchport mode [access|trunk]`，将这些交换机端口配置为中继或接入链路；
- 作为可选步骤，如该接口或这些接口已被配置为接入端口，就要使用命令 `switchport access vlan [number]`，将其指派到同样的VLAN中。而如该接口或这些接口已被配置为中继端口，就要通过执行接口配置命令 `switchport trunk allowed vlan [range]`，选择允许通过该中继的那些VLANs；而如VLAN 1将不作为原生VLAN（802.1Q的），就要通过执行接口配置命令 `switchport trunk native vlan [number]`，输入原生VLAN。此项配置在所有端口通道成员接口上必须一致。
- 下一配置步骤就是通过接口配置命令 `channel-group [number] mode on`，将这些接口配置为无条件中继（the next configuration step is to configure the interfaces to unconditionally trunk via the `channel-group [number] mode on` interface configuration command）。

用到上述步骤的无条件以太网通道配置，将基于下图33.5中所演示的网络拓扑。

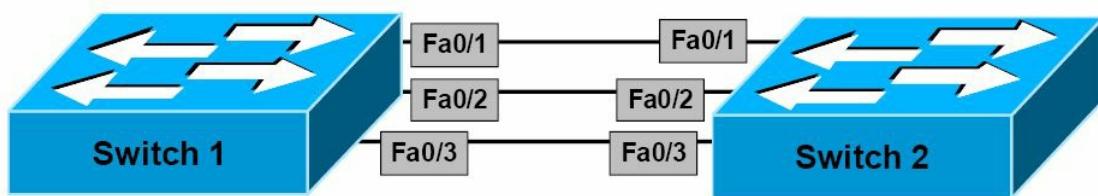


图 33.5 -- 以太网通道配置输出示例的网络拓扑

下面的输出演示了如何在Switch 1及Switch 2上，基于图33.5中所描述的网络拓扑，配置无条件通道操作。该以太网通道将配置成一个使用默认参数的二层802.1Q中继。

```

Switch-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-1(config)#interface range fa0/1 - 3
Switch-1(config-if-range)#no shutdown
Switch-1(config-if-range)#switchport
Switch-1(config-if-range)#switchport trunk encapsulation dot1q
Switch-1(config-if-range)#switchport mode trunk
Switch-1(config-if-range)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1
Switch-1(config-if-range)#exit
Switch-1(config)#exit

```

**注意：**注意到该交换机自动默认创建出 `interface port-channel 1` (根据下面的输出)。没有要配置该接口的显式用户配置 (notice that the switch automatically creates `interface port-channel 1` by default(refer to the output below). No explicit user configuration is required to configure this interface)。

```
Switch-2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-2(config)#interface range fa0/1 - 3
Switch-2(config-if-range)#switchport
Switch-2(config-if-range)#switchport trunk encapsulation dot1q
Switch-2(config-if-range)#switchport mode trunk
Switch-2(config-if-range)#channel-group 1 mode on
Creating a port-channel interface Port-channel 1
Switch-2(config-if-range)#exit
Switch-2(config)#exit
```

命令 `show EtherChannel [options]` 此时即可用于验证该以太网通道的配置。下面的输出中打印了可用选项 (依据不同平台会有不同)。

```
Switch-2#show EtherChannel ?
<1-6>          Channel group number
detail           Detail information
load-balance    Load-balance/frame-distribution scheme among ports in port-channel
port             Port information
port-channel    Port-channel information
protocol        protocol enabled
summary         One-line summary per channel-group
|
<cr>
```

下面的输出对命令 `show EtherChannel summary` 进行了演示。

```
Switch-2#show EtherChannel summary
Flags: D - down
      I - stand-alone
      H - Hot-standby (LACP only)
      R - Layer3
      u - unsuitable for bundling
      U - in use
      d - default port
      P - in port-channel
      S - suspended
      S - Layer2
      f - failed to allocate aggregator
Number of channel-groups in use: 1
Number of aggregators: 1
Group  Port-channel  Protocol    Ports
-----+-----+-----+
1      Po1(SU)       -          Fa0/1(Pd)   Fa0/2(P)   Fa0/3(P)
```

在上面的输出中，可以看到在通道组1 (Channel Group 1) 中有三条链路。接口FastEthernet0/1是默认端口；该端口将用于发送比如的STP数据包。如果该端口失效，FastEthernet0/2就将被指定为默认端口，如此延续 (this port will be used to send STP packets, for example. If this port fails, FastEthernet0/2 will be designated as the default port, and so forth)。同时通过看看 Po1 后面的 su 标志，还可以看到该端口组是一个活动的二层以太网通道。下面的输出实现了由 `show EtherChannel detail` 命令所打印出的信息。

```

Switch-2#show EtherChannel detail
      Channel-group listing:
      -----
      Group: 1
      -----
      Group state = L2
      Ports: 3    Maxports = 8
      Port-channels: 1 Max Port-channels = 1
      Protocol:   -
                  Ports in the group:
      -----
      Port: Fa0/1
      -----
      Port state      = Up Mstr In-Bndl
      Channel group  = 1           Mode = On/FEC          Gcchange = -
      Port-channel   = Po1         GC   = -             Pseudo port-channel = Pol
      Port index     = 0           Load  = 0x00          Protocol = -
      Age of the port in the current state: 0d:00h:20m:20s
      Port: Fa0/2
      -----
      Port state      = Up Mstr In-Bndl
      Channel group  = 1           Mode = On/FEC          Gcchange = -
      Port-channel   = Po1         GC   = -             Pseudo port-channel = Pol
      Port index     = 0           Load  = 0x00          Protocol = -
      Age of the port in the current state: 0d:00h:21m:20s
      Port: Fa0/3
      -----
      Port state      = Up Mstr In-Bndl
      Channel group  = 1           Mode = On/FEC          Gcchange = -
      Port-channel   = Po1         GC   = -             Pseudo port-channel = Pol
      Port index     = 0           Load  = 0x00          Protocol = -
      Age of the port in the current state: 0d:00h:21m:20s
      Port-channels in the group:
      -----
      Port-channel: Po1
      -----
      Age of the Port-channel      = 0d:00h:26m:23s
      Logical slot/port   = 1/0           Number of ports = 3
      GC                 = 0x00000000       HotStandBy port = null
      Port state        = Port-channel Ag-Inuse
      Protocol          = -
      Ports in the Port-channel:
      Index  Load  Port    EC state      No of bits
      -----+-----+-----+-----+
      0      00    Fa0/1  On/FEC       0
      0      00    Fa0/2  On/FEC       0
      0      00    Fa0/3  On/FEC       0
      Time since last port bundled: 0d:00h:21m:20s      Fa0/3
  
```

在上面的输出中，可以看出这是一个带有通道组中最多8个可能端口中的三个的二层以太网通道。还可以看出，以太网通道模式是 `on`，这是基于由一条短横线所表示的协议字段看出的。此外，同样可以看出这是一个FastEtherChannel(FEC) (in the output above, you can see that this is a Layer 2 EtherChannel with three out of a maximum of eight possible ports in the channel group. You can also see that the EtherChannel mode is on, based on the protocol being denoted by a hash(-). In addition, you can also see that this is a FastEtherChannel(FEC))。

最后，还可以通过执行命令 `show interface port-channel [number] switchport`，对该逻辑的port-channel接口的二层运行状态进行检查。这在下面的输出中进行了演示。在上面的输出中，可以看到这是一个带有通道组中最多8个中的3个端口的二层以太网通道。还可以从由短横所表示的协议，看出以太网通道模式是 `on`。此外，还可以看到这是一个FastEtherChannel(FEC)。

最后，还可通过执行命令 `show interfaces port-channel [number] switchport`，对该逻辑的端口通道接口（the logical port-channel interface）的二层运作状态进行查看。这在下面的输出中有所演示。

```
Switch-2#show interfaces port-channel 1 switchport
Name: Po1
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
Trunking Native Mode VLAN: 1 (default)
Voice VLAN: none
Administrative private-vlan host-association: none
Administrative private-vlan mapping: none
Administrative private-vlan trunk native VLAN: none
Administrative private-vlan trunk encapsulation: dot1q
Administrative private-vlan trunk normal VLANs: none
Administrative private-vlan trunk private VLANs: none
Operational private-vlan: none
Trunking VLANs Enabled: ALL
Pruning VLANs Enabled: 2-1001
Protected: false
Appliance trust: none
```

## 配置并验证PAgP以太网通道

### Configuring and Verifying PAgP EtherChannels

此部分对PAgP二层以太网通道的配置进行了说明。为配置并建立一个PAgP以太网通道，需要执行以下步骤。

1. 第一个配置步骤是通过全局配置命令 `interface [name]` 或 `interface range [range]`，进入到所需的这些以太网接口的接口配置模式；
2. 配置的第二步，是通过接口配置命令 `switchport`，将这些接口配置为二层交换端口；
3. 第三个配置步骤，是通过接口配置命令 `switchport mode [access|trunk]`，将这些交换端口，配置为中继或接入链路；
4. 作为可选步骤，如果已将这些端口配置为接入端口，那么就要使用命令 `switchport access vlan [number]`，将其指派到同一个VLAN中；而如果这些接口已被配置为中继端口，那么就要通过执行接口配置命令 `switchport trunk allowed vlan [range]`，来选择所允许通过该中继的那些VLANs；如未打算将VLAN 1用作原生VLAN（对于802.1Q），就要通过执行接口配置命令 `switchport trunk native vlan [number]`，输入原生VLAN。此项配置在所有端口通道的成员接口上一致。
5. 作为可选项，通过执行接口配置命令 `channel-protocol pagg`，将PAgP配置作为以太网通道协议（the EtherChannel protocol）。因为以太网通道默认是PAgP的，所以此命令被认为是可选的而无需输入。但执行该命令被看作是良好实践，因为可以令到配置绝对确定（it is considered good practice to issue this command just to be absolutely sure of your configuration）。
6. 下一步就是通过接口配置命令 `channel-group [number] mode`，将这些接口配置为无条件中继。

下面的输出演示了如何在基于上面的图33.5中所给出的网络拓扑的Switch 1和Switch 2上，配置PAgP的通道（PAgP channelling）。该以太网通道将被配置为使用默认参数的二层802.1Q中继。

```

Switch-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-1(config)#interface range fa0/1 - 3
Switch-1(config-if-range)#switchport
Switch-1(config-if-range)#switchport trunk encapsulation dot1q
Switch-1(config-if-range)#switchport mode trunk
Switch-1(config-if-range)#channel-group 1 mode desirable
Creating a port-channel interface Port-channel 1
Switch-1(config-if-range)#exit

```

**注意：**在上面的输出中，选择了端口通道的 `desirable` 模式。可以在此命令（`channel-group 1 mode desirable`）之后加上一个额外关键字 `[non-silent]`。这是因为，默认情况下，PAgP 的 `auto` 模式默认是安静模式。当交换机被连接到一台不兼容PAgP的设备时，就用到安静模式，且绝不会传送数据包(an additional keyword, `[non-silent]`, may also be appended to the end of this command. This is because, by default, PAgP auto and desirable modes default to a silent mode. The silent mode is used when the switch is connected to a device that is not PAgP-capable and that seldom, if ever transmits packets)。一台安静相邻设备的例子 (an example of a silent partner)，就是一台文件服务器或未有生成流量的数据包分析器。而如果一台设备不会发出PAgP数据包（比如处于 `auto` 模式），也用到安静模式。

在此示例中，在一个连接到一台安静相邻设备的物理端口上运行PAgP阻止了那个交换机端口成为运作端口；但是，该安静设置允许PAgP运行，从而将该接口加入到一个通道组，同时利用该接口进行传输。在本例中，因为Switch 2将被配置为 `auto` 模式（被动模式），该端口采用默认的安静模式运作，就是首先的了 (In this case, running PAgP on a physical port connected to a silent partner prevents that switch port from ever becoming operational; however, the silent setting allows PAgP to operate, to attach the interface to a channel group, and to use the interface for transmission. In this example, because Switch 2 will be configured for auto mode(passive mode), it is preferred that the port uses the default silent mode operation)。这在下面的PAgP以太网通道配置中进行了演示。

```

Switch-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-1(config)#interface range fa0/1 - 3
Switch-1(config-if-range)#switchport
Switch-1(config-if-range)#switchport trunk encapsulation dot1q
Switch-1(config-if-range)#switchport mode trunk
Switch-1(config-if-range)#channel-group 1 mode desirable ?
    non-silent Start negotiation only after data packets received
<cr>
Switch-1(config-if-range)#channel-group 1 mode desirable non-silent
Creating a port-channel interface Port-channel 1
Switch-1(config-if-range)#exit

```

继续进行PAgP以太网通道的配置，则Switch 2被配置为以下这样。

```

Switch-2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-2(config)#int range fa0/1 - 3
Switch-2(config-if-range)#switchport
Switch-2(config-if-range)#switchport trunk encapsulation dot1q
Switch-2(config-if-range)#switchport mode trunk
Switch-2(config-if-range)#channel-group 1 mode auto
Creating a port-channel interface Port-channel 1
Switch-2(config-if-range)#exit

```

以下输出演示了怎样通过在Switch 1及Switch 2上使用命令 `show EtherChannel summary`，验证该PAgP以太网通道的配置。

```
Switch-1#show EtherChannel summary
Flags: D - down
      I - stand-alone
      H - Hot-standby (LACP only)
      R - Layer3
      u - unsuitable for bundling
      U - in use
      d - default port
      P - in port-channel
      s - suspended
      S - Layer2f - failed to allocate aggregator
Number of channel-groups in use:    1
Number of aggregators:             1
Group  Port-channel  Protocol     Ports
-----+-----+-----+
1      Po1(SU)       PAgP        Fa0/1(Pd)  Fa0/2(P)  Fa0/3(P)
```

还可以通过执行命令 `show pagg [options]`，查看到PAgP以太网通道的配置及统计数据。下面的输出，演示了此命令下可用的选项。

```
Switch-1#show pagg ?
<1-6>   Channel group number
counters  Traffic information
internal  Internal information
neighbor  Neighbor information
```

**注意：** 对需要的端口通道编号的进入，提供上面所打印出的后三个选项。这在下面的输出中进行了演示。

```
Switch-1#show pagg 1 ?
counters  Traffic information
internal  Internal information
neighbor  Neighbor information
```

关键字 [counters] 提供了有关PAgP发出及接收到的数据包的信息。关键字 [internal] 提供了诸如端口状态、Hello间隔时间、PAgP端口优先级以及端口学习方式等的信息。下面的输出对命令 `show pagg internal` 的使用进行了演示。

```
Switch-1#show pagg 1 internal
Flags: S - Device is sending Slow hello.  C - Device is in Consistent state.
      A - Device is in Auto mode.          d - PAgP is down.
Timers: H - Hello timer is running.      Q - Quit timer is running.
      S - Switching timer is running.    I - Interface timer is running.
Channel group 1
                                         Hello          Partner PAgP      Learning  Group
Port   Flags   State   Timers Interval   Count  Priority Method   Ifindex
Fa0/1  SC      U6/S7  H       30s       1      128    Any      29
Fa0/2  SC      U6/S7  H       30s       1      128    Any      29
Fa0/3  SC      U6/S7  H       30s       1      128    Any      29
```

关键字 [neighbor] 打印出邻居名称、PAgP邻居的ID、邻居设备ID（MAC）以及邻居端口。同时在比如邻居是一台物理学习设备时（a physical learner），这些标志同样表明了邻居运行的模式。下面的输出对命令 `show pagg neighbor` 的使用，进行了演示。

```

Switch-1#show pagg 1 neighbor
Flags: S - Device is sending Slow hello. C - Device is in Consistent state.
       A - Device is in Auto mode.          P - Device learns on physical port.
Channel group 1 neighbors
      Partner      Partner      Partner      Partner Group
Port    Name     Device ID    Port     Age Flags   Cap.
Fa0/1  Switch-2  0014.a9e5.d640 Fa0/1   19s SAC    10001
Fa0/2  Switch-2  0014.a9e5.d640 Fa0/2   24s SAC    10001
Fa0/3  Switch-2  0014.a9e5.d640 Fa0/3   18s SAC    10001

```

## 配置并验证LACP以太网通道

### Configuring and Verifying LACP EtherChannels

此部分对LACP的二层以太网通道的配置进行了讲述。为配置并建立一个LACP以太网通道，需要执行下面这些步骤。

1. 第一个配置步骤是通过全局配置命令 `interface [name]` 或 `interface range [range]`，进入到所需要的以太网通道接口的接口配置模式；
2. 第二个配置步骤时通过接口配置命令 `switchport`，将这些接口配置为二层交换端口；
3. 第三个配置步骤，时通过接口配置命令 `switchport mode [access|trunk]`，将这些交换端口配置为中继或接入链路；
4. 作为可选步骤，如该接口或这些接口已被配置为接入端口，就要使用命令 `switchport access vlan [number]` 将其指派到同样的VLAN中。而如该接口或这些接口已被配置为中继端口，就要通过执行接口配置命令 `switchport trunk allowed vlan [range]`，选择允许通过该中继的VLANs；而如将不使用VLAN 1作为原生VLAN（802.1Q的），就要通过执行接口配置命令 `switchport trunk native vlan [number]`，输入该原生VLAN。此项配置在所有的端口通道成员接口上一致；
5. 通过执行接口配置命令 `channel-protocol lacp`，将LACP配置作为以太网通道协议。因为以太网通道协议默认时PAgP，该命令被认为时LACP所强制的，同时也是所要求输入的（because EtherChannels default to PAgP, this command is considered mandatory for LACP and is required）；
6. 下一配置步骤时通过接口配置命令 `channel-group [number] mode`，将这些接口配置为无条件中继（the next configuration step is to configure the interfaces to unconditionally trunk via the `channel-group [number] mode interface configuration command`）。

下面的输出对在Switch 1和Switch 2上如何配置基于图33.5中所给出的网络拓扑的LACP通道，进行了演示，该以太网通道将被配置为一个使用默认参数的二层802.1Q中继，如下面的输出所示。

```

Switch-1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-1(config)#int range FastEthernet0/1 - 3
Switch-1(config-if-range)#switchport
Switch-1(config-if-range)#switchport trunk encapsulation dot1q
Switch-1(config-if-range)#switchport mode trunk
Switch-1(config-if-range)#channel-protocol lacp
Switch-1(config-if-range)#channel-group 1 mode active
Creating a port-channel interface Port-channel 1
Switch-1(config-if-range)#exit
Switch-2#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Switch-2(config)#interface range FastEthernet0/1 - 3
Switch-2(config-if-range)#switchport
Switch-2(config-if-range)#switchport trunk encapsulation dot1q
Switch-2(config-if-range)#switchport mode trunk
Switch-2(config-if-range)#channel-protocol lacp
Switch-2(config-if-range)#channel-group 1 mode passive
Creating a port-channel interface Port-channel 1
Switch-2(config-if-range)#exit

```

下面的输出演示了如何通过在 Switch 1 及 Switch 2 上执行 `show EtherChannel summary` 命令，来对该 LACP 以太网通道配置进行验证。

```

Switch-1#show EtherChannel summary
Flags: D - down
      I - stand-alone
      H - Hot-standby (LACP only)
      R - Layer3
      u - unsuitable for bundling
      U - in use
      d - default port
      P - in port-channel
      s - suspended
      S - Layer2
      f - failed to allocate aggregator
Number of channel-groups in use: 1
Number of aggregators: 1
Group  Port-channel  Protocol    Ports
-----+-----+-----+
1      Po1(SU)       LACP        Fa0/1(Pd)  Fa0/2(P)   Fa0/3(P)

```

默认 LACP 允许最多 16 个端口进入到一个端口通道组中 (by default, LACP allows up to 16 ports to be entered into a port channel group)。前 8 个运作接口将为 LACP 所使用，而剩下的 8 个接口将被置为热备份状态。命令 `show EtherChannel detail` 显示出一个 LACP 以太网通道中所支持的链路最大数量，如下面的输出所示。

```

Switch-1#show EtherChannel 1 detail
Group state = L2
Ports: 3 Maxports = 16
Port-channels: 1 Max Port-channels = 16
Protocol: LACP
          Ports in the group:
          -----
Port: Fa0/1
-----
Port state      = Up Mstr In-Bndl
Channel group   = 1           Mode = Active      Gcchange = -
Port-channel    = Po1          GC   = -           Pseudo port-channel = Po1
Port index      = 0           Load = 0x00        Protocol = LACP
Flags: S - Device is sending Slow LACPDU.   F - Device is sending fast
                                         LACPDU.
                                         A - Device is in active mode.       P - Device is in passive mode.
Local information:
          LACP port      Admin      Oper      Port      Port
Port   Flags  State Priority   Key       Key      Number  State
Fa0/1  SA    bndl  32768     0x1      0x1      0x0     0x3D
Partner's information
          Partner      Partner      Partner
Port   System ID      Port Number   Age      Flags
Fa0/1  00001,0014.a9e5.d640 0x1      4s       SP
          LACP Partner      Partner      Partner
          Port Priority      Oper Key    Port State
                                         32768     0x1      0x3C
Age of the port in the current state: 00d:00h:00m:35s
Port: Fa0/2
-----
Port state      = Up Mstr In-Bndl
Channel group   = 1           Mode = Active      Gcchange = -
Port-channel    = Po1          GC   = -           Pseudo port-channel = Po1
Port index      = 0           Load = 0x00        Protocol = LACP
Flags: S - Device is sending Slow LACPDU.   F - Device is sending fast
                                         LACPDU.
                                         A - Device is in active mode.       P - Device is in passive mode.
Local information:
          LACP port      Admin      Oper      Port      Port
Port   Flags  State Priority   Key       Key      Number  State
Fa0/2  SA    bndl  32768     0x1      0x1      0x1     0x3D
Partner's information
          Partner      Partner      Partner
Port   System ID      Port Number   Age      Flags
Fa0/2  00001,0014.a9e5.d640 0x2      28s       SP
          LACP Partner      Partner      Partner
          Port Priority      Oper Key    Port State
                                         32768     0x1      0x3C
Age of the port in the current state: 00d:00h:00m:33s
Port: Fa0/3
-----
Port state      = Up Mstr In-Bndl
Channel group   = 1           Mode = Active      Gcchange = -
Port-channel    = Po1          GC   = -           Pseudo port-channel = Po1
Port index      = 0           Load = 0x00        Protocol = -
Flags: S - Device is sending Slow LACPDU.   F - Device is sending fast
                                         LACPDU.
                                         A - Device is in active mode.       P - Device is in passive mode.
Local information:
          LACP port      Admin      Oper      Port      Port
Port   Flags  State Priority   Key       Key      Number  State
Fa0/3  SA    bndl  32768     0x1      0x1      0x2     0x3D
Partner's information:
          Partner      Partner      Partner
Port   System ID      Port Number   Age      Flags

```

```

Fa0/3      00001,0014.a9e5.d640 0x3          5s      SP
          LACP Partner       Partner       Partner
          Port Priority     Oper Key     Port State
          32768           0x1        0x3C
Age of the port in the current state: 00d:00h:00m:29s
Port-channels in the group:
-----
Port-channel: Po1    (Primary Aggregator)
-----
Age of the Port-channel = 00d:00h:13m:50s
Logical slot/port = 1/0          Number of ports = 3
HotStandBy port = null
Port state      = Port-channel Ag-Inuse
Protocol        = LACP
Ports in the Port-channel:
Index  Load  Port   EC state
-----+-----+-----+
0      00    Fa0/1  Active
0      00    Fa0/2  Active
0      00    Fa0/3  Active
Time since last port bundled: 00d:00h:00m:32s   Fa0/3
Time since last port Un-bundled: 00d:00h:00m:49s  Fa0/1

```

LACP的配置及统计数据也可以通过执行 `show lacp [options]` 命令进行查看。此命令可用的选项在下面的输出中进行了演示。

```

Switch-1#show lacp ?
<1-6>    Channel group number
counters  Traffic information
internal   Internal information
neighbor   Neighbor information
sys-id    LACP System ID

```

`[counters]` 关键字提供了有关LACP发出和接收到的数据包的信息。该命令的打印输出如下面所示。

```

Switch-1#show lacp counters
      LACPDU          Marker      Marker Response      LACPDU
      Sent    Recv    Sent    Recv    Sent    Recv    Pkts Err
Port
-----+
Channel group: 1
Fa0/1   14    12    0     0     0     0     0
Fa0/2   21    18    0     0     0     0     0
Fa0/3   21    18    0     0     0     0     0

```

而 `[internal]` 关键字提供了诸如端口状态、管理密钥（administrative key）、LACP端口优先级，以及端口编号等信息。下面的输出对此进行了演示。

```

Switch-1#show lacp internal
Flags: S - Device is sending Slow LACPDU. F - Device is sending Fast
          LACPDU.
A - Device is in Active mode.      P - Device is in Passive mode.
Channel group 1
      LACP port      Admin      Oper      Port      Port
      Port    Flags  State  Priority    Key    Key    Number  State
Fa0/1    SA    bndl  32768     0x1    0x1    0x0    0x3D
Fa0/2    SA    bndl  32768     0x1    0x1    0x1    0x3D
Fa0/3    SA    bndl  32768     0x1    0x1    0x2    0x3D

```

关键字 [neighbor] 打印出邻居名称、LACP邻居的ID、邻居的设备ID（MAC），以及邻居端口等信息。这些标志还表明邻居运行所处状态，以及其是否时一个物理学习设备（the flags also indicate the mode the neighbor is operating in, as well as whether it is a physical learner, for example）。下面的输出对此进行了演示。

```
Switch-1#show lacp neighbor
Flags: S - Device is sending Slow LACPDU. F - Device is sending Fast
          LACPDU.
          A - Device is in Active mode.      P - Device is in Passive mode.
Channel group 1 neighbors
Partner's information
  Partner          Partner          Partner
Port   System ID    Port Number    Age   Flags
Fa0/1  00001,0014.a9e5.d640 0x1     11s   SP
        LACP Partner    Partner
        Port Priority   Oper Key    Port State
        32768           0x1       0x3C
Partner's information:
  Partner          Partner          Partner
Port   System ID    Port Number    Age   Flags
Fa0/2  00001,0014.a9e5.d640 0x2     19s   SP
        LACP Partner    Partner
        Port Priority   Oper Key    Port State
        32768           0x1       0x3C
Partner's information:
  Partner          Partner          Partner
Port   System ID    Port Number    Age   Flags
Fa0/3  00001,0014.a9e5.d640 0x3     24s   SP
        LACP Partner    Partner
        Port Priority   Oper Key    Port State
        32768           0x1       0x3C
```

最后，关键字 [sys-id] 提供了本地交换机的系统ID（finally, the [sys-id] keyword provides the system ID of the local switch）。这是一个该交换机MAC地址和LACP优先级的结合体，如下面的输出所示。

```
Switch-1#show lacp sys-id
1 ,000d.bd06.4100
```

## 第33天问题

1. What type of ports does a FastEtherChannel contain?
2. How many ports can a standard EtherChannel contain?
3. What are the two protocol options you have when configuring EtherChannels on a Cisco switch?
4. Which of the protocols mentioned above is Cisco proprietary?
5. PagP packets are sent to the destination Multicast MAC address 01-00-0C-CC-CC-CC . True or false?
6. What are the two port modes supported by PagP?
7. What are the two port modes supported by LACP?
8. If more than eight links are assigned to an EtherChannel bundle running LACP, the protocol uses the port priority to determine which ports are placed into a standby mode. True or false?
9. LACP automatically configures an administrative key value on each port configured to use LACP. The administrative key defines the ability of a port to aggregate with other ports. Only ports that have the same administrative key are allowed to be aggregated into the same port channel group. True or false?
10. What is the command used to assign a port to a channel group?

## 第33天答案

1. 100 Mbps ports.
2. Up to eight ports.
3. PagP and LACP.
4. PagP.
5. True.
6. Auto and desirable.
7. Active and passive.
8. True.
9. True.
10. The `channel-group [number] mode` command in Interface Configuration mode.

## 第33天实验

### 以太网通道实验

#### EtherChannel Lab

在一个包含了两台直接相连的交换机（它们至今至少有两条链路）上，对本课程模块中出现的配置命令进行测试。通过Fa1/1及Fa2/2将它们连接起来（Fa1/1到Fa1/1及Fa2/2到Fa2/2）。

- 在两条链路上以 `auto-desirable` 模式配置PAgP
- 将该以太网通道配置为一条中继并允许一些VLANs通过它
- 执行一条 `show etherchannel summary` 命令，并验证该端口通道是运行的
- 执行一条 `show mac-address-table` 命令，并看看在两台交换机上所学习到的MAC地址
- 执行一条 `show pagp neighbor` 命令，并检查结果
- 采用LACP的 `passive-active` 模式，重复上述步骤
- 使用命令 `show EtherChannel detail` 及 `show lacp neighbor` 命令，对配置进行验证
- 使用 `show interface port-channel [number] switchport` 命令，对配置进行验证
- 通过端口通道发出一些流量（ping），并使用 `show lacp counters` 命令对计数器进行检查
- 配置一个不同的 `lacp system-priority` 输出，并使用 `show lacp sys-id` 命令予以验证
- 配置一个不同的 `lacp port-priority` 输出，并使用命令 `show lacp internal` 予以验证
- 使用命令 `port-channel load-balance`，对LACP的负载均衡进行配置，并使用 `show etherchannel load-balance` 命令对此进行验证

## 第34天 第一跳冗余协议

### First Hop Redundancy Protocols

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第34天任务

- 阅读今天的课文
- 回顾昨天的课文
- 完成今天的实验
- 阅读ICND2记诵指南
- 在[subnetting.org](http://subnetting.org)上花15分钟

在设计和部署交换网络时，**高可用性**（High Availability, HA）是一项不可或缺的考虑。作为思科IOS软件中所提供的一项技术，高可用性确保了网络层面的弹性与恢复能力，从而提升了IP网络的可用性。所有网段都必须具备弹性和恢复能力，以便网络能足够快地从故障中恢复过来，且此恢复过程要对用户及网络应用无感知及透明。这里的这些第一跳冗余协议（First Hop Redundancy Protocols, FHRPs），就提供了在不同的包交换局域网环境下的冗余。

今天将学习以下内容：

- 热备份路由器协议（Hot Standby Router Protocol）
- 虚拟路由器冗余协议（Virtual Router Redundancy Protocol）
- 网关负载均衡协议（Gateway Load Balancing Protocol）

这节课对应了一下ICND2考试大纲要求：

- 认识高可用性（FHRP）
  - HSRP
  - VRRP
  - GLBP

## 热备份路由器协议

### Hot Standby Router Protocol

热备份路由器协议是一项思科公司专有的第一跳冗余协议。HSRP令到两台配置在同样HSRP组中的物理网关，使用同样的虚拟网关地址。而位处这两台网关所在的子网中的网络主机，就以该虚拟网关IP地址，作为其默认网关地址。

当HSRP运作时，由主网关（the primary gateway）转发该HSRP组的那些以虚拟网关IP地址为目的地址的数据包。而加入主网关失效，则从网关（the secondary gateway）就接过主网关角色，并转发那些发送到虚拟网关IP地址的数据包。下面的图34.1演示了某网络中HSRP的运作：

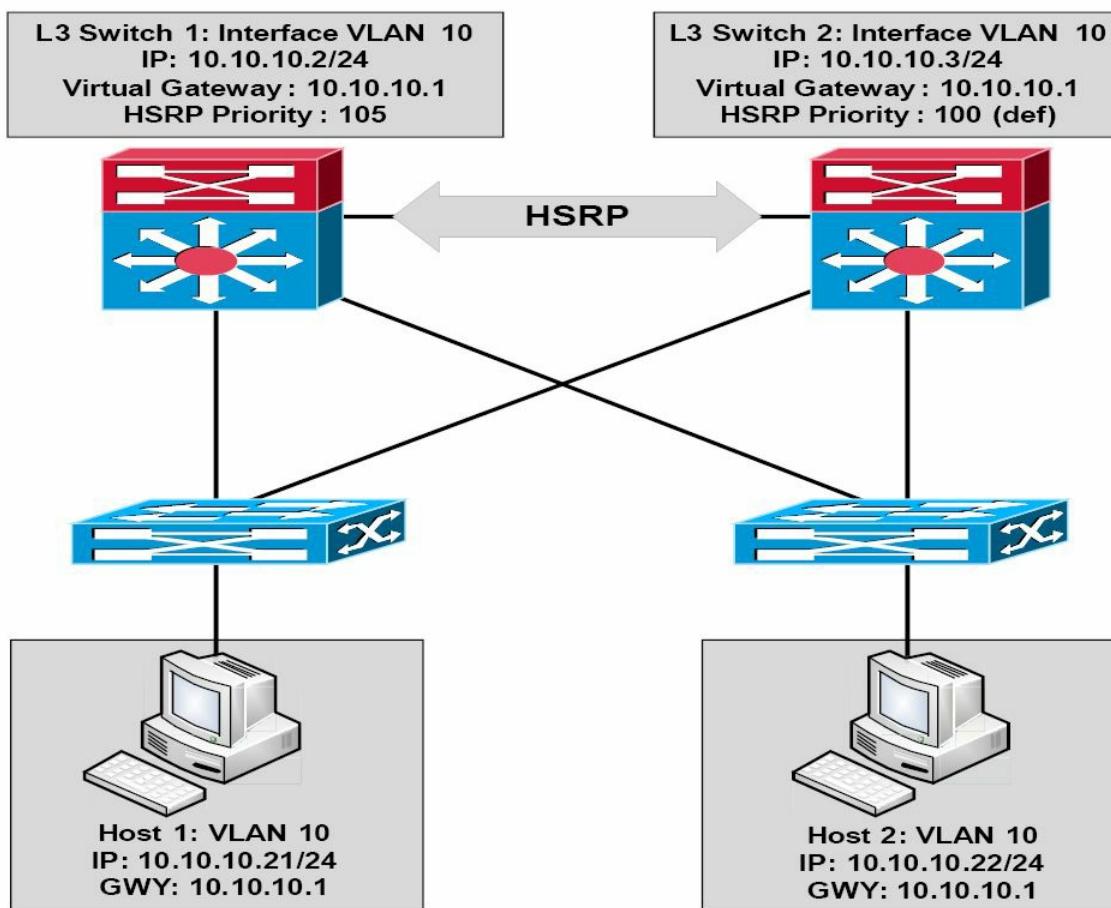


图 34.1 -- 热备份路由器协议的运作

参阅图 34.1，HSRP 是在三层（分发/分布层，the Layer 3, Distribution Layer）交换机之间的，给 VLAN 10 提供了网关的冗余性。分配给三层交换机 Switch 1 上的交换机虚拟借口（the Switch Virtual Interface, SVI）的IP地址是 10.10.10.2/24，同时分配给三层交换机 Switch 2 的交换机虚拟接口的IP地址是 10.10.10.3/24。两台交换机都被配置为同一HSRP组的组成部分，并共用了该虚拟网关 10.10.10.1。

Switch 1 已被配置了优先级值 105，而 Switch 2 使用的是默认优先级值 100。因为三层交换机 Switch 1 有着更高的优先级值，其就被选作主交换机，同时三层交换机 Switch 2 被选作从交换机。在 VLAN 10 上的所有主机，都配置了默认网关地址 10.10.10.1。因此，假如 Switch 1 失效，Switch 2 就将接过网关的职责。此过程对这些网络主机完全透明无感知。

#### 真实世界应用 Real-World Implementation

在生产网络中配置各种FHRPs，确保子网的活动（主）网关同时也是该特定VLAN的生成树根桥，被认为是一种好的做法。比如参阅图 34.1 中的图例，Switch 1 在作为 VLAN 10 的 HSRP 主网关的同时，也应被配置为该 VLAN 的根桥。

如此做法带来的是一个确切的网络（a deterministic network），从而避免在二层或三层上的次优转发。比如假设 VLAN 10 的根桥是 Switch 2，而 VLAN 10 的主网关又是 switch 1，那么从网络主机发送到默认网关IP地址的数据包就将如下图34.2那样被转发了：

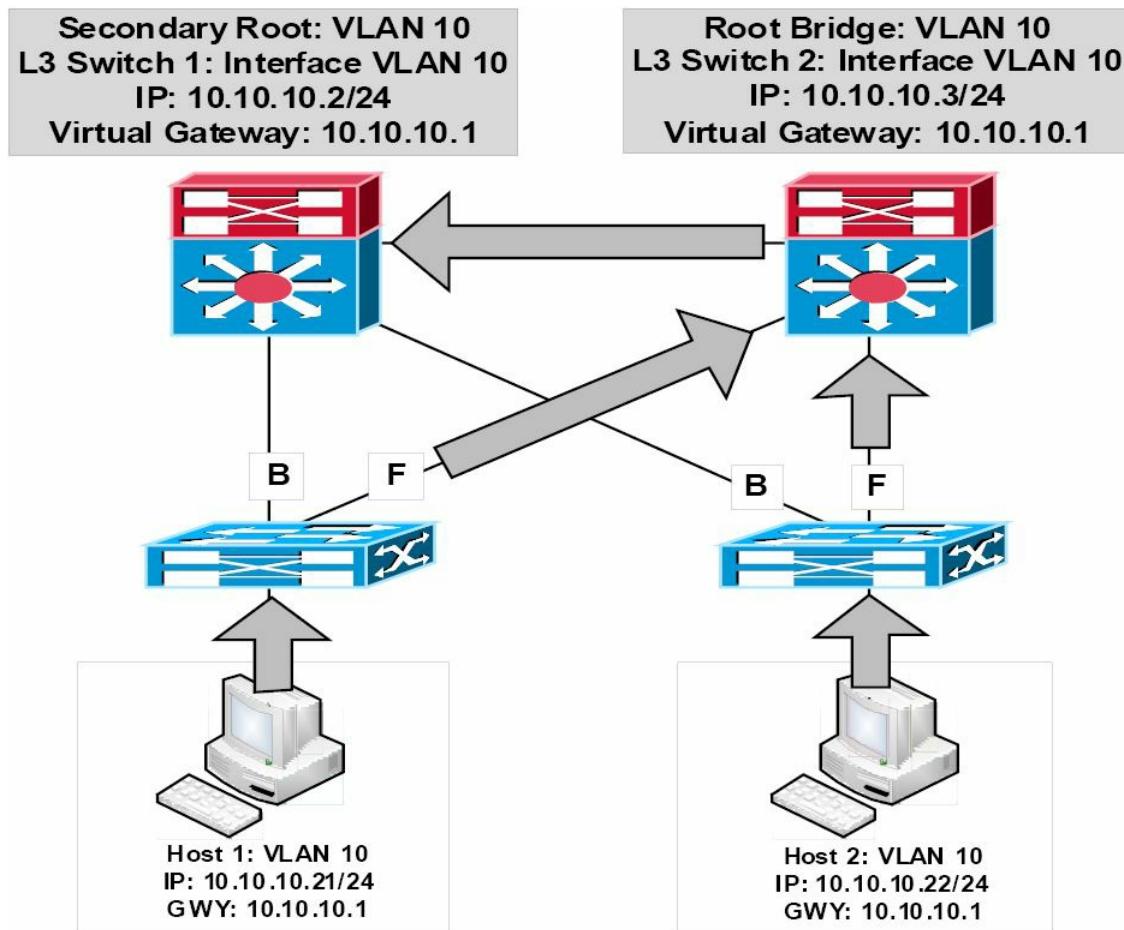


图 34.2 -- STP 拓扑与 HSRP 拓扑的同步，Synchronising the STP Topology with HSRP

在上面的网络中，从 Host 1 到 10.10.10.1 的数据包将被如下这样转发：

1. 接入层交换机收到一个来自 Host 1、以虚拟网关IP地址之MAC地址为目标的数据帧。此数据帧实在 VLAN 10 中收到的，同时该接入交换机经由其根端口，已学习到了虚拟网关的MAC地址。
2. 因为 VLAN 10 的根桥是 Switch 2，那么到 switch 1 （也就是HSRP的主路由器）的上行线路也被置于阻塞状态。此时该接入交换机就将该数据帧经由到 Switch 2 的上行链路予以转发。
3. Switch 2 又经由连接到 Switch 1 的指定端口，转发该数据帧。对于来自 Host 2 的数据帧，会用上述的相同次优路径。

思科IOS软件当前支持两个版本的HSRP：版本1及版本2。后续章节将对它们的相似点和不同点进行说明。

## HSRP版本1

默认情况下，当在思科IOS软件中开启热备份路由器协议时，是开启的版本1。HSRP版本1将可配置的HSRP分组限制在最多255个。HSRP版本1的那些路由器之间的通信，是通过往多播组地址（Multicast group address） 224.0.0.2 上，使用UDP端口 1985 发送报文进行的。下面的图34.3显示了HSRP版本1的报文：

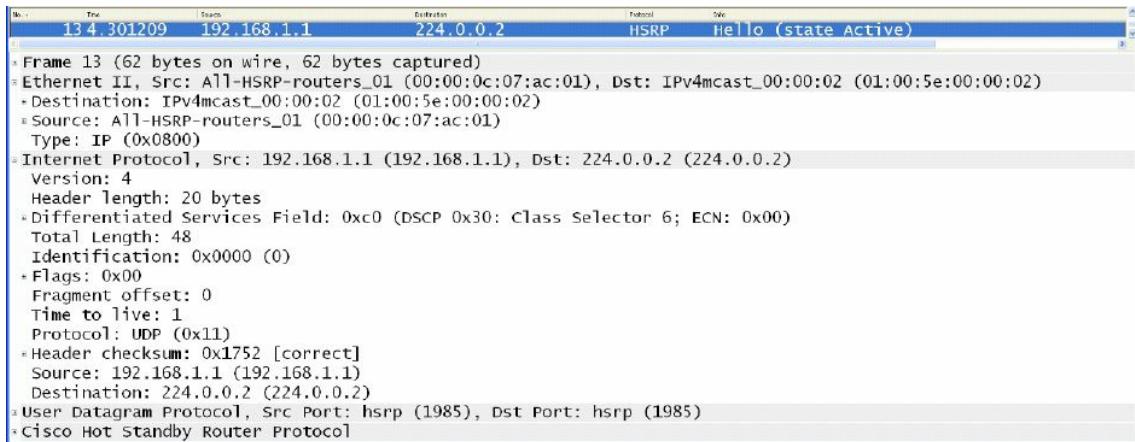


图 34.3 -- HSRP 版本1多播组地址

对HSRP数据包格式的深入探讨，是超出CCNA考试要求的范围的，下图34.4仍然给出了HSRP版本1数据包的信息：

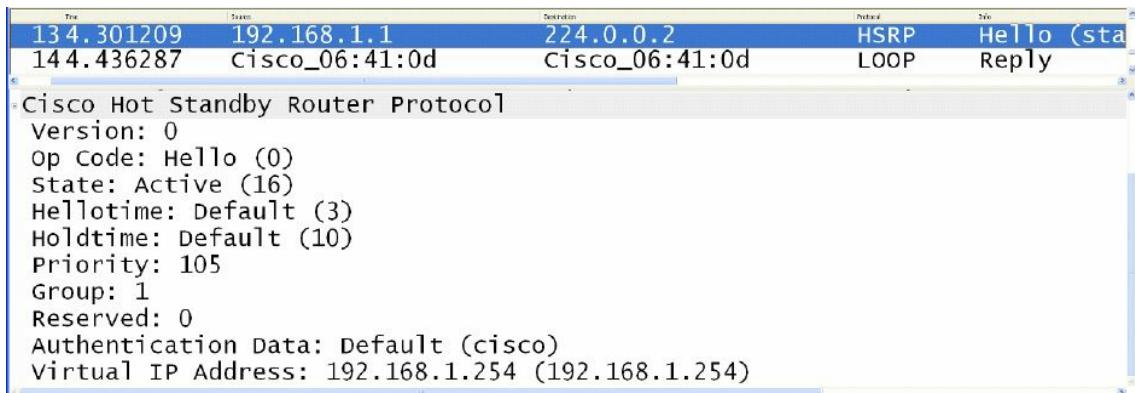


图 34.4 -- HSRP 版本1数据包的字段

在图34.4中，要注意版本字段显示的是数值0。这是在版本1开启时该字段的默认值；不过仍然要知道这里使用的是HSRP版本1。

## HSRP版本2

HSRP版本2使用了新的多播地址 224.0.0.102，而不是版本1的多播地址 224.0.0.2，来发送 Hello 数据包。不过其所用到的UDP端口号仍然一样（1985）。同时此新地址在IP数据包及以太网数据帧中都得以编码，如下图34.5所示：

No.	Time	Source	Destination	Protocol	Info
8 3.349709		192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
15 6.350550		192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
19 9.351449		192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)

\* Frame 8 (94 bytes on wire, 94 bytes captured)  
 \* Ethernet II, Src: Cisco\_9f:f0:01 (00:00:0c:9f:f0:01), Dst: IPv4mcast\_00:00:66 (01:00:5e:00:00:66)  
 - Destination: IPv4mcast\_00:00:66 (01:00:5e:00:00:66)  
 - Source: Cisco\_9f:f0:01 (00:00:0c:9f:f0:01)  
 Type: IP (0x0800)  
 - Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.102 (224.0.0.102)  
 Version: 4  
 Header length: 20 bytes  
 Differentiated Services Field: 0xc0 (DSCP 0x30; Class Selector 6; ECN: 0x00)  
 Total Length: 80  
 Identification: 0x0000 (0)  
 Flags: 0x00  
 Fragment offset: 0  
 Time to live: 1  
 Protocol: UDP (0x11)  
 Header checksum: 0x16ce [correct]  
 Source: 192.168.1.1 (192.168.1.1)  
 Destination: 224.0.0.102 (224.0.0.102)  
 User Datagram Protocol, Src Port: hsrp (1985), Dst Port: hsrp (1985)  
 Cisco Hot Standby Router Protocol

图 34.5 -- HSRP版本2多播组地址

对HSRP版本2数据包格式的深入探讨，也是超出CCNA考试要求范围的，但要记住HSRP版本2并未使用与版本1相同的数据包格式。

版本2数据包使用了一直类型/长度/值的格式（a Type/Length/Value format, TLV format）。被HSRP版本1的路由器接收到的版本2数据包，会将类型字段映射到HSRP版本1的版本字段，而被忽略掉。下图34.6给出了HSRP版本2数据包中所包含的信息：

No.	Time	Source	Destination	Protocol	Info
8 3.349709		192.168.1.1	224.0.0.102	HSRPv2	Hello (state Act)
15 6.350550		192.168.1.1	224.0.0.102	HSRPv2	Hello (state Act)

Cisco Hot Standby Router Protocol  
 - Group State TLV: Type=1 Len=40  
 Version: 2  
 Op Code: Hello (0)  
 State: Active (6)  
 IP Ver.: IPv4 (4)  
 Group: 1  
 Identifier: Cisco\_86:0a:20 (00:13:19:86:0a:20)  
 Priority: 105  
 Hellotime: Default (3000)  
 Holdtime: Default (10000)  
 Virtual IP Address: 192.168.1.254 (192.168.1.254)  
 - Text Authentication TLV: Type=3 Len=8  
 Authentication Data: Default (cisco)

图 34.6 -- HSRP版本2的数据包字段

## HSRP版本1与版本2的比较

HSRP 版本2包括了一些对版本1的增强。本小节将对这些增强及与版本1的不同进行说明。

尽管HSRP版本1通告了计时器数值，但这些数值都是整秒的，因为版本1无法通告或学习到毫秒的计时器数值。而版本2既可以通告也可以学习毫秒的计时器数值了。下面的图34.7与图34.8分别着重表示了HSRP版本1与版本2在计时器字段上的不同：

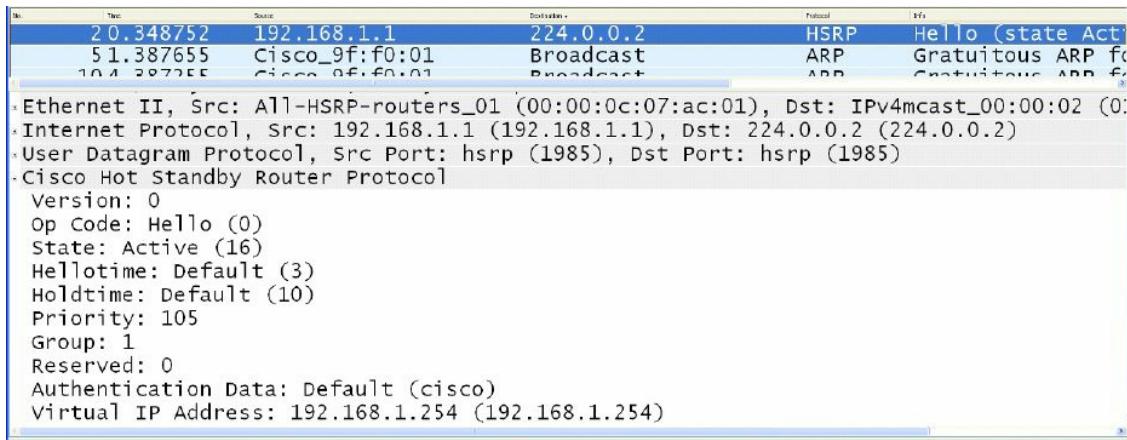


图 34.7 -- HSRP版本1的计时器字段

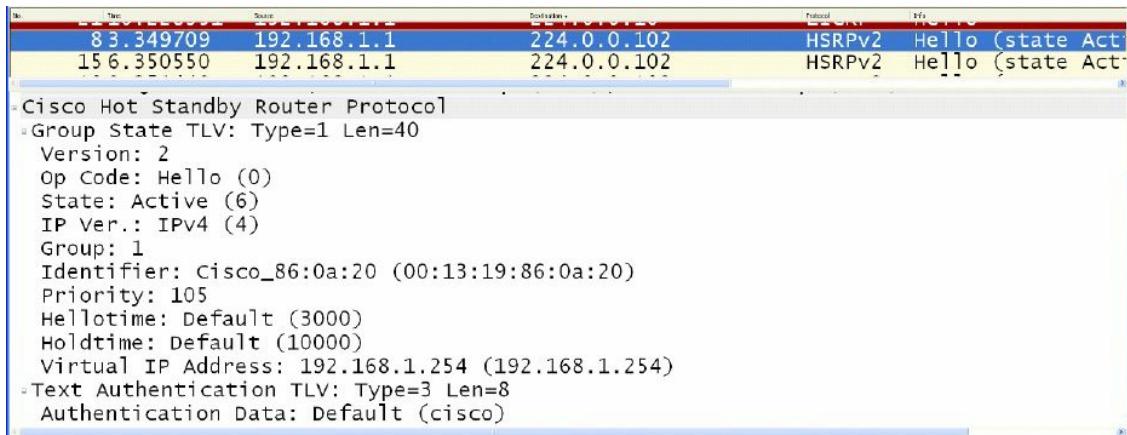


图 34.8 -- HSRP版本2的计时器字段

HSRP版本1的分组编号是限制在 0 到 255 的，而版本 2 的分组编号则已拓展到 0 到 4095 了。本课程模块后面的HSRP配置示例中，将就此差异进行演示。

版本2通过包含一个由物理路由器接口的MAC地址生成、用于对HSRP活动 Hello 报文来源的唯一性识别的6字节识别符字段（a 6-byte Identifier field），提供了改进的管理与故障排除功能。在版本1中，这些 Hello 报文所包含的源MAC地址，都是虚拟MAC地址，那就是说无法找出是哪台HSRP路由器发送的 HSRP Hello 报文。下图34.9给出了HSRP版本2，而非版本1数据包中出现的识别符字段：

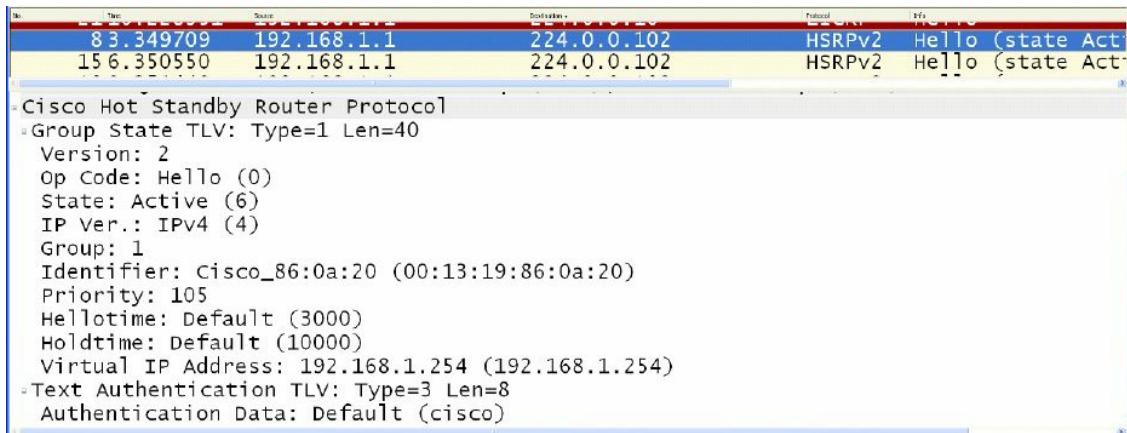


图 34.9 -- HSRP版本2中的识别符字段

在HSRP版本1中，虚拟IP地址所使用的二层地址将是一个由 `0000.0C07.ACxx` 构成的虚拟MAC地址，这里的 `xx` 就是HSRP分组编号的十六进制值，同时是基于相应接口的。而在HSRP版本2中，虚拟网关IP地址则是使用了新的MAC地址范围 `0000.0C9F.F000` 到 `0000.0C9F.FFFF`。下图34.10给出了这些不同，该图现实了 HSRP Group 1 的版本1的虚拟MAC地址，同时在图34.11中显示了版本2的虚拟MAC地址，也是HSRP Group 1 的：

Time	Src	Dest	Protocol	Info
20.348752	192.168.1.1	224.0.0.2	HSRP	Hello (state Active)
83.349709	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
Ethernet II, Src: All-HSRP-routers_01 (00:00:0c:07:ac:01), Dst: IPv4mcast_00:00:02 (01:00:5e:00:00:02)				
Destination: IPv4mcast_00:00:02 (01:00:5e:00:00:02) Address: IPv4mcast_00:00:02 (01:00:5e:00:00:02) .... .1 .... .... .... = IG bit: Group address (multicast/broadcast) .... .0 .... .... .... = LG bit: Globally unique address (factory default)				
Source: All-HSRP-routers_01 (00:00:0c:07:ac:01) Address: All-HSRP-routers_01 (00:00:0c:07:ac:01) .... .0 .... .... .... = IG bit: Individual address (unicast) .... .0 .... .... .... = LG bit: Globally unique address (factory default)				
Type: IP (0x0800)				
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.2 (224.0.0.2)				
User Datagram Protocol, Src Port: hsrp (1985), Dst Port: hsrp (1985)				
Cisco Hot Standby Router Protocol				
Version: 0				
Op Code: Hello (0)				
State: Active (16)				
Hello time: Default (3)				
Holdtime: Default (10)				
Priority: 105				
Group: 1				
Reserved: 0				

图 34.10 -- HSRP 版本1的虚拟MAC地址格式

Time	Src	Dest	Protocol	Info
83.349709	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
156.350550	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
Ethernet II, Src: Cisco_9f:f0:01 (00:00:0c:9f:f0:01), Dst: IPv4mcast_00:00:66 (01:00:5e:00:00:66)				
Destination: IPv4mcast_00:00:66 (01:00:5e:00:00:66) Address: IPv4mcast_00:00:66 (01:00:5e:00:00:66) .... .1 .... .... .... = IG bit: Group address (multicast/broadcast) .... .0 .... .... .... = LG bit: Globally unique address (factory default)				
Source: Cisco_9f:f0:01 (00:00:0c:9f:f0:01) Address: Cisco_9f:f0:01 (00:00:0c:9f:f0:01) .... .0 .... .... .... = IG bit: Individual address (unicast) .... .0 .... .... .... = LG bit: Globally unique address (factory default)				
Type: IP (0x0800)				
Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.102 (224.0.0.102)				
User Datagram Protocol, Src Port: hsrp (1985), Dst Port: hsrp (1985)				
Cisco Hot Standby Router Protocol				
- Group State TLV: Type=1 Len=40				
Version: 2				
Op Code: Hello (0)				
State: Active (6)				
IP Ver.: IPv4 (4)				
Group: 1				
Identifier: Cisco_86:0a:20 (00:13:19:86:0a:20)				
Priority: 105				

图 34.11 -- HSRP 版本2的虚拟MAC地址格式

## HSRP的主网关选举

可通过将默认HSRP优先级值 `100`，修改为 `1` 到 `255` 之间的任何值，对HSRP主网关的选举施加影响。有着最高优先级的路由器将被选举为该HSRP分组的主网关。

而在两个网关都使用默认优先级值时，或两个网关上的优先级值被手工配置为相等时，那么有着最高IP地址的路由器将被选举为主网关。在HSRP数据帧中，HSRP优先级值与该路由器的当前状态（比如是主路由器还是备份路由器），都有进行传送。下图34.12演示了一台配置了非默认优先级值 `105`，此优先级令到该路由器被选举为此HSRP组的活动网关，的网关的优先级和状态字段：

No.	Time	Source	Destination	Protocol	Info
101	33.009872	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
96	30.008988	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
92	27.008057	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
74	24.007110	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
49	21.006350	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)
24	18.005333	192.168.1.1	224.0.0.102	HSRPv2	Hello (state Active)

\* Frame 96 (94 bytes on wire, 94 bytes captured)  
 \* Ethernet II, Src: Cisco\_9f:f0:01 (00:00:0c:9f:f0:01), Dst: IPv4mcast\_00:00:66 (01:00:5e:00:00:66)  
 \* Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.102 (224.0.0.102)  
 \* User Datagram Protocol, Src Port: hsrp (1985), Dst Port: hsrp (1985)  
 \* Cisco Hot Standby Router Protocol  
 - Group State TLV: Type=1 Len=40  
 Version: 2  
 Op Code: Hello (0)  
 State: Active (6)  
 IP Ver.: IPv4 (4)  
 Group: 1  
 Identifier: Cisco\_86:0a:20 (00:13:19:86:0a:20)  
 Priority: 105  
 Hellotime: Default (3000)  
 Holdtime: Default (10000)  
 Virtual IP Address: 192.168.1.254 (192.168.1.254)  
 - Text Authentication TLV: Type=3 Len=8  
 Authentication Data: Default (cisco)

图 34.12 -- HSRP的优先级与状态字段

## HSRP报文

HSRP路由器之间就下列三种类型的报文进行交换：

- Hello报文
- Coup报文
- Resign报文

**Hello** 报文是经由多播进行交换的，这些报文告诉另一网关本地路由器的HSRP状态和优先级数值。**Hello** 报文还包含了组ID（the Group ID）、各种HSRP计时器数值、HSRP版本，以及认证信息。前面给出的所有报文，都是HSRP的Hello报文。

HSRP Coup报文实在当前备份路由器打算接过该HSRP组的活动网关角色时发出的。这与现实生活中的一次篡位（a coup d'état）类似。

而HSRP的 **Resign** 报文，则是在活动路由器即将关闭，以及在一台有着更高优先级的网关发出一个 **Hello** 报文或 **Coup** 报文时发出的。也就是说，在活动网关交出其作为主网关角色时，发出此报文。

## HSRP的抢占

### HSRP Preemption

在已有一台网关被选举为活动网关的情况下，作为HSRP组一部分的另一网关被重新配置了一个更高的HSRP优先级数值时，当前活动网关会保留主转发角色。这是HSRP的默认行为。

而为了在某HSRP组中已有一个主网关的情形下，令到具有更高优先级的网关接过活动网关功能，就必须将该路由器配置上抢占功能。这样做就允许该网关发起一次抢占，并接过该HSRP组的活动网关角色。

HSRP抢占在接着的配置示例中有演示。

**注意：** 抢占并不意味着生成树拓扑也会发生改变（译者注：这将导致次优路径）。

## HSRP的各种状态

与开放最短路径优先（Open Shortest Path First, OSPF）的方式类似，当在某个接口上开启了HSRP时，该网关接口会经历以下一系列状态的改变：

1. 关闭 ( `Disabled` )
2. 初始化 ( `Init` )
3. 倾听 ( `Listen` )
4. `Speak`
5. 备份 ( `Standby` )
6. 活动 ( `Active` )

**注意：**这些接口状态过度并无设置时间数值（There are no set time values for these interface transitions）。

在关闭及初始化状态中，该网关处于尚未准备妥当或是无法参与到HSRP组情形，可能的原因在于相关接口没有开启。

而倾听状态是适用于备份网关的。仅有备份网关才会监听来自活动网关的 `Hello` 报文。假如备份网关在 10 秒内未能收到 `Hello` 报文，其就假定活动网关已经宕机，并接过活动网关角色。如有在统一网段上存在其它网关，这些网关也会倾听 `Hello` 报文，且如果它们有着下一最高优先级值或IP地址，那么它们就会被选举为该分组的活动网关。

在 `Speak` 阶段，备份网关与活动网关进行报文交换。在此阶段完成后，主网关就过渡到活动状态，同时备份网关过渡到备份状态。备份状态表明该网关已准备好在主网关阵亡时接过活动网关角色，同时活动状态表明该网关已准备好进行数据包的转发。

以下输出给出了在一台刚开启HSRP的网关上，`debug standby` 命令中显示的状态变化：

```

R2#debug standby
HSRP debugging is on
R2#
R2#conf t
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.
End with CNTL/Z.
R2(config)#logging con
R2(config)#int f0/0
R2(config-if)#stand 1 ip 192.168.1.254
R2(config-if)#
*Mar 1 01:21:55.471: HSRP: Fa0/0 API 192.168.1.254 is not an HSRP address
*Mar 1 01:21:55.471: HSRP: Fa0/0 Grp 1 Disabled -> Init
*Mar 1 01:21:55.471: HSRP: Fa0/0 Grp 1 Redundancy "hsrp-Fa0/0-1" state Disabled -> Init
*Mar 1 01:22:05.475: HSRP: Fa0/0 Interface up
...
[Truncated Output]
...
*Mar 1 01:22:06.477: HSRP: Fa0/0 Interface min delay expired
*Mar 1 01:22:06.477: HSRP: Fa0/0 Grp 1 Init: a/HSRP enabled
*Mar 1 01:22:06.477: HSRP: Fa0/0 Grp 1 Init -> Listen
*Mar 1 01:22:06.477: HSRP: Fa0/0 Redirect adv out, Passive, active 0 passive 1
...
[Truncated Output]
...
*Mar 1 01:22:16.477: HSRP: Fa0/0 Grp 1 Listen: d/Standby timer expired (unknown)
*Mar 1 01:22:16.477: HSRP: Fa0/0 Grp 1 Listen -> Speak
...
[Truncated Output]
...
*Mar 1 01:22:26.478: HSRP: Fa0/0 Grp 1 Standby router is local
*Mar 1 01:22:26.478: HSRP: Fa0/0 Grp 1 Speak -> Standby
*Mar 1 01:22:26.478: %HSRP-5-STATECHANGE: FastEthernet0/0 Grp 1 state Speak -> Standby
*Mar 1 01:22:26.478: HSRP: Fa0/0 Grp 1 Redundancy "hsrp-Fa0/0-1" state Speak -> Standby

```

## HSRP地址分配

### HSRP Addressing

在本课程模块的早期，已了解到HSRP版本1中，用于虚拟IP地址的二层地址将是一个由 000.0C07.ACxx 构成的虚拟MAC地址，其中的 xx 就是该HSRP组的编号，且是基于相应接口的。而在HSRP版本2中，使用了一个新的MAC地址范围，从 0000.0C9F.F000 到 0000.0C9F.FFFF，作为虚拟网关IP地址的虚拟MAC地址。

而在某些情况下，我们并不期望使用这些默认的地址范围。比如在连接到一个配置了端口安全的交换机端口的某个路由器接口上，配置了好几个HSRP组时。在此情况下，该路由器就应对不同HSRP组使用不同的MAC地址，那么结果就是这些MAC地址都需要满足（accommodate）交换机端口的安全配置。该项配置在每次将HSRP组加入到路由器接口时都必须进行修改；否则就会触发端口安全冲突（otherwise, a port security violation would occur）。

为解决此问题，思科IOS软件允许管理员将HSRP配置为使用其所配置上的物理接口的实际MAC地址。那么结果就是一个单独的MAC地址为所有HSRP组所使用（也就是活动网关所使用的MAC地址），且在每次往连接到这些交换机上的路由器添加HSRP组的时候，无需对端口安全配置进行修改。此操作是通过使用接口配置命令 `standby use-bia` 命令完成的。下面的输出演示了命令 `show standby`，该命令给出了一个配置了两个不同HSRP组的网关接口的信息：

```

Gateway-1#show standby
FastEthernet0/0 - Group 1
  State is Active
    8 state changes, last state change 00:13:07
    Virtual IP address is 192.168.1.254
    Active virtual MAC address is 0000.0c07.ac01
      Local virtual MAC address is 0000.0c07.ac01 (v1 default)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 2.002 secs
    Preemption disabled
    Active router is local
    Standby router is 192.168.1.2, priority 100 (expires in 9.019 sec)
    Priority 105 (configured 105)
    IP redundancy name is "hsrp-Fa0/0-1" (default)
FastEthernet0/0 - Group 2
  State is Active
    2 state changes, last state change 00:09:45
    Virtual IP address is 172.16.1.254
    Active virtual MAC address is 0000.0c07.ac02
      Local virtual MAC address is 0000.0c07.ac02 (v1 default)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 2.423 secs
    Preemption disabled
    Active router is local

```

在上面的输出中，由于是默认的HSRP版本，那么HSRP Group 1 的虚拟MAC地址就是 0000.0c07.ac01，同时HSRP组2的就是 0000.0c07.ac02。这就意味着连接此网关的交换机端口要学习三个不同地址：物理接口 Fastethernet0/0 的实际或出厂地址、HSRP Group 1 的虚拟MAC地址，以及HSRP组2的虚拟MAC地址。

下面的输出，演示了如何将HSRP配置为使用该网关接口的实际MAC地址，作为不同HSRP分组的虚拟MAC地址：

```

Gateway-1#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line. End with CNTL/Z.
Gateway-1(config)#int f0/0
Gateway-1(config-if)#standby use-bia
Gateway-1(config-if)#exit

```

基于上面的输出中的配置，命令 show standby 会反应出HSRP组的新MAC地址，如下面的输出所示：

```

Gateway-1#show standby
FastEthernet0/0 - Group 1
  State is Active
    8 state changes, last state change 00:13:07
    Virtual IP address is 192.168.1.254
    Active virtual MAC address is 0013.1986.0a20
      Local virtual MAC address is 0013.1986.0a20 (bia)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 2.756 secs
    Preemption disabled
    Active router is local
    Standby router is 192.168.1.2, priority 100 (expires in 9.019 sec)
    Priority 105 (configured 105)
    IP redundancy name is "hsrp-Fa0/0-1" (default)

FastEthernet0/0 - Group 2
  State is Active
    2 state changes, last state change 00:09:45
    Virtual IP address is 172.16.1.254
    Active virtual MAC address is 0013.1986.0a20
      Local virtual MAC address is 0013.1986.0a20 (bia)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 0.188 secs
    Preemption disabled
    Active router is local
    Standby router is unknown
    Priority 105 (configured 105)
    IP redundancy name is "hsrp-Fa0/0-2" (default)

```

那么这里两个HSRP组所用的MAC地址，都是 0013.1986.0a20，就是分配给物理网关接口的MAC地址了。这在下面的输出中有证实：

```

Gateway-1#show interface FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0013.1986.0a20 (bia 0013.1986.0a20)
  Internet address is 192.168.1.1/24
  MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation ARPA, loopback not set
...
[Truncated Output]

```

**注意：**除了将HSRP配置为使用出厂地址（the burnt-in address, BIA），管理员亦可经由接口配置命令 `standby [number] mac-address [mac]`，静态指定虚拟网关要使用的MAC地址。但一般不会这样做，因为这可能会导致交换网络中的重复MAC地址，这就会引起严重的网络故障，甚至造成网络中断。

## HSRP的明文认证

### HSRP Plain Text Authentication

HSRP报文默认以明文密钥字串(the plain text key string) `cisco` 发送，以此作为一种对HSRP成员（HSRP peers）进行认证的简单方式。如报文中的密钥字串与HSRP成员路由器上所配置的密钥匹配，报文就被接受。否则，HSRP就忽略那些未认证的报文。

明文密钥提供了最低的安全性，因为使用诸如Wireshark或Ethereal这样的简单抓包软件，它们就可被抓包捕获。下图34.13显示了HSRP报文中所使用的默认命令认证密钥：

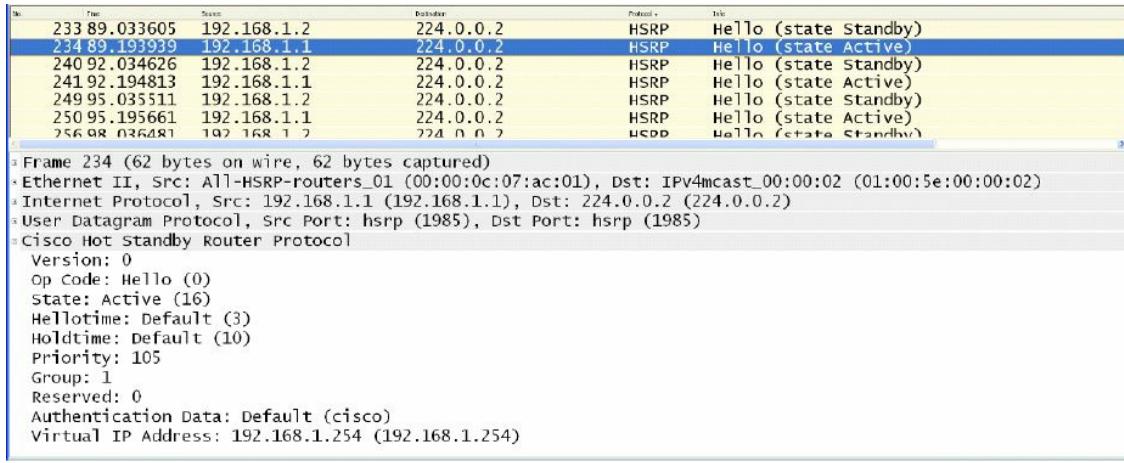


图 34.13 -- 查看HSRP默认明文密钥

因为明文认证提供很低的安全性，那么下面介绍的消息摘要5（message digest 5, MD5），就是推荐的 HSRP 认证方式了。

## HSRP MD5 认证

这并非CCNA题目，放在这里是为了完整性及那些要实际从事网络方面工作的人的考虑。

消息摘要5认证通过生成一个多播HSRP协议数据包的HSRP部分的摘要，提供了HSRP比起明文认证更强的安全性。在举行了MD5认证后，就允许各个HSRP组成员使用一个密钥，来生成一个加密了的MD5散列值，并作为发出数据包的一部分。而接收到的HSRP数据包也会产生一个加密的散列值，如果接收到的数据包的加密散列值与MD5生成值不匹配，接收路由器就会忽略此数据包。

既可以通过在配置使用一个密钥字串直接提供MD5散列值的密钥，也可以通过密钥链（a key chain）来提供到。本课程模块稍后会对这两种方式进行讲解。在应用了明文或是MD5认证时，在出现以下情形之后，网关都会拒绝那些HSRP数据包：

- 路由器与收到的数据包认证方案不一致时
- 路由器与收到的数据包的MD5摘要不同时
- 路由器与收到的数据包的明文认证字串不一致时

## HSRP接口跟踪

### HSRP Interface Tracking

HSRP允许管理员对当前活动网关上的接口状态进行追踪，所以在有接口失效时，网关就会将其优先级降低一个特定数值，默认为 10，这样就可以让其它网关接过HSRP组的活动网关角色。此概念在下图34.14中进行了演示：

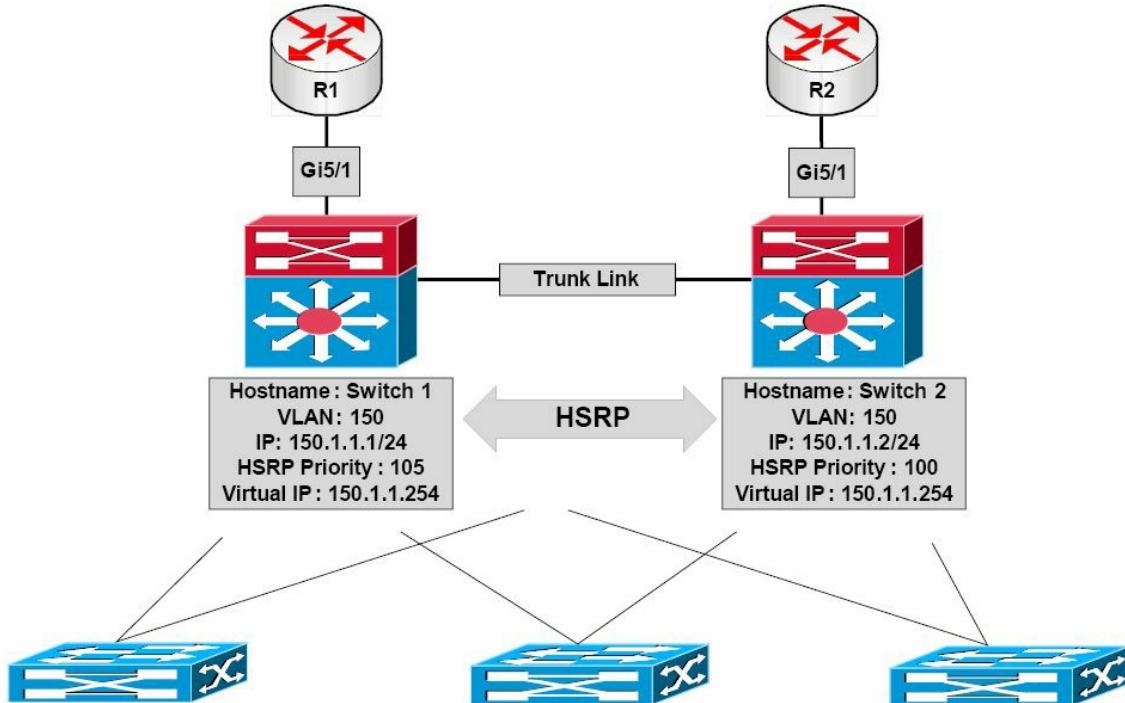


图 34.14 -- HSRP 接口追踪

参考图34.14, 对于 VLAN 150 ,已在 Switch 1 及 Switch 2 上开启了HSRP。而基于当前的优先级配置, Switch 1 有着优先级数值 105 ,已被选举为该VLAN的主交换机。 Switch 1 与 Switch 2 都通过其各自的 Gigabitethernet5/1 接口, 分别连接到两台路由器。这里假定这两台与其它外部网络相连, 比如互联网。

在没有HSRP接口跟踪功能时, 如果 Switch 1 与 R1 之间的 Gigabitethernet5/1 接口失效, 那么 Switch 1 仍将保持其主网关状态。此时就必须将所有接收到的、比如前往互联网的数据包, 使用 Switch 1 本身与 Switch 2 之间的连接, 转发到 Switch 2 上。这些数据包将会通过 R2 转发到它们本来的目的地。这就造成了网络中的次优流量路径。

HSRP接口跟踪功能令到管理员可将HSRP配置为追踪某个接口的状态, 并据此将活动网关的优先级降低一个默认 10 的值, 亦可指定该降低值。同样参考图34.14, 如果在 Switch 1 上采用默认值配置了HSRP接口跟踪, 那么就令到该交换机对接口 Gigabitethernet5/1 的状态进行跟踪, 在那个接口失效后, Switch 1 就会将其该HSRP组的优先级降低 10 ,得到一个 95 的优先级。

又假设 Switch 2 上配置了抢占 ( preempt ), 在此情形下是强制性要配置的, 那么它就会注意到自己有着更高的优先级 ( 100 比 95 ), 就会执行一次篡位, 结果该HSRP组的活动网关角色。

**真实场景应用** 在生产网络中, 思科Catalyst交换机还支持增强对象跟踪 (Enhanced Object Tracking, EOT) 功能, 可用于所有FHRP (也就是HSRP、VRRP及GLBP) 上。增强对象跟踪功能令到管理员可以将交换机配置为对以下参数进行跟踪:

- 某个接口的IP路由状态, The IP routing state of an interface
- IP路由的可达性, IP route reachability
- IP路由度量值阈值, The threshold of IP route metrics
- IP SLA 的运作, IP SLA operations([Service-Level Agreements](#), 服务等级协议)

对于这些FHRPs, 比如HSRP, 可被配置为对这些增强对象进行跟踪, 以令到在部署FHRP失效情形时具有更大的灵活性。比如, 在采用EOT时, 可将活动HSRP路由器配置为在网络或主机路由不可达时 (也就是出现在路由表中), 降低其优先级某个数值。EOT功能是超出了CCNA考试要求的, 在配置示例中不会涉及。

## HSRP的负载均衡

HSRP允许管理员在一些物理接口上配置多个HSRP组，以实现负载均衡。默认情况下，在两台网关之间配置HSRP时，在任何时期都只有一台网关对那个组的流量进行转发。这样就导致了备份网关链路上带宽的浪费。这在下图34.15中进行了演示：

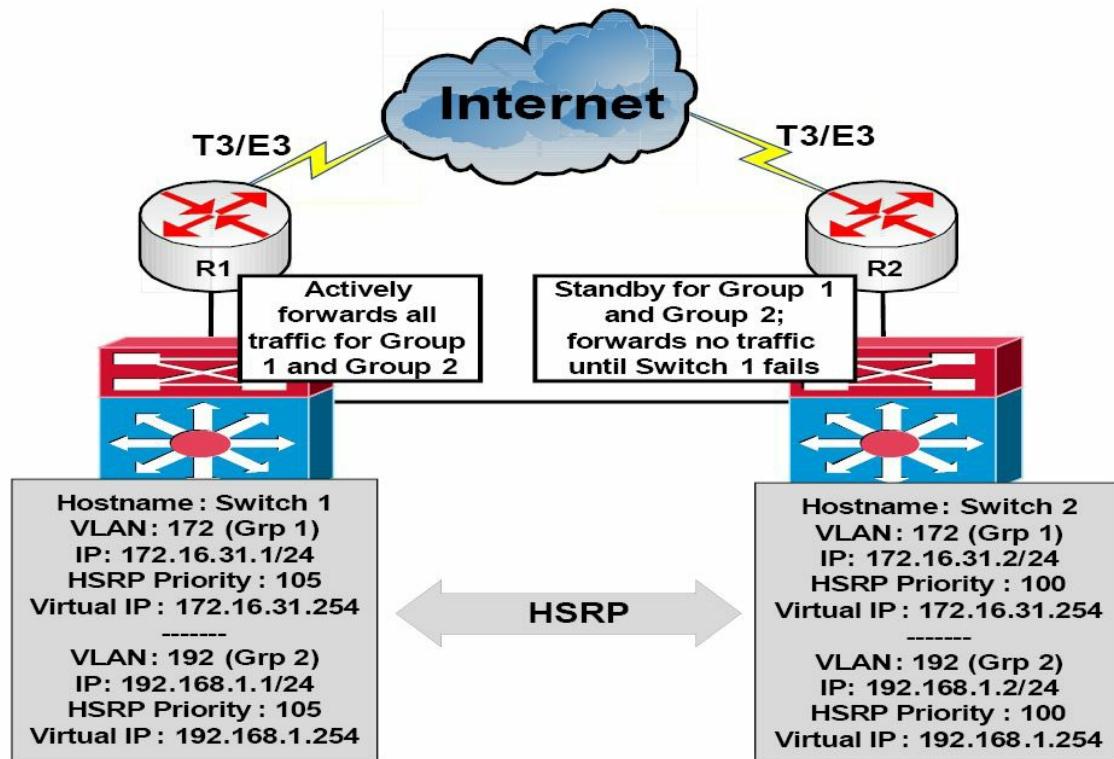


图 34.15 -- 不具备 HSRP 负载均衡的一个网络

在图34.15中，在 Switch 1 和 Switch 2 上配置了两个HSRP组。Switch 1 已被配置为两个组的活动（主）网关--这是基于其有着较高的优先级值。Switch 1 与 switch 2 都相应的连接到了路由器 R1 和 R2 上。这两台路由器都通过各自的 T3/E3 线路，连接到互联网。因为 Switch 1 是两个HSRP组的活动网关，它就会转发两个组的流量，直到其失效后， Switch 2 才会结果活动（主）网关的角色。

尽管这样做满足了网络的冗余需求，但也造成 R2 上昂贵的 T3/E3 线路的空闲，除非在 Switch 2 成为活动网关并开始经由它来转发流量。自然，这就出现了一定数量带宽的浪费。

而通过配置多个HSRP组，每个组使用不同的活动网关，管理员就可以有效的防止不必要的资源浪费，并在 Switch 1 与 Switch 2 之间实现负载均衡。这在下图34.16中进行了演示：

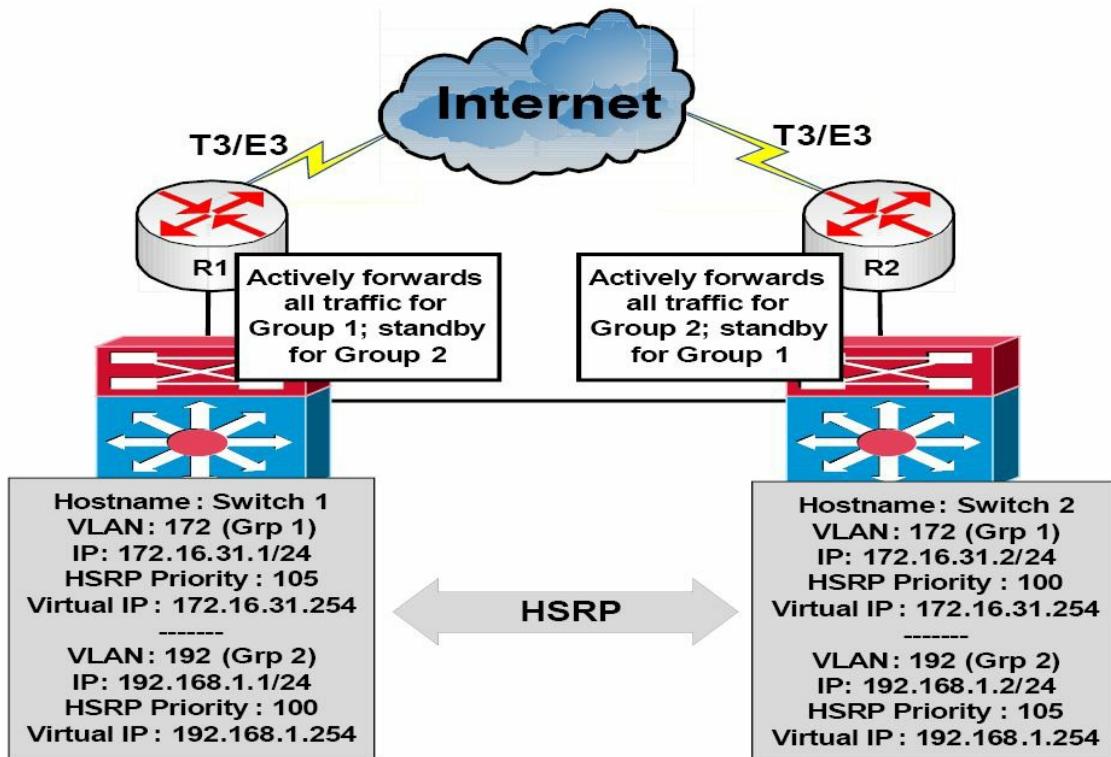


图 34.16 -- 一个采用 HSRP 实现负载均衡的网络

这里通过将 `Switch 1` 配置为 HSRP `Group 1` 的活动网关，将 `Switch 2` 配置为 HSRP 组 `2` 的活动网关，管理员就令到来自两个不同组的流量，在 `Switch 1` 与 `Switch 2` 之间实现了负载均衡，并最终通过这两条专用 `T3/E3` 广域网连接。同时每台交换机又互为对方 HSRP 组的备份。比如在 `Switch 2` 失效时，`Switch 1` 就将接过 HSRP 组 `2` 活动网关的角色，相反亦然。

#### 真实世界的部署

在生产网络中，需要记住多个 HSRP 组的建立，会造成网关上 CPU 使用率的上升，以及有 HSRP 报文交换所造成的网络带宽占用的增加。诸如 Catalyst 4500 及 6500 系列的思科 Catalyst 交换机，提供了对 HSRP 客户组（HSRP client groups）的支持。

在前面的小节中，了解到 HSRP 允许在单个的网关接口上配置多个 HSRP 组。而在网关接口上允许许多不同 HSRP 组的主要问题，就是这样做会导致网关上 CPU 使用率的上升，并也因为 HSRP 每隔 3 秒的 Hello 数据包，而潜在可能增加网络流量。

为解决这个潜在的问题，HSRP 就还允许客户或从组的配置（the configuration of client or slave groups）。这些组是一些简单的 HSRP 组，它们跟随某个主 HSRP 组（a master HSRP group），而不参与 HSRP 选举。这些客户或从组跟随主组的允许与状态，因此它们本身无需周期性地交换 Hello 数据包。这样在运用多个 HSRP 组时，降低 CPU 与网络的使用。

但是，为了刷新那些交换机的虚拟 MAC 地址，这些客户组仍然要发送周期性的报文。不过与主组的协议选举报文相比，这些刷新报文是以低得多的频率发送的。尽管 HSRP 客户组的配置是超出 CCNA 考试要求的，下面的输出还是演示两个客户组的配置，这两个客户组被配置为跟随主组 HSRP `Group 1`，该主组又被命名为 `SWITCH-HSRP` 组：

```

Gateway-1(config)#interface vlan100
Gateway-1(config-if)#ip address 192.168.1.1 255.255.255.0
Gateway-1(config-if)#ip address 172.16.31.1 255.255.255.0 secondary
Gateway-1(config-if)#ip address 10.100.10.1 255.255.255.0 secondary
Gateway-1(config-if)#standby 1 ip 192.168.1.254
Gateway-1(config-if)#standby 1 name SWITCH-HSRP
Gateway-1(config-if)#standby 2 ip 172.16.31.254
Gateway-1(config-if)#standby 2 follow SWITCH-HSRP
Gateway-1(config-if)#standby 3 ip 10.100.10.254
Gateway-1(config-if)#standby 3 follow SWITCH-HSRP
Gateway-1(config-if)#exit

```

在上面的输出配置中， Group 1 被配置为了主HSRP组， 同时 Group 2 与 Group 3 被配置为了客户组或叫做从组。

## 网关上HSRP的配置

在网关上配置HSRP，需要完成以下步骤：

1. 使用接口配置命令 `ip address [address] [mask] [secondary]` 配置网关接口的IP地址及掩码。
2. 通过接口配置命令 `standby [number] ip [virtual address] [secondary]`，在网关接口上建立一个HSRP组，以及给该HSRP组指派虚拟IP地址。关键词（keyword）`[secondary]` 将该IP地址指定为指定组的次网关IP地址。
3. 这里作为可选项，使用接口配置命令 `standby [number] name [name]`，为HSRP组指派一个名称。
4. 作为可选项，如打算对活动网关的选举施加影响，就要经由接口配置命令 `standby [number] priority [value]`，对组优先级进行配置。

本章中的后续HSRP配置输出，将建立在下图34.17中的网络：

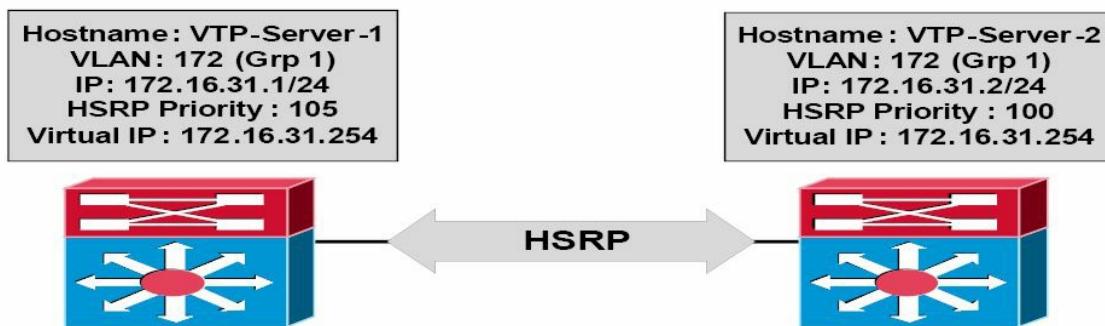


图 34.17 -- HSRP示例配置的拓扑

**注意：**这里假定在 VTP-Server-1 与 VTP-Server-2 之间的VLAN与中继已有配置妥当，同时交换机之间可以经由VLAN172 ping 通。为简短起见，这些配置已在配置示例中省略。

```

VTP-Server-1(config)#interface vlan172
VTP-Server-1(config-if)#ip address 172.16.31.1 255.255.255.0
VTP-Server-1(config-if)#standby 1 ip 172.16.31.254
VTP-Server-1(config-if)#standby 1 priority 105
VTP-Server-1(config-if)#exit
VTP-Server-2(config)#interface vlan172
VTP-Server-2(config-if)#ip address 172.16.31.2 255.255.255.0
VTP-Server-2(config-if)#standby 1 ip 172.16.31.254
VTP-Server-2(config-if)#exit

```

**注意：**这里应用到 VTP-Server-2 的HSRP配置并没有手动指派优先级数值。默认情况下，HSRP将使用一个 100 的优先级值，以允许带有优先级值 105 的 VTP-Server-1，在选举中胜选，从而被选举为该HSRP组的主网关。

在配置应用后，就可使用 `show standby [interface brief]` 命令，对HSRP的配置进行验证。下面的输出对 `show standby brief` 命令进行了展示：

```
VTP-Server-1#show standby brief
    P indicates configured to preempt.
    |
Interface  Grp      Pri  P State   Active   Standby       Virtual IP
Vl172        1        105  Active   local    172.16.31.2    172.16.31.254
VTP-Server-2#show standby brief
    P indicates configured to preempt.
    |
Interface  Grp      Pri  P State   Active   Standby       Virtual IP
Vl172        1        100  Standby  local    172.16.31.1    172.16.31.254
```

基于此种配置，只有在 VTP-Server-1 失效时， VTP-Server-2 才会成为活动网关。此外，因为没有配置抢占（preemption），那么即使在 VTP-Server-1 重新上线时，就算在该HSRP组中，其比起 VTP-Server-2 有着更高的优先级，它仍然无法强制性地接过活动网关角色。

## HSRP抢占的配置

### Configuring HSRP Preemption

抢占特性令到某台网关在本身比当前活动网关有着更高优先级时，强制性地接过活动网关的角色。使用命令 `standby [number] preempt` 命令，来配置HSRP抢占特性。下面的输出，演示了在 VTP-Server-1 上的此项配置：

```
VTP-Server-1(config)#interface vlan172
VTP-Server-1(config-if)#standby 1 preempt
```

这里同样使用命令 `show standby [interface [name] |brief]`，来验证在某个网关上已有配置抢占特性。是通过下面的 `show standby brief` 命令输出中的“P”字样演示的：

```
VTP-Server-1#show standby brief
    P indicates configured to preempt.
    |
Interface  Grp Pri  P State   Active   Standby       Virtual IP
Vl172        1   105  P Active  local    172.16.31.2    172.16.31.254
```

有了这个修改，在因 VTP-Server-1 失效而导致 VTP-Server-2 接过VLAN172的活动网关角色时，一旦 VTP-Server-1 再度上线，其就将强制性再度接手那个角色。在配置抢占特性时，思科IOS软件允许指定在交换机抢占及强制重新获得活动网关角色之前的时间间隔。

默认下抢占是立即发生的。但可使用接口配置命令 `standby [number] preempt delay [minimum|reload|sync]` 对此时间间隔进行修改。关键字 `[minimum]` 用于指定在抢占前等待的最短时间（秒）。下面的输出展示了如何配置在抢占前等待30秒钟：

```
VTP-Server-1(config)#interface vlan172
VTP-Server-1(config-if)#standby 1 preempt delay minimum 30
```

此配置可使用命令 `show standby [interface]` 进行验证。下面的输出对此进行了演示：

```
VTP-Server-1#show standby vlan172
Vlan172 - Group 1
  State is Active
    5 state changes, last state change 00:00:32
  Virtual IP address is 172.16.31.254
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 0.636 secs
  Preemption enabled, delay min 30 secs
  Active router is local
  Standby router is 172.16.31.2, priority 100 (expires in 8.629 sec)
  Priority 105 (configured 105)
  IP redundancy name is "hsrp-Vl172-1" (default)
```

而关键字 [reload] 用于指定网关在其重启后需要等待的时间 (the [reload] keyword is used to specify the amount of time the gateway should wait after it initiates following a reload)。关键字 [sync] 是与IP冗余客户端配合使用的。此配置超出了CCNA考试要求，但在生产环境中是十分有用的，因为在出现某个正在被跟踪的抖动接口，或类似情况下，此配置可以阻止不必要的角色切换 (this configuration is beyond the scope of the CCNA exam requirements but is very useful in production environments because it prevents an unnecessary change of roles in the case of a flapping interface that is being tracked, or similar activity)。

## 配置HSRP接口跟踪

HSRP接口跟踪特性，令到管理员可以将HSRP配置为追踪接口状态，从而将当前优先级降低一个默认数值（10）或指定数值，以允许另一网关接过指定HSRP组的主网关角色。

在下面的输出中， VTP-Server-1 被配置为对连接到假想WAN路由器的接口 Gigabitethernet5/1 的状态，进行跟踪。在那个接口状态转变为 down 时，该网关就将其优先级值降低10（默认的）：

```
VTP-Server-1#show standby vlan172
Vlan172 - Group 1
  State is Active
    5 state changes, last state change 00:33:22
  Virtual IP address is 172.16.31.254
  Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
  Hello time 3 sec, hold time 10 sec
    Next hello sent in 1.085 secs
  Preemption enabled
  Active router is local
  Standby router is 172.16.31.2, priority 100 (expires in 7.616 sec)
  Priority 105 (configured 105)
  IP redundancy name is "hsrp-Vl172-1" (default)
  Priority tracking 1 interfaces or objects, 1 up:
  Interface or object      Decrement      State
  GigabitEthernet5/1          10            Up
```

而要将该网关降低值配置为比如50，就可以执行命令 standby [name] track [interface] [decrement value]，如下面的输出所示：

```
VTP-Server-1(config)#interface vlan172
VTP-Server-1(config-if)#standby 1 track GigabitEthernet5/1 50
```

此项配置可使用命令 show standby [interface] 进行验证。下面对此进行了演示：

```
VTP-Server-1#show standby vlan172
Vlan172 - Group 1
  State is Active
    5 state changes, last state change 00:33:22
    Virtual IP address is 172.16.31.254
    Active virtual MAC address is 0000.0c07.ac01
    Local virtual MAC address is 0000.0c07.ac01 (v1 default)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 1.085 secs
    Preemption enabled
    Active router is local
    Standby router is 172.16.31.2, priority 100 (expires in 7.616 sec)
    Priority 105 (configured 105)
    IP redundancy name is "hsrp-Vl172-1" (default)
    Priority tracking 1 interfaces or objects, 1 up:
    Interface or object      Decrement   State
    GigabitEthernet5/1        50          Up
```

## 配置HSRP的版本

如同在本课程模块先前指出的那样， 默认当HSRP开启时， 是启用的版本1。但可通过接口配置命令 `standby version [1|2]` 来手动开启HSRP版本2。下面的输出演示了HSRP版本2的配置：

```
VTP-Server-1(config)#interface vlan172
VTP-Server-1(config-if)#standby version 2
```

使用命令 `show standby [interface]`， 可对此配置进行验证。下面的输出对此进行了演示：

```
VTP-Server-1#show standby vlan172
Vlan172 - Group 1 (version 2)
  State is Active
    5 state changes, last state change 00:43:42
    Virtual IP address is 172.16.31.254
    Active virtual MAC address is 0000.0c9f.f001
      Local virtual MAC address is 0000.0c9f.f001 (v2 default)
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 2.419 secs
    Preemption enabled
    Active router is local
    Standby router is 172.16.31.2, priority 100 (expires in 4.402 sec)
    Priority 105 (configured 105)
    IP redundancy name is "hsrp-Vl172-1" (default)
```

而HSRP的开启，就自动将HSRP所使用的MAC地址范围，从 `0000.0C07.ACXX`， 改变为 `0000.0C9F.F000` 到 `0000.0C9F.FFFF`。因此务必要记住这将导致生产网络中的一些数据包丢失，因为网络中的设备必须要掌握到网关的新MAC地址。这类导致包丢失的变动，都推荐在维护窗口或几乎的断网窗口来进行。

## 虚拟路由器冗余协议

### Virtual Router Redundancy Protocol

虚拟路由器冗余协议（Virtual Router Redundancy Protocol, VRRP），是一个动态地将一个或多个网关的职责，指派给LAN上的VRRP路由器的网关选举协议（a gateway election protocol），其令到在诸如以太网这样的某个多路访问网段（a Multi-Access segment）上的数台路由器，能够使用同一个虚拟IP地址，作为它们的默认网关。

VRRP以与HSRP类似的方式运作；但与HSRP不同，VRRP是一个定义在RFC 2338中的开放标准，RFC 2338在[RFC 3768](#)中被废弃。VRRP将通告发送到多播目的地址 224.0.0.18（VRRP），使用的是IP协议编号 112。在数据链路层，通告是从主虚拟路由器（the master virtual router）的MAC地址 00-00-5e-00-01-xx发出的，这里的"xx"表示了两位十六进制的组编号。这在下图34.18中进行了演示：

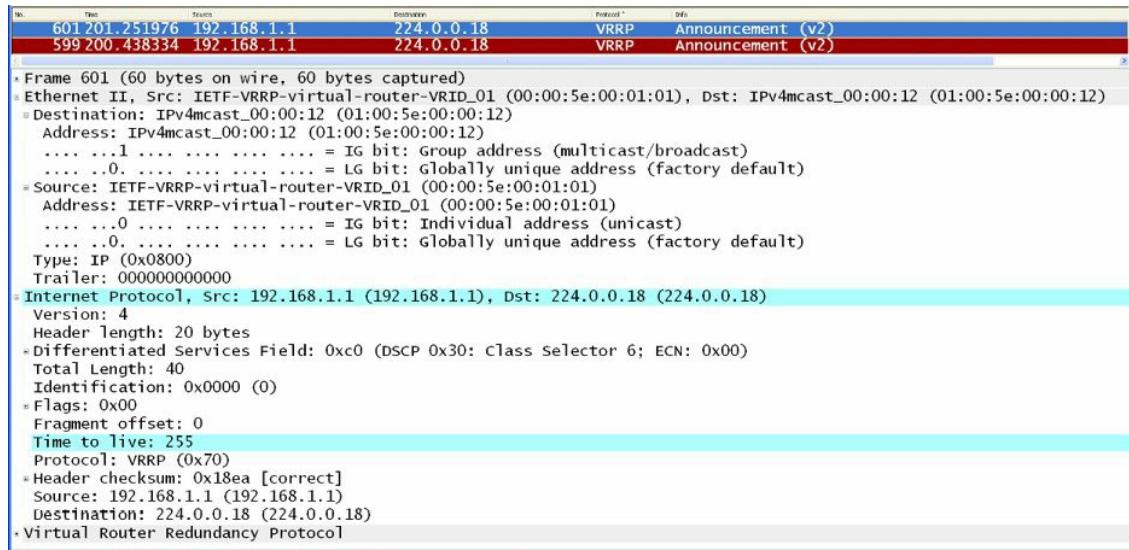


图 34.18 -- VRRP 的多播地址，VRRP Multicast Addresses

**注意：**这里的协议编号是十六进制形式的。而十六进制值 `0x70` 就等于是进制的112。与此类似，数据链路层目的地址 `01-00-5e-00-00-12` 中的十六进制值 `12` 就是十进制值18（也就是 `224.0.0.18`）了。如你仍对这些数值是如何转换的没有掌握，那么本CCNA手册的十六进制到十进制转换在网上是很详细的。

### 真是世界的部署

与HSRP不同，VRRP并没有允许网关使用出厂地址（Burnt-in Address, BIA）或静态配置的地址作为VRRP组的MAC地址的选项。因此，在带有多于VRRP组的生产网络中，对在某个特定接口上应用多个MAC地址的理解掌握，尤其是当部署了诸如端口安全这样的特性时，是重要的。记得要着重于整体上；否则就会发现，尽管有正确配置，一些特性或协议也不会如预期那样跑起来。

一个VRRP网关是在一台或多台连接到LAN的路由器上，配置用于运行VRRP协议的（A VRRP gateway is configured to run the VRRP protocol in conjunction with one or more other routers attached to a LAN）。在VRRP配置中，一台网关被选举为主虚拟路由器（the master router），而其它网关则扮演在主虚拟路由器失效时的备份虚拟路由器。下图34.19对此概念进行了演示：

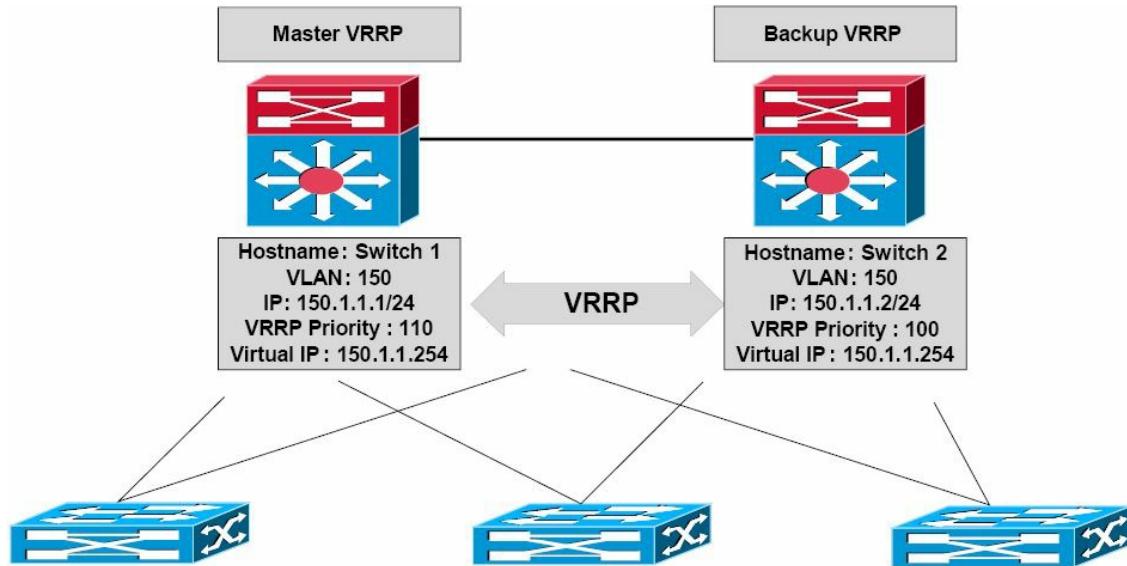


图 34.19 -- VRRP 的基本运作

## VRRP的多虚拟路由器支持特性

可在某个接口上配置多大255个的虚拟路由器。而某个路由器接口实际能支持的虚拟路由器数目，由以下因素决定：

- 路由器的处理能力，Router processing capability
- 路由器的内存容量，Router memory capability
- 路由器接口对多MAC地址的支持情况，Router interface support of multiple MAC addresses

## VRRP的主路由器选举

### VRRP Master Router Election

VRRP默认使用优先级值来决定哪台路由器将被选举为主虚拟路由器。默认的VRRP优先级值为100; 但此数值可被手工修改为一个到254之间的数值。而如多台网关有着相同的优先级数值，那么有着最高IP地址的网关将被选举为主虚拟路由器，同时有着较低IP地址的那台就成为备份虚拟路由器。

加入有多于两台的路由器被配置为VRRP组的组成部分，那么备份虚拟路由器中有着第二高优先级的，就会在当前主虚拟路由器失效或不可用时，被选举为主虚拟路由器。又假如那些备份虚拟路由器又有着相同的优先级，那么这些备份路由器中有着最高IP地址的那台，将被选举为主路由器。下图34.20对此概念进行了演示：

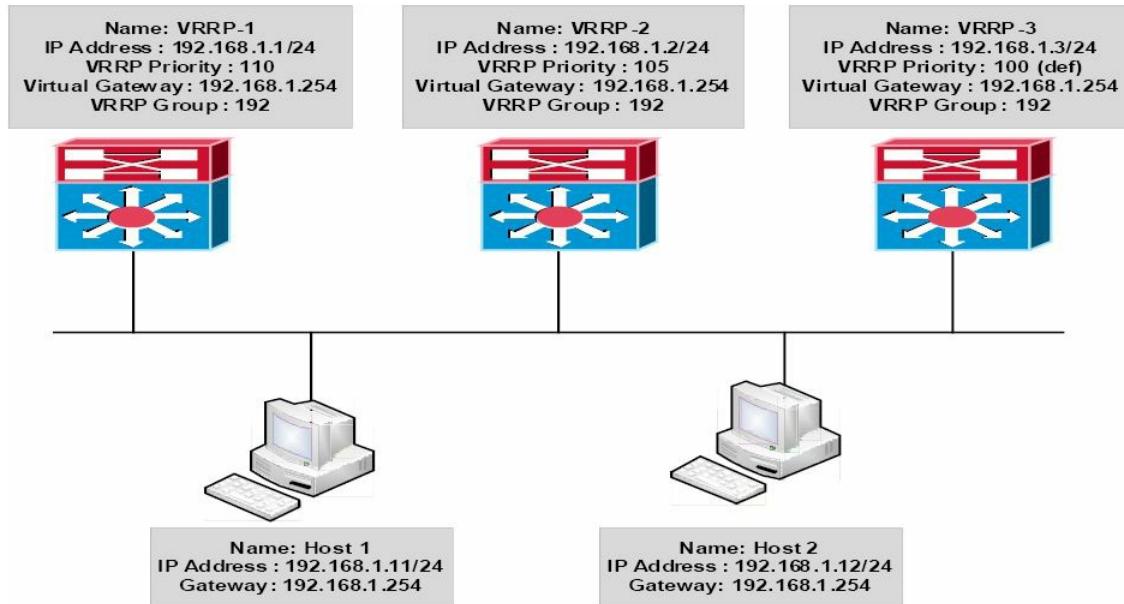


图 34.20 -- VRRP 主虚拟路由器及备份虚拟路由器的选举

图34.20演示了一个采用了VRRP作为网关冗余的网络。主机1与主机2都配置了默认 192.168.1.254 作为默认网关，此网关就是配置在交换机 VRRP-1、VRRP-2 及 VRRP-3 上，给VRRP group 192 的虚拟IP地址。

交换机 VRRP-1 已被配置了优先级值110, VRRP-2 的是105, VRRP-3 的是默认VRRP优先级100。基于此种配置， VRRP-1 就被选举为主虚拟路由器，同时 VRRP-2 和 VRRP-3 就成为备份虚拟路由器。

在 VRRP-1 失效时，因为 VRRP-2 有着比起 VRRP-3 更高的优先级，所以它就成为主虚拟路由器。但如果 VRRP-2 与 VRRP-3 有着相同优先级的话， VRRP-3 将被选举为主虚拟路由器，因为它有着更高的IP地址。

## VRRP的抢占

与HSRP不同，VRRP的抢占特性是默认开启的，因此无需管理员为开启此功能而进行显式的配置。但此功能可经由使用接口配置命令 `no vrrp [number] preempt` 进行关闭。

## VRRP的负载均衡

VRRP允许以与HSRP类似的方式，实现负载均衡。比如，在一个于某台网关上配置了多个虚拟路由器（VRRP组）的网络中，一个接口可作为某个虚拟路由器（VRRP组）的主接口（虚拟路由器），同时又可作为另一或更多虚拟路由器（VRRP组）的备份（虚拟路由器）。下图34.21对此进行了演示：

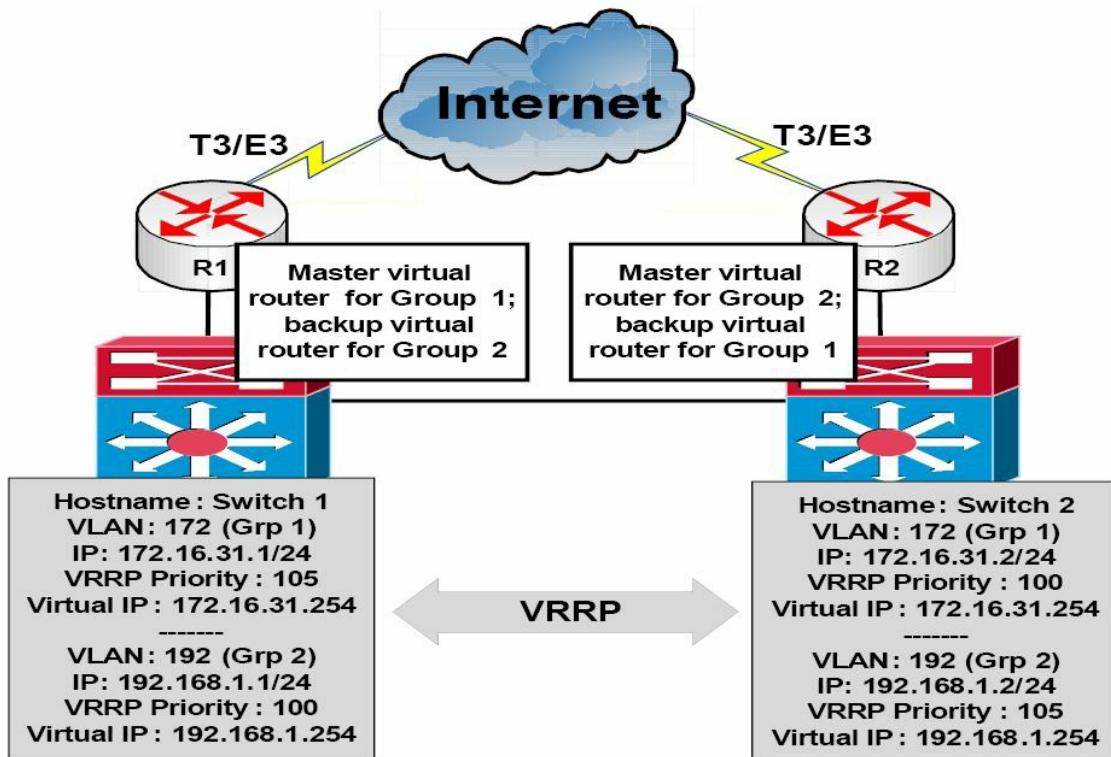


图 34.21 -- VRRP 的负载均衡

## VRRP的版本

默认情况下，当在某台运行思科IOS软件的网关上配置了VRRP时，开启的是VRRP版本2（见下图）。版本2正是默认的以及当前的VRRP版本。这里并不能如同在HSRP中那样改变版本，因为并没有VRRP版本1的标准。

**注意:** 在本手册编写过程中，为IPv4与IPv6定义VRRP的版本3，正处于草案阶段，且并未标准化。

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.1	224.0.0.18	VRRP	Advertisement
2	0.965574	192.168.1.1	224.0.0.18	VRRP	Advertisement
5	1.927097	192.168.1.1	224.0.0.18	VRRP	Advertisement
7	2.876659	192.168.1.1	224.0.0.18	VRRP	Advertisement
8	3.826235	192.168.1.1	224.0.0.18	VRRP	Advertisement

```

- Virtual Router Redundancy Protocol
- Version 2, Packet type 1 (Advertisement)
  0010 .... = VRRP protocol version: 2
  .... 0001 = VRRP packet type: Advertisement (1)
  Virtual Rtr ID: 1
  Priority: 110 (Non-default backup priority)
  Count IP Addrs: 1
  Auth Type: No Authentication (0)
  Adver Int: 1
  Checksum: 0xae55 [correct]
  IP Address: 192.168.1.254 (192.168.1.254)

```

图 34.22 -- VRRP 版本2的数据包

## VRRP的各种通告

### VRRP Advertisements

主虚拟路由器将通告发送给同一VRRP组中的其它VRRP路由器。通告就主虚拟路由器的优先级与状态进行通信。VRRP的通告是以IP数据包进行封装的，并被发送到在图34.18中所演示的那个指派给该VRRP组的IPv4多播地址。通告默认以每秒的频率发送；不过此时间间隔是可被用户配置的，因而可以改变。同时备份虚拟路由器收听主虚拟路由器通告的间隔，也可进行配置。

## 在网关上配置VRRP

在网关上配置VRRP，需要以下步骤：

1. 使用接口配置命令 `ip address [address] [mask] [secondary]`，给网关接口配置正确的IP地址与子网掩码。
2. 通过接口配置命令 `vrrp [number] ip [virtual address] [secondary]`，在该网关接口上建立一个VRRP组，并为其指派一个虚拟IP地址。关键字 `[secondary]` 将该虚拟IP地址配置为指定VRRP组的次网关地址。
3. 作为可选项，使用接口配置命令 `vrrp [number] description [name]`，为该VRRP组指派一个描述性名称。
4. 作为可选项，在打算对主虚拟路由器及备份虚拟路由器的选举进行控制时，就要经由接口配置命令 `vrrp [number] priority [value]`，对该组的优先级进行配置。

本小节的VRRP配置输出，将基于下图34.23的网络：

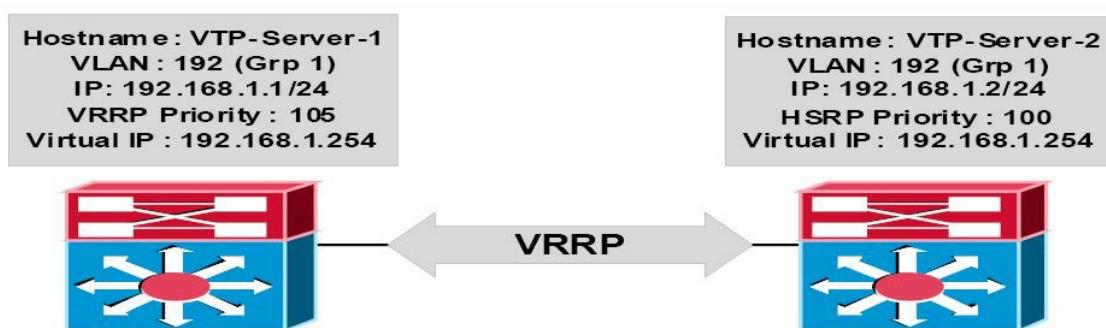


图 34.23 -- VRRP 配置示例的拓扑

**注意：**这里假定在 VTP-Server-1 与 VTP-Server-2 之间的VLAN与中继已有配置妥当，同时交换机之间可以经由VLAN192 ping 通。为简短起见，这些配置已在配置示例中省略。

```
VTP-Server-1(config)#interface vlan192
VTP-Server-1(config-if)#ip address 192.168.1.1 255.255.255.0
VTP-Server-1(config-if)#vrrp 1 ip 192.168.1.254
VTP-Server-1(config-if)#vrrp 1 priority 105
VTP-Server-1(config-if)#vrrp 1 description 'SWITCH-VRRP-Example'
VTP-Server-1(config-if)#exit
VTP-Server-2(config)#interface vlan192
VTP-Server-2(config-if)#ip address 192.168.1.2 255.255.255.0
VTP-Server-2(config-if)#vrrp 1 ip 192.168.1.254
VTP-Server-2(config-if)#vrrp 1 description 'SWITCH-VRRP-Example'
VTP-Server-2(config-if)#exit
```

**注意：**这里没有为 VTP-Server-2 上所应用的VRRP配置手动指派优先级数值。那么默认情况下，VRRP将使用100的优先级数值，这就令到带有优先级数值105的 VTP-Server-1，在选举中获胜而被选举为该VRRP组的主虚拟路由器。此外，这里还为该VRRP组配置了一个描述信息。

下面还使用命令 `show vrrp [all|brief|interface]`，对此配置进行了验证。关键字 `[all]` 展示了有关该 VRRP 配置的所有信息，包括了组的状态、描述信息（在配置了的情况下）、本地网关优先级，以及主虚拟路由器和其它信息。关键字 `[brief]` 则会列印出该 VRRP 配置的摘要信息。而 `[interface]` 关键字会列印出特定接口的 VRRP 信息。下面的输出展示了 `show vrrp all` 命令的输出：

```
VTP-Server-1#show vrrp all
Vlan192 - Group 1
' SWITCH-VRP-Example'
  State is Master
    Virtual IP address is 192.168.1.254
    Virtual MAC address is 0000.5e00.0101
    Advertisement interval is 1.000 sec
    Preemption enabled
    Priority is 105
    Master Router is 192.168.1.1 (local), priority is 105
    Master Advertisement interval is 1.000 sec
    Master Down interval is 3.589 sec
VTP-Server-2#show vrrp all
Vlan192 - Group 1
' SWITCH-VRP-Example'
  State is Backup
    Virtual IP address is 192.168.1.254
    Virtual MAC address is 0000.5e00.0101
    Advertisement interval is 1.000 sec
    Preemption enabled
    Priority is 100
    Master Router is 192.168.1.1, priority is 105
    Master Advertisement interval is 1.000 sec
    Master Down interval is 3.609 sec (expires in 3.328 sec)
```

下面的输出展示了由命令 `show vrrp brief` 所列印出的信息：

```
VTP-Server-1#show vrrp brief
Interface      Grp Pri Time Own Pre State   Master addr   Group addr
Vl192          1   105 3589     Y  Master   192.168.1.1   192.168.1.254
VTP-Server-2#show vrrp brief
Interface      Grp Pri Time Own Pre State   Master addr   Group addr
Vl192          1   100 3609     Y  Backup   192.168.1.1   192.168.1.254
```

## 配置 VRRP 的接口跟踪特性

为将 VRRP 配置为对某个接口进行跟踪，就必须要在全局配置模式下，为接口跟踪而使用全局配置命令 `track [object number] interface [line-protocol|ip routing]`，或为 IP 前缀追踪而使用全局配置命令 `track [object number] ip route [address | prefix] [reachability | metric threshold]`，建立一个被跟踪的对象。依据软件与平台的不同，交换机上可对高达 500 个的被跟踪对象进行跟踪。随后再使用接口配置命令 `vrrp [number] track [object]`，实现 VRRP 对被跟踪对象的跟踪。

**注意：** CCNA 考试不要求完成这些高级对象追踪的配置。

下面的输出展示了如何配置 VRRP 的跟踪，引用了对象 1，该被跟踪对象对 `Loopback0` 接口的线路协议进行跟踪：

```
VTP-Server-1(config)#track 1 interface Loopback0 line-protocol
VTP-Server-1(config-track)#exit
VTP-Server-1(config)#interface vlan192
VTP-Server-1(config-if)#vrrp 1 track 1
VTP-Server-1(config-if)#exit
```

而下面的输出则展示了如何将VRRP配置为对引用对象2的追踪，此被追踪对象追踪了到前缀 1.1.1.1/32 的可达性。一个被追踪的IP路由对象在存在一个该路由的路由表条目时，被认为是在线且可达的，同时该路由不是无法访问的（无法访问就是说有着255的路由度量值），当发生无法访问时，该路由就会从路由信息数据库中被移除（a tracked IP route object is considered to be up and reachable when a routing table entry exists for the route and the route is not accessible(i.e., has a route metric of 255)，in which case the route is removed from the Routing Information Base(RIB) anyway）。

```
VTP-Server-1(config)#track 2 ip route 1.1.1.1/32 reachability
VTP-Server-1(config-track)#exit
VTP-Server-1(config)#interface vlan192
VTP-Server-1(config-if)#vrrp 1 track 2
```

VRRP跟踪的配置，是通过使用命令 `show vrrp interface [name]` 命令进行验证的。下面的输出对此进行了演示：

```
VTP-Server-1#show vrrp interface vlan192
Vlan192 - Group 1
'SWITCH-VRRP-Example'
  State is Master
  Virtual IP address is 192.168.1.254
  Virtual MAC address is 0000.5e00.0101
  Advertisement interval is 0.100 sec
  Preemption enabled
  Priority is 105
    Track object 1 state Up decrement 10
    Track object 2 state Up decrement 10
  Authentication MD5, key-string
  Master Router is 192.168.1.1 (local), priority is 105
  Master Advertisement interval is 0.100 sec
  Master Down interval is 0.889 sec
```

而要查看被追踪对象的各项参数，就使用命令 `show track [number] [brief] [interface] [ip] [resolution] [timers]`。下面是 `show track` 命令输出的演示：

```
VTP-Server-1#show track
Track 1
  Interface Loopback0 line-protocol
  Line protocol is Up
    1 change, last change 00:11:36
  Tracked by:
    VRRP Vlan192 1
Track 2
  IP route 1.1.1.1 255.255.255.255 reachability
  Reachability is Up (connected)
    1 change, last change 00:08:48
  First-hop interface is Loopback0
  Tracked by:
    VRRP Vlan192 1
```

**注意：**这些被追踪对象亦可与HSRP和GLBP配合使用。GLBP在下面的小节进行说明。

## VRRP的调试

命令 `debug vrrp` 提供给管理员用于查看有关VRRP运作情况实时信息的诸多选项。这些选项如下面的输出所示：

```
VTP-Server-1#debug vrrp ?
  all Debug all VRRP information
  auth VRRP authentication reporting
  errors VRRP error reporting
  events Protocol and Interface events
  packets VRRP packet details
  state VRRP state reporting
  track Monitor tracking
<cr>
```

## 网关负载均衡协议

### Gateway Load Balancing Protocol

与HSRP一样，网关负载均衡协议也是一种思科专有的协议。GLBP以与HSRP和VRRP类似的方式，提供了高的网络可用性。但与HSRP与VRRP在任何时候都由单一网关来转发特定组的流量不同，GLBP允许在同一GLBP组中的多台网关，同时进行流量的转发。

GLBP网关之间的通信，是通过以每隔3秒的频率，往多播地址 224.0.0.102 上，使用UDP端口3322发送Hello报文进行的。下图34.24对此进行了演示：

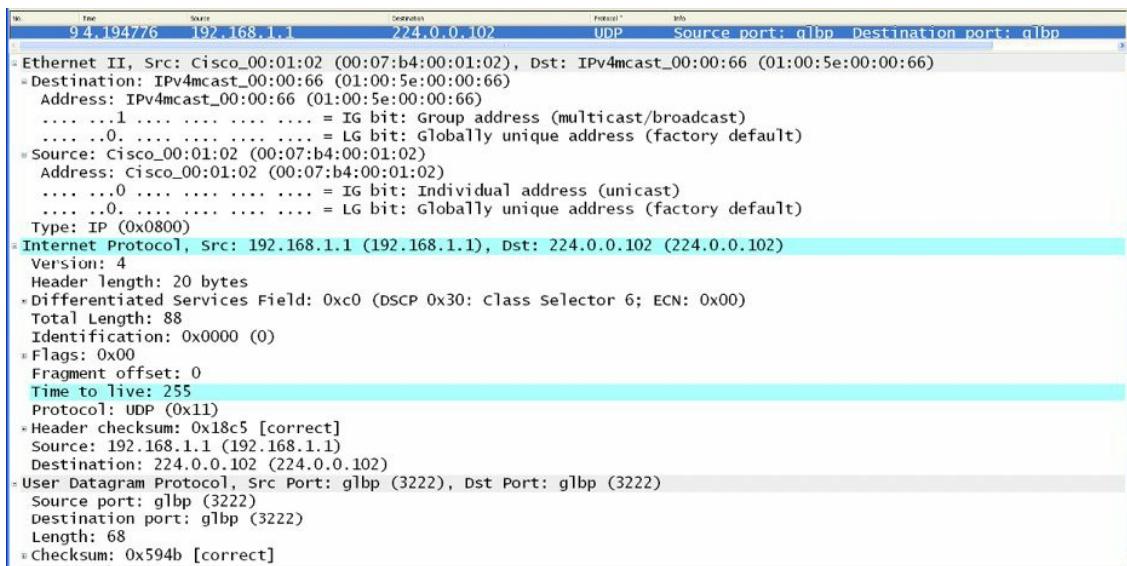


图 34.24 -- GLBP的三层及四层协议与地址, GLBP Layer 3 and Layer 4 Protocols and Addresses

### GLBP的运作

在启用了GLBP后，该GLBP组的那些成员就选举出一台网关，作为改组的活动虚拟网关（the active virtual gateway, AVG）。该活动网关有着最高的优先级值。在成员优先级值相等时，组中带有最高IP地址的活动虚拟网关将被选举为网关。组中剩下的其它网关，就会在活动虚拟网关不可用时，提供活动虚拟网关的备份。

活动虚拟网关将应答所有对虚拟路由器地址的地址解析协议（Address Resolution Protocol, ARP）请求。此外活动虚拟网关还会为GLBP组的每个成员网关，都分配一个虚拟MAC地址。因此每个成员网关都要负责转发发送到由活动虚拟网关所指派的虚拟MAC地址上的数据包了。这些网关一起，作为它们所分配到的虚拟MAC地址所对应的活动虚拟转发器（active virtual forwarders, AVFs）被看待。这就令到GLBP能够提供负载的共同承担。下图34.25对此概念进行了演示：

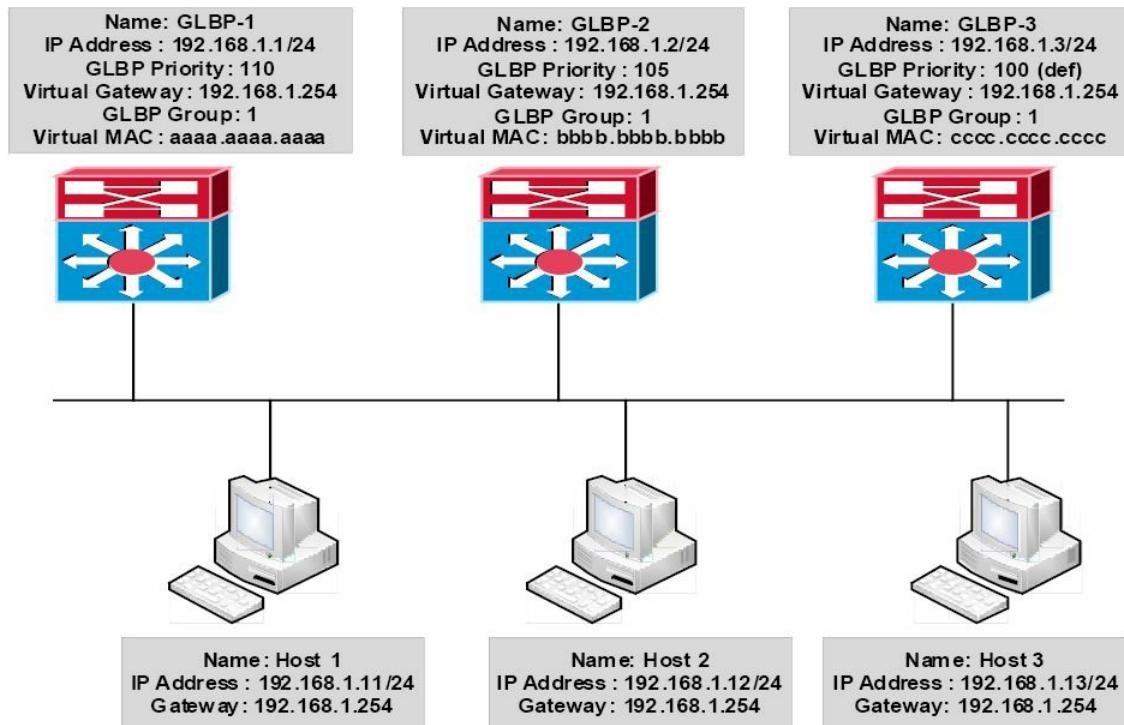


图 34.25 -- GLBP 的活动虚拟网关与活动虚拟转发器，GLBP Active Virtual Gateway and Active Virtual Forwarders

图34.25展示了使用GLBP作为FHRP的网络。这里的三台网关都被配置在GLBP Group 1 中。网关 GLBP-1 配置了110的优先级值，网关 GLBP-2 配置的优先级值是105，网关 GLBP-3 使用了默认的优先级值100。那么 GLBP-1 就被选举为活动虚拟网关，同时 GLBP-2 和 GLBP-3 又被分配到相应的虚拟MAC地址 bbbb.bbbb.bbbb.bbbb 及 cccc.cccc.cccc，且各自成为这些虚拟MAC地址对应的活动虚拟转发器。GLBP-1 也是其本身虚拟MAC地址 aaaa.aaaa.aaaa 的活动虚拟转发器。

主机1、2、3都配置了默认网关地址 192.168.1.254，此IP地址正是指派给该GLBP组的虚拟IP地址。主机1发出了查询其网关IP地址的ARP广播。此查询被活动虚拟网关（GLBP-1）接收到，GLBP-1 就以其自身的虚拟MAC地址 aaaa.aaaa.aaaa 加以响应。主机1于是就将到 192.168.1.254 的流量，转发到这个MAC地址了。

主机2发出一个查询其网关IP地址的ARP广播。此查询被活动虚拟网关（GLBP-1）接收，进而以虚拟MAC地址 bbbb.bbbb.bbbb 进行响应。那么主机2就将那些到 192.168.1.254 的流量，都转发到这个MAC地址了，并由 GLBP-2 来进一步转发这些流量。

主机3的情况与此类似，将会把到 192.168.1.254 的流量，转发到虚拟MAC地址 cccc.cccc.cccc，由 GLBP-3 来转发这些流量。

通过使用上组中的所有网关，GLBP实现了无需像在HSRP或VRRP中那样需要配置多个组，就能做到负载均衡。

## GLBP的虚拟MAC地址分配

一个GLBP允许每组有4个的虚拟MAC地址。由活动虚拟网关来负责将虚拟MAC地址分配给组中的各个成员。其它组成员是在它们发现了活动虚拟网关后，精油Hello报文，请求到虚拟MAC地址的。

这些网关是依序分配到下一个虚拟MAC地址的。已通过活动虚拟网关分配到了虚拟MAC地址的网关，被称作主虚拟转发器（a primary virtual forwarder），而已学习到某个虚拟MAC地址的网关，被称作是从虚拟转发器（a secondary virtual forwarder）。

## GLBP的冗余

在GLBP组中，是单一一台网关被选举为活动虚拟网关，有另一网关被选举为备份虚拟网关（the standby virtual gateway）的。组中剩下的其它网关，都被置于侦听状态（a Listen state）。在活动虚拟网关失效时，备份虚拟网关将接过该虚拟IP地址的角色。于此同时，又会再进行一次选举，此时将从那些处于侦听状态的网关中选出一个新的备份虚拟网关。

在该活动虚拟网关失效时，处于侦听状态的某台从虚拟转发器，会接过该虚拟MAC地址的职责。但是因为新的活动虚拟转发器已是使用了另一虚拟MAC地址的转发器，GLBP就需要确保原有的转发器MAC地址停止使用，同时那些主机已从此MAC地址迁移。这是通过使用下面的两个计时器实现的（in the event the AVF fails, one of the secondary virtual forwarders in the Listen state assumes responsibility for the virtual MAC address. However, because the new AVF is already a forwarder using another virtual MAC address, GLBP needs to ensure that the old forwarder MAC address ceases being used and hosts are migrated away from this address. This is achieved using the following two timers）：

- 重定向计时器，the redirect timer
- 超时计时器，the timeout timer

重定向时间是指在活动虚拟网关持续将主机重新到原有该虚拟转发器MAC地址的间隔。在此计时器超时后，活动虚拟网关就在ARP应答中停止使用原有的虚拟转发器MAC地址了，就算该虚拟转发器仍将发送到原有虚拟转发器MAC地址的数据包（the redirect time is the interval during which the AVG continues to redirect hosts to the old virtual forwarder MAC address. When this timer expires, the AVG stops using the old virtual forwarder MAC address in ARP replies, although the virtual forwarder will continue to forward packets that were sent to the old virtual forwarder MAC address）。

而在超时计时器超时后，该虚拟转发器就被从该GLBP组的所有网关中移除。那些仍在使用ARP缓存中原有MAC地址的客户端，就必须刷新此项项目，以获取到新的虚拟MAC地址。GLBP使用Hello报文，来就这两个计时器的当前状态进行通信（when the timeout timer expires, the virtual forwarder is removed from all gateways in the GLBP group. Any clients still using the old MAC address in their ARP caches must refresh the entry to obtain the new virtual MAC address. GLBP uses Hello messages to communicate the current state of these two timers）。

## GLBP的负载抢占

GLBP抢占默认是关闭的，也就是说仅在当前活动虚拟网关失效时，备份虚拟网关才能成为活动虚拟网关，这与分配给那些虚拟网关的优先级无关。这种运作方式，与HSRP中用到的类似。

思科IOS软件允许管理员开启GLBP的抢占特性，这就令到在备份虚拟网关被指派了一个比当前活动虚拟网关更高的优先级值时，成为活动虚拟网关。默认GLBP的虚拟转发器抢占性方案是开启的，有一个30秒的延迟（By default, the GLBP virtual forwarder preemptive scheme is enabled with a delay of 30 seconds）。但这个延迟可由管理员手动调整。

## GLBP的权重

### GLBP Weighting

GLBP采用了一种权重方案（a weighting scheme），来确定GLBP组中各台网关的转发容量。指派给GLBP组中某台网关的权重，可用于确定其是否要转发数据包，因此就可以依比例来确定该网关所要转发的LAN中主机的数据包了（the weighting assigned to a gateway in the GLBP group can be used to determine whether it will forward packets and, if so, the proportion of hosts in the LAN for which it will forward packets）。

每台网关都默认指派了100的权重。管理员可通过配置结合了GLBP的对象跟踪，比如接口及IP前缀跟踪，来进一步将网关配置为动态权重调整。在某个接口失效时，权重就被动态地降低一个指定数值，如此令到那些有着更高权重值的网关，用于转发比那些有着更低权重值的网关更多的流量。

此外，在某个GLBP组（成员）的权重降低到某个值时，还可设置一个阈值，用于关闭数据包的转发，且在权重上升到另一与之时，又可自动开启转发。在当前活动虚拟转发器的权重掉到低权重阈值30秒时，备份虚拟转发器将成为活动虚拟转发器。

## GLBP负载共同分担

### GLBP Load Sharing

GLBP支持以下三种方式的负载分担：

- 有赖于主机的， Host-dependent
- 轮转调度的， Round-robin
- 加权的， Weighted

在有赖于主机的负载共担下，生成虚拟路由器地址ARP请求的各台客户端，总是会在响应中收到同样的虚拟MAC地址。此方式为客户端提供了一致的网关MAC地址。

而轮询的负载共担机制，将流量平均地分发到组中作为活动虚拟转发器的所有网关（the round-robin load-sharing mechanism distributes the traffic evenly across all gateways participating as AVFs in the group）。这是默认的负载分担机制。

加权的负载分担机制，使用权重值来确定发送到某个特定AVF的流量比例。较高的权重值会带来更频繁的包含那台网关虚拟MAC地址的ARP响应。

## GLBP的客户端缓存

GLBP的客户端缓存，包含了使用到某个GLBP组作为默认网关的那些网络主机的信息。此缓存项目包含了关于发送了IPv4 ARP或IPv6 邻居发现（Neighbor Discovery, ND）请求主机，以及AVG指派了哪个转发器给它的信息，还有每台网络主机已被分配的GLBP转发器的编号，和当前分配给GLBP组中各台转发器的网络主机总数。

可以开启某个GLBP组的活动虚拟网关，来存储一个使用到此GLBP组的所有LAN客户端的客户端缓存数据库（a client cache database）。客户端缓存数据库最多可以存储2000个条目，但建议条目数不要超过1000。同时GLBP缓存的配置，是超出CCNA考试要求的，此特性可使用命令 `glbp client-cache` 进行配置，使用命令 `show glbp detail` 进行验证。

## 在网关上配置GLBP

在网关上配置GLBP，需要以下步骤：

1. 使用接口配置命令 `ip address [address] [mask] [secondary]`，为网关接口配置正确的IP地址与子网掩码。
2. 通过接口配置命令 `glbp [number] ip [virtual address] [secondary]`，在网关接口上建立一个GLBP组，并给该组指派上虚拟IP地址。关键字 `[secondary]` 将该虚拟IP地址配置为指定组的第二网关地址。
3. 作为可选项，可通过接口配置命令 `glbp [number] name [name]`，为该GLBP组指派一个名称。

4. 作为可选项，如打算对活动虚拟网关的选举进行控制，就要通过接口配置命令 `glbp [number] priority [value]`，配置该组的优先级。

本小节中的GLBP示例，将基于下图34.26的网络：

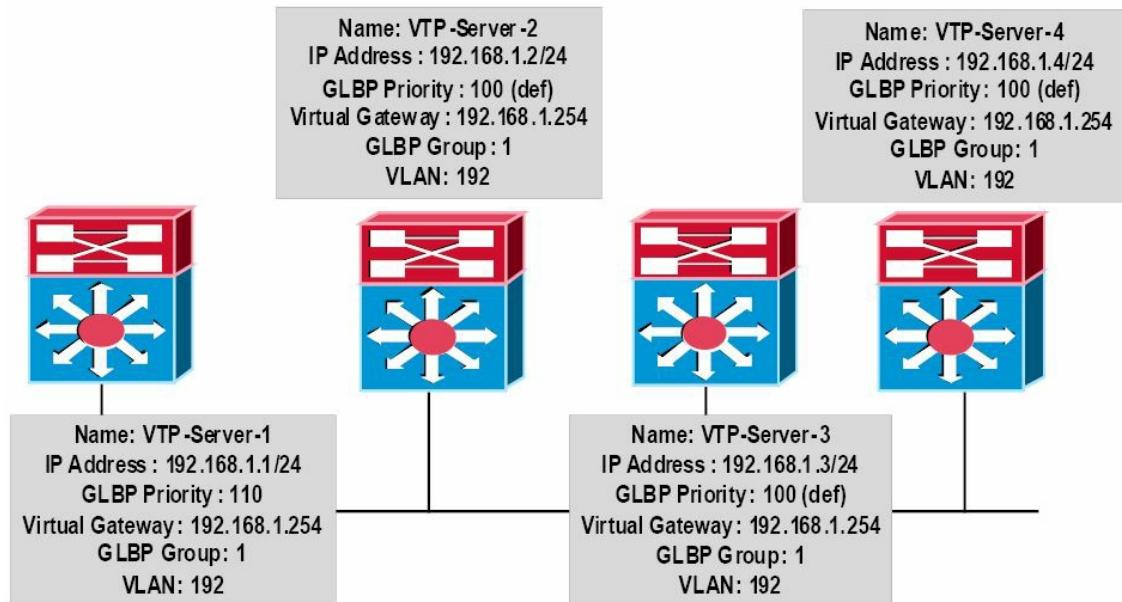


图 34.26 -- GLBP 配置示例的拓扑

**注意：**这里假定在 VTP-Server-1 与 VTP-Server-2 之间的VLAN与中继已有配置妥当，同时交换机之间可以经由VLAN192 ping 通。为简短起见，这些配置已在配置示例中省略。

```
VTP-Server-1(config)#interface vlan192
VTP-Server-1(config-if)#glbp 1 ip 192.168.1.254
VTP-Server-1(config-if)#glbp 1 priority 110
VTP-Server-1(config-if)#exit
VTP-Server-2(config)#interface vlan192
VTP-Server-2(config-if)#glbp 1 ip 192.168.1.254
VTP-Server-2(config-if)#exit
VTP-Server-3(config)#interface vlan192
VTP-Server-3(config-if)#glbp 1 ip 192.168.1.254
VTP-Server-3(config-if)#exit
VTP-Server-4(config)#interface vlan192
VTP-Server-4(config-if)#glbp 1 ip 192.168.1.254
VTP-Server-4(config-if)#exit
```

一旦该GLBP组已被配置，就可使用命令 `show glbp brief` 来查看该GLBP配置的摘要信息了，如同下面的输出所示：

```
VTP-Server-1#show glbp brief
Interface  Grp  Fwd Pri State      Address      Active router  Standby router
Vl192      1    -   110 Active     192.168.1.254 local        192.168.1.4
Vl192      1    1   - Active      0007.b400.0101 local        -
Vl192      1    2   - Listen     0007.b400.0102 192.168.1.2        -
Vl192      1    3   - Listen     0007.b400.0103 192.168.1.3        -
Vl192      1    4   - Listen     0007.b400.0104 192.168.1.4        -

VTP-Server-2#show glbp brief
Interface  Grp  Fwd Pri State      Address      Active router  Standby router
Vl192      1    -   100 Listen    192.168.1.254 192.168.1.1 192.168.1.4
Vl192      1    1   - Listen     0007.b400.0101 192.168.1.1        -
Vl192      1    2   - Active     0007.b400.0102 local        -
Vl192      1    3   - Listen     0007.b400.0103 192.168.1.3        -
Vl192      1    4   - Listen     0007.b400.0104 192.168.1.4        -

VTP-Server-3#show glbp brief
Interface  Grp  Fwd Pri State      Address      Active router  Standby router
Vl192      1    -   100 Listen    192.168.1.254 192.168.1.1 192.168.1.4
Vl192      1    1   - Listen     0007.b400.0101 192.168.1.1        -
Vl192      1    2   - Listen     0007.b400.0102 192.168.1.2        -
Vl192      1    3   - Active     0007.b400.0103 local        -
Vl192      1    4   - Listen     0007.b400.0104 192.168.1.4        -

VTP-Server-4#show glbp brief
Interface  Grp  Fwd Pri State      Address      Active router  Standby router
Vl192      1    -   100 Standby  192.168.1.254 192.168.1.1 local
Vl192      1    1   - Listen     0007.b400.0101 192.168.1.1        -
Vl192      1    2   - Listen     0007.b400.0102 192.168.1.2        -
Vl192      1    3   - Listen     0007.b400.0103 192.168.1.3        -
Vl192      1    4   - Active     0007.b400.0104 local        -
```

从上面的输出可以看出，基于 VTP-Server-1 (192.168.1.1) 有着优先级值110, 该值高于所有其它网关的优先级值，而已被选举作为活动虚拟网关。网关 VTP-Server-4 (192.168.1.4)，由于有着剩下三台网关中最高的IP地址，而就算这三台网关有着同样的优先级值，被选举作备份虚拟网关。因此网关 VTP-Server-2 与 VTP-Server-3 都被置于侦听状态了。

命令 show glbp 将有关该GLBP组状态的详细信息打印了出来，下面对此命令的输出进行了演示：

```
VTP-Server-1#show glbp
Vlan192 - Group 1
  State is Active
    2 state changes, last state change 02:52:22
    Virtual IP address is 192.168.1.254
    Hello time 3 sec, hold time 10 sec
      Next hello sent in 1.465 secs
    Redirect time 600 sec, forwarder time-out 14400 sec
    Preemption disabled
    Active is local
    Standby is 192.168.1.4, priority 100 (expires in 9.619 sec)
    Priority 110 (configured)
    Weighting 100 (default 100), thresholds: lower 1, upper 100
    Load balancing: round-robin
    Group members:
      0004.c16f.8741 (192.168.1.3)
      000c.ce47.f3a0 (192.168.1.2)
      0013.1986.0a20 (192.168.1.1) local
      0030.803f.ea81 (192.168.1.4)
    There are 4 forwarders (1 active)
    Forwarder 1
      State is Active
        1 state change, last state change 02:52:12
        MAC address is 0007.b400.0101 (default)
        Owner ID is 0013.1986.0a20
        Redirection enabled
        Preemption enabled, min delay 30 sec
        Active is local, weighting 100
    Forwarder 2
      State is Listen
      MAC address is 0007.b400.0102 (learnt)
      Owner ID is 000c.ce47.f3a0
      Redirection enabled, 599.299 sec remaining (maximum 600 sec)
      Time to live: 14399.299 sec (maximum 14400 sec)
      Preemption enabled, min delay 30 sec
      Active is 192.168.1.2 (primary), weighting 100 (expires in 9.295 sec)
    Forwarder 3
      State is Listen
      MAC address is 0007.b400.0103 (learnt)
      Owner ID is 0004.c16f.8741
      Redirection enabled, 599.519 sec remaining (maximum 600 sec)
      Time to live: 14399.519 sec (maximum 14400 sec)
      Preemption enabled, min delay 30 sec
      Active is 192.168.1.3 (primary), weighting 100 (expires in 9.515 sec)
    Forwarder 4
      State is Listen
      MAC address is 0007.b400.0104 (learnt)
      Owner ID is 0030.803f.ea81
      Redirection enabled, 598.514 sec remaining (maximum 600 sec)
      Time to live: 14398.514 sec (maximum 14400 sec)
      Preemption enabled, min delay 30 sec
      Active is 192.168.1.4 (primary), weighting 100 (expires in 8.510 sec)
```

当在活动虚拟网关上执行时，命令 `show glbp` 除了展示其它内容外，还会给出备份虚拟网关的地址和组中所有活动虚拟转发器的数目，以及由活动虚拟网关所指派给这些活动虚拟转发器的状态。同时还显示了各台活动虚拟转发器的虚拟MAC地址。

## 第34天问题

1. Name two FHRP protocols that are Cisco proprietary.
2. Name the open standard FHRP protocol.

3. By default, when HSRP is enabled in Cisco IOS software, version 1 is enabled. True or false?
4. Which Multicast address does HSRP version 2 use to send Hello packets?
5. HSRP version 1 group numbers are restricted to the range of 0 to 255, whereas the version 2 group numbers have been extended from 0 to 4095. True or false?
6. Which parameter can be adjusted in order to influence the HSRP primary gateway election?
7. How does HSRP interface tracking influence the primary gateway election process?
8. Which command can you use to configure an HSRP address on an interface?
9. Just like HSRP, VRRP has the option of allowing the gateway to use the BIA or a statically configured address as the MAC address for VRRP groups. True or false?
10. Which command can you use to configure a GLBP group IP address on a router interface?

## 第34天问题答案

1. HSRP and GLBP.
2. VRRP.
3. True.
4. 224.0.0.102.
5. True.
6. HSRP priority.
7. It modifies HSRP priority based on interface status.
8. The `standby [number] ip [virtual address]` command.
9. False.
10. The `glbp [number] ip [virtual address]` command.

## 第34天实验

### HSRP实验

在包含了两台直连路由器的场景中（也就是 `Fa0/0` 连接到 `Fa0/0`），对本课程模块中有解释的那些命令进行测试。这两天应都经由比如端口 `Fa0/1`，连接到一台交换机。便在交换机上连接一台工作站（workstation）。

- 在两台路由器上配置某种一致的IP分址方案（configure a consistent IP addressing scheme on the two routers），比如 `192.168.0.1/24` 与 `192.168.0.2/24`
- 使用地址 `192.168.0.10`，在面向LAN的接口上配置HSRP 10（configure HSRP 10 on LAN-facing interfaces）
- 将该HSRP组命名为 `CCNA`
- 使用命令 `standby 10 priority 110`，来对主HSRP网关的选举进行控制
- 使用命令 `show standby [brief]`，对HSRP配置进行验证
- 在两台路由器都配置上HSRP抢占
- 关闭 `Router 1`，观察 `Router 2` 如何成为主路由器
- 重启 `Router 1`，并观察其如何因为开启了抢占，而再度成为主路由器
- 将工作站的IP地址配置为 `192.168.0.100/24`，网关地址为 `192.168.0.10`；并从该工作站对网关进行 `ping` 操作

- 配置接口跟踪：使用命令 `standby 10 track [int number]` 对路由器上的一个未使用接口进行跟踪；将该接口循环置于不同根状态，进而对基于该接口状态，而发生的相应路由器优先级变化进行观察。
- 使用命令 `standby version 2`，配置上HSRP版本2
- 通过命令 `standby 10 timers x y`，在两台路由器上对不同HSRP计时器进行修改
- 在两台路由器之间配置MD5的HSRP验证
- 在一台路由器的主网关状态变化时，使用命令 `debug standby` 对HSRP进行调试，从而观察另一台是如何被选举为主网关的

## VRRP实验

重复上一实验，但这次在适用的命令改变下，用VRRP代替HSRP（repeat the previous lab but this time using VRRP instead of HSRP, with the applicable command changes）。

## GLBP实验

重复第一个实验，在适用的命令改变下，用GLBP代替HSRP。在两台路由器上使用 `glbp 10 load-balancing round-robin` 命令，配置GLBP的负载共担，并观察LAN中流量是如何同时到达两台路由器的。

访问[www.in60days.com](http://www.in60days.com), 看看作者是怎样完成此实验的。

# 第35天 系统启动引导过程与 IOS

## Booting and IOS

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

# 第35天任务

- 阅读以下今天的课文
- 复习昨天的课文
- 完成后面的实验
- 阅读ICND2记诵指南

架构（architecture）指的是制造路由器所用的部件，以及在路由器启动过程中它们的用法。这些知识全是一名思科CCNA工程师所要掌握的基础知识，思科CCNA工程师需要知道路由器中的各种存储器完成什么功能，以及怎样使用IOS命令来对各种存储器进行备份或对其存储内容进行操作。

今天将学习以下内容：

- 路由器存储器及各种文件
- 管理IOS

本课程对应了以下ICND2大纲要求：

- 对思科IOS路由器的启动过程进行描述
- 加电自检过程，Power-On Self-Test, POST
- 路由器的启动过程，Router bootup process
- 管理思科IOS的各种文件
- 各种启动选项，Boot preferences
- 各种思科IOS镜像，Cisco IOS images(15)
- 软件许可，licensing
- 展示/修改许可证，show/change license

## 路由器存储与各种文件

下图35.1对路由器内部的主要存储器部件进行了演示。每种存储器都扮演了不同角色，且包含了不同的文件：

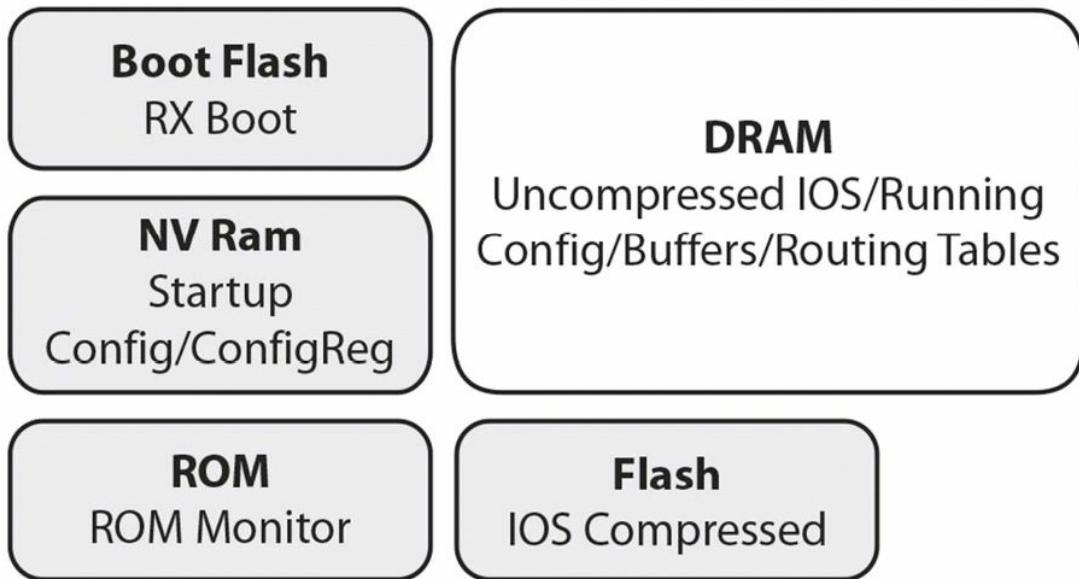


图 35.1 -- 路由器的各种存储器部件

在将路由器盖子打开后，在其内部常能见到不同的存储器插槽。还能发现一些闪存卡插在路由器插槽中。



图35.2 -- 在某台路由器主板上的DRAM单列直插内存模组 (Dynamic Random Access Memory Single In-line Mememory Module on a Router Motherboard)

以下是每种内存及文件类型的作用：

引导ROM (boot ROM) -- 是电可擦可编程只读存储器 (Electrically Erasable Programmable Read-Only Memory, EEPROM, 一种掉电后数据不丢失的存储芯片)，用于启动图/Rommon (startup diagram/Rommon) 的存储及IOS的装入。在路由器启动时，如缺少IOS文件，那么就会启动一种叫做 Rommon 的紧急模式 (an emergency mode)，此模式下允许输入一些有限的几个命令，以对路由器进行恢复及装入其它IOS。此模式又叫做启动模式 (bootstrap mode)，在以下两种路由器提示符下，就可以明白是在此模式：

```
>
Rommon>
```

非易失性随机访问存储器 (Non-Volatile Random Access Mememory, NVRAM) -- 用于启动配置与配置寄存器的存储。启动配置是用于存储已保存的路由器配置的文件。其在路由器重启是不被擦除。

闪存/PCMCIA (Personal Computer Mememory Card International Association) 卡 -- 包含了IOS及一些配置文件。闪存存储器还被当作EEPROM，同时思科IOS就以某种压缩形式存放在那里。在闪存容量充足时，甚至可以在闪存存储器上保存多个版本的思科IOS。

DRAM (内存) -- 也就是RAM，其存储完整的IOS、运行中的配置，及路由表。其为运行内存，在路由器重启后数据被擦除。

ROM监测程序 (ROM Monitor) -- 用于系统诊断及启动。ROM监测程序中有着名为启动器或启动帮助器的一套甚为小型的代码，用于对安装的各种存储器及接口进行检查 (The ROM Monitor has a very small code called bootstrap or boohelper in it to check for attached mememory and interfaces)。

RxBoot程序 -- 小型的IOS (Mini-IOS)，在此程序模式下允许上传一个完整的IOS。其又被称为启动装载器 (the boot loader)，可用于完成一些路由器维护操作 (参见[这里](#))。

路由器配置 -- 尽管严格来说这并非一类路由器组件，其存储在NVRAM中，在启动时拉入到DRAM中。可将DRAM中的配置，经由命令 `copy run start`，放入到NVRAM，同时也可使用命令 `copy start run`，将NVRAM中的配置文件放到内存中。

配置寄存器 (the Configuration Register) -- 设置启动中的一些指令 (sets instructions for booting)。因为在实验中要对用到的路由器上的配置寄存器进行修改 (比如无配置的干净启动)，或是要完成一次口令恢复，所以对配置寄存器的掌握是非常重要的。虽然在某些模型上有所不同，但下面是两个最常见的设置：

- 配置寄存器值 `0x2142` -- 启动并忽略启动配置
- 配置寄存器值 `0x2102` -- 正常启动

通过命令 `show version`，就可以查看到当前的配置寄存器设置：

```
Router#show version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-JS-L), Version 12.1(17), RELEASE SOFTWARE (fc1) Copyright (c) 1986-2002
Compiled Wed 04-Sep-02 03:08 by kellythw Image text-base: 0x03073F40, data-base: 0x00001000
ROM: System Bootstrap, Version 11.0(10c)XB2, PLATFORM SPECIFIC RELEASE SOFTWARE (fc1) BOOTLDR: 3000 Boot

Router uptime is 12 minutes
System returned to ROM by reload
System image file is "flash:c2500-js-l.121-17.bin"

Cisco 2500 (68030) processor (revision L) with 14336K/2048K bytes of memory.
Processor board ID 01760497, with hardware revision 00000000 Bridging software.
X.25 software, Version 3.0.0.
SuperLAT software (copyright 1990 by Meridian Technology Corp).
TN3270 Emulation software.
2 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
16384K bytes of processor board System flash (Read ONLY)

Configuration register is 0x2102
```

命令还显示了该路由器已在线多长时间及上次重启的原因--在对启动问题进行故障排除时，这些信息是有用的。

```
Router uptime is 12 minutes
System returned to ROM by reload
```

同时改命令将显示处路由器上不同类型的存储器：

```
Router#show version
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-IS-L), Version 12.2(4)T1, RELEASE SOFTWARE Copyright (c) 1986-2001 by Ci:
ROM: System Bootstrap, Version 11.0(10c), SOFTWARE-- ROM code
BOOTLDR: 3000 Bootstrap Software (IGS-BOOT-R), Version 11.0(10c)
System image file is "flash:c2500-is-l_122-4_T1.bin"-- Flash image
Cisco 2522 (68030) processor CPU-- CPU
with 14336K/2048K bytes of memory. -- DRAM
Processor board ID 18086064, with hardware revision 00000003
32K bytes of non-volatile configuration memory.-- NVRAM
16384K bytes of processor System flash (Read ONLY) -- EEPROM/FLASH
```

下面是路由器启动过程的一个图形化再现：

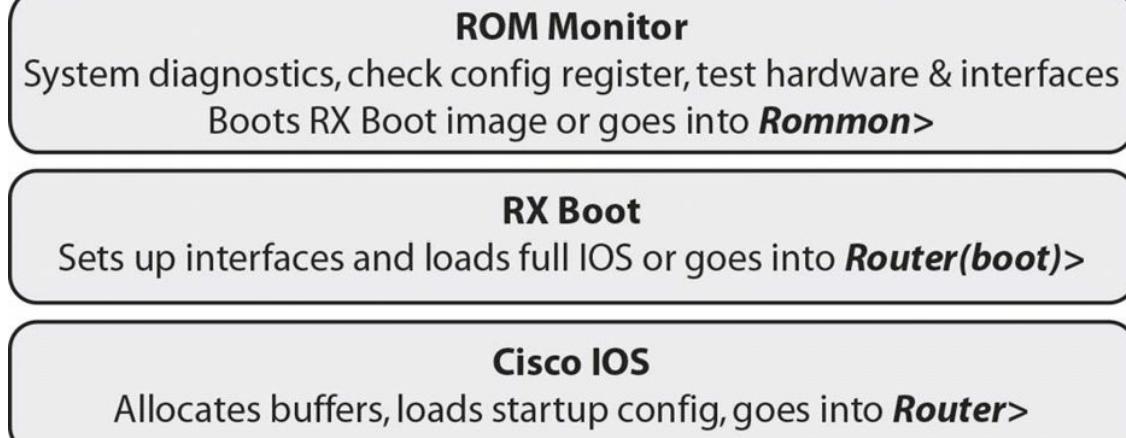


图 35.3 -- 路由器的启动过程

## 管理IOS

做好一些简单的路由器及交换机日常工作，就可避免许多的网络灾难（many network disasters could have been avoided with simple router and switch housekeeping）。如路由器配置文件对于你及你的业务比较重要，那么就应对其进行备份。

如觉得路由器的当前的运行配置，可作为工作版本，就可以使用命令 `copy run start`，将其拷贝到NVRAM中。

而为了将该路由器配置保存起来，就需要在网络上保有一台运行着TFTP服务器软件的PC及或服务器。可从诸如SolarWinds这类公司下载到免费版的TFTP服务器软件。升级闪存镜像也需要有TFTP服务器。

路由器配置可在路由器或网络上的PC机或服务器之间移动。路由器上的运行配置保存在DRAM中。对配置做出的任何修改，都将保存在DRAM中，此时由于任何的原因而导致的重启，这些运行配置都会丢失。

你可以将运行配置拷贝到一台运行了TFTP服务器软件的PC机或服务器上：

```
Router#copy startup-config tftp:-- You need to include the colon
```

还可以将IOS镜像复制到某台TFTP服务器上。如要将服务器IOS更新到另一较新版本，就必须这样做，以防新版本可能带来的问题（管理员经常将一个路由器现有闪存装不下的IOS镜像放上去）。

```
Router#copy flash tftp:
```

路由器将提示输入TFTP服务器的IP地址，建议服务器与路由器位处同一子网。而如打算从TFTP服务器下载IOS镜像，就只需简单地逆转一下命令即可：

```
Router#copy tftp flash:
```

这些命令的问题在于大多数工程师一年也就用两三次，或者只在出现网络灾难时才用到。通常，你会发现在你的网络宕机时，互联网接入也没有了，所以必须要将路由器存储器中将它们备份出来！

作者强烈建议在家庭网络上对配置完成一些备份及恢复的联系。此外，还建议观看一下作者在Youtube上的恢复实验：

[www.youtube.com/user/paulwbrowning](http://www.youtube.com/user/paulwbrowning)

通过 `show version` 或 `show flash` 命令，或者经由 `dir flash:` 进入到flash目录，进入到flash目录将显示出闪存中所有的文件，就可以查看到闪存的文件名。

```
RouterA#show flash
System flash directory:
File      Length      Name/status
1        14692012    c2500-js-l.121-17.bin
[14692076 bytes used, 2085140 available, 16777216 total]
16384K bytes of processor board System flash (Read ONLY)
```

作者本打算对此方面进行深入，但你应着重于CCNA考试本身及日常工作。不过灾难恢复应在深入研究及实验的目标清单当中。

## 各种启动选项

### Booting Options

在路由器启动时，有着许多可选选项。通常在闪存中都只有一个IOS镜像，因此路由器将使用那个镜像进行启动。在有着多个镜像，或者路由器闪存对于镜像太小而无法放下镜像时，就可能需要路由器从网络上的某台保存了IOS镜像的TFTP服务器启动了。

取决于所要配置的启动选项，命令可能有些许不同。所以要在一台开启的路由器上对所有选项都进行尝试。

```
RouterA(config)#boot system ?
WORD          TFTP filename or URL
flash         Boot from flash memory
mop          Boot from a Decnet MOP server
ftp           Boot from server via ftp
rcp           Boot from server via rcp
tftp          Boot from tftp server
```

对于闪存来说：

```
RouterA(config)#boot system flash ? WORD System image filename <cr>
```

而对于TFTP：

```
Enter configuration commands, one per line. End with CNTL/Z.  
RouterB(config)#boot system tftp: c2500-js-l.121-17.bin ? Hostname or A.B.C.D Address from which to do  
RouterA(config)#boot system tftp:
```

## 启动过程及加电自检

### Booting Process and POST

一次标准的路由器启动顺序，看起来像下面这样：

1. 设备开机并将首先执行加电自检（Power on Self Test）动作。加电自检对硬件进行测试，以确保所有组件都不缺少且是正常的（包括各种接口、存储器、CPU、专用集成电路(ASICs)等等）。加电自检程序是存储在ROM中，并自ROM运行的。
2. 启动引导程序（the bootstrap）查找并装入思科IOS软件。启动引导程序是ROM中的一个程序，用于执行一些其它程序，并负责查找各个IOS软件所处位置，接着就装入IOS镜像文件。启动引导程序找到思科IOS软件并将其装入到RAM中。思科IOS文件可在这三个地方找到：闪存、某台TFTP服务器，或在启动配置文件中所指定的另一位置。在所有思科路由器中，IOS软件默认都是从闪存装入的。要从其它位置进行装入，就必须对配置设置进行修改。
3. IOS软件在NVRAM中查找一个可用的配置文件（也就是启动配置文件(the startup-config file)）。
4. 如在NVRAM中确实有着一个启动配置文件，路由器就会装入此文件，此时路由器就将成为可运作的了。而如果在NVRAM中没有启动配置文件，路由器将启动设置模式的配置（the setup-mode configuration）。

在运行路由器上所作的任何修改，就将保存在RAM中，这里就需要手动执行命令 `copy running-config startup-config`，令到当前配置作为在每次启动路由器时的启动配置。

## IOS许可

### IOS Licensing

自思科为其第一台路由器构建首个互联网络操作系统（the first Internetwork Operating System, IOS）以来，其都遵循了以下方式：每种型号的路由器都有着其自己的版本与软件发布构建。大的版本都赋予了12.0的编号系统。对这些版本的改动，就被编号作12.1、12.2等等。这些小的版本是一些漏洞修复，或对一些模块的支持及引入其它特性，比如12.1(1a)。

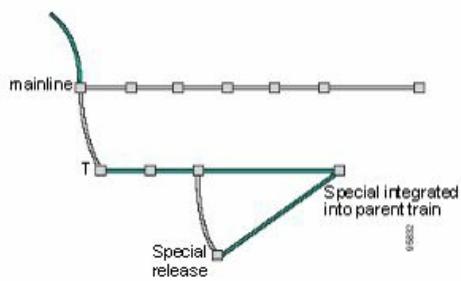
不幸的是，随着支持的加入以及漏洞的修复，这些发布就被拆分成了路线，以致每个型号有其自己的IOS，这样就导致许多不同版本与发布（unfortunately, as support was added and bugs fixed, the releases were split into trains so each model had its own IOS, which led to various versions and releases）。假如需要一个安全或是语音镜像，那么就必须购买对于手头路由器正确版本的特定镜像，同时有着正确的特性支持与漏洞修复。

思科公司最终发布了完整的训练工具与演示，这样就可以搞清楚IOS软件的命名约定、发布级别及支持的模块（the naming conventions, release levels, and supported modules）。而根据软件的测试及成熟情况，其还有着不同的名字，比如ED表示处于早期部署阶段(Early Deployment, ED)，而GD则表示处于一般部署阶段（General Deployment, GD）！这些对于消费者来讲都是非常迷惑的。下面是一张从思科官方文档中摘取的对IOS发布进行解释的图片：

## Special Releases

Cisco.com

- Are similar to rebuilds but instead of quick fixes, special releases introduce new features or additional platform support to quickly meet market demands.
- A branch from a train code base.
- Does not conform to a strict naming convention. They use a double letter after the release number.
- The first letter could be a one-time release, the train identifier, or the technology identifier.
- The second letter could be a sequential revision or a one-time release.
- Special releases do not have an EoL, they are integrated back into the parent train.



Identifier	Target Technology or Platform
X	
Y	
Z	Varies—one time release
A	Aggregation/Access Server/Dial
D	xDSL
H	SDH/SONET
J	Aironet Wireless Networking
M	Mobile Wireless
W	ATM/LAN Switching/Layer 3 Switching

图 35.4 -- IOS 软件的特别发布, IOS Special Releases (Image Copyright Cisco Systems Inc.)

作者在思科技术支持中心（Technical Assistance Center, TAC）就解决了数不清的那些买了一台路由器及一套IOS软件，却发现到手的产品无法支持其对网络设施所要求的那些特性，而迷惑的或是愤怒的客户。还要记住对于大型、企业网络，必须要提前数月来安排IOS升级，把IOS升级放到一个很小的维护窗口。

## 一个新的型号

### A New Model

思科公司现在已经改变了其IOS型号，且从IOS发布12跳到了15（Cisco have now changed their IOS model and jumped from IOS release 12 to 15）。当前，对于每个型号的路由器，有着一个通用镜像。此镜像有着你需要的所有特性，但为了获取到那些真正所需的高级特性的使用，就需要购买适当的许可证，并在具体设备上对许可证加以验证。这样做的目的是为了思科公司及其客户的便利，以及阻止对思科软件的窃取与未授权共享，可以想象，这些思科软件是花了可观成本去开发的。

购买自思科（零售商）的所有新型号路由器，都带有安装好的基础镜像，以及启用了的许可证。而如果客户想要开启高级安全或是语音特性，那么就需要开启这些特性。这通常是经由使用一个名为思科许可证管理器（Cisco License Manager, CLM）的免费思科应用完成的。在[Cisco.com](http://Cisco.com)网站可轻易搜寻到此应用：

The screenshot shows the Cisco License Manager Software download page. At the top, there's a search bar and navigation links for 'Download Cart (0 items)', 'Feedback', and 'Help'. Below the header, the page title is 'Download Software' with a breadcrumb trail: 'Downloads Home > Products > Cloud and Systems Management > Other Network Management Products > Cisco License Manager > License Manager Software > Windows-3.2.4'. The main content area is titled 'Cisco License Manager' and shows 'Release 3.2.4'. On the left, there's a sidebar with a search bar, 'Expand All | Collapse All' buttons, and a tree view of releases: 'Latest Releases' (selected, showing '3.2.4'), '3.1.1', '3.0.12', '2.2', and 'All Releases' (showing '3.2', '3.1', '3', and '2'). The main table lists two packages: 'Cisco License Manager 3.2.4 Client and Server Package (Windows)' (size 300.10 MB) and 'Cisco License Manager 3.2.4 Java Software Developer Kit (SDK)' (size 2.75 MB). Each row has 'Download' and 'Add to cart' buttons.

图 35.5 -- 思科许可证管理器的下载页面

可在某台允许客户在他们的设备与思科公司的许可证门户之间进行操作的服务器或主机上，安装CLM。CLM专注于对当前许可证及各台设备的特性的跟踪，使用图形界面。

The screenshot shows the Cisco License Manager 2.2 graphical user interface. The window title is 'Cisco License Manager 2.2 - User [admin] connected to host [localhost]'. The menu bar includes File, Edit, Manage, License, Troubleshoot, and Help. The toolbar includes 'Get License', 'Manage Licenses', 'Manage Devices', and 'Refresh'. The left sidebar is titled 'Quick Links' and contains sections for Overview (Getting Started), Manage (Authorization Keys, Licenses, Devices, Users), Common Tasks (Get License, Deploy License, Add Device, Discover Devices), Troubleshoot (View Alerts, View Audit Log, Check Device Conn, Check System Conn), and Other (Transfer License, Policy Licensing). The main panel is titled 'Manage Licenses' and shows a hierarchical tree of groups and devices. A table below the tree lists license details: Group/Device, Deploy Sta..., Type, State, Device Name, and PAK Name. The table shows entries like 'Default' (asa5520, clm-2821-1, clm-2821-3, clm-pixar-1, 3XPXR474927, pixfirewall, switch), 'Others' (ipbase, advipservices, ipservices), and '3XPXR474927' (ipbase, advipservices, Others, ipservices). The bottom right corner of the window says 'No new alerts'.

图 35.6 -- 思科许可证管理器的图形界面 (Image Copyright Cisco Systems Inc.)

## 许可证的激活

### License Activation

每种型号的思科路由器（支持许可证的），都已分配了一个叫做唯一设备标识符（the unique device identifier, UDI）的，唯一识别编号（a unique identifying number）。唯一设备标识符是由序列号及产品身份证件组成的（this is compromised of the serial number(SN) and the product identification(PID)）。执行 show license udi 命令，来查看此信息。

```
Router#show license ?
all      Show license all information
detail   Show license detail information
feature  Show license feature information
udi      Show license udi information
Router#show license udi
Device#  PID          SN           UDI
-----
*0       CISCO1941/K9    FTX15240000  CISCO1941/K9:FTX15240000
```

在[www.cisco.com/go/license](http://www.cisco.com/go/license)处将IOS于思科公司进行注册时，就需要输入UDI。还需要把由经销商在你为IOS付款后提供给你的许可证（产品授权密钥，Product Authorization Key, PAK）加入进去，此许可证将与UDI进行比对检查。在验证通过后，思科将发送给你一封许可证密钥的电子邮件。

在下面可以看到有哪些特性也被激活。特性 ipbasek9 将总是开启的。

```
Router#show license all
License Store: Primary License Storage
StoreIndex: 0  Feature: ipbasek9          Version: 1.0
              License Type: Permanent
              License State: Active, In Use
              License Count: Non-Counted
              License Priority: Medium
License Store: Evaluation License Storage
StoreIndex: 0  Feature: securityk9        Version: 1.0
              License Type: Evaluation
              License State: Inactive
                  Evaluation total period: 208 weeks 2 days
                  Evaluation period left: 208 weeks 2 days
              License Count: Non-Counted
              License Priority: None
StoreIndex: 1  Feature: datak9            Version: 1.0
              License Type:
              License State: Inactive
                  Evaluation total period: 208 weeks 2 days
                  Evaluation period left: 208 weeks 2 days
              License Count: Non-Counted
              License Priority: None
```

命令 show license feature 将打印出已开启的特性摘要信息：

```
Router#show license feature
Feature name      Enforcement  Evaluation  Subscription  Enabled
ipbasek9         no          no          no          yes
securityk9        yes         yes         no          no
datak9            yes         no          no          no
```

一旦许可证得到验证，就必须通过U盘或网络服务器，及在命令行执行 license install [url]，将该许可证密钥添加到路由器。需要注意“.lic”这个文件名。

```
Router#dir usbflash0:  
  
Directory of usbflash0:/  
  
1 -rw-        3064  Apr 18 2013 03:31:18 +00:00  FHH1216P07R_20090528163510702.lic  
  
255537152 bytes total (184524800 bytes free)  
Router#  
Router#license install usbflash0:FHH1216P07R_20090528163510702.lic  
Installing...Feature:datak9...Successful:Supported  
1/1 licenses were successfully installed  
0/1 licenses were existing licenses  
0/1 licenses were failed to install  
Router#  
*Jun 25 11:18:20.234: %LICENSE-6-INSTALL: Feature datak9 1.0 was installed in this device. UDI=CISCO29!  
*Jun 25 11:18:20.386: %IOS_LICENSE_IMAGE_APPLICATION-6-LICENSE_LEVEL: Module name = c2951 Next reboot
```

此时将必须重启该路由器，以激活新的特性集。

## 第35天问题

1. Which files would you usually find in DRAM?
2. Where is the compressed IOS held?
3. You want to boot the router and skip the startup configuration. Which command do you use to modify the configuration register?
4. Which command puts the running configuration into NVRAM?
5. Which command will copy your startup configuration onto a network server?
6. You want to boot your router from a network server holding the IOS. Which command will achieve this?
7. The universal image includes all the feature sets you require, but in order to gain access to the advanced features you need to buy the appropriate license and verify it on the actual device. True or false?
8. The ROM monitor has a very small code called bootstrap or boohelper in it to check for attached memory and interfaces. True or false?
9. Which command do you use to view the files stored on the flash memory on a Cisco router?
10. What is the purpose of the POST?

## 第35天答案

1. Uncompressed IOS, running configuration, and routing tables.
2. On the flash memory.
3. The `config-register [version]` command in Global Configuration mode.
4. The `copy run start` command.
5. The `copy start tftp:` command.
6. The `boot system [option]` command.
7. True.
8. True.
9. The `show flash/dir` command.
10. The POST tests the hardware in order to verify that all the components are present and healthy (interfaces, memory, CPU, ASICs, etc.).

## 第35天实验

对本课程模块中讲到的那些配置命令进行测试：

- 在某台思科设备上执行一下 `show version` 命令，并对输出进行检查；将这些输出项与课程中详细解释进行联系
- 将启动配置拷贝到一台TFTP服务器上
- 从某台TFTP服务器拷贝配置文件到路由器上
- 从某台TFTP服务器拷贝一个IOS镜像到路由器的闪存中
- 使用 `show flash` 命令，对闪存中的内容进行检查
- 以 `boot system flash: [name]` 命令，使用新的IOS文件启动设备

访问[www.in60days.com](http://www.in60days.com)网站，免费观看作者完成此实验。

# 第36天 增强的内部网关路由协议

## Enhanced Interior Gateway Routing Protocol, EIGRP

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

# 第36天任务

- 阅读今天的课文
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2记诵指南

增强的内部网关路由协议是一种由思科开发的、专有内部网关协议（a proprietary Interior Gateway Protocol(IGP) that was developed by Cisco）。EIGRP包含了一些传统的距离矢量特性，比如水平拆分（split horizon），还包含了与那些被链路状态路由协议所用到的类似特性，比如增量更新（EIGRP includes traditional Distance Vector characteristics, such as split horizon, as well as characteristics that are similar to those used by Link State routing protocols, such as incremental updates）。

尽管有着链路状态路由协议的某些特性，但EIGRP是被分类到距离矢量路由协议类别中的，被指为一种高级的距离矢量路由协议（an advanced Distance Vector routing protocol）。EIGRP直接在IP上运行，使用协议编号88。

今天将学习以下内容：

- 思科公司EIGRP概述与基础知识, Cisco EIGRP overview and fundamentals
- EIGRP配置基础, EIGRP configuration fundamentals
- EIGRP的各种报文, EIGRP messages
- EIGRP的邻居发现与邻居维护, EIGRP neighbour discovery and maintenance
- 各种度量值、弥散更新算法（DUAL）与拓扑表, Metrics, DUAL, and the topology table
- 相等与不相等开销下的负载均衡, equal cost and unequal cost load sharing
- 采用EIGRP作为默认路由, default routing using EIGRP
- EIGRP网络中的水平分割, split horizon in EIGRP networks
- EIGRP的存根路由, EIGRP stub routing
- EIGRP的路由汇总, EIGRP route summarisation
- 掌握被动接口, understanding passive interfaces
- 掌握EIGRP路由器ID的用法, understanding the use of the EIGRP router ID
- EIGRP的日志与报表, EIGRP logging and reporting

本课程对应了以下CCNA大纲要求：

- 配置并验证EIGRP（单一自治系统），configure and verify EIGRP(single AS)
- 可行距离/可行的后续路由/报告的距离/通告的距离分别是什么，Feasible Distance/Feasible Successor routes/Reported Distance/Advertised Distance
- 可行性条件，Feasibility condition
- 度量值综合，Metric composition
- 路由器ID，Router ID
- 自动汇总，Auto summary
- 路径选择，path selection
- EIGRP的负载均衡，load balancing
  - 开销一样时
  - 开销不同时
- 什么是EIGRP的被动接口，passive interfaces

## 思科EIGRP概述与基础知识

为解决其先前的专有距离矢量路由协议 -- 内部网关路由协议 (Interior Gateway Routing Protocol, IGRP) 的某些缺陷，思科公司开发了EIGRP。相比路由信息协议 (Routing Information Protocol, RIP)，IGRP确实有着一些改进，比如对更多跳数的支持；但仍然有着那些传统距离矢量路由协议的局限，这些局限如下所示：

- 发送完整的周期性路由更新，sending full periodic routing updates
- 跳数限制，a hop limitation
- 缺少对变长子网掩码的支持，the lack of VLSM support
- 收敛速度慢，slow convergence
- 缺少防止环回形成的机制，the lack of loop prevention mechanisms

与传统距离矢量路由协议往邻居发送包含所有路由信息的周期性路由更新不同，EIGRP发送的是非周期性的、增量式路由更新，以将路由信息在整个路由域中分发 (unlike the traditional Distance Vector routing protocols, which send their neighbours periodic routing updates that contain all routing information, EIGRP sends non-periodic incremental routing updates to distribute routing information throughout the routing domain)。只有在网络拓扑发生变化时，才会发送EIGRP增量更新。

默认下的RIP（一种以前的CCNA考试项目）有着15的跳数限制，这就令到RIP只适合于小型网络。EIGRP默认跳数限制为100；但此数值可被管理员在配置EIGRP时，使用路由器配置命令 `metric maximum-hops <1-255>`，予以手动调整。这就令到EIGRP具备对有着多达数百台路由器网络的支持能力，使其具备了更大的可伸缩性，从而对较大型网络也是适合的。

增强的IGRP采用了两个独特的**类型/长度/数值三联体数据结构**来表示和传输路由条目 (Enhanced IGRP uses two unique Type/Length/Value(TLV) triplets to carry route entries)。这两个TLVs分别是**内部EIGRP路由TLV**与**外部EIGRP路由TLV** (the Internal EIGRP Route TLV and the External EIGRP Route TLV)，分别用于内部及外部的EIGRP路由。两种TLVs都包含了一个8位的前缀长度字段 (an 8-bit Prefix Length field)，用于指明用于目的网络子网掩码的位数。包含在此字段中的该信息，就令到EIGRP能够支持不同的子网划分了。

增强的IGRP比起传统的距离矢量路由协议收敛得快得多。除了仅仅依赖于计时器，EIGRP还使用其拓扑表中的信息，来找出那些替代路径。EIGRP也可以在未能于本地路由器的拓扑表中找出替代路径的情况下，向邻居路由器查询信息。本课程模块后面会讲到EIGRP的拓扑表。

而为了确保整个网络中没有环回路径，EIGRP使用了**弥散更新算法** (Diffusing Update Algorithm, DUAL)，使用此算法来对邻居通告的所有路由进行追踪，并随后选出到目的网络最优的无环回路径。弥散更新算法是EIGRP的一个核心概念，将在本课程模块的稍后讲到。

## EIGRP配置基础

### EIGRP Configuration Fundamentals

在思科IOS软件中，是通过使用全局配置命令 `router eigrp [ASN]`，来开启增强的IGRP的。关键字 `[ASN]` 指定EIGRP的自治系统编号（autonomous system number, ASN），这是一个32位整数，大小介于1-65535之间。除了本章后面将涉及的其它因素之外，**运行EIGRP的那些路由器都必须位处同一自治系统中**，以成功形成邻居关系。在全局配置命令 `router eigrp [ASN]` 之后，路由器就转变为EIGRP路由器配置模式（EIGRP Router Configuration mode）了，在这里就可以对那些与EIGRP有关的参数进行配置了。所配置的ASN，可在命令 `show ip protocols` 的输出中进行验证，如下面所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
...
[Truncated Output]
```

而除了 `show ip protocols` 命令，命令 `show ip eigrp neighbours` 会打印出所有EIGRP邻居，以及这些邻居各自自治系统的相关信息。该命令及其可用选项，将在本课程模块的后面进行详细讲解。在那些运行了多个EIGRP实例的路由器上，可使用 `show ip eigrp [ASN]` 命令，来查看只与在此命令中所指定的自治系统有关的信息。下面的输出演示了这个命令的使用：

```
R1#show ip eigrp 150 ?
  interfaces  IP-EIGRP interfaces
  neighbors   IP-EIGRP neighbors
  topology    IP-EIGRP topology table
  traffic     IP-EIGRP traffic statistics
```

在上面的输出中，150就是自治系统编号（ASN）。如 `show ip eigrp` 命令没有指定自治系统，那么在思科IOS软件中该命令默认将打印出所有EIGRP实例的信息。

而一旦处于路由器配置模式（Router Configuration mode），就要使用 `network` 命令，来指明要在哪些网络（接口上）开启EIGRP路由了（once in Router Configuration mode, the `network` command is used to specify the network(s) (interfaces) for which EIGRP routing will be enabled）。在使用 `network` 命令并指明了一个大的有类网络后，该启用了EIGRP的路由器将完成以下动作：

- 对位处该指明的有类网络范围里的网络，开启EIGRP，EIGRP is enabled for networks that fall within the specified classful network range.
- 以这些直连子网，生成EIGRP的拓扑表，the topology table is populated with these directly connected subnets.
- 从这些与子网关联的接口，发出EIGRP Hello 数据包，EIGRP Hello packets are sent out of the interfaces associated with these subnets.
- EIGRP将这些网络，经由更新报文，通告给EIGRP邻居，EIGRP advertises the network(s) to EIGRP neighbours in Update messages.
- 在报文交换的基础上，那些EIGRP路由，此时就被加入到路由器的IP路由表中，Based on the exchange of messages, EIGRP routes are then added to the IP routing table.

比如假设某台路由器上配置了以下这些环回接口：

- Loopback0 -- IP地址 10.0.0.0/24
- Loopback1 -- IP地址 10.0.0.1/24
- Loopback2 -- IP地址 10.0.0.2/24
- Loopback3 -- IP地址 10.0.0.3/24

如EIGRP已开启使用，且将路由器配置命令 `network` 与大的有类 10.0.0.0/8 网络一道进行了使用，同时所有4个环回接口（all four Loopback interfaces）又都开启了EIGRP路由的话，那么下面就给出了此种情况下 `show ip eigrp interfaces` 的输出演示：

```
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 150
          Xmit Queue   Mean      Pacing Time    Multicast      Pending
Interface    Peers Un/Reliable SRTT    Un/Reliable     Flow Timer    Routes
Lo0          0       0/0        0        0/10          0            0
Lo1          0       0/0        0        0/10          0            0
Lo2          0       0/0        0        0/10          0            0
Lo3          0       0/0        0        0/10          0            0
```

可使用 `show ip protocols` 命令，来对大的有类 10.0.0.0/8 网络上EIGRP的启用情况，进行验证。此命令的输出如下所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway          Distance      Last Update
    Distance: internal 90 external 170
```

使用命令 `show ip eigrp topology`，可查看到EIGRP的拓扑表。此命令的输出如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
      r - reply Status, s - sia Status
P 10.3.3.0/24, 1 successors, FD is 128256
      via Connected, Loopback3
P 10.2.2.0/24, 1 successors, FD is 128256
      via Connected, Loopback2
P 10.1.1.0/24, 1 successors, FD is 128256
      via Connected, Loopback1
P 10.0.0.0/24, 1 successors, FD is 128256
      via Connected, Loopback0
```

**注意：**本课程模块稍后会对拓扑表、EIGRP的Hello数据包及更新数据包进行详细讲解。本小节仅着重于EIGRP的配置实施（EIGRP configuration implementation）。

使用 `network` 命令来指明一个大的有类网络（a major classful network），就令到位于该有类网络中的多个子网，得以在最小配置下同时被通告出去。但可能存在管理员不想对某个有类网络中的所有子网，都开启EIGRP路由的情形。比如，参考前一示例中 R1 上所配置的环回接口，假设只打算对 10.1.1.0/24 及 10.3.3.0/24 子网开启EIGRP路由，而不愿在 10.0.0.0/24 及 10.2.2.0/24 开启EIGRP路由。那么很明显这在使用 `network` 命令时，对这些网络（也就是 10.1.1.0 及 10.3.3.0）予以指明就可以做到，思科IOS软件仍会将这些语句，转换成大的有类 10.0.0.0/8 网络，如下所示：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.0
R1(config-router)#network 10.3.3.0
R1(config-router)#exit
```

尽管有着上面的配置，但 `show ip protocols` 命令给出的确实下面的输出：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway      Distance      Last Update
  Distance: internal 90 external 170
```

**注意：**一个常见的误解就是，关闭EIGRP的自动汇总特性，就能解决此问题；但是，这与 `auto-summary` 命令一点关系都没有。比如，假设对在前一示例中的配置执行了 `no auto-summary` 命令，如下所示：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.0
R1(config-router)#network 10.3.3.0
R1(config-router)#no auto-summary
R1(config-router)#exit
```

`show ip protocols` 命令仍将显示对网络 10.0.0.0/8 开启了EIGRP，如下面的输出所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
  Routing Information Sources:
    Gateway      Distance      Last Update
Distance: internal 90 external 170
```

为了提供到对那些开启EIGRP路由的网络进行更细粒度的控制，思科IOS软件支持在对EIGRP进行配置时，将通配符掩码与 `network` 语句一起配合使用（in order to provide more granular control of the networks that are enabled for EIGRP routing, Cisco IOS software supports the use of wildcard masks in conjunction with the `network` statement when configuring EIGRP）。这里的通配符掩码，以与ACLS中用到的通配符掩码类似的方式运作，而与网络的子网掩码是不相干的。

作为一个示例，命令 `network 10.1.1.0 0.0.0.255` 将匹配到网络 `10.1.1.0/24`、`10.1.1.0/26` 及 `10.1.1.0/30` 网络。参考上一输出中所配置的那些环回借口（the Loopback interfaces），为将 R1 配置为对 `10.1.1.0/24` 及 `10.3.3.0/24` 子网开启EIGRP路由，且不对 `10.0.0.0/24` 子网或 `10.2.2.0` 子网开启，就应将其如下面那样进行配置：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#network 10.3.3.0 0.0.0.255
R1(config-router)#exit
```

使用命令 `show ip protocols`，就可对此配置进行验证，如下所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.1.1.0/24
    10.3.3.0/24
  Routing Information Sources:
    Gateway      Distance      Last Update
Distance: internal 90 external 170
```

此外，还可以使用命令 `show ip eigrp interfaces`，确认到仅已对 `Loopback1` 与 `Loopback3` 开启了EIGRP路由由：

```
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 150
      Xmit Queue   Mean     Pacing Time    Multicast    Pending
Interface    Peers Un/Reliable SRTT    Un/Reliable   Flow Timer  Routes
Lo1          0       0/0        0        0/10          0           0
Lo3          0       0/0        0        0/10          0           0
```

如上面所示，因为这里的通配符掩码配置，而仅在 `Loopback1` 和 `Loopback3` 上启用了EIGRP路由。

这里重要的是记住除了使用通配符掩码，**也可以使用子网掩码来配置 `network` 命令**。在此情况下，思科IOS软件将翻转子网掩码，而使用通配符掩码来保存该命令。比如，参照路由器上同样的环回借口，路由器 `r1` 也可被如下这样进行配置：

```
R1(config-router)#router eigrp 150
R1(config-router)#network 10.1.1.0 255.255.255.0
R1(config-router)#network 10.3.3.0 255.255.255.0
R1(config-router)#exit
```

基于此种配置，就在运行配置中输入了下面的参数（这里使用了管道（pipe），取得运行配置中感兴趣的部分）：

```
R1#show running-config | begin router eigrp
router eigrp 150
network 10.1.1.0 0.0.0.255
network 10.3.3.0 0.0.0.255
auto-summary
```

通过上面的配置可以看出，可与那些 `show` 命令一道运用管道，来获得更细的粒度。这对于那些有着编程知识的人来说，是一种熟悉的概念。

如将某个网络上的特定地址与通配符一起使用，那么思科IOS软件将执行一次逻辑与运算（a logical AND operation），从而确定出那个要启用EIGRP的网络。比如，在执行了 `network 10.1.1.15 0.0.0.255` 命令时，思科IOS软件会执行以下动作：

- 将通配符掩码翻转为子网掩码值 `255.255.255.0`
- 执行一次逻辑与操作
- 将命令 `network 10.1.1.0 0.0.0.255` 加入到配置中

本示例中所用到的 `network` 配置，如下面输出所示：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.15 0.0.0.255
R1(config-router)#exit
```

那么基于此配置，路由器上的运行配置，就会显示如下内容：

```
R1#show running-config | begin router eigrp
router eigrp 150
network 10.1.1.0 0.0.0.255
auto-summary
```

如上面的配置所示，不管是使用通配符子网掩码，还是子网掩码，都会在思科IOS软件中造成同样的操作，并得到同样的 network 语句配置。

### 真实世界的部署

当在生产网络中对EIGRP进行配置时，一般做法都是使用全0的通配符掩码或全1的子网掩码。比如，`network 10.1.1.1 0.0.0.0` 及 `network 10.1.1.1 255.255.255.255`，两个命令都会执行同样的动作。全0的通配符掩码或全1的子网掩码的使用，就将思科IOS软件配置为与一个具体接口地址进行匹配，而不考虑在接口本身上所配置哦子网掩码了。这两个命令都会匹配到配置了比如 `10.1.1.1/8`、`10.1.1.1/16`、`10.1.1.1/24`，以及 `10.1.1.1/30` 等地址的接口。这些命令的用法如下面的输出所示：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.0.0.1 0.0.0.0
R1(config-router)#network 10.1.1.1 255.255.255.255
R1(config-router)#exit
```

`show ip protocols` 命令将验证到路由器对于两个 network 语句，都是以相似的方式进行处理的，如下所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.1/32
    10.1.1.1/32
  Routing Information Sources:
    Gateway      Distance      Last Update
  Distance: internal 90 external 170
```

在使用了全1的子网掩码或全1的通配符掩码时，就会在所指定的（匹配的）接口上开启EIGRP，同时将通告那个接口所位处的网络。也就是说，EIGRP不会通告上面输出中的 /32 地址，而是通告基于配置在匹配接口上的子网掩码的具体网络。此配置的用法，是独立于配置所匹配具体接口子网掩码的（when a subnet mask with all ones or a wildcard mask with all zeros is used, EIGRP is enabled for the specified(matched) interface and the network the interface resides on is advertised. In other words, EIGRP will not advertise the /32 address in the above but, instead, the actual network based on the subnet mask configured on the matched interface. The use of this configuration is independent of the subnet mask configuration on the actual interface matched）。

## EIGRP的各种报文

### EIGRP Messages

本小节将对EIGRP所用到的各种类型的报文进行说明。但是，在深入到各种不同报文类型前，首要的是对EIGRP数据包头部有扎实的掌握，这正是包含这些报文的地方。

## EIGRP数据包头部

### EIGRP Packet Header

尽管对EIGRP数据包格式的具体了解，是超出CCNA考试要求的，但对EIGRP数据包头部的扎实掌握，对于完整理解EIGRP这种路由协议的整体运作原理，是很重要的。下图36.1对EIGRP数据包头部进行了演示：

Version	OPCode	Checksum
Flags		
Sequence		
Acknowledgement		
Autonomous System Number		
TLVs		

图 36.1 -- EIGRP数据包头部的各种字段

在EIGRP数据包头部，其中的4位版本字段（the 4-bit Version field）用于表明该协议的版本。当前的思科 IOS镜像都支持EIGRP版本 1.x。后面的4为OPcode字段（the 4-bit OPCode field），则指定了EIGRP数据包或报文的类型。不同EIGRP数据包类型都被分配了一个唯一的OPcode数值，以将其与其它数据包类型进行区分。本课程模块后面会对这些报文类型进行说明。

而那个24位的校验和字段（the 24-bit Checksum field），是用于对该EIGRP数据包做完整性检查的（a sanity check）。该字段基于完整的EIGRP数据包的，是排除了IP头部的。32位的标志字段（the 32-bit Flags field），用于表明该EIGRP数据包或报文是一个新EIGRP邻居的 INIT，还是EIGRP的可靠传输协议（Reliable Transport Protocol, RTP）条件接收下（the Conditional Receive, CR）的 INIT。这里的RTP及CR都将在本课程模块稍后讲到。

接着便是32位的序列字段（the 32-bit Sequence field），其指明了被EIGRP可靠传输协议所用到的顺序编号（the sequence number），用于可靠数据包的顺序投送。而32位的确认字段（the 32-bit Acknowledgment field），则被用于EIGRP可靠数据包的接收确认。

后面的32位自治系统编号字段（the 32-bit Autonomous System Number field），指定了该EIGRP域（the EIGRP domain）的自治系统编号（ASN）。最后的32位类型/长度/值三联体字段，就被用于路由条目运送及EIGRP弥散更新算法信息的提供了（Finally, the 32-bit Type/Length/Value(TLV) triplet field is used to carry route entries and provides EIGRP DUAL information）。EIGRP支持几种不同类型的TLVs，最常用的是下面几种：

- 参数TLV，有着建立邻居关系的那些参数，the Parameters TLV, which has the parameters to establish neighbour relationships

- 序号TLV，为RTP所用到的TLV，the Sequence TLV, which is used by RTP
- 下一次多播序号TLV，RTP使用的TLV，the Next Multicast Sequence TLV, which is used by RTP
- EIGRP内部路由TLV，用于内部EIGRP路由，the EIGRP Internal Route TLV, Which is used for internal EIGRP routes
- EIGRP外部路由TLV，用于外部的EIGRP路由，the EIGRP External Route TLV, which is used for external EIGRP routes

**注意：** 并不要求对EIGRP的各种TLVs有详细了解。

下图36.2演示了一个抓包到的EIGRP数据包的所呈现的不同字段：

```
Cisco EIGRP
Version      = 2
Opcode       = 5 (Hello)
Checksum     = 0xee36
Flags        = 0x00000000
Sequence     = 0
Acknowledge   = 0
Autonomous System : 150
EIGRP Parameters
Software Version: IOS=12.4, EIGRP=1.2
```

图 36.2 -- 对EIGRP数据包头部的抓包

在该EIGRP数据包头部，4位的OPCode字段被用于指明该EIGRP数据包类型或报文。EIGRP使用到不同的报文或数据包类型，它们是**Hello数据包**（Hello packets）、**确认数据包**（Acknowledgment packets）、**更新数据包**（Update packets）、**查询数据包**（Query packets）、**应答数据包**（Reply packets）以及**请求数据包**（Request packets），共计6种报文或数据包类型。将在随后的小节对这些类型的数据包进行说明。

## Hello数据包

### Hello Packets

在某台路由器上对某个特定网络开启了增强的IGRP后，其就会发送Hello数据包（Enhanced IGRP sends Hello packets once it has been enabled on a router for a particular network）。这些报文被用于邻居的识别，同时邻居一经识别后，Hello报文就用于在邻居间作为一种保持活动机制，发挥作用（these messages are used to identify neighbours and, once identified, serve or function as a keepalive mechanism between neighbours）。**EIGRP的邻居发现与维护机制**，将在本课程模块的后面进行说明。

EIGRP的Hello数据包，是发送到**链路本地多播组地址**（the Link Local Multicast group address）**224. 0. 0. 10** 上的。由EIGRP发出的Hello数据包，是不需要发出确认数据包来确认其已收到的（Hello packets sent by EIGRP do not require an Acknowledgment to be sent confirming that they were received）。因为Hello数据包不需要显式的确认，所以它们被分类为**不可靠的EIGRP数据包**（Hello packets are classified as unreliable EIGRP packets）。EIGRP Hello数据包的OPCode为5。

## 确认数据包

### Acknowledgment Packets

EIGRP确认数据包，就是一个**不包含数据的EIGRP Hello数据包**。EIGRP使用确认数据包来对EIGRP数据包的可靠送达进行确认。这些确认数据包（the ACK packets）总是发送到一个单播地址（a Unicast address），该地址就是可靠数据包发送方的源地址（the source address of the sender of the reliable packet），而并不是EIGRP的多播组地址了。此外，确认数据包将总是会包含一个非零的确认编号（a non-zero acknowledgment number）。确认数据包使用了Hello数据包相同的OPCode，因为其本来就是一个不包含任何信息的Hello数据包。其OPCode为5。

## 更新数据包

### Update Packets

增强IGRP的更新数据包被用于传送目标的可达性（used to convey reachability of destinations）。也就是说，更新数据包包含了EIGRP的路由更新。在发现了一个新的邻居时，就会通过单播发出更新数据包（往该新的邻居），如此新的邻居就能够建立起自己的EIGRP拓扑表了。在其它情况，比如某链路的开销改变时，就会经由多播发出更新数据包。重要的是记住更新数据包都是可靠地传输的，且总是要求显式的确认。分配给更新数据包的OPCode是1。下图36.3演示了一个EIGRP的更新数据包：

```
Cisco EIGRP
Version      = 2
Opcode = 1 (Update)
Checksum     = 0x1629
Flags        = 0x00000008
Sequence     = 7
Acknowledge   = 10
Autonomous System : 150
IP internal route = 1.0.0.0/8
Type = 0x0102 (IP internal route)
Size = 26 bytes
Next Hop     = 0.0.0.0
Delay        = 128000
Bandwidth    = 256
MTU          = 1514
Hop Count    = 0
Reliability  = 255
Load         = 1
Reserved
Prefix Length = 8
Destination  = 1.0.0.0
```

图 36.3 -- EIGRP 的更新数据包

**注意：** 并不要求对EIGRP各种数据包中的所包含的信息有深入了解。

## 查询数据包

### Query Packets

增强IGRP的查询数据包是多播的，并被用于请求可靠的路由信息。EIGRP的查询数据包是在某条路由不可用，但该路由器却需要为快速收敛，而需要就该路由的状态进行询问时，所发送给其邻居的数据包。如发出查询数据包的路由器未能从其某些邻居收到响应，那么其就会再度向那些未响应的邻居发出一次查询。如在16次尝试后都没有响应，那么该EIGRP邻居关系将被重置。本课程模块后面将对此概念进行更为深入的说明。分配给EIGRP查询数据包的OPCode为3。

## 应答数据包

### Reply Packets

增强IGRP的应答数据包是作为对查询数据包的响应发送的。应答数据包用于可靠地响应某个查询数据包。应答数据包是到查询发起方的单播数据包。分配给EIGRP应答数据包的OPCode为4。

## 请求数据包

### Request Packets

增强IGRP的请求数据包，用于从一个或多个邻居处获取特定信息，且是在路由服务器应用中用到的（used in route server applications）。这些数据包既可通过单播、也可通过多播进行发送，但它们总是以不可靠方式传输。也就是说，它们无需显式确认。

**注意：**尽管这里的Hello数据包和确认数据包是作为两种独立的数据包类型的，但重要的是记住在某些课本中，EIGRP的Hello数据包与确认数据包被认为是同一中类型的数据包。这是因为，正如在本小节中指出的那样，确认数据包就是不包含数据的Hello数据包。

命令 `debug eigrp packets`，可用于打印出本小节中所讲到的各种不同EIGRP数据包的实时调试信息。要知道此命令还包括了一些这里并没有说到的其它数据包，因为这些其它类型数据包超出了当前CCNA考试要求。下面的输出对此命令的用法进行了演示：

```
R1#debug eigrp packets ?
SIAquery EIGRP SIA-Query packets
SIAreply EIGRP SIA-Reply packets
ack EIGRP ack packets
hello EIGRP hello packets
ipxsap EIGRP ipxsap packets
probe EIGRP probe packets
query EIGRP query packets
reply EIGRP reply packets
request EIGRP request packets
retry EIGRP retransmissions
stub EIGRP stub packets
terse Display all EIGRP packets except Hellos
update EIGRP update packets
verbose Display all EIGRP packets
<cr>
```

而 `show ip eigrp traffic` 命令，则是用于对本地路由器所发送及接收到的EIGRP数据包的数量进行查看的命令。该命令同时还是一个强大的故障排除工具。比如假设某路由器在发出Hello数据包，却并未收到任何回复，这可能表明其尚未配置好预期的邻居，或者甚至有可能某个确认数据包阻塞了EIGRP数据包（For example, if the router is sending out Hello packets but is not receiving any back, this could indicate that the intended neighbour is not configured, or even that an ACK may be blocking EIGRP packets）。下面的输出对此命令进行了演示：

```
R2#show ip eigrp traffic
IP-EIGRP Traffic Statistics for AS 150
    Hellos sent/received: 21918/21922
    Updates sent/received: 10/6
    Queries sent/received: 1/0
    Replies sent/received: 0/1
    Acks sent/received: 6/10
    SIA-Queries sent/received: 0/0
    SIA-Replies sent/received: 0/0
    Hello Process ID: 178
    PDM Process ID: 154
    IP Socket queue: 0/2000/2/0 (current/max/highest/drops)
    Eigrp input queue: 0/2000/2/0 (current/max/highest/drops)
```

下表36.1对本小节中讲到的这些EIGRP的数据包进行了总结，以及各自是否以可靠或不可靠方式进行发送：

表 36.1 -- EIGRP数据包总结

报文类型	说明	发送方式
Hello	用于邻居发现、维护及保持存活	不可靠
确认数据包 (Acknowledgment)	用于对信息接收的确认	不可靠
更新数据包 (Update)	用于传达路由信息	可靠的
查询数据包 (Query)	用于请求指定的路由信息	可靠的
应答数据包 (Reply)	用于对查询数据包的响应	可靠的
请求数据包 (Request)	用于路由服务器应用中的信息请求	不可靠

## EIGRP的邻居发现与邻居维护

### EIGRP Neighbour Discovery and Maintenance

可将增强的IGRP配置为动态地发现相邻路由器（这是默认选项）（discover neighbouring routers dynamically(default)），或者经由管理员手动配置来发现相邻路由器。下面的小节将对这两种方式，以及其它有关EIGRP邻居相关话题，进行讨论。

### 动态的邻居发现

#### Dynamic Neighbour Discovery

动态邻居发现，是通过往目的多播组地址（the destination Multicast group address）`224.0.0.10`，发送EIGRP的Hello数据包完成的。动态邻居发现，又是紧跟着在路由器配置EIGRP时`network`命令的执行，而完成的。此外，如早先指出的那样，EIGRP数据包直接透过IP、使用协议编号88发送。下图36.4对此基本的EIGRP邻居发现与路由交换过程，进行了演示：

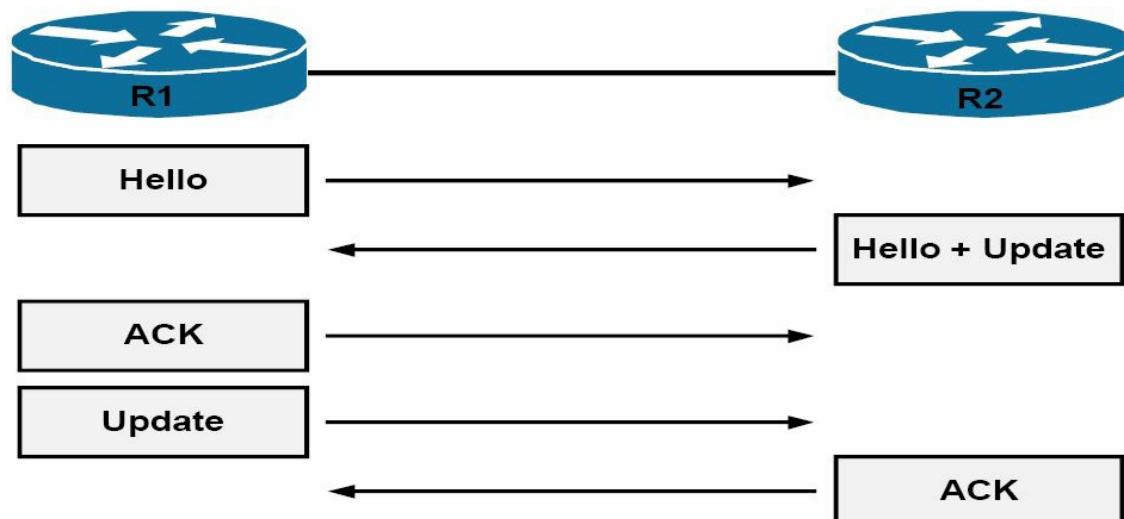


图 36.4 -- EIGRP的邻居发现与路由交换, EIGRP Neighbour Discovery and Route Exchange

参考图36.4，在初始化时，这些EIGRP邻居便发出Hello数据包，以发现其它邻居（Referencing Figure 36.4, upon initialisation, the EIGRP neighbours send Hello packets to discover other neighbours）。随后邻居们就通过完整更新，就其整个路由表进行交换。这些更新包含了所有已知路由信息。因为更新包是可靠发送的，接收方就必须对其进行显式确认。

在邻居们完成它们的路由信息交换后，还将持续交换Hello数据包，以维护邻居关系。此外，这些EIGRP邻居路由器以后就将仅发送增量更新了，通过增量更新来将其状态或路由变化，通告给它们的邻居们。它们再也不会发送完整的更新给邻居们了。

这里重要的是要明白仅简单地在两台或多台路由器上开启EIGRP，并不能确保邻居关系的建立。而是还需要一些参数必须要匹配，这样这些路由器才能成为邻居。**在以下几种情况下，就不能建立EIGRP邻居关系：**

- EIGRP认证参数不匹配（在有配置时）， Mismatched EIGRP authentication parameters(if configured)
- EIGRP的那些K值不一致， Mismatched EIGRP K values
- EIGRP自治系统编号不一致， Mismatched EIGRP autonomous system number
- 在EIGRP邻居关系中，使用了接口的第二地址， Using secondary addresses for EIGRP neighbour relationships
- 邻居并不位于同一子网中， the neighbours are not on a common subnet

尽管 `show ip eigrp neighbours` 命令在动态与静态配置的邻居间没有区别，但 `show ip eigrp interfaces detail <name>` 命令却可以用于对路由器接口是否在发出多播数据包来发现和维护邻居关系，进行检查。下面演示了在一台开启了动态邻居发现的路由器上该命令的输出：

```
R2#show ip eigrp interfaces detail FastEthernet0/0
IP-EIGRP interfaces for process 150
          Xmit Queue   Mean      Pacing Time      Multicast      Pending
Interface    Peers Un/Reliable SRTT    Un/Reliable     Flow Timer    Routes
Fa0/0         1       0/0        1        0/1           50            0

Hello interval is 5 sec
Next xmit serial <none>
Un/reliable mcasts: 0/2  Un/reliable ucasts: 2/2
Mcast exceptions: 0  CR packets: 0  ACKs suppressed: 0
Retransmissions sent: 1  Out-of-sequence rcvd: 0
Authentication mode is not set
Use multicast
```

**注意：** `show ip eigrp neighbours` 命令将在后面讲到。在查看 `show ip eigrp interfaces detail <name>` 命令的输出时，要注意因为EIGRP同时用到多播及单播数据包（both Multicast and Unicast packets），所以该命令的计数器将包含两种类型数据包的数值，如上面输出所示。

## 静态的邻居发现

### Static Neighbour Discovery

与动态EIGRP邻居发现过程不同，静态EIGRP邻居关系需要在路由器上手动配置邻居。在配置好静态EIGRP邻居后，本地路由器就使用单播邻居地址，往这些路由器发送数据包。

在EIGRP网络中，静态邻居关系的使用是十分罕见的。这主要是因为手动邻居配置在大型网络中无法适应其规模。但重要的是弄明白**为何思科IOS软件中还是有此选项，以及在什么情况下可以运用到此特性**。使用静态邻居配置的一个主要，就是在那些没有广播或多播数据包原生支持的传输介质，比如帧中继上，部署EIGRP的情况下（A prime example of when static neighbour configuration could be used would be in a situation where EIGRP is being deployed across media that does not natively support Broadcast or Multicast packets, such as Frame Relay）。

另一实例就是在譬如以太网这样的多路访问网络上，仅有少数几台开启了EIGRP的路由器时，为了防止发送不必要的EIGRP数据包的情形。在此情况下，除开基本的EIGRP配置，还必须在本地路由器上对**所有静态EIGRP邻居配置** `neighbour` 命令。**如果一台路由器被配置为使用单播（静态），而其它路由器又被配置**

为使用多播（动态），那么这些开启了EIGRP的路由器是不会建立邻接关系的（EIGRP-enabled routers will not establish an adjacency if one router is configured to use Unicast(static) while another uses Multicast(dynamic)）。

在思科IOS软件中，静态EIGRP的邻居，是通过使用路由器配置命令（router configuration command）`neighbour <address> <interface>`进行配置的。记住这只是简单的对基础EIGRP配置的补充（this is simply in addition to the basic EIGRP configuration）。下图36.5中给出的简单网络拓扑，将用于静态EIGRP邻居的演示和验证。

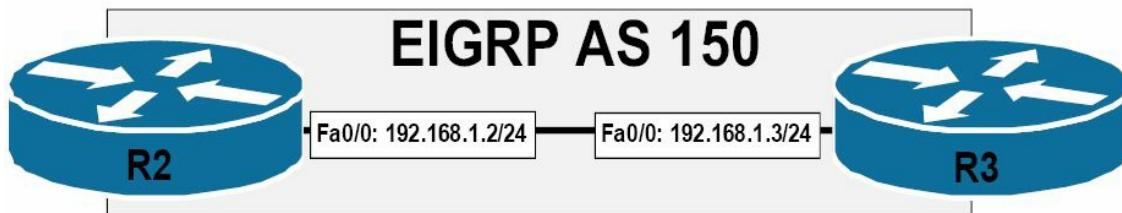


图 36.5 -- 配置静态的EIGRP邻居

参考图36.5中所给出的拓扑，路由器 R2 将作如下配置：

```
R2(config)#router eigrp 150
R2(config-router)#network 192.168.1.0 0.0.0.255
R2(config-router)#neighbor 192.168.1.3 FastEthernet0/0
R2(config-router)#no auto-summary
R2(config-router)#exit
```

而应用于路由器 R3 上的配置则如下：

```
R3(config)#router eigrp 150
R3(config-router)#network 192.168.1.0 0.0.0.255
R3(config-router)#neighbor 192.168.1.2 FastEthernet0/0
R3(config-router)#no auto-summary
R3(config-router)#exit
```

可使用 `show ip eigrp interfaces detail <name>` 命令，对路由器接口使用多播（动态），还是使用单播（静态）数据包来进行邻居发现与维护进行判断。下面的输出对此进行了演示：

```
R2#show ip eigrp interfaces detail FastEthernet0/0
IP-EIGRP interfaces for process 150
      Xmit Queue   Mean    Pacing Time    Multicast      Pending
Interface     Peers Un/Reliable SRTT    Un/Reliable    Flow Timer    Routes
Fa0/0          1       0/0        2        0/1           50            0

Hello interval is 5 sec
Next xmit serial <none>
Un/reliable mcasts: 0/1  Un/reliable ucasts: 3/8
Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 2
Retransmissions sent: 1  Out-of-sequence rcvd: 0
Authentication mode is not set
Use unicast
```

此外，可使用 `show ip eigrp neighbours [detail]` 命令来判断EIGRP邻居的类型。在本课程模块的后面将对此命令进行详细讲解。

## EIGRP的Hello及保持计时器

### EIGRP Hello and Hold Timers

增强的IGRP对不同传输介质，使用不同的Hello及保持计数器。Hello计时器被用于确定发送EIGRP Hello数据包的时间间隔。保持计数器则用于确定下路由器在认为其某个邻居宕机前，要经历的时间（the Hold timer is used to determine the time that will elapse before a router consider an EIGRP neighbour as down）。默认保持时间是Hello间隔的3倍。

在广播网络、点对点串行网络、点对点子接口网络及高于T1线路速率的多点电路网络上，增强的IGRP每5秒发送一次Hello数据包（Enhanced IGRP sends Hello packets every 5 seconds on Broadcast, Point-to-Point Serial, Point-to-Point subinterfaces, and Multipoint circuits greater than T1 speed）。而默认的保持时间就是15秒。在其它链路类型网络上，包括速率低于T1线路的低带宽的WAN链路，EIGRP每60秒发送Hello数据包。在这些链路上的邻居关系保持时间也是Hello间隔的3倍，也就是180秒。

在那些相邻路由器上，不必为了形成邻居关系而要求EIGRP计时器数值保持一致（Enhanced IGRP timer value do not have to be the same on neighbouring routers in order for a neighbour relationship to be established）。此外，对于保持时间是Hello间隔的3倍这一点，也没有强制性要求。这只是一个建议的做法（a recommended guideline），因此可在思科IOS软件中进行手动修改。可使用接口配置命令 `ip hello-interval eigrp <ASN> <secs>`，对EIGRP的Hello时间进行调整，使用借口配置命令 `ip hold-time eigrp <ASN> <secs>`，对EIGRP的保持时间进行调整。

掌握EIGRP中Hello计时器及保持计时器的用法，是重要的。保持时间是在EIGRP的Hello数据包中通告的，同时Hello时间值告诉本地路由器往其邻居发送Hello数据包的频率。而保持时间，则告诉其邻居路由器，在等待多长时间后，就可以宣布其已“死亡”（the hold time, on the other hand, tells the neighbour router(s) of the local router how long to wait before declaring the local router "dead"）。下图36.6演示了EIGRP的Hello数据包，以及保持时间字段（the Hold Time field）：

```
Cisco EIGRP
Version      = 2
Opcode       = 5 (Hello)
Checksum     = 0xee36
Flags        = 0x00000000
Sequence     = 0
Acknowledge   = 0
Autonomous System : 150
-EIGRP Parameters
  Type = 0x0001 (EIGRP Parameters)
  Size = 12 bytes
  K1 = 1
  K2 = 0
  K3 = 1
  K4 = 0
  K5 = 0
  Reserved
  Hold Time = 15
-Software Version: IOS=12.4, EIGRP=1.2
```

图 36.6 -- EIGRP Hello 数据包中的EIGRP保持时间

参考图36.6，除开其它方面，该EIGRP Hello数据包（OPCode 5）包含了所配置的保持时间数值。图36.6中所显示的值15，是一个使用接口配置命令 `ip hold-time eigrp <ASN> <secs>` 所配置的非默认数值。重要的是记住，在Hello数据包中，是不包含Hello时间间隔的。但可使用 `show ip eigrp interfaces detail <name>` 命令，查看到所配置的Hello时间。下面演示了此命令所打印出的信息：

```
R2#show ip eigrp interfaces detail FastEthernet0/0
IP-EIGRP interfaces for process 150
          Xmit Queue  Mean    Pacing Time   Multicast      Pending
Interface    Peers Un/Reliable SRTT  Un/Reliable Flow Timer  Routes
Fa0/0         1     0/0        7       0/1           50            0

Hello interval is 5 sec
Next xmit serial <none>
Un/reliable mcasts: 0/1  Un/reliable ucasts: 2/5
Mcast exceptions: 1  CR packets: 1  ACKs suppressed: 0
Retransmissions sent: 1  Out-of-sequence rcvd: 0
Authentication mode is not set
Use multicast
```

调整默认EIGRP计时器数值的最常见原因，就是要加速路由协议的收敛。比如在一条低速WAN链路上，180秒的保持时间将是一个在EIGRP宣告某台邻居路由器宕机之前，所要等待的相当长的时间。相反，在某些情形中，为了确保得到一个稳定的路由拓扑，就可能有必要在某些高速链路上增加EIGRP计时器数值。在部署某种活动粘滞路由方案（a solution for Stuck-In-Active routes）时，这是常见的做法。本课程模块后面或详细讲解活动粘滞路由。

## EIGRP的邻居表

### EIGRP Neighbour Table

运行EIGRP的路由器使用EIGRP邻居表，来维护有关EIGRP邻居的状态信息。在学习到新近发现的邻居时，该邻居的地址和接口就被记录下来。这对于动态发现的邻居与静态定义的邻居，都是适用的。对每种协议相关模组（Protocol-Dependent Module, PDM, [Wikipedia上的PDM](#)），都有着一个唯一的EIGRP邻居表。

在某个EIGRP邻居发出了一个Hello数据包时，其就通告出一个保持时间，此保持时间就是某台路由器将一个邻居视为可达且运作中的时间长短。在某台路由器收到一个Hello数据包后，该保持时间就开始减少，直到计数到0。此时就会收到另一个Hello数据包，同时保持时间又开始从头减少，同时该过程继续重复。而如果在保持时间内未能收到Hello数据包，那么保持时间就超时了（到0）。在保持时间超时后，弥散更新算法就被告知拓扑的变化，同时EIGRP宣告该邻居已宕机。路由器将打印并记录如下的一条类似消息：

```
%DUAL-5-NBRCHANGE: IP-EIGRP(0) 1: Neighbor 10.1.1.2 (Serial0/0) is down: holding time expired
```

EIGRP邻居表条目还包含了可靠传输协议（the Reliable Transport Protocol, RTP）所需要的信息。EIGRP使用可靠传输协议来确保更新、查询及应答数据包的可靠发送。此外还使用了顺序编号来匹配数据包与确认。EIGRP邻居表条目中记录了从该邻居收到的最后一个顺序编号，以便检测出那些次序被打乱了的数据包（In addition, sequence numbers are also used to match acknowledgments with data packets. The last sequence number received from the neighbour is recorded in order to detect out-of-order packets）。这样做确保了可靠的数据包送达。

**注意：**本课程模块后面详细讲到了RTP。

邻居表包含了每个邻居的一个在可能需要重传时，用于对数据包进行排队的传输清单。此外，在邻居数据结构中还有着一些往返计时器，使用这些计时器来估算出最优重传间隔（the neighbour table includes a transmission list that is used to queue packets for possible retransmission on a per-neighbour basis. Additionally, round-trip timers are kept in the neighbour data structure to estimate an optimal retransmission interval）。所有这些信息都在 `show ip eigrp neighbours` 命令的输出中有打印出来。如下面所示：

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address       Interface   Hold      Uptime     SRTT      RTO      Q      Seq
                (sec)           (ms)          Cnt      Num
  0   192.168.1.3   Fa0/0        14    00:43:08      2      200      0      12
```

对此命令所打印出的信息的掌握，是相当重要的，既是作为对一项核心EIGRP组件能力进行演示的基础，同时也是对EIGRP故障进行排除的基础（It is important to understand the information printed by this command, both as a basis for demonstrating competency on a core EIGRP component and for troubleshooting EIGRP issues）。下表36.2对此命令输出中所包含的那些字段，进行了列出和说明：

表 36.2 -- EIGRP邻居表的各个字段

字段	说明
H	邻居清单（编号），以所学习到的先后顺序，以“0”开始
Address	邻居的IP地址
Interface	经由其学习到邻居的那个接口（the interface via which the neighbour is learned）
Hold	邻居的保持计数器；在其到0时，邻居就已宕机
Uptime	邻居关系已建立了多长时间
SRTT	平滑往返时间（Smooth Round-Trip Time），发送并接收到一个可靠EIGRP数据包所花费的时间
RTO	重传超时（Retransmission Timeout），在未收到一次确认时，路由器重新传送EIGRP可靠数据包所要等待的时间
Q Cnt	等待发送中的EIGRP数据包（Update, Query and Reply）数量
Sequence Number	从该邻居所接收到的上一个EIGRP可靠数据包的顺序编号，用以确保自该邻居接收到的数据包是有序的

尽管 `show ip eigrp neighbours` 命令打印出有关已知EIGRP邻居信息，其在动态发现的邻居和手动配置的邻居上是没有区别的。比如，在路由器 R2 上的该 `show ip eigrp neighbours` 命令的输出表明该路由器有着两个EIGRP邻居关系。在此配置下，其中一个是静态配置的邻居，而另一个则是动态发现的。可以看出，从下面的输出是没法判断出哪个是哪个的：

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address       Interface   Hold      Uptime     SRTT      RTO      Q      Seq
                (sec)           (ms)          Cnt      Num
  1   150.2.2.2   Se0/0        13    00:00:48      153     918      0      4
  0   192.168.1.3   Fa0/0        10    08:33:23      1      200      0      20
```

在路由器同时有着动态发现与静态配置的邻居关系环境中，可以使用 `show ip eigrp neighbours detail` 命令，来判断出哪个邻居是静态配置的，哪个是动态发现的，如下面所示：

```
R2#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
          H   Address       Interface   Hold   Uptime     SRTT      RT0      Q      Seq
                           (sec)           (ms)      Cnt      Num
1   150.2.2.2     Se0/0        11    00:04:22    153     918      0      4
Version 12.3/1.2, Retrans: 0, Retries: 0, Prefixes: 1
0   192.168.1.3   Fa0/0        10    08:33:23      1     200      0      20
Static neighbour
Version 12.4/1.2, Retrans: 0, Retries: 0, Prefixes: 1
```

参考上面的输出，邻居 192.168.1.3 就是手动配置的邻居，而邻居 150.2.2.2 则是动态发现的邻居了。还可以通过使用 `show ip eigrp neighbours static <interface>`，来查看到那些静态邻居，如下所示：

```
R2#show ip eigrp neighbors static FastEthernet0/0
IP-EIGRP neighbors for process 150
Static Address           Interface
192.168.1.3             FastEthernet0/0
```

## 可靠传输协议

### Reliable Transport Protocol

增强的IGRP需要自己的传输协议，来确保数据包的可靠送达。EIGRP使用可靠传输协议，来确保更新（Update）、查询（Query）及应答（Reply）数据包的可靠发送。顺序编号的使用，还确保了EIGRP数据包的有序接收。

当可靠EIGRP数据包发送到某个邻居时，发送路由器期望从接收路由器收到一个表明该数据包已收到的确认。在使用可靠传输协议时，EIGRP维护着一个未确认数据包的传输窗口（a transport window of one unacknowledged packet），这就意味着所发出的所有可靠数据包，都需要进行确认，之后才能发出下一个数据包。发送方路由器将对未收到确认的可靠数据包进行重传，直到其收到一个确认。

但重要的是需注意到，**未经确认的数据包只会重传16次**。如在16次重传后仍没有确认，EIGRP将对该邻居关系进行重置。可靠传输协议使用多播及单播数据包。在诸如以太网这样的广播多路访问网络，EIGRP就会向网段上的每台路由器发送多播数据包，而不是单个数据包（单播）（On Broadcast Multi-Access networks such as Ethernet, EIGRP uses Multicast packets instead of sending an individual packet(Unicast) to each router on the segment）。但在没有从多路访问网段上一台或更多的邻居收到响应时，还是会用单播发送数据包。下面结合图36.7中的图表，对此进行了说明：

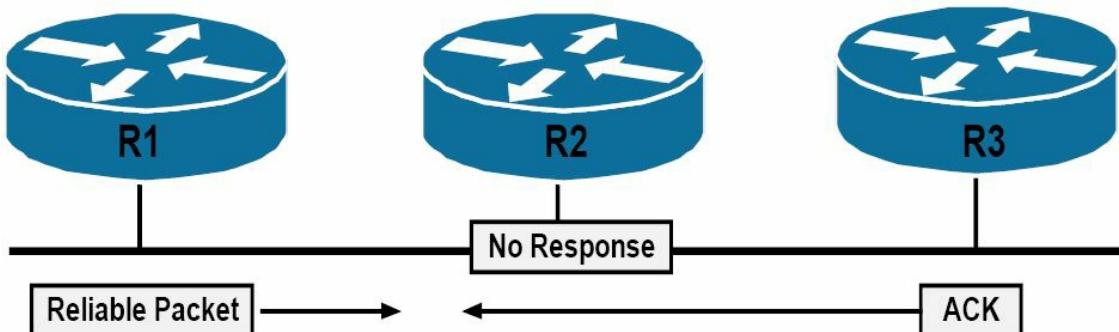


图 36.7 -- EIGRP 可靠传输协议的运作

在图36.7中，路由器 R1、R2 与 R3 位于多路访问网段上的同一子网中。在给定的传输介质下，EIGRP将使用多播，在这些路由器直接发送可靠数据包。这里假定，比如路由器 R1 发出了一个需要确认的数据包给路由器 R2 和 R3。随后 R1 就等待来自 R2 和 R3 收到此数据包的确认。

假设路由器 R3 响应了， R2 却无法对此数据包进行响应。在EIGRP维护了一个未确认数据包传输窗口的情况下，就是说每个发出的单独可靠数据包，在发送下一个可靠数据包之前，都必须要邻居路由器进行显式的确认，而由于路由器 R3 将无法在收到来自 R2 的确认前，发出数据包，这样就在该多路访问网段上出现了一个可能的问题。因此路由器 R3 就间接受到 R2 上故障的影响了。

为了避免这种坑，路由器 R1 将等待连接到该多路访问网段上的以太网接口的多播流计时器超时（To avoid this potential pitfall, R1 will wait for the Multicast Flow Timer(MFT) on the Ethernet interface connected to the Multi-access segment to expire）。多播流计时器，或简单的说就是流计时器（the Flow Timer），是发送方路由器等待自某个组成员的确认数据包的最长时间。在该计数器超时后，路由器 R1 将以多播方式，发出一个特殊的名为顺序TLV的EIGRP数据包（when the timer expires, R1 will Multicast a special EIGRP packet called a Sequence TLV）。此数据包列出了路由器 R2 （也就是例外的那台路由器，the offender），且表明其是一个顺序错乱的多播数据包（this packet lists R2 (the offender) and indicates an out-of-order Multicast packet）。而因为路由器 R3 未被列入到该数据包，所以其就进入到条件接收模式（the Conditional Receive(CR) mode），并继续侦听多播数据包。路由器 R1 此时就使用单播，将该数据包重传给 R2 。重传超时（the Retransmission Timeout, RTO）表示等待那个单播数据包的确认的时间。如在总共16次尝试后，仍没有来自路由器 R2 的响应，EIGRP将重置该邻居。

**注意：**当前的CCNA考试不要求对MFT及RTO有深入了解。

## 各种度量值、弥散更新算法及拓扑表

### Metrics, DUAL, and the Topology Table

在部署EIGRP时，对于路由被真正放入到IP路由表中之前，所用到的EIGRP本身及为其所用到的方方面面的概念、方法及数据结构等的掌握，是重要的。在本小节中，将学到有关EIGRP的综合度量值及其计算方式。还将学习影响度量值计算，及对计算出的度量值进行调整的不同方式（when implementing EIGRP, it is important to understand the various aspects used within and by the protocol before routes are actually placed into the IP routing table. In this section, you will learn about the EIGRP composite metric and how it is calculated. You will also learn about the different ways to influence metric calculation, as well as to adjust the calculated metric）。

在那之后，将学习到弥散更新算法（the Diffusing Update Algorithm, DUAL）与EIGRP的拓扑表。此小节包括了一个有关如何在一台运行着EIGRP的路由器上，将所有这些信息进行配合，以最终产生出IP路由表的讨论。

## EIGRP综合度量值的计算

### EIGRP Composite Metric Calculation

增强的EIGRP使用了一种综合度量值（a composite metric），该度量值包含了以不同的K值所表示的不同变量（Enhanced IGRP uses a composite metric, which includes different variables referred to as the K values）。这些K值是一些常量，用于赋予路径的不同方面以不同的权重，这些路径的不同方面，都可能包含在该综合EIGRP度量值中。这些K值的默认值为 K1=K3=1，K2=K4=K5=0。也就是说，K1与K3被默认被设置为1，同时K2、K4和K5默认被设置为0。

假定在这些默认的K值下，那么完整的EIGRP度量值就可以使用下面的数学公式算出来：

$$[K1 * \text{带宽} + (K2 * \text{带宽}) / (256 - \text{负载}) + K3 * \text{延迟}] * [K5 / (\text{可靠性} + K4)]$$

但在仅有K1和K3有着默认的正值的情况下，默认的EIGRP度量值是由下面的数学公式计算出来的：

$$[(10^7 / \text{路径上的最低带宽}) + (\text{所有延迟总和})] \times 256$$

这实际上就是说，EIGRP使用了到目的网络的路径上的最小带宽，以及总的累积延迟，来计算理由度量值。不过思科IOS软件允许管理员将其它K值设置为非零值，以将其它变量结合到该综合度量值中。通过使用路由器配置命令 `metric weights [tos] k1 k2 k3 k4 k5`，就可完成此操作。

在使用 `metric weights` 命令时，`[tos]` 表示服务类型（Type of Service）。尽管思科IOS软件显示可以使用任何0到8之间的数值，但在撰写本手册时，该字段（`[tos]`）当前却只能被设置为0。而这些K值，就可以被设置为0到255之间的任何数值。通过执行 `show ip protocols` 命令，就可查看默认的这些EIGRP K值。下面的输出对此进行了演示：

```
R2#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    192.168.1.0
  Routing Information Sources:
    Gateway          Distance      Last Update
    192.168.1.3      90           00:00:15
  Distance: internal 90 external 170
```

在对这些EIGRP的K值进行调整时，重要的是记住在EIGRP域中的所有路由器上，都要配置上同样的这些数值。如这些**K值不匹配**，那么**EIGRP的邻居关系就不会建立**。

**注意：**不建议对这些默认的K值进行调整。对这些K值的调整，只应在那些对网络中这类行为造成的后果有扎实了解老练的高级工程师的指导下，或在思科公司技术支持中心的建议下完成。

## 使用接口带宽来影响EIGRP的度量值

### Using Interface Bandwidth to Influence EIGRP Metric Calculation

可通过使用 `bandwidth` 命令对指定到那些单个接口的默认带宽进行调整，从而直接对EIGRP度量值计算施加影响。通过此命令指定的带宽，是以千位（Kilobits）计的。在EIGRP的度量值计算中，带宽也是以千位计的。下图36.8演示了一个由两台路由器通过两条带宽为1544Kbps, 的串行（T1）链路连接所组成的网络。

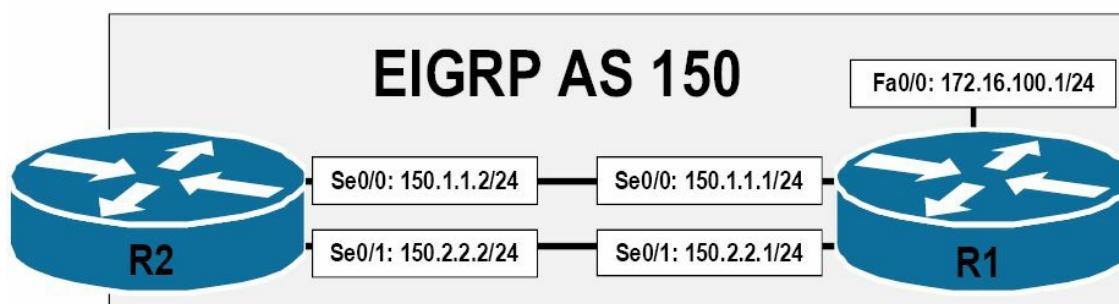


图 36.8 -- EIGRP度量值的带宽修改

参考图36.8中的图示，因为路由器 R1 与 R2 之间两条链路的带宽（及延迟）是相等的，所以从路由器 R2 到子网 172.16.100.0/24 将同时继承到这两条路径的相同EIGRP度量值 (because of the equal bandwidth (and delay) values of the links between R1 and R2 , the same EIGRP metric will be derived for both paths from R2 to the 172.16.100.0/24 subnet)。EIGRP将在这两条链路之间进行流量负载均衡，如下面路由器 R2 上的输出所示：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 2172416, type internal
Redistributing via eigrp 150
Last update from 150.2.2.1 on Serial0/1, 00:48:09 ago
Routing Descriptor Blocks:
  150.2.2.1, from 150.2.2.1, 00:48:09 ago, via Serial0/1
    Route metric is 2172416, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
* 150.1.1.1, from 150.1.1.1, 00:48:09 ago, via Serial0/0
  Route metric is 2172416, traffic share count is 1
  Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops
```

对两条链路中的任何一条的带宽进行调整，都会直接影响到EIGRP对到目的网络路径的度量值计算。**这样的操作，可用于更为大型网络中路径的控制（也就是基于管理员定义的数值与配置，对流量所要采取的路径进行控制）** (Such actions can be used for path control within larger networks(i.e., controlling the path that traffic takes based on administrator-defined values and configurations))。比如这里要令到EIGRP优先使用 Serial0/0 作为前往目的网络的主要路径，而将 Serial0/1 作为到目的网络的备份路径，就要采取两种操作之一。

第一种操作，就是可以增加 Serial0/0 上的带宽值，造成该路径的一个更好（更低）的度量值。那么第二种方法，就是可以降低 Serial0/1 上的带宽值，造成该路径的一个更差（更高）的度量值。两种选项都是可接受的，同时都将达成所需的结果。下面的输出演示了如何将 Serial0/0 上的默认带宽进行降低，从而有效地确保 Serial0/0 作为路由器 R2 及 172.16.100.0/24 网络之间的主要路径。

```
R2(config)#interface Serial0/1
R2(config-if)#bandwidth 1024
R2(config-if)#exit
```

**注意：**如同在第1天指出的，该配置并不意味着接口 Serial0/1 上仅容许1024Kbps速率的流量通过该接口 (As stated in Day 1, this configuration does not mean that Serial0/1 is now capable of only 1024Kbps of throughput through this interface)。

该配置的结果就是接口 Serial0/0 成为路由器 R2 到达目的网络 172.16.100.0/24 网络的主要路径。这在下面的输出中有所演示：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 2172416, type internal
Redistributing via eigrp 150
Last update from 150.1.1.1 on Serial0/0, 00:01:55 ago
Routing Descriptor Blocks:
* 150.1.1.1, from 150.1.1.1, 00:01:55 ago, via Serial0/0
  Route metric is 2172416, traffic share count is 1
  Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

**注意：**这里星号（the asterisk, \*）指向的接口，就是下一数据包要发送出去的接口。而在路由表中有着多个开销相等的路由时，星号的位置就会在这些开销相等的路径之间轮转。

在将EIGRP作为路由协议时，尽管经由 serial0/1 接口的路径未被安装到**路由表**中，重要的是记住该路径并未被完全忽略掉（Although the path via the `Serial0/1` interface is not installed into the routing table, when using EIGRP as the routing protocol, it is important to remember that this path is not completely ignored）。而是该路径被存储在**EIGRP的拓扑表**中，EIGRP的拓扑表包含了到那些远端目网络的主要及替代（备份）路径。本课程模块后面将对EIGRP的拓扑表予以讲解。

**注意：**默认在开启了EIGRP时，其可能会用到高达接口带宽的50%来发送EIGRP本身的数据包（EIGRP是一种非常话痨的协议，所以在可能的带宽使用上进行了限制，EIGRP is a very chatty protocol, so it limits itself in possible bandwidth usage）。EIGRP是基于接口配置命令 `bandwidth`，来判断带宽数量的。因此在对接口带宽数值进行调整时，就要记住这点。而该默认设置，可使用接口配置命令 `ip bandwidth-percent eigrp [ASN] [percentage]`，进行修改。

总的来说，在应用带宽命令 `bandwidth` 对EIGRP的度量值计算施加影响时，重要的是记住，EIGRP会使用到目的网络路径上的最小带宽，以及延迟的累计值，来计算路由度量值（EIGRP uses the minimum bandwidth on the path to a destination network, along with the cumulative delay, to compute routing metrics）。同时还要对网络拓扑有牢固掌握，以对在何处使用 `bandwidth` 命令，从而实现对EIGRP度量值计算的影响。**但在真实世界中，对EIGRP度量值施加影响的首选方法，不是修改带宽，而是修改延迟。**

## 运用接口的延迟来对EIGRP的度量值计算进行影响

### Using Interface Delay to Influence EIGRP Metric Calculation

接口延迟数值，是用微妙来表示的。但在EIGRP度量值计算中用到的延迟数值，是以10微妙计的（in tens of microseconds）。因此，为了计算出EIGRP的度量值，接口上的延迟数值，就必须除以10。下面的表36.3对思科IOS软件中使用到的默认接口带宽及延迟数值，进行了演示：

表 36.3 -- 默认的接口带宽与延迟数值

接口类型	带宽 (Kilobits)	延迟 (Microseconds, us)
以太网 (Ethernet)	10000	1000
快速以太网 (FastEthernet)	100000	100
千兆以太网 (GigabitEthernet)	1000000	10
万兆以太网 (Ten-GigabitEthernet)	10000000	10
串行线路 (T1)	1544	20000
串行线路 (E1)	2048	20000
串行线路 (T3)	44736	200
串行线路 (E3)	34010	200

在对接口的带宽及延迟数值进行计算时，重要的是记住**对接口带宽的调整并不会自动地调整到接口的延迟，相反也是这样。这两个数值是相互独立的。**比如，下面的输出展示了一个快速以太网接口的默认带宽及延迟数值：

```
R2#show interfaces FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0013.1986.0a20 (bia 0013.1986.0a20)
  Internet address is 192.168.1.2/24
    MTU 1500 bytes, BW 100000 Kbit/sec, DLY 100 usec,
      reliability 255/255, txload 1/255, rxload 1/255
...
[Truncated Output]
```

为对此概念进行强化，下面使用接口配置命令 `bandwidth`，将该快速以太网接口的带宽调整为1544Kbps：

```
R2(config)#interface FastEthernet0/0
R2(config-if)#bandwidth 1544
R2(config-if)#exit
```

此时显示在 `show interfaces` 命令的输出中的带宽数值，反应了该已应用下去的配置，但默认的接口延迟数值却仍然保持原来的大小，如下面的输出所示：

```
R2#show interfaces FastEthernet0/0
FastEthernet0/0 is up, line protocol is up
  Hardware is AmdFE, address is 0013.1986.0a20 (bia 0013.1986.0a20)
  Internet address is 192.168.1.2/24
    MTU 1500 bytes, BW 1544 Kbit/sec, DLY 100 usec,
      reliability 255/255, txload 1/255, rxload 1/255
```

而EIGRP所使用的累积延迟，则是在源网络与目的网络之间所有接口延迟的和。对路径中任何一个延迟数值的修改，都会影响到EIGRP度量值的计算。接口延迟数值，是通过使用接口配置命令 `delay` 进行调整的。该数值在用于EIGRP度量值计算时，会除以10。下图36.9演示了一个由两台通过两条有着1544Kbps的带宽，及默认20000微秒的延迟串行 (T1) 链路，连接的路由器所组成的网络。此外，网络 172.16.100.0/24 是直接连接到一个快速以太网接口的，该以太网接口有着默认100000Kbps的带宽，以及默认100微秒的延迟数值：

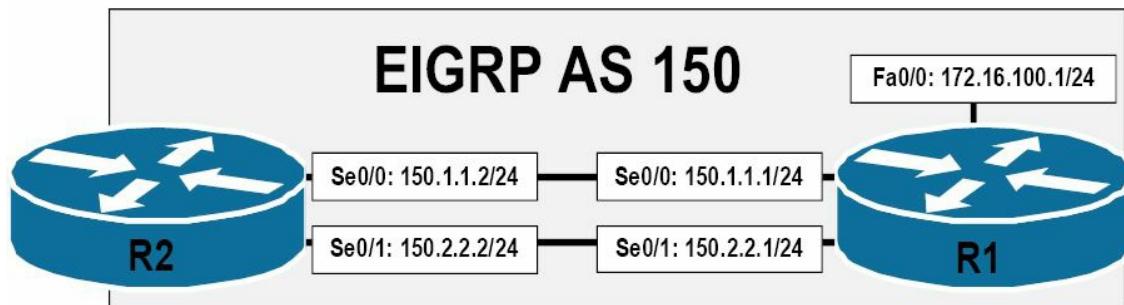


图 36.9 -- EIGRP 度量值中延迟的修改

那么从路由器 R2 到网络 172.16.100.0/24 的EIGRP度量值的计算如下：

$$\text{度量值} = [(10^7/\text{路径上的最小带宽}) + (\text{延迟综合})] \times 256$$

$$\text{度量值} = [(10000000/1544) + (2000+10)] \times 256$$

**注意：**记住在EIGRP度量值计算中，要将接口延迟数值除以10。

**注意：**这里计算出的数值应总是要向下取到最接近的整数。

$$\text{度量值} = [(10000000/1544) + (2000+10)] \times 256$$

$$\text{度量值} = [6476 + 2010] \times 256$$

$$\text{度量值} = 8486 \times 256$$

$$\text{度量值} = 2172416$$

可使用 `show ip route` 命令对此计算进行验证，如下所示：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 2172416, type internal
Redistributing via eigrp 150
Last update from 150.2.2.1 on Serial0/1, 00:03:28 ago
Routing Descriptor Blocks:
  150.2.2.1, from 150.2.2.1, 00:03:28 ago, via Serial0/1
    Route metric is 2172416, traffic share count is 1
    Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
* 150.1.1.1, from 150.1.1.1, 00:03:28 ago, via Serial0/0
  Route metric is 2172416, traffic share count is 1
  Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

与使用 `bandwidth` 命令一样，为了对EIGRP的度量值计算施加影响，我们既可以使用 `delay` 命令对接口延迟数值进行提升，也可以对其进行降低。比如，为了将路由器 R2 配置为使用链路 `Serial0/0` 到达 172.16.100.0/24 网络，而将 `Serial0/1` 仅用作一条备份链路，那么就可以如下将 `Serial0/0` 上的延迟数值进行降低：

```
R2(config)#int s0/0
R2(config-if)#delay 100
R2(config-if)#exit
```

此配置就对经由 `Serial0/0` 的路径的EIGRP度量值进行了调整，如下所示：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 1686016, type internal
Redistributing via eigrp 150
Last update from 150.1.1.1 on Serial0/0, 00:01:09 ago
Routing Descriptor Blocks:
* 150.1.1.1, from 150.1.1.1, 00:01:09 ago, via Serial0/0
  Route metric is 1686016, traffic share count is 1
  Total delay is 1100 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

而经由 `Serial0/1` 的路径，则被保留在拓扑表中，作为到该网络的一条替代路径（an alternate path）。

## 关于弥散更新算法

### The Diffusing Update Algorithm(DUAL)

本小节涉及以下术语：

- 可行距离, the Feasible Distance
- 后继路由器, the Successor
- 报告的距离, the Reported Distance
- 通告的距离, the Advertised Distance, 与报告的距离一样
- 可行后继, the Feasible Successor
- 可行条件, the Feasible Condition

弥散更新算法是EIGRP路由协议的关键所在。弥散更新算法会对从邻居路由器处接收到的所有路由进行观察与比较，然后选出有着到目的网络最低度量值（最优）--该度量值就是可行距离（the Feasible Distance, FD）--的无环回路径，从而得到后续路由（the Successor route）。可行距离既包含了由相连的路由所通告的某网络的度量值，还要加上到那台特定邻居路由器的开销。

由邻居路由器所通告的度量值，就被叫做到目的网路报告的距离（the Reported Distance, RD），又被叫做到目的网络通告的距离（the Advertised Distance, AD, 所以  $RD == AD$ ）。因此，可行距离就包含了报告的距离，加上到达那台特定邻居路由器的开销。那么该后续路由的下一跳路由器，就被叫做是后续路由器（the Successor）。后续路由器就被放入到IP路由表（the IP routing table）及EIGRP拓扑表（the EIGRP topology table）中，并指向到后继路由器。

到相同目的网络的其它那些有着比起后继路由器路径的可行距离高一些的报告距离，却仍然是无环回的那些路由，就被叫做是可行后继路由器路由了（Any other routes to the same destination network that have a lower RD than the FD of the Successor path are guaranteed to be loop-free and are referred to as Feasible Successor(FS) routes, 这里疑似原作者笔误，“lower”应该是“higher”）。这些路由不会被放入到IP路由表中；但它们仍然会与其它后继路由器路由，一起被放入到EIGRP的拓扑表。

为了某条路由成为可行后继路由，则其必须满足可行性条件（the Feasible Condition, FC），该可行性条件仅会在到目的网络的报告距离少于可行距离时，才会发生。在报告距离高于可行距离时，该路由不会被选作可行后继（FS）。EIGRP这么做是为了防止可能出现的环回。下图36.10中所演示的网络拓扑，将会用于对本小节中出现的术语进行解释。

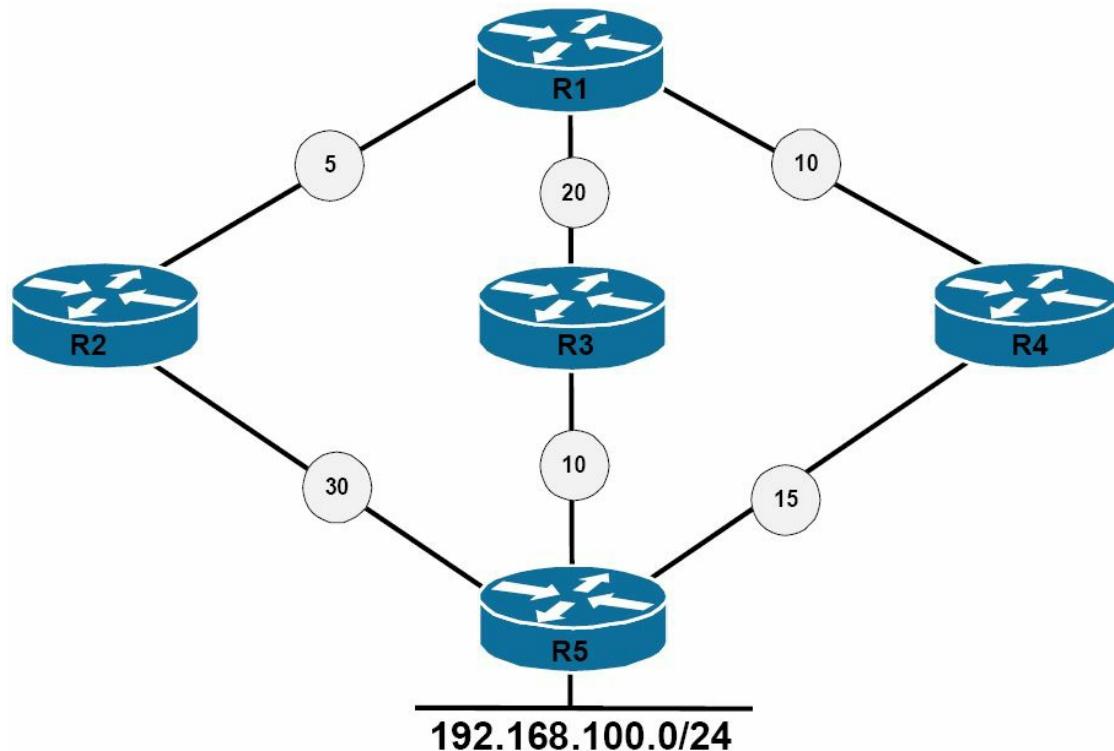


图 36.10 -- 掌握弥散更新算法

参考图36.10, 下表36.4给出了在路由器 R1 上观察到的到 192.168.100.0/24 网络的可行距离及报告距离数值：

表 36.4 -- R1 的那些路径及距离, R1 Paths and Distances

网络路径	R1 的邻居	邻居度量值（报告距离）	R1 的可行距离
R1 - R2 - R5	R2	30	35
R1 - R3 - R5	R3	10	30
R1 - R4 - R5	R4	15	25

基于表36.4中的信息，路由器 R1 将选择经由 R4 的路径，作为后继路由，这是根据该路由的可行距离得出的。此路由将被放入到IP路由表以及EIGRP拓扑表中。路由器 R1 随后将对那些到 192.168.100.0/24 网络的替代路径进行查看。这里邻居路由器 R3 到 192.168.100.0/24 网络的度量值，又被叫做是报告的距离或通告距离，就是10。该距离小于（当前的）可行距离，所以该路由满足到可行条件（FC），那么就被放入到 EIGRP的拓扑表中。而邻居路由器 R2 到 192.168.100.0/24 的度量值为30。该值高于了当前的可行距离 25。此路由则不能满足可行条件，就不被看作是一个可行后继（FS）。但该路由仍然会被放入到EIGRP的拓扑表中。这将在后面的EIGRP拓扑表小节，进行演示。

当某个邻居路由器改变了度量值，或拓扑发生了改变，以及后继路由被移除或改变时，弥散更新算法会检查那些可行后继路由器的路由，在发现了一台可行后继路由器时，弥散更新算法就使用该可行后继路由器，以避免不必要的重新计算路由。执行一次本地运算，节省了CPU处理能力，因为在当前后继或主路由失效时，可行后继路由本身已选出且已经存在了。此过程就叫做**本地运算**（When a neighbor changes a metric, or when a topology change occurs, and the Successor route is removed or changes, DUAL checks for FSs for the route and if one is found, then DUAL uses it to avoid re-computing the route unnecessarily. This is referred to as local computation. Performing a **local computation** saves CPU power because the FS has been chosen and already exists before the Successor or primary route fails）。

而当目的网络的可行后继路由器不存在时，本地路由器将向邻居路由器发出一次查询，对邻居路由器是否有着关于目的网络的信息。如邻居路由器有该信息，同时另一路由器确实有着到目的网络的路由，此时该路由器将执行一次**弥散运算**，以确定出一台新的后继路由器（If the information is available and another neighbour does have a route to the destination network, then the router performs a **diffusing computation** to determine a new Successor）。

## EIGRP的拓扑表

### The EIGRP Topology Table

EIGRP的拓扑表，是由EIGRP的各种协议相关模块，在**弥散更新算法的有限状态机**之上，运算得到的（The EIGRP topology table is populated by **EIGRP PDMs** acted upon by the **DUAL Finite State Machine**）。由已形成邻居关系的EIGRP路由器（neighbouring EIGRP routers）通告的所有目的网络与子网，都被存储在EIGRP拓扑表中。该表包含了后继路由器路由、可行后继路由器路由，甚至那些并不满足可行条件的路由（This includes Successor routes, FS routes, and even routes that have not met the FC）。

正是拓扑表，才令到所有EIGRP路由器对整个网络，有着一致的视图。其还实现了EIGRP网络中的快速收敛。在拓扑表中的每个条目，都包含了某个目的网络及那些通告该目的网络的**那些邻居**。可行距离及通告距离，都被存储在拓扑表中。EIGRP的拓扑表，包含了构建出一个到所有可达网络的距离与矢量集合的所有信息（The topology table allows all EIGRP routers to have a consistent view of the entire network. It also allows for rapid convergence in EIGRP networks. Each individual entry in the topology table contains the destination network and the neighbour(s) that have advertised the destination network. Both the FD and the RD are stored in the topology table. The EIGRP topology table contains the information needed to build a set of distances and vectors to each reachable network），这些信息包括以下内容：

- 到目的网络的最低带宽, the lowest bandwidth on the path to the destination network
- 到目的网络的总/累积延迟, the total or cumulative delay to the destination network
- 到目的网络路径的可靠性, the reliability of the path to the destination network
- 到目的网络路径的负载, the loading of the path to the destination network
- 到目的网络的最大传输单元的最小值, The minimum Maximum Transmission Unit(MTU) to the destination network
- 到目的网络的可行距离, the Feasible Distance to the destination network
- 由邻居路由器所报告的到目的网络的报告距离, the Reported Distance by the neighbour router to the destination network
- 目的网络的路由源（仅针对那些外部路由），The route source(only external routes) of the destination network

**注意：**尽管在拓扑表中包含了最大传输单元（MTU），但EIGRP并不会在实际的度量值计算中使用到该数值。而是该MTU仅简单地作为判断到目的网络数据包大小最小值而被追踪。接口的最大传输单元指定了经某条链路，在无需将数据报或数据包拆分到更小片的情况下，所能传输的数据报最大大小（The interface MTU specifies the largest size of datagram that can be transferred across a certain link without the need of fragmentation, or breaking the datagram or packet into smaller pieces）。

使用 `show ip eigrp topology` 命令，就可查看到EIGRP拓扑表的内容。该命令下可用的选项如下所示：

```
R2#show ip eigrp topology ?
<1-65535>      AS Number
A.B.C.D          IP prefix <network>/<length>, e.g., 192.168.0.0/16
A.B.C.D          Network to display information about
active           Show only active entries
all-links        Show all links in topology table
detail-links    Show all links in topology table
pending          Show only entries pending transmission
summary          Show a summary of the topology table
zero-successors Show only zero successor entries
|                Output modifiers
<cr>
```

不带选项的 `show ip eigrp topology` 命令，将仅打印出那些拓扑表中路由的、且是该路由器上所有开启的 EIGRP 实例的后继路由器及可行后继的信息（The `show ip eigrp topology` command with no options prints only the Successor and Feasible Successor information for routes in the topology table and for all of the EIGRP instances enabled on the router）。下面演示了该命令的打印输出：

```
R2#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(2.2.2.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 150.2.2.0/24, 1 successors, FD is 20512000
      via Connected, Serial0/1
      via 150.1.1.1 (2195456/2169856), Serial0/0
P 150.1.1.0/24, 1 successors, FD is 1683456
      via Connected, Serial0/0
P 172.16.100.0/24, 1 successors, FD is 1686016
      via 150.1.1.1 (1686016/28160), Serial0/0
```

而 `show ip eigrp topology [network]/[prefix]` 及 `show ip eigrp topology [network] [mask]` 两个命令，则将打印出其各自所指定路由的后继路由、可行后继路由以及未能满足可行条件的那些其它路由（The `show ip eigrp topology [network]/[prefix]` and `show ip eigrp topology [network] [mask]` commands print Successor routes, FS routes, and routes that have not met the FC for the route specified in either command）。下面的输出演示了 `show ip eigrp topology [network]/[prefix]` 命令的用法：

```
R2#show ip eigrp topology 172.16.100.0/24
IP-EIGRP (AS 150): Topology entry for 172.16.100.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 1686016
Routing Descriptor Blocks:
150.1.1.1 (Serial0/0), from 150.1.1.1, Send flag is 0x0
  Composite metric is (1686016/28160), Route is Internal
  Vector metric:
    Minimum bandwidth is 1544 Kbit
    Total delay is 1100 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
150.2.2.1 (Serial0/1), from 150.2.2.1, Send flag is 0x0
  Composite metric is (2167998207/2147511807), Route is Internal
  Vector metric:
    Minimum bandwidth is 128 Kbit
    Total delay is 83906179 microseconds
    Reliability is 255/255
    Load is 1/255
    Minimum MTU is 1500
    Hop count is 1
```

在上面的输出中，可以看出经由 `Serial0/1` 的路径并没有满足可行条件 (FC)，因为其报告的距离 (RD) 超过了可行距离 (FD)。这就是该路径没有在 `show ip eigrp topology` 命令的输出中打印出来的原因。而为了判断出那些后继路由、可行后继路由，以及未能满足可行条件的那些路由，就可以使用 `show ip eigrp topology all-links` 命令，而不是对单个前缀进行查看，从而查看到在EIGRP拓扑表中所有前缀的所有可能的路由。下面对此命令的输出进行了演示：

```
R2#show ip eigrp topology all-links
IP-EIGRP Topology Table for AS(150)/ID(2.2.2.2)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 150.2.2.0/24, 1 successors, FD is 20512000, serno 42
    via Connected, Serial0/1
    via 150.1.1.1 (2195456/2169856), Serial0/0
P 150.1.1.0/24, 1 successors, FD is 1683456, serno 32
    via Connected, Serial0/0
    via 150.2.2.1 (21024000/2169856), Serial0/1
P 172.16.100.0/24, 1 successors, FD is 1686016, serno 47
    via 150.1.1.1 (1686016/28160), Serial0/0
    via 150.2.2.1 (2167998207/2147511807), Serial0/1
```

在该EIGRP拓扑表中的路由条目，可能被标记为被动的（Passive, P）或主动的（Active, A）状态。处于被动状态的某条路由，表明EIGRP已经完成了该路由度量值的主动计算，同时可以使用该后继路由将流量转发到目的网络。此状态是拓扑表中所有路由的首选状态。

当后继路由丢失，且路由器为确定出可行后继而发出了一个查询数据包时，那些EIGRP的路由就处于主动状态了。通常情况下，是存在着可行后继的，且EIGRP会将那个可行后继提升为后继路由。那么在此情况下，就无需涉及到网络中其它路由器，该路由器就可以完成收敛。此过程就叫做一次本地运算（Enhanced IGRP routes are in Active state when the Successor route has been lost and the router sends out a Query packet to determine an FS. Usually, an FS is present and EIGRP promotes that to the Successor route. This way, the router converges without involving other routers in the network. This process is referred to as a **local computation**）。

不过，如后继路由已丢失或被移除，而又没有可行后继时，那么路由器就将开始弥散运算（diffused computation）。在弥散运算中，EIGRP将往所有邻居路由器、从除开连接到后继路由的接口外的所有接口发出一次查询。当某台EIGRP邻居路由器收到某条路由的查询时，如那个邻居的EIGRP拓扑表中没有包含该被查询路由的条目，那么这个邻居就立即对该查询应答一条不可达报文，指出经过此邻居处并无这条路由的路径。

而如果邻居路由器上的EIGRP拓扑表已将发出查询的路由器，列为所查询路由的后继路由器，同时存在一条可行后继路由，那么该可行后继路由器路由就被安装起来，同时该邻居路由器对该邻居查询做出其有着一条到已丢失目的网络路由的应答（If the EIGRP topology table on the neighbour lists the router sending the Query as the Successor for that route, and an FS exists, then the FS is installed and the router replies to the neighbour Query that it has a route to the lost destination network）。

但如果该EIGRP拓扑表虽然将发出查询的路由器列为了该路由的后继路由器，却没有可行后继时，收到查询的路由器就会查询其所有的EIGRP邻居，除开那些作为其先前后继路由器而发出的同样接口。在尚未收到对此路由的所有查询的一条应答时，该路由器不会对先前的查询进行响应（However, if the EIGRP topology table lists the router sending the Query as the Successor for this route and there is no FS, then the router queries all of its EIGRP neighbors, except those that were sent out of the same interface as its former Successor. The router will not reply to the Query until it has received a Reply to all Queries that it originated for this route）。

最后，如某个非此目的网络后继的邻居收到了此次查询，随后该路由器以其自己的后继信息予以了应答。而假如这些邻居路由器仍然没有该已丢失的路由信息，那么这些邻居路由器就会向它们自己的邻居路由器发出查询，直到抵达查询边界。所谓查询边界，既可以是网络的末端、分发清单的边界，或者汇总的边界（Finally, if the Query was received from a neighbour that is not the Successor for this destination, then the router replies with its own Successor information. If the neighbouring routers do not have the lost route information, then Queries are sent from those neighbouring routers to their neighbouring routers until the Query boundary is reached. The Query boundary is either the end of the network, the distribute list boundary, or the summarization boundary）。

查询一旦发出，那么发出查询的EIGRP路由器就必须在计算后继路由前，等待完成所有应答的接收。如有任何邻居在三分钟之内没有应答，那么该路由就被称作处于活动粘滞状态（If any neighbour has not replied within three minutes, the route is said to be Stuck-In-Active(SIA)）。而当某条路由成为活动粘滞路由时，该（这些）未对查询进行响应的路由器的邻居关系，就将被重置。在此情况下，可以观察到路由器记录下了如下类似的一条消息：

```
%DUAL-5-NBRCHANGE: IP-EIGRP 150:  
    Neighbor 150.1.1.1(Serial0/0) is down: stuck in active  
%DUAL-3-SIA:  
    Route 172.16.100.0/24 stuck-in-active state in IP-EIGRP 150.  
Cleaning up
```

造成（这些）EIGRP邻居路由器未能对查询进行响应的原因有几种，包括下面这些：

- 该邻居路由器的CPU过载了，因此而无法及时响应
- 该邻居路由器本身就没有关于那条丢失路由的信息
- 电路质量问题，造成数据包丢失
- 某些低带宽链路拥塞，从而造成数据包的延迟

而为了防止因为延迟响应造成的来自其它EIGRP邻居的活动粘滞方面的故障，可使用路由器配置模式中的 `timers active-time` 命令，将本地路由器配置为等待多于默认的三分钟，以接收到返回给其查询数据包的响应。

**注意：**重要的是应注意在对网络中某台路由器上的该默认参数进行修改时，就必须对EIGRP路由域中的所有路由器上的该参数进行修改（It is important to note that if you change this default parameter on one EIGRP router in your network, you must change it on all the other routers within your EIGRP routing domain）。

## 相等开销及不相等开销下的负载均衡

### Equal Cost and Unequal Cost Load Sharing

思科IOS软件对所有路由协议支持默认下至多4条路径的相等开销负载均衡（Cisco IOS software supports equal cost load sharing for a default of up to four paths for all routing protocols）。下面的 `show ip protocols` 命令的输出，对此进行了演示：

```
R2#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    150.1.1.2/32
    150.2.2.2/32
  Routing Information Sources:
    Gateway        Distance      Last Update
    Gateway        Distance      Last Update
    150.2.2.1      90           00:00:52
    150.1.1.1      90           00:00:52
  Distance: internal 90 external 170
```

而又可以使用路由器配置命令 `maximum-paths <1-6>`，对默认的最多4条路径，修改为最多6条相等开销的路径。在进行相等开销的负载均衡时，路由器将负载在所有路径直接进行均匀地分配。有着一个流量分享计数，对每条路径上传输的数据包进行识别（The traffic share count identifies the number of outgoing packets on each path）。在进行相等开销的负载均衡时，单个的完整数据包是在某条单独路径上发出的，如下面的输出所示：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
  Known via "eigrp 150", distance 90, metric 2172416, type internal
  Redistributing via eigrp 150
  Last update from 150.2.2.1 on Serial0/1, 00:04:00 ago
  Routing Descriptor Blocks:
    150.2.2.1, from 150.2.2.1, 00:04:00 ago, via Serial0/1
      Route metric is 2172416, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    * 150.1.1.1, from 150.1.1.1, 00:04:00 ago, via Serial0/0
      Route metric is 2172416, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1544 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

除了相等开销下的负载均衡能力，EIGRP还能完成不相等开销下的负载均衡。这种特别的能力令到EIGRP能够使用那些不相等开销的路径，基于不同的流量分享权重数值，从而将传输中的数据包发送到目的网络（This unique ability allows EIGRP to use unequal cost paths to send outgoing packets to the destination network based on weighted traffic share values）。通过使用路由器配置命令 `variance <multiplier>`，就可以开启不相等开销下的负载均衡特性。

关键字 `<multiplier>` 是一个介于1到128之间的整数。默认的倍数（multiplier）1，就是说不进行不相等开销下的负载均衡。此默认设置在下面的 `show ip protocols` 命令的输出中进行了演示：

```

R2#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    150.1.1.2/32
    150.2.2.2/32
  Routing Information Sources:
    Gateway          Distance      Last Update
    150.2.2.1           90          00:00:52
    150.1.1.1           90          00:00:52
  Distance: internal 90 external 170

```

该倍数是一个可变的整数，告诉路由器在那些有着低于最小度量值乘以该倍数的那些路由上，进行负载均衡  
(The multiplier is a variable integer that tells the router to load share across routes that have a metric that is less than the minimum metric multiplied by the multiplier)。比如，在指定了5的变化时 (specifying a variance of 5)，就指示路由器在那些度量值低于最小度量值5倍的路由上，进行负载均衡。在使用了 variance 命令，同时指定了除开1之外的倍数时，该路由器就将在这些满足条件的路由上，按照各条路由的度量值，成比例的分配流量。也就是说，对于那些有着较低度量值的路由，比起那些有着较高度量值的路由，要经由其发送更多的流量。

下图36.11演示了一个基本的运行EIGRP的网络。其中的路由器 R1 和 R2，是通过背靠背的串行链路连接起来的 (R1 and R2 are connected via back-to-back Serial links)。两台路由器间的链路 150.1.1.0/24，有着1024Kbps的带宽。两台路由器之间的 150.2.2.0/24 链路，有着768Kbps的带宽。路由器 R1 是通过EIGRP将 172.16.100.0/24 前缀，通告给 R2 的。

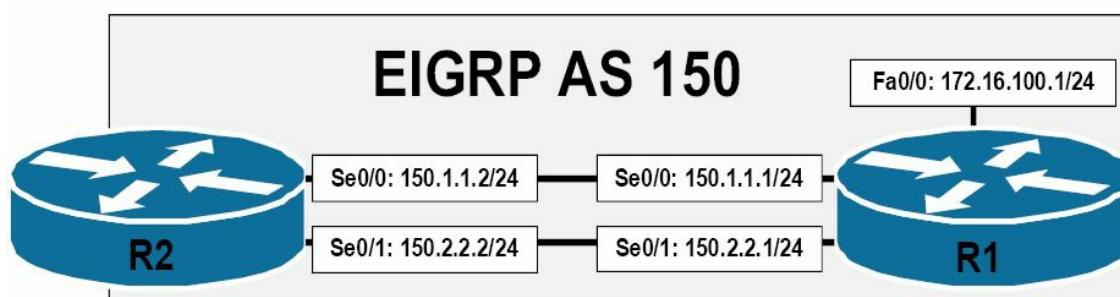


图 36.11 -- 掌握EIGRP的非等价负载均衡，Understanding EIGRP Variance

基于图36.11中所演示的拓扑，下面的输出演示在路由器 R2 上 172.16.100.0/24 前缀的路由表：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 3014400, type internal
Redistributing via eigrp 150
Last update from 150.1.1.1 on Serial0/0, 00:00:11 ago
Routing Descriptor Blocks:
* 150.1.1.1, from 150.1.1.1, 00:00:11 ago, via Serial0/0
  Route metric is 3014400, traffic share count is 1
  Total delay is 20100 microseconds, minimum bandwidth is 1024 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

下面的EIGRP拓扑表，同时显示了后继与可行后继路由：

```
R2#show ip eigrp topology 172.16.100.0 255.255.255.0
IP-EIGRP (AS 150): Topology entry for 172.16.100.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3014400
Routing Descriptor Blocks:
 150.1.1.1 (Serial0/0), from 150.1.1.1, Send flag is 0x0
   Composite metric is (3014400/28160), Route is Internal
   Vector metric:
     Minimum bandwidth is 1024 Kbit
     Total delay is 20100 microseconds
     Reliability is 255/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 1
 150.2.2.1 (Serial0/1), from 150.2.2.1, Send flag is 0x0
   Composite metric is (3847680/28160), Route is Internal
   Vector metric:
     Minimum bandwidth is 768 Kbit
     Total delay is 20100 microseconds
     Reliability is 255/255
     Load is 1/255
     Minimum MTU is 1500
     Hop count is 1
```

这里为了确定要配置到路由器上的非等价负载均衡数值 (the variance value to configure on the router) , 就可使用下面的公式：

非等价负载均衡数值 (Variance) = 考虑要用到路径的度量值中最高的/最优路径的度量值 (最小的度量值)

使用此公式，就可以像下面这样，计算出在路由器 R2 上，所要配置的非等价负载均衡数值：

非等价负载均衡数值 (Variance) = 考虑要用到路径的度量值中最高的/最优路径的度量值 (最小的度量值)

非等价负载均衡数值 (Variance) =  $3847680/3014400$

非等价负载均衡数值 (Variance) = 1.28

随后必须对此数值进行向上取整，这里就是2了。那么就可以在路由器配置模式中，通过应用将下面的配置，将路由器 R2 配置为进行非等价的负载均衡了：

```
R2(config)#router eigrp 150
R2(config-router)#variance 2
R2(config-router)#exit
```

而根据此配置，此时 172.16.100.0/24 前缀的路由表条目就变成如下所示了：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
Known via "eigrp 150", distance 90, metric 3014400, type internal
Redistributing via eigrp 150
Last update from 150.2.2.1 on Serial0/1, 00:00:36 ago
Routing Descriptor Blocks:
  150.2.2.1, from 150.2.2.1, 00:00:36 ago, via Serial0/1
    Route metric is 3847680, traffic share count is 47
    Total delay is 20100 microseconds, minimum bandwidth is 768 Kbit
    Reliability 255/255, minimum MTU 1500 bytes
    Loading 1/255, Hops 1
* 150.1.1.1, from 150.1.1.1, 00:00:36 ago, via Serial0/0
  Route metric is 3014400, traffic share count is 60
  Total delay is 20100 microseconds, minimum bandwidth is 1024 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

其中的流量分配计数（The traffic share count）表明，每从 serial0/0 转发60个数据包，路由器就将从 serial0/1 转发47个数据包。数据包的转发，是依两条路径的路由度量值的比例完成的。这是在应用了 variance 命令后的默认行为。而通过路由器配置命令 **traffic-share balanced**（the **traffic-share balanced router configuration command**），就可以开启此智能流量分配功能（this intelligent traffic sharing functionality），该命令无需显式配置（默认是开启的）。

**注意：**该 **traffic-share balanced** 命令默认是开启的，且就算对其进行了显式配置，其也不会在运行配置中出现。这一点在下面进行了演示：

```
R2(config)#router eigrp 150
R2(config-router)#vari 2
R2(config-router)#traffic-share balanced
R2(config-router)#exit
R2(config)#do show run | begin router
router eigrp 150
variance 2
network 150.1.1.2 0.0.0.0
network 150.2.2.2 0.0.0.0
no auto-summary
```

如同本小节前面指出的那样，在使用了 **variance** 命令时，所有那些满足了可行条件、且有着低于最小度量值乘以那个倍数的路径，都将被安装到路由表中。路由器随后将使用到所有路径，并依据其各自的路由度量值来按比例地进行负载均衡。

在某些情况下，可能打算将那些替代路由，比如可行后继路，预先放入到路由表中，却只在后继路由被移除时，才使用它们。执行这类操作的典型目的，是在开启EIGRP的网络中降低收敛时间。要弄明白此概念，就要回忆一下，路由器默认是只将后继路由放入到路由表中的。而在后继路由已不可用时，可行后继路由就被提升为后继路由。随后该路由才被装入到路由表中，作为到目的网络的主要路径。

可将路由器配置命令 **traffic-share min across-interfaces**，与 **variance** 命令结合使用，从而将那些有着少于最小度量值乘以所指定倍数所有路由，都安装到路由表中，却仅使用最小（最优）度量值的那条路由（后继路由），来转发数据包，直到该路由变为不可用。此配置的主要目的，就是在主路由丢失时，保证替代路由已经处于路由表中，且立即可用（从而达到减少收敛时间的目的）。

下面的配置示例，使用了以上图36.11中展示的拓扑，用于对如何将路由器配置为把那些度量值少于最小度量值两倍的路由，放入到路由表中，而仅使用有着最低度量值的那条路由（后继路由）来转发数据包：

```
R2(config)#router eigrp 150
R2(config-router)#vari 2
R2(config-router)#traffic-share min across-interfaces
R2(config-router)#exit
```

此配置造成路由表中 172.16.100.0/24 前缀的以下输出：

```
R2#show ip route 172.16.100.0 255.255.255.0
Routing entry for 172.16.100.0/24
  Known via "eigrp 150", distance 90, metric 3014400, type internal
  Redistributing via eigrp 150
  Last update from 150.2.2.1 on Serial0/1, 00:09:01 ago
  Routing Descriptor Blocks:
    150.2.2.1, from 150.2.2.1, 00:09:01 ago, via Serial0/1
      Route metric is 3847680, traffic share count is 0
      Total delay is 20100 microseconds, minimum bandwidth is 768 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
    * 150.1.1.1, from 150.1.1.1, 00:09:01 ago, via Serial0/0
      Route metric is 3014400, traffic share count is 1
      Total delay is 20100 microseconds, minimum bandwidth is 1024 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 1
```

如上面的输出所示，基于该种不同开销下的负载均衡配置，两条不同度量值的路由，已被安装到路由表中。但注意到经由 serial0/1 的流量分享计数（the traffic share count）是0，而经由 serial0/0 的计数为1。这就意味着，尽管该路由条目已被安装到路由表中，该路由器不会通过 Serial0/1，向 172.16.100.0/24 网络发送任何数据包，除非经由 Serial0/0 的路径不再可用。

## 使用EIGRP作默认路由

### Default Routing Using EIGRP

在将**最终网关**或网络动态地通告给路由域中的其它路由器方面，EIGRP支持多种不同方式。最终网关，或**默认路由**，是在目的网络未在路由表中特别列出时，路由器用于引导流量的一种方式（Enhanced IGRP supports numerous ways to advertise dynamically the gateway or network of last resort to other routers within the routing domain. A **gateway of last resort**, or default route, is a method for the router to direct traffic when the destination network is not specifically listed in the routing table）。而在此种情形下，路由器引导流量的方式有：

- 使用 `ip default-network` 命令, using the `ip default-network` command
- 使用 `network` 命令来对网络 `0.0.0.0/0` 通告, using the `network` command to advertise network `0.0.0.0/0`
- 对默认静态路由进行重分发, Redistributing the default static route
- 使用命令 `ip summary-address eigrp [asn] [network] [mask]` , using the `ip summary-address eigrp [asn] [network] [mask]` command

而第一种，使用 `ip default-network` 命令，被认为是一种EIGRP下对默认路由进行动态通告的过时方式。但因为在当前的IOS软件中仍然支持这种方式，所以这里有必要提到。

`ip default-network` 配置命令通过把一个星号，插入到路由表中某个网络旁边，而将该网络标记为默认网络。那些目的地没有明确路由表条目的流量，就会被路由器转发到这个网络（Traffic for destinations to which there is no specific routing table entry is then forwarded by the router to this network）。下面参考图36.12中的EIGRP拓扑，对此种部署进行了演示：

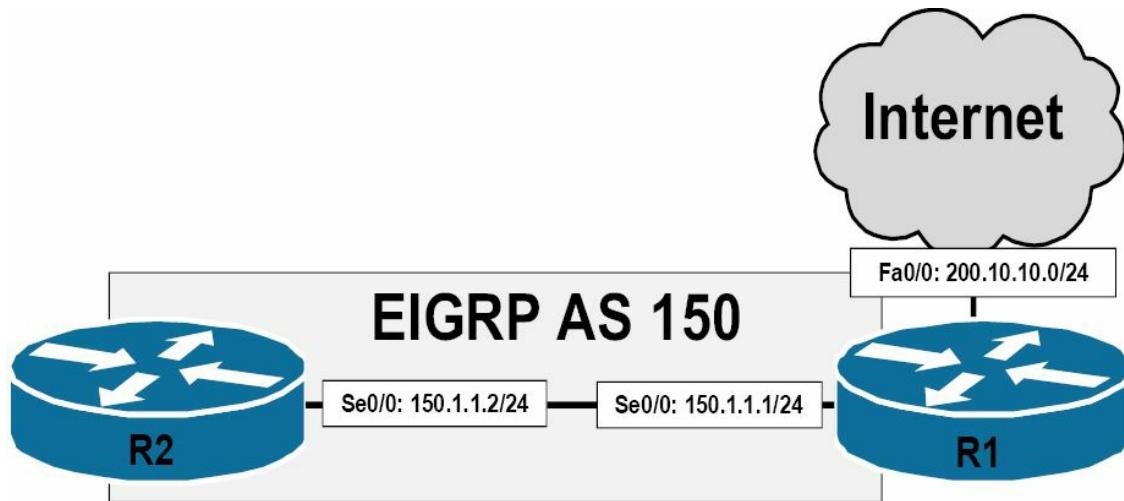


图 36.12 -- EIGRP 的默认路由

参考图36.12, 假设子网 200.10.10.0/24 是连接到互联网的。该子网位于路由器 R1 的 Fastethernet0/0 侧。路由器 R1 与 R2 相应地通过一条背靠背的串行连接相连。两台路由器都是处于EIGRP AS 150中。为了将 200.10.10.0/24 标记为最终网络, 就要在路由器 R1 上进行如下配置:

```
R1(config)#router eigrp 150
R1(config-router)#network 200.10.10.0 0.0.0.255
R1(config-router)#exit
R1(config)#ip default-network 200.10.10.0
R1(config)#exit
```

基于此配置, 路由器 R2 就会将 200.10.10.0/24 作为最终网络接收下来, 如下所示:

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 150.2.2.1 to network 200.10.10.0
D*   200.10.10.0/24 [90/2172416] via 150.2.2.1, 00:01:03, Serial0/0
      150.1.0.0/24 is subnetted, 1 subnets
C       150.1.1.0 is directly connected, Serial0/0
```

`network` 命令可用于对某条既有的指向某个物理或逻辑接口, 通常是 `Null0` 接口的静态默认路由, 进行通告 (The `network` command can be used to advertise an existing static default route point to either a physical or a logical interface, typically the `Null0` interface)。

**注意:** `Null0` 接口是路由器上的一个虚拟接口, 会将路由至该接口的所有流量进行抛弃处理。如果有了一条指向 `Null0` 的静态路由, 那么所有以该静态路由中所指定网络为目的的流量, 都将被简单地做丢弃处理。可将 `Null0` 接口看作是一个黑洞: 数据包进入了, 但不会有任何东西离开那里。其基本上就是路由器上的一个数位垃圾桶 (It is essentially a bit-bucket on the router)。

参考上面的图36.12, `network` 命令与一条既有默认静态路由的结合使用, 在以下路由器 R1 的配置中进行了演示:

```
R1(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
R1(config)#router eigrp 150
R1(config-router)#network 0.0.0.0
R1(config-router)#exit
```

基于此种配置，下面的输出，演示了路由器 R2 上的IP路由表：

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
      Gateway of last resort is 150.1.1.1 to network 0.0.0.0
D    200.10.10.0/24 [90/2172416] via 150.1.1.1, 00:01:11, Serial0/0
      150.1.0.0/24 is subnetted, 1 subnets
C          150.1.1.0 is directly connected, Serial0/0
D*   0.0.0.0/0 [90/2172416] via 150.1.1.1, 00:00:43, Serial0/0
```

尽管路由重分发 (route redistribution) 不是CCNA考试部分，这里仍将对其进行一个概述。路由重分发正是通过EIGRP对一条默认路由进行通告的第三种方式。为将现有的静态默认路由重分发到EIGRP中，就要使用路由器配置命令 `redistribute static metric [bandwidth] [delay] [reliability] [load] [MTU]`。用于本小节中前面那些输出的同一网络拓扑，将用于对这种方式的部署进行演示，如下图36.13所示：

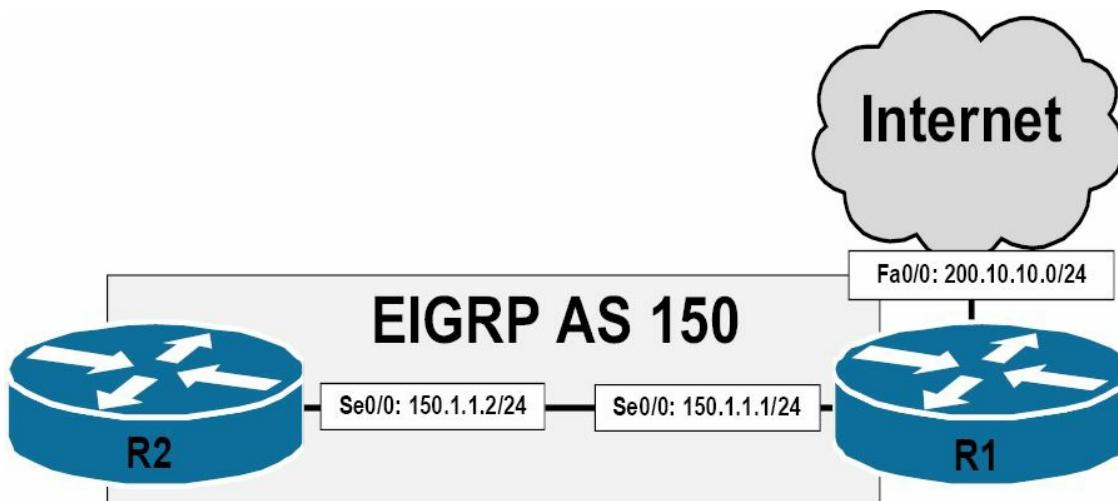


图 36.13 -- EIGRP 的默认路由 (续)

参考图36.13，该图与图36.12是一样的，在路由器 R1 上完成以下配置：

```
R1(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet0/0
R1(config)#router eigrp 150
R1(config-router)#redistribute static metric 100000 100 255 1 1500
R1(config-router)#exit
```

**注意：** 这里度量值中所用到的那些数值，可从接口上进行继承到，或可以在使用此命令时指定想要的任意数值。

基于此种配置，路由器 R2 上的路由表就如下所示了：

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 150.1.1.1 to network 0.0.0.0
      150.1.0.0/24 is subnetted, 1 subnets
C          150.1.1.0 is directly connected, Serial0/0
D*EX 0.0.0.0/0 [170/2195456] via 150.1.1.1, 00:01:16, Serial0/0
```

因为该路由是在路由器 R1 上被重分发到EIGRP中的，所以如同上面所反应出的，其就是一条外部EIGRP 路由了。对于那些外部路由，EIGRP拓扑表中就包含了诸如该路由所起源的路由器、该路由是为何种协议接收的，以及该外部路由的度量值等信息。下面的输出对此进行了演示：

```
R2#show ip eigrp topology 0.0.0.0/0
IP-EIGRP (AS 150): Topology entry for 0.0.0.0/0
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2195456
  Routing Descriptor Blocks:
    150.1.1.1 (Serial0/0), from 150.1.1.1, Send flag is 0x0
      Composite metric is (2195456/51200), Route is External
      Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 21000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
      External data:
        Originating router is 1.1.1.1
        AS number of route is 0
        External protocol is Static, external metric is 0
        Administrator tag is 0 (0x00000000)
        Exterior flag is set
```

可以看出该默认路由是一条在路由器 R1 上重分发到EIGRP中的静态路由。该路由有着度量值0。此外还可以看出路由器 R1 的EIGRP路由器ID (the EIGRP router ID, RID) 为 1.1.1.1。

对默认路由进行通告的最后一种方法，就是使用接口配置命令 `ip summary-address eigrp [asn] [network] [mask]` 了。在本课程模块的后面，将对EIGRP的路由汇总 (EIGRP route summarization) 进行详细讲解。这里要着重于在应用EIGRP时，使用此命令来对默认路由进行通告的用途。

参考上面图36.13中所演示的网络拓扑图示，这里使用了接口配置命令 `ip summary-address eigrp [asn] [network] [mask]`，将路由器 R1 配置为把默认路由通告给 R2，如下所示：

```
R1(config)#interface Serial0/0
R1(config-if)#description 'Back-to-Back Serial Connection To R2 Serial0/0'
R1(config-if)#ip summary-address eigrp 150 0.0.0.0 0.0.0.0
R1(config-if)#exit
```

使用这个命令的主要优势在于，无需为了让EIGRP将网络 0.0.0.0/0 通告给邻居路由器，而将某条默认路由或某个默认网络放入到路由表中 (The primary advantage to using this command is that a default route or network does not need to exist in the routing table in order for EIGRP to advertise network 0.0.0.0/0 to its neighbour routers)。在执行了此命令后，本地路由器将生成一条到 Null0 接口的汇总路由，并将该条目标记为备选默认路由 (the candidate default route)。如下所示：

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 0.0.0.0 to network 0.0.0.0
      150.1.0.0/24 is subnetted, 1 subnets
C        150.1.1.0 is directly connected, Serial0/0
D*    0.0.0.0/0 is a summary, 00:02:26, Null0
```

于是在路由器 R2 上就作为一条内部EIGRP路由 (an internal EIGRP route) , 接收到该汇总路由, 如下所示:

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is 150.1.1.1 to network 0.0.0.0
      150.1.0.0/24 is subnetted, 1 subnets
C        150.1.1.0 is directly connected, Serial0/0
D*    0.0.0.0/0 [90/2297856] via 150.1.1.1, 00:03:07, Serial0/0
```

## EIGRP网络中的水平分割

### Split Horizon in EIGRP Networks

前面已获悉水平分割是一项距离矢量协议的特性, 该特性强制路由信息无法从接收到的接口再发送回去, 这样做就阻止了路由信息重新通告到学习到其的同一接口 (Previously, you learned that split horizon is a Distance Vector protocol feature mandating that routing information cannot be sent back out of the same interface through which it was received. This prevents the re-advertising of information back to the source from which it was learned) 。虽然这个特征是一种很好的环回防止机制 (a great loop prevention mechanism) , 但其也是一项明显的缺陷, 特别是在星形网络中 (especially in hub-and-spoke networks) 。为更好地理解此特性的不足, 就要参考下面图36.14中的EIGRP星形网络:

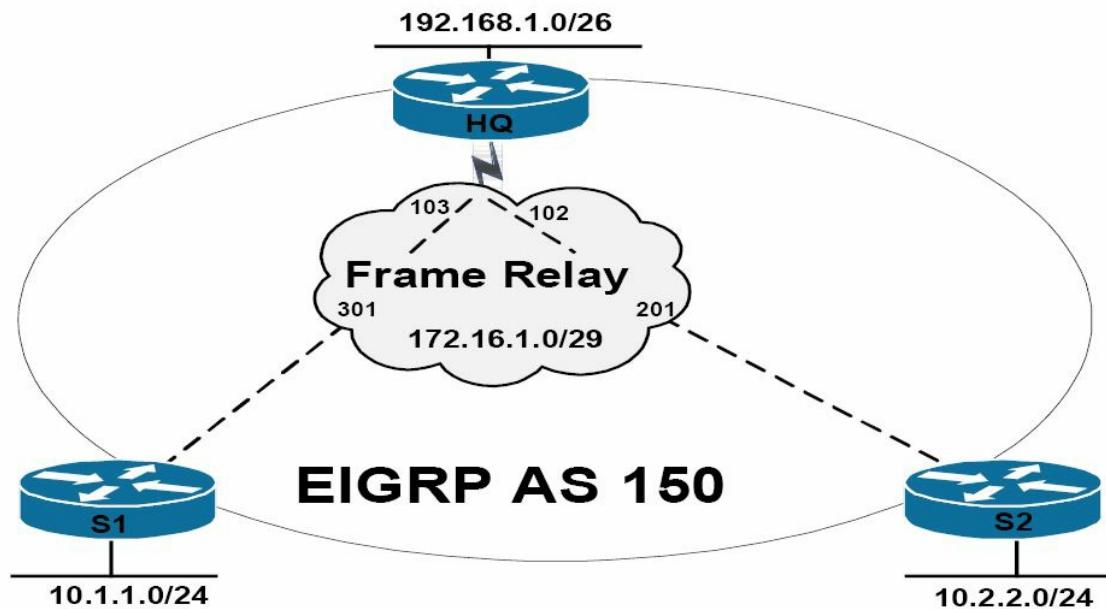


图 36.14 -- EIGRP 的水平分割

图36.14中的拓扑，演示了一个典型的星形网络，其中的总部路由器（router HQ）是中心路由器（the hub router），路由器 s1 与 s2 则是分支路由器(the spoke router)。在该帧中继广域网上，每台分支路由器都有着一个在部分网状拓扑（a partial-mesh topology）中，各自与中心路由器之间的，所提供的数据链路层连接标识（Data Link Connection Identifier，这是个6位标识，表示正在进行的客户和服务器之间的连接。用于RFCOMM 层。On the Frame Relay WAN, each spoke router has a single DLCI provisioned between itself and the HQ router in a partial-mesh topology）。下面对这些路由器上的帧中继配置进行了检查：

```

HQ#show frame-relay map
Serial0/0 (up): ip 172.16.1.2 dlci 102(0x66,0x1860), static,
                  broadcast,
                  CISCO, status defined, active
Serial0/0 (up): ip 172.16.1.1 dlci 103(0x67,0x1870), static,
                  broadcast,
                  CISCO, status defined, active

S1#show frame-relay map
Serial0/0 (up): ip 172.16.1.2 dlci 301(0x12D,0x48D0), static,
                  broadcast,
                  CISCO, status defined, active
Serial0/0 (up): ip 172.16.1.3 dlci 301(0x12D,0x48D0), static,
                  broadcast,
                  CISCO, status defined, active

S2#show frame-relay map
Serial0/0 (up): ip 172.16.1.1 dlci 201(0xC9,0x3090), static,
                  broadcast,
                  CISCO, status defined, active
Serial0/0 (up): ip 172.16.1.3 dlci 201(0xC9,0x3090), static,
                  broadcast,
                  CISCO, status defined, active

```

在后面的广域网章节，将涉及到帧中继。这里在所有三台路由器上都开启了EIGRP，使用了自治系统编号 150。下面的输出演示了中心路由器与分支路由器之间的EIGRP邻居关系：

```
HQ#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address           Interface      Hold  Uptime     SRTT    RTO    Q    Seq
                (sec)          (ms)          Cnt Num
  1   172.16.1.1       Se0/0          165   00:01:07   24     200    0    2
  0   172.16.1.2       Se0/0          153   00:01:25  124    744    0    2
```

下面的输出对第一台分支路由器 s1 与中心路由器之间的EIGRP邻居关系：

```
S1#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address           Interface      Hold  Uptime     SRTT    RTO    Q    Seq
                (sec)          (ms)          Cnt Num
  0   172.16.1.3       Se0/0          128   00:00:53  911    5000   0    4
```

下面的输出对第二台分支路由器 s2 与中心路由器之间的EIGRP邻居关系：

```
S2#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address           Interface      Hold  Uptime     SRTT    RTO    Q    Seq
                (sec)          (ms)          Cnt Num
  0   172.16.1.3       Se0/0          156   00:02:20    8     200    0    4
```

默认EIGRP的水平分割是开启的，但在**局部网状网络的非广播多路访问网络**上，EIGRP的水平分割是不需要的（By default, EIGRP split horizon is enabled, which is undesirable in **partial-mesh NBMA networks**）。这就意味着对于那些在 `Serial0/0` 上学习到的路由信息，中心路由器不会再从相同接口（`Serial0/0`）进行通告。而这种默认行为的效果，就是中心路由器不会将接收自路由器 s1 的 `10.1.1.0/24` 前缀，通告给 s2，因为该路由是经由 `Serial0/0` 接口接收到的，而水平分割特性阻止了该路由器对从该接口学习到的信息，在通告出同一接口。这一点对于中心路由器接收自路由器 s2 的 `10.2.2.0/24` 前缀同样适用。

这种默认行为意味着尽管中心路由器注意到了这两条前缀，但分支路由器却只有局部的路由表。中心路由器上的路由表如下：

```
HQ#show ip route eigrp
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
  D        10.1.1.0/24 [90/2195456] via 172.16.1.1, 00:12:04, Serial0/0
  D        10.2.2.0/24 [90/2195456] via 172.16.1.2, 00:12:06, Serial0/0
```

分支路由器 s1 上的路由表如下：

```
S1#show ip route eigrp
  192.168.1.0/26 is subnetted, 1 subnets
  D        192.168.1.0 [90/2195456] via 172.16.1.3, 00:10:53, Serial0/0
```

分支路由器 s2 上的路由表如下：

```
S2#show ip route eigrp
  192.168.1.0/26 is subnetted, 1 subnets
  D        192.168.1.0 [90/2195456] via 172.16.1.3, 00:10:55, Serial0/0
```

这种默认行为的结果，就是尽管中心（总部）路由器能够到达两个分支路由器的网络，但两台分支路由器却都无法到达对方的网络。解决此种情况的方法有几种，如下所示：

- 在中心路由器上关闭水平分割, Disabling split horizon on the HQ(hub) router
- 从中心路由器往分支路由器通告一条默认路由, Advertising a default route from the HQ router to the spoke routers
- 在路由器上手动配置EIGRP邻居, Manually configuring EIGRP neighbours on the routers

通过在中心路由器的接口级别使用接口配置命令 `no ip split-horizon eigrp [AS]`，就可以完成关闭水平分割。命令 `show ip split-horizon interface_name` 不会显示EIGRP的水平分割状态，因为该命令是作用于RIP的。所以要查看到EIGRP的水平分割状态，就必须对接口配置部分进行检查（也就是执行 `show run interface_name` 命令）。参考上面图36.14中所演示的网络拓扑，此接口配置命令就应在中心路由器上的 `Serial0/0` 接口上应用。应像下面这样完成：

```
HQ(config)#interface Serial0/0
HQ(config-if)#no ip split-horizon eigrp 150
```

在水平分割关闭后，中心路由器就可以把在某个接口上接收到的路由信息，再在该接口上发送出去了。比如，分支路由器 `s2` 上的路由表现在就显示了一个由分支 `s1` 通告给中心路由器的 `10.1.1.0/24` 前缀了：

```
S2#show ip route eigrp
 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 D       10.1.1.0/24 [90/2707456] via 172.16.1.3, 00:00:47, Serial0/0
 192.168.1.0/26 is subnetted, 1 subnets
 D       192.168.1.0 [90/2195456] via 172.16.1.3, 00:00:47, Serial0/0
```

可使用一个简单的从分支路由器 `s2` 到 `10.1.1.0/24` 的 `ping` 操作，对连通性进行检查，如下所示：

```
S2#ping 10.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 24/27/32 ms
```

关闭水平分割的第二种方法，就是简单地从中心路由器往分支路由器通告一条默认路由。在这种情况下，可以将接口配置命令 `ip summary-address eigrp 150 0.0.0.0 0.0.0.0`，应用到中心路由器的 `Serial0/0` 接口。这样做就令到分支路由器能够经由中心路由器，到达对方的网络，中心路由器包含了完整的路由表，从而消除了对关闭水平分割的需求。

而关闭水平分割的第三种替代方式，就是在**所有路由器**上，使用路由器配置命令 `neighbor` 来手动配置一些EIGRP的邻居语句。因为在使用了此种配置后，邻居间的更新就是单播的了，所以水平分割的限制条件就得以消除。此选项在小型网络中用起来是不错的；但随着网络的增大，以及分支路由器数量的增加，工作量也会增加。

因为EIGRP默认路由与静态邻居方面的配置在本课程模块的前面小节都已详细介绍过了，所以这里为了简洁期间，这些特性的配置就做了省略处理。

## EIGRP的路由汇总

### EIGRP Route Summarisation

路由汇总特性减少了路由器必须处理的信息量，这样就实现了网络更快的收敛。通过将网络中具体区域的详细拓扑信息的隐藏，汇总还对由网络变动而造成的影响范围有所限制。最后，如同本课程模块前面所指出的那样，汇总还用于**EIGRP查询边界**的定义，而**EIGRP查询边界又支持两种类型的路由汇总**（Route summarisation reduces the amount of information that routers must process, which allows for faster

convergence within the network. Summarisation also restricts the size of the area that is affected by network changes by hiding detailed topology information from certain areas within the network. Finally, as was stated earlier in this module, summarisation is used to define a **Query boundary for EIGRP, which supports two types of route summarisation**），如下所示：

- 自动的路由汇总, Automatic route summarisation
- 手动的路由汇总, Manual route summarisation

默认情况下，当在路由器上开启了EIGRP时，自动路由汇总就生效了。这是通过使用 `auto-summary` 命令应用的。该命令允许EIGRP在**有类边界** (at classful boundaries) 进行自动路由汇总。参考下图36.15中的网络拓扑，对此默认特性的运作进行了演示：

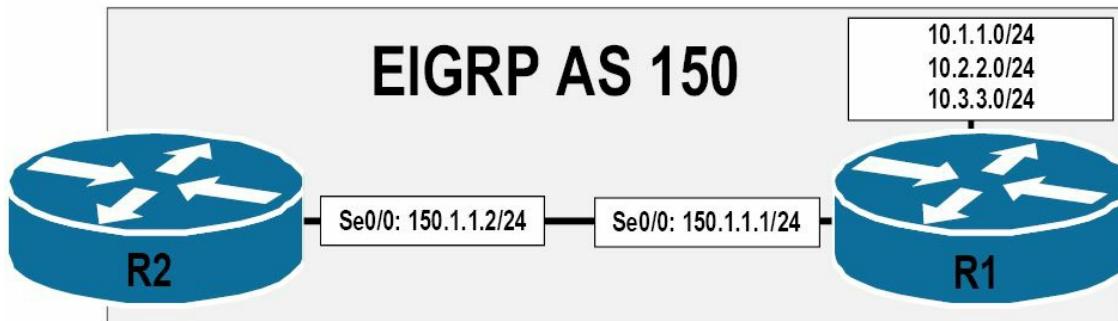


图 36.15 -- EIGRP 的自动路由汇总

参考图36.15中的EIGRP网络，路由器 R1 与 R2 运行着EIGRP，使用着自治系统编号 150。10.1.1.0/24、10.2.2.0/24 与 10.3.3.0/24 子网都是直接连接到路由器 R1 上的。路由器 R1 正将这些路由通告给 R2。路由器 R1 与 R2 是使用了一条在 150.1.1.0/24 子网（该子网是一个与 10.1.1.0/24、10.2.2.0/24 及 10.3.3.0/24 子网都不相同的主要网络(major network))，上的背靠背的串行链路连接起来的。根据连接在这些路由器上的网络，默认EIGRP将执行自动汇总，如下所示：

- 10.1.1.0/24、10.2.2.0/24 与 10.3.3.0/24 子网将被汇总到 10.0.0.0/8
- 150.1.1.0/24 子网将被汇总到 150.1.0.0/16

可以查看 `show ip protocols` 命令的输出，来对此默认行为进行验证。路由器 R1 上该命令的输出如下所示：

```

R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    150.1.0.0/16 for Loopback1, Loopback2, Loopback3
      Summarizing with metric 2169856
    10.0.0.0/8 for Serial0/0
      Summarizing with metric 128256
  Maximum path: 4
  Routing for Networks:
    10.1.1.0/24
    10.2.2.0/24
    10.3.3.0/24
    150.1.1.0/24
  Routing Information Sources:
    Gateway          Distance      Last Update
    (this router)      90          00:03:12
    150.1.1.2        90          00:03:12
  Distance: internal 90 external 170

```

在上面的输出中，`10.1.1.0/24`、`10.2.2.0/24` 与 `10.3.3.0/24` 子网已被自动汇总到 `10.0.0.0/8`。该汇总地址就被通告出 `Serial0/0` 接口。而 `150.1.1.0/24` 子网已被汇总到 `150.1.0.0/16`。该汇总地址就被通告出 `Loopback1`、`Loopback2` 与 `Loopback3` 这三个环回接口。这里要记住，默认EIGRP将在所有EIGRP路由开启的接口上，发出路由更新（Remember, by default, EIGRP will send out updates on all interfaces for which EIGRP routing is enabled）。

参考上面的打印输出，可以看到环回接口上所发出的更新，就是一种资源的浪费，因为设备是无法物理连接到路由器环回接口上去监听此类更新的（Referencing the output printed above, you can see that sending updates on a Loopback interface is a waste of resource because a device cannot be connected physically to a router Loopback interface listening for such updates）。那么就可以通告使用路由器配置命令 `passive-interface`，来关闭此默认行为，如下所示：

```

R1(config)#router eigrp 150
R1(config-router)#passive-interface Loopback1
R1(config-router)#passive-interface Loopback2
R1(config-router)#passive-interface Loopback3
R1(config-router)#exit

```

这样的配置的结果，就是不再往这些环回发出EIGRP数据包了。因此，如下所示，汇总地址就不会在这些接口上通告出去了：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    10.0.0.0/8 for Serial0/0
      Summarizing with metric 128256
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    150.1.0.0
  Passive Interface(s):
    Loopback0
    Loopback1
    Loopback2
    Loopback3
  Routing Information Sources:
    Gateway          Distance      Last Update
    (this router)      90          00:03:07
    150.1.1.2        90          00:01:12
  Distance: internal 90 external 170
```

**注意：**本课程模块稍后会对 `passive-interface` 命令进行详细讲解。

继续有关自动汇总方面的内容，在对有类边界进行自动汇总后，EIGRP就将一条到汇总地址的路由，安装到EIGRP的拓扑表与IP路由表中（Continuing with automatic summarisation, following automatic summarisation at the classful boundary, EIGRP installs a route to the summary address into the EIGRP topology table and the IP routing table）。下面的EIGRP拓扑表中就包含了此汇总地址的路由，以及更具体的路由条目以及这些路由条目各自所直接连接的接口：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/8, 1 successors, FD is 128256
  via Summary (128256/0), Null0
P 10.3.3.0/24, 1 successors, FD is 128256
  via Connected, Loopback3
P 10.2.2.0/24, 1 successors, FD is 128256
  via Connected, Loopback2
P 10.1.1.0/24, 1 successors, FD is 128256
  via Connected, Loopback1
...
[Truncated Output]
```

在路由表中，汇总路由是直接连接到 `Null0` 接口上的。该路由有着一个默认为 5 的管理距离数值（a default administrative distance value of 5）。下面的输出对此进行了演示：

```
R1#show ip route 10.0.0.0 255.0.0.0
Routing entry for 10.0.0.0/8
Known via "eigrp 150", distance 5, metric 128256, type internal
Redistributing via eigrp 150
Routing Descriptor Blocks:
* directly connected, via Null0
  Route metric is 128256, traffic share count is 1
  Total delay is 5000 microseconds, minimum bandwidth is 10000000 Kbit
  Reliability 255/255, minimum MTU 1514 bytes
  Loading 1/255, Hops 0
```

在EIGRP完成自动汇总时，路由器将对汇总路由进行通告，从而抑制了那些更为具体路由的通告（When EIGRP performs automatic summarisation, the router advertises the summary route and suppresses the more specific routes）。换句话说，在汇总路由得以通告时，那些更为具体的前缀，在发往EIGRP邻居的更新中就被省略了（the more specific prefixes are suppressed in updates to EIGRP neighbours）。这一点可通过查看路由器 R2 上的路由表，加以验证，如下所示：

```
R2#show ip route eigrp
D 10.0.0.0/8 [90/2298856] via 150.1.1.1, 00:29:05, Serial0/0
```

这种默认行为在一些基本的网络，比如上图36.15中演示的网络中，运作不错。但其可能在**不连续网络**--那种由两个分离的主要网络所组成的网络中，如下图36.16中所演示的，有着不利影响（However, it can have an adverse impact in a **discontiguous network**, which comprises a major network that separates another major network）：

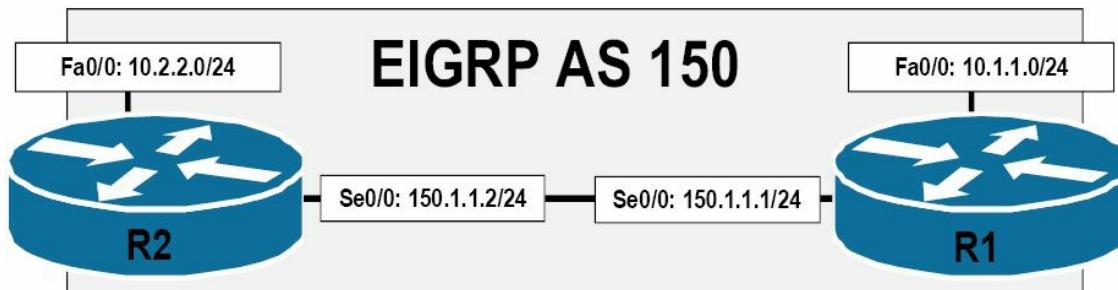


图 36.16 -- 不连续网络, Discontiguous Network

参考图36.16中所演示的图例，一个大的 150.1.0.0/16 网络将这里的两个大的 10.0.0.0/8 网络分开了。在开启自动汇总时，路由器 R1 与 R2 将把 10.1.1.0/24 及 10.2.2.0/24 子网，相应地都汇总到 10.0.0.0/8 地址。该汇总路由将以下一跳接口 Null0，被安装（到路由表中）。而 Null0 接口又是一个“数位垃圾桶(bit-bucket)”。所有发送到此接口的数据包，就将实实在在地被丢弃。

**译者注：**本小节及前面小节中所提到的大网络（the major network），是指按网络大类分的网络，也就是A、B、C、D及E类网络。

因为两台路由器就只把汇总地址通告给对方，所以两台路由器都将无法到达对方的 10.x.x.x/24 子网。为掌握到图36.16所演示网络中自动汇总的衍生问题（ramifications），下面从在路由器 R1 及 R2 上的配置开始，一步一步的走一下这些步骤，如下所示：

```
R1(config)#router eigrp 150
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#network 150.1.1.0 0.0.0.255
R1(config-router)#exit
```

```
R2(config)#router eigrp 150
R2(config-router)#network 10.2.2.0 0.0.0.255
R2(config-router)#network 150.1.1.0 0.0.0.255
R2(config-router)#exit
```

因为两台路由器上有类边界处的自动汇总都是默认开启的，所以这两台路由器都将生成两个汇总地址：一个是  $10.0.0.0/8$  的，另一个是  $150.1.0.0/16$  的（Because automatic summarisation at the classful boundary is enabled by default on both of the routers, they will both generate two summary addresses: one for  $10.0.0.0/8$  and another for  $150.1.0.0/16$ ）。这两个汇总地址都将指向到  $\text{Null}\emptyset$  接口，同时路由器 R1 上的路由表将显示下面这些条目：

```
R1#show ip route eigrp
 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 D      10.0.0.0/8 is a summary, 00:04:51, Null0
 150.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
 D      150.1.0.0/16 is a summary, 00:06:22, Null0
```

与此类似，路由器 R2 上反应了同样的情况，如下所示：

```
R2#show ip route eigrp
 10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
 D      10.0.0.0/8 is a summary, 00:01:58, Null0
 150.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
 D      150.1.0.0/16 is a summary, 00:01:58, Null0
```

虽然汇总地址  $150.1.0.0/16$  被安装到了IP路由表中，路由器 R1 与 R2 是仍可 ping 通对方的，因为这里的更为具体路由条目（the more route-specific entry,  $150.1.1.0/24$ ）是位处于直连接口之上的。可通过执行 `show ip route [address] [mask] longer-prefixes` 命令，来查看到这些某个汇总路由中的详细条目。下面演示了对于  $150.1.0.0/16$  汇总，该命令的输出：

```
R1#show ip route 150.1.0.0 255.255.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
 150.1.0.0/16 is variably subnetted, 2 subnets, 2 masks
 C      150.1.1.0/24 is directly connected, Serial0/0
 D      150.1.0.0/16 is a summary, 00:10:29, Null0
```

因为那条更详细的  $150.1.1.0/24$  路由条目是存在的，所以发送到  $150.1.1.2$  地址的数据包将经由  $\text{Serial}0/0$  接口加以转发。这就令到路由器 R1 与 R2 的连通性没有问题，如下所示：

```
R1#ping 150.1.1.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 150.1.1.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
```

但是，那些到这个大的  $150.1.0.0/16$  网络的所有其它子网的数据包，都将发送到  $\text{Null}\emptyset$  接口，因为没有其它具体路由条目存在（于路由表中）。

那么到这里，一切都说得通了（So far, everything appears to be in order）。可以看出，因为大的 150.1.0.0/16 网络的这条更具体路由条目（150.1.1.0/24）的存在，路由器 R1 与 R2 之间是能 ping 通的。不过问题在于路由器 R1 与 R2 上大的网络 10.0.0.0/8 的那些子网。路由器 R1（的路由表）显示出下面其生成的 10.0.0.0/8 汇总地址的具体路由条目：

```
R1#show ip route 10.0.0.0 255.0.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.1.1.0/24 is directly connected, FastEthernet0/0
D        10.0.0.0/8 is a summary, 00:14:23, Null0
```

于此类似，路由器 R2（的路由表）显示出其生成的 10.0.0.0/8 汇总地址的以下具体路由条目：

```
R2#show ip route 10.0.0.0 255.0.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C        10.2.2.0/24 is directly connected, FastEthernet0/0
D        10.0.0.0/8 is a summary, 00:14:23, Null0
```

可以看出，两台路由器都没有到对方的 10.x.x.x/24 子网的路由。假设在路由器 R1 尝试往 10.2.2.0/24 发送数据包时，将使用那个汇总地址，那么数据包就将转发到 Null0 接口。下面的输出对此进行了演示：

```
R1#show ip route 10.2.2.0
Routing entry for 10.0.0.0/8
  Known via "eigrp 150", distance 5, metric 28160, type internal
  Redistributing via eigrp 150
  Routing Descriptor Blocks:
    * directly connected, via Null0
      Route metric is 28160, traffic share count is 1
      Total delay is 100 microseconds, minimum bandwidth is 100000 Kbit
      Reliability 255/255, minimum MTU 1500 bytes
      Loading 1/255, Hops 0
```

路由器 R1 将无法 ping 通 R2 上的 10.x.x.x/24 子网，反之亦然，如下所示：

```
R1#ping 10.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

而要解决这个问题，则有两种方案，如下：

- 在两台路由器上手动配置  $10.x.x.x/24$  的静态路由, Manually configure static routes for the  $10.x.x.x/24$  subnets on both routers
- 关闭EIGRP的自动有类网络汇总, Disable EIGRP automatic classful network summarisation

第一个选项是相当简单粗暴基础的 (very basic) 。但是, 静态路由配置不具备可伸缩性, 且在大型网络中需要大量费时费力的配置。而第二选项, 也就是**推荐做法**, 除了具备伸缩性, 比起第一选项只需较少的时间精力。通过执行 `no auto-summary` 命令, 就可将自动汇总予以关闭 (较新版本的IOS中, 该特性已被默认关闭了), 如下所示:

```
R1(config)#router eigrp 150
R1(config-router)#no auto-summary
R1(config-router)#exit
```

```
R2(config)#router eigrp 150
R2(config-router)#no auto-summary
R2(config-router)#exit
```

此配置的结果, 就是大的网络上的那些具体子网, 在两台路由器上都有通告。不会生成汇总路由, 如下所示:

```
R2#show ip route eigrp
 10.0.0.0/24 is subnetted, 2 subnets
 D       10.1.1.0 [90/2172416] via 150.1.1.1, 00:01:17, Serial0/0
```

这些  $10.x.x.x/24$  子网之间的连通性, 可使用一个简单的 `ping` 操作, 加以验证, 如下所示:

```
R2#ping 10.1.1.1 source 10.2.2.2 repeat 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 10.1.1.1, timeout is 2 seconds:
Packet sent with a source address of 10.2.2.2
!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 1/3/4 ms
```

在深入讨论手动路由汇总前, 重要的是先要知道EIGRP是不会自动对外部网络进行汇总的, 除非某个内部网络将包含在那个汇总中 (Before we go into the details pertaining to manual summarisation, it is important to know that EIGRP will not automatically summarise external networks unless there is an internal network that will be included in the summary) 。为更好地掌握此概念, 就要参考下图36.17, 该图片演示了一个基本的EIGRP网络:

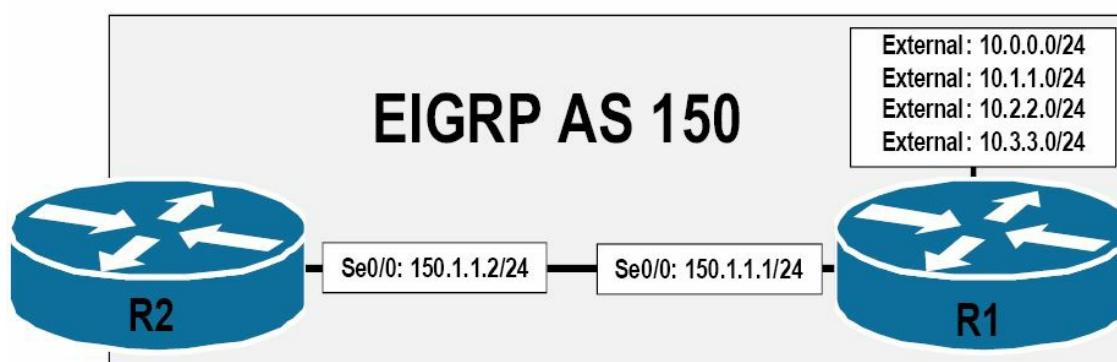


图 36.17 -- 对外部网络的汇总, Summarising External Networks

参考图36.17, 路由器 R1 正在进行重分发 (redistributing, 这令到这些网络成为外部的网络) , 并在随后将外部网络 10.0.0.0/24 、 10.1.1.0/24 、 10.2.2.0/24 与 10.3.3.0/24 这些外部网络经由EIGRP进行通告。路由器 R1 上开启了自动路由汇总。路由器 R1 上的初始配置是下面这样的:

```
R1(config)#router eigrp 150
R1(config-router)#redistribute connected metric 8000000 5000 255 1 1514
R1(config-router)#network 150.1.1.1 0.0.0.0
R1(config-router)#exit
```

`show ip protocols` 命令显示出路由器 R1 的 `Serial0/0` 接口上开启了EIGRP, 同时正通告着那些连接的网络。同时自动汇总是开启的, 如下所示:

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
Outgoing update filter list for all interfaces is not set
Incoming update filter list for all interfaces is not set
Default networks flagged in outgoing updates
Default networks accepted from incoming updates
EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
EIGRP maximum hopcount 100
EIGRP maximum metric variance 1
Redistributing: connected, eigrp 150
EIGRP NSF-aware route hold timer is 240s
Automatic network summarization is in effect
Maximum path: 4
Routing for Networks:
  150.1.1.1/32
Routing Information Sources:
  Gateway          Distance      Last Update
  150.1.1.2           90          00:00:07
  Distance: internal 90 external 170
```

就像前面输出示例所演示的那样, 因为这些 `10.x.x.x/24` 前缀都是外部路由, 所以EIGRP不会对这些前缀进行自动汇总。又因此EIGRP不会往拓扑表, 也不会往IP路由表中添加一条这些前缀的汇总路由。下面的输出对此进行了演示:

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
P 10.1.1.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
P 10.2.2.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
P 10.3.3.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
...
[Truncated Output]
```

这些具体路由条目, 是作为外部EIGRP路由, 通告给路由器 R2 的, 如下所示:

```
R2#show ip route eigrp
 10.0.0.0/24 is subnetted, 4 subnets
D EX    10.3.3.0 [170/3449856] via 150.1.1.1, 00:07:02, Serial0/0
D EX    10.2.2.0 [170/3449856] via 150.1.1.1, 00:07:02, Serial0/0
D EX    10.1.1.0 [170/3449856] via 150.1.1.1, 00:07:02, Serial0/0
D EX    10.0.0.0 [170/3449856] via 150.1.1.1, 00:07:02, Serial0/0
```

现在，假设  $10.0.0.0/24$  是一个内部网络，而  $10.1.1.0/24$ 、 $10.2.2.0/24$  与  $10.3.3.0/24$  三个子网却是外部网络。因为这些将构成有类汇总地址  $10.0.0.0/8$  的路由中有一条是内部路由，所以EIGRP将创建出一个汇总地址，并将其包含在EIGRP拓扑表与IP路由表中。命令 `show ip protocols` 显示出  $10.0.0.0/24$  网络此时就是一个内部EIGRP网络了，如下所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: connected, eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is in effect
  Automatic address summarization:
    150.1.0.0/16 for Loopback0
      Summarizing with metric 2169856
    10.0.0.0/8 for Serial0/0
      Summarizing with metric 128256
  Maximum path: 4
  Routing for Networks:
    10.0.0.1/32
    150.1.1.1/32
  Routing Information Sources:
    Gateway          Distance      Last Update
    (this router)     90            00:00:05
    150.1.1.2        90            00:00:02
  Distance: internal 90 external 170
```

在上面的输出中，EIGRP的自动汇总，已经生成一个  $10.0.0.0/8$  的汇总地址，因为  $10.0.0.0/24$  这个内部子网是该聚合地址（the aggregate address）的一部分。而EIGRP的拓扑表，将显示出这些外部与内部条目，以及生成的汇总地址，如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/8, 1 successors, FD is 128256
  via Summary (128256/0), Null0
P 10.0.0.0/24, 1 successors, FD is 128256
  via Connected, Loopback0
P 10.1.1.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
P 10.2.2.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
P 10.3.3.0/24, 1 successors, FD is 1280256
  via Rconnected (1280256/0)
...
[Truncated Output]
```

此时，就仅有一条路由通告给路由器 R2 了，如下面的输出所示：

```
R2#show ip route eigrp
D 10.0.0.0/8 [90/2297856] via 150.1.1.1, 00:04:05, Serial0/0
```

从路由器 R2 的角度看，该路由就是一条简单的内部EIGRP路由。也就是说，路由器 R2 并不知道这个汇总地址还有着一些外部路由，如下所示：

```
R2#show ip route 10.0.0.0 255.0.0.0
Routing entry for 10.0.0.0/8
Known via "eigrp 150", distance 90, metric 2297856, type internal
Redistributing via eigrp 150
Last update from 150.1.1.1 on Serial0/0, 00:05:34 ago
Routing Descriptor Blocks:
* 150.1.1.1, from 150.1.1.1, 00:05:34 ago, via Serial0/0
  Route metric is 2297856, traffic share count is 1
  Total delay is 25000 microseconds, minimum bandwidth is 1544 Kbit
  Reliability 255/255, minimum MTU 1500 bytes
  Loading 1/255, Hops 1
```

经由所接收到的这条汇总路由，路由器 R2 是可以同时到达这个内部的 10.0.0.0/24 与那些其它的外部 10.x.x.x/24 网络，如下所示：

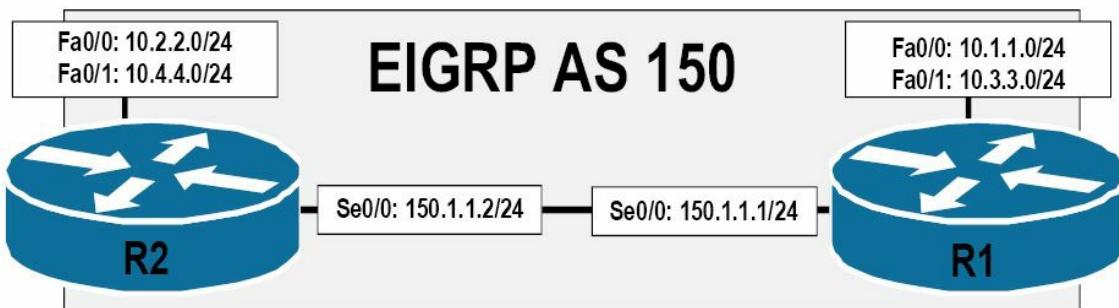
```
R2#ping 10.0.0.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.0.0.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/4 ms

R2#ping 10.3.3.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.3.3.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/3/4 ms
```

与EIGRP的自动汇总不同，**EIGRP的手动路由汇总，是在接口级别使用接口配置命令 ip summary-address eigrp [ASN] [network] [mask] [distance] [leak-map <name>]**，进行配置和部署的。默认情况下，分配给 EIGRP 汇总地址的默认**管理距离**值为 5 (By default, an EIGRP summary address is assigned a default **administrative distance** value of 5)。可通过由 [distance] 关键字所指定的数值，从而指定所需的管理距离值，来改变此默认分配的值。

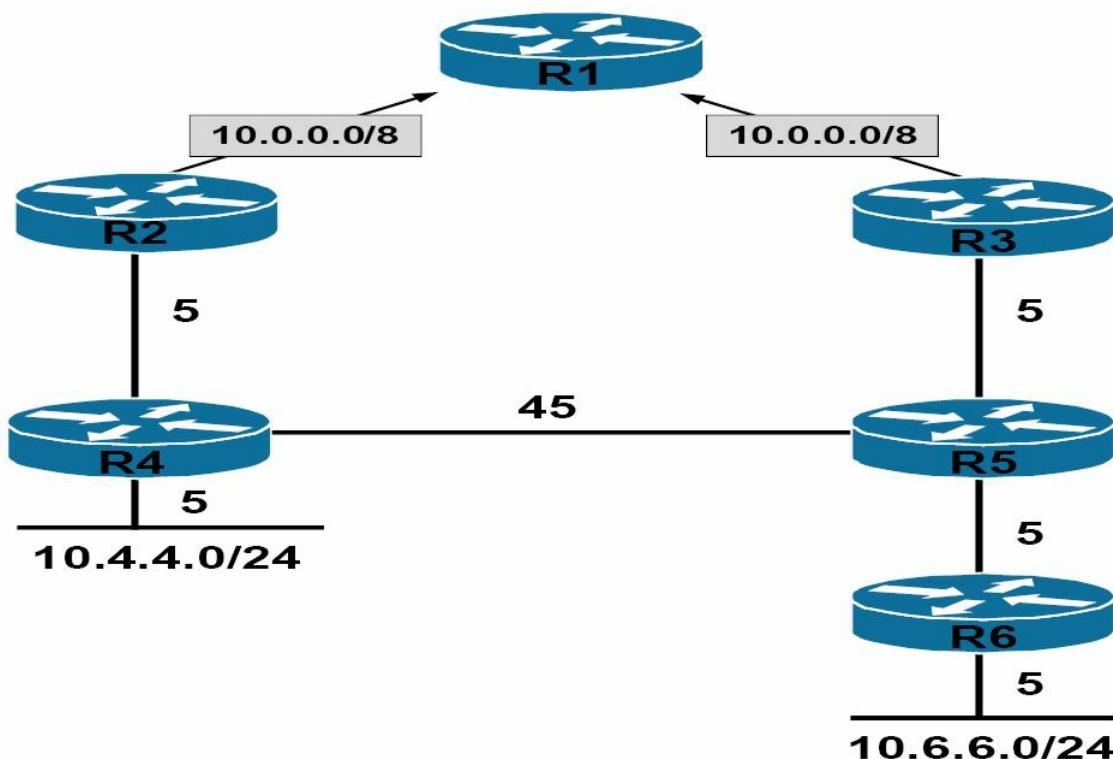
默认在配置了手动路由汇总时，EIGRP将不会就包含在已汇总网络条目中的那些更具体路由条目，进行通告了。可将关键字 [leak-map <name>] 配置为允许EIGRP路由泄漏，有了此特性，EIGRP就允许那些指定的具体路由条目，与汇总地址一起得以通告。而那些未在泄露图谱中指定的条目，则仍然会被压减 (By default, when manual route summarisation is configured, EIGRP will not advertise the more specific route entries that fall within the summarised network entry. The [leak-map <name>] keyword can be configured to allow EIGRP route leaking, wherein EIGRP allows specified specific route entries to be advertised in conjunction with the summary address. Those entries that are not specified in the leak map are still suppressed)。

在对路由进行手动汇总时，重要的是要足够具体 (When manually summarising routes, it is important to be as specific as possible)。否则，所作出的配置将导致与先前所讲到的不连续网络示例中类似的流量黑洞问题。下图36.18中对此概念进行了演示：

图 36.18 -- 不良路由汇总下的流量黑洞问题, *Black-Holing traffic with Poor Route Summarisation*

参考图36.18, 假如在两台路由器上都手动配置了一个  $10.0.0.0/8$  的汇总地址, 那么更为具体的那些前缀就将被压减掉了。因为同时EIGRP将一条到该汇总地址的路由, 与下一跳接口 `Null0`, 安装到EIGRP的拓扑表及IP路由表, 那么在此网络中将经历与不连续网络中自动汇总同样的问题, 两台路由器上相应子网将无法与对方联系上。

此外, 同样重要的是需要明白, 如网络中有着不良部署, 手动汇总还将导致网络的次优路由问题 (suboptimal routing)。下图36.19对此进行了演示:

图 36.19 -- 路由汇总带来的次优路由问题, *Suboptimal Routing with Route Summarisation*

默认情况下, 在EIGRP的一条汇总路由创建出来后, 路由器就将该汇总地址以其中所有更具体路由度量值中最小的度量值进行通告。也就是说, 汇总地址将有着包含在汇总地址建立中最低的、最具体路由的度量值 (By default, when a summary route is created for EIGRP, the router advertises the summary address with a metric equal to the minimum of all the more specific routes. In other words, the summary address will have the same metric as the lowest, most specific route included in the creation of the summary address)。

参考图36.19中所演示的网络拓扑，路由器 R2 与 R3 都正将汇总地址 10.0.0.0/8 通告给 R1。此汇总是有更具体的 10.4.4.0/24 与 10.6.6.0/24 前缀构成的。该汇总地址所使用的度量值，在两台路由器上分别如下表36.5那样计算出来：

表 36.5 -- 汇总路由的度量值计算

起始点（路由器）	到 10.4.4.0/24 的度量值	到 10.6.6.0/24 的度量值
R2	5+5=10	5+45+5+5=60
R3	5+45+5=55	5+5+5=15

基于表36.5中的度量值计算，对于从路由器 R1 起始的流量，R2 显然有着到 10.4.4.0 的最低度量值路径，同时 R3 有着对于起始自 R1、到 10.6.6.0/24 网络流量的最低度量值路径。但在汇总地址 10.0.0.0/8 被通告到路由器 R1 时，该汇总地址使用了构成该汇总路由的所有路由中最低最小度量值。在这个示例中，路由器 R2 以度量值 10，将该汇总地址通告给 R1。R3 以同样逻辑，以 15 的度量值将该汇总路由通告给 R1。

在路由器 R1 从 R2 及 R3 处接收到两条汇总路由后，它将使用最低的度量值来转发那些以包含在大的（主要）有类网络 10.0.0.0/8 中的那些子网为目的的流量。下图36.20对此进行了演示：

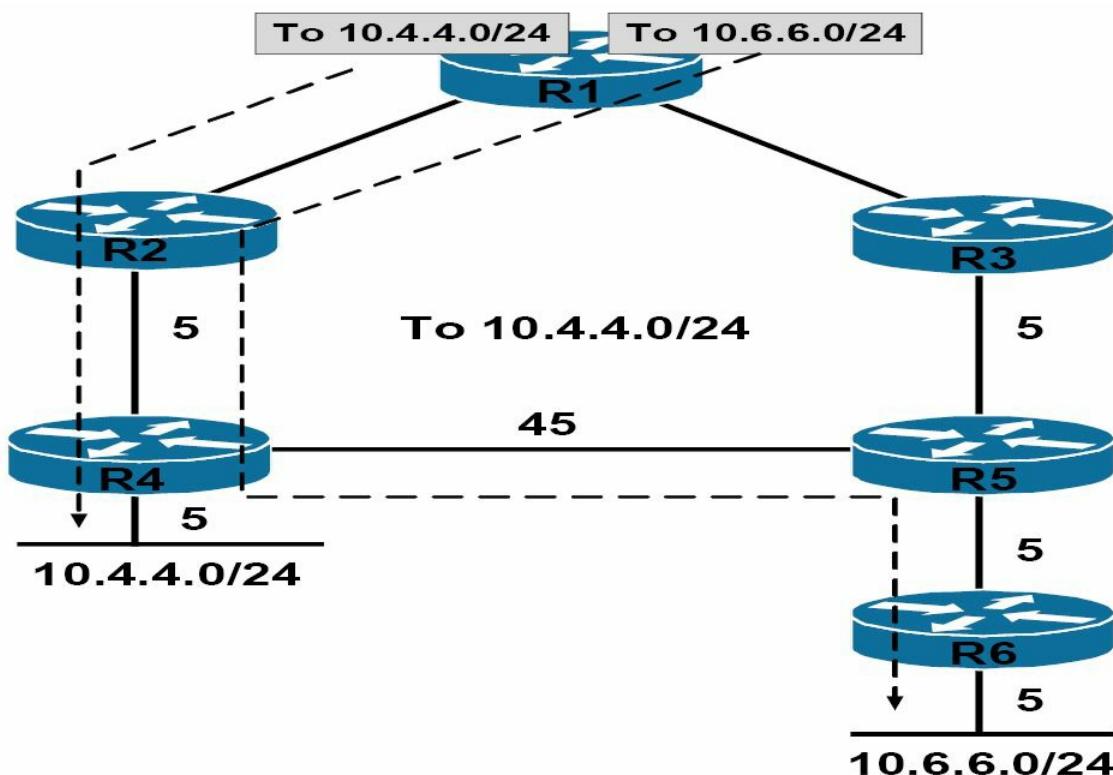


图 36.20 -- 路由汇总下的次优路由，Suboptimal Routing with Route Summarisation

参考图36.20，可清楚地发现，尽管这对于 10.4.4.0/24 子网来说，是一条最优路径，而对于 10.6.6.0/24 子网却是一条次优路径。因此，在网络中部署路由汇总时，先弄清楚网络拓扑是非常重要的。

回到采用EIGRP时的手动路由汇总的配置上，下图36.21所演示的网络拓扑，将用于对手动路由汇总及**路由泄露特性**（route leaking）的演示：

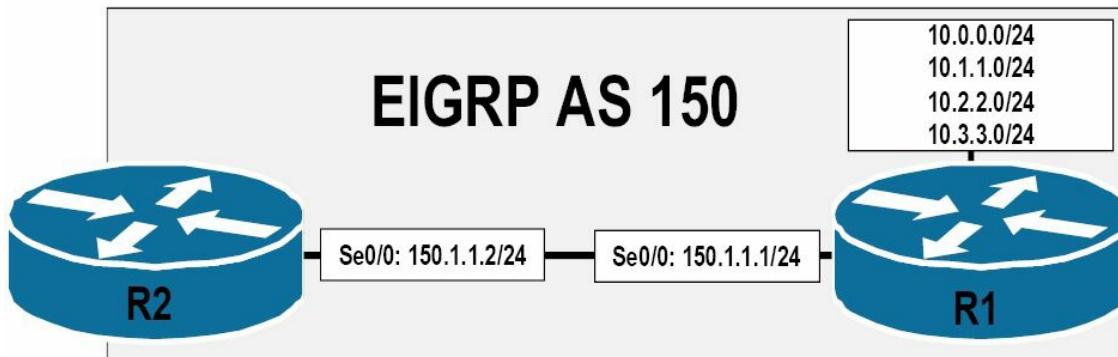


图 36.21 -- EIGRP 手动路由汇总的配置，Configuring EIGRP Manual Route Summarisation

基于路由器 R1 上所配置的接口，R2 上的路由表将显示出以下条目：

```
R2#show ip route eigrp
 10.0.0.0/24 is subnetted, 4 subnets
 D      10.3.3.0 [90/2297856] via 150.1.1.1, 00:00:14, Serial0/0
 D      10.2.2.0 [90/2297856] via 150.1.1.1, 00:00:14, Serial0/0
 D      10.1.1.0 [90/2297856] via 150.1.1.1, 00:00:14, Serial0/0
 D      10.0.0.0 [90/2297856] via 150.1.1.1, 00:00:14, Serial0/0
```

为将路由器 R1 上的这些路由条目进行汇总并通告出一条单一的特定（译者注，这里可能是“汇总”，而不是具体(specific)）路由，就要在 R1 的 Serial0/0 接口上应用如下配置：

```
R1(config)#interface Serial0/0
R1(config-if)#ip summary-address eigrp 150 10.0.0.0 255.252.0.0
R1(config-if)#exit
```

在此配置下，该汇总条目 10.0.0.0/14 就将被安装到路由器 R1 的EIGRP拓扑表与IP路由表中。该EIGRP拓扑表条目如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/14, 1 successors, FD is 128256
    via Summary (128256/0), Null0
P 10.3.3.0/24, 1 successors, FD is 128256
    via Connected, Loopback3
P 10.2.2.0/24, 1 successors, FD is 128256
    via Connected, Loopback2
P 10.0.0.0/24, 1 successors, FD is 128256
    via Connected, Loopback0
P 10.1.1.0/24, 1 successors, FD is 128256
    via Connected, Loopback1
...
[Truncated Output]
```

而该路由表条目，也同样反应出来此汇总路由的下一跳接口为 Null0，如下所示：

```
R1#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.3.3.0/24 is directly connected, Loopback3
C        10.2.2.0/24 is directly connected, Loopback2
C        10.1.1.0/24 is directly connected, Loopback1
C        10.0.0.0/24 is directly connected, Loopback0
D*       10.0.0.0/14 is a summary, 00:02:37, Null0
  150.1.0.0/24 is subnetted, 1 subnets
C        150.1.1.0 is directly connected, Serial0/0
  150.2.0.0/24 is subnetted, 1 subnets
C        150.2.2.0 is directly connected, Serial0/1
```

可再次使用 `show ip route [address] [mask] longer-prefixes` 命令，来查看组成该聚合或者说汇总路由的那些具体路由条目，如下面路由器 R1 上的输出所示：

```
R1#show ip route 10.0.0.0 255.252.0.0 longer-prefixes
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
  10.0.0.0/8 is variably subnetted, 5 subnets, 2 masks
C        10.3.3.0/24 is directly connected, Loopback3
C        10.2.2.0/24 is directly connected, Loopback2
C        10.1.1.0/24 is directly connected, Loopback1
C        10.0.0.0/24 is directly connected, Loopback0
D*       10.0.0.0/14 is a summary, 00:04:03, Null0
```

而在路由器 R2 上，则接收到汇总地址 `10.0.0.0/14` 的一条单一的路由条目，如下所示：

```
R2#show ip route eigrp
  10.0.0.0/14 is subnetted, 1 subnets
D        10.0.0.0 [90/2297856] via 150.1.1.1, 00:06:22, Serial0/0
```

为加强该汇总路由度量值概念，这里假设路由器 R1 上的那些路由都是有着不同度量值的外部路由（也就是说，这些路由是已被重分发到EIGRP中的）。那么 R1 上的EIGRP拓扑表，将显示以下条目：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/24, 1 successors, FD is 10127872
    via Rconnected (10127872/0)
P 10.1.1.0/24, 1 successors, FD is 3461120
    via Rconnected (3461120/0)
P 10.2.2.0/24, 1 successors, FD is 2627840
    via Rconnected (2627840/0)
P 10.3.3.0/24, 1 successors, FD is 1377792
    via Rconnected (1377792/0)
...
[Truncated Output]
```

此时再度在路由器 R1 上做先前示例中的相同汇总地址配置，如下所示：

```
R1(config)#int s0/0
R1(config-if)#ip summary-address eigrp 150 10.0.0.0 255.252.0.0
R1(config-if)#exit
```

那么基于此配置，该汇总路由就以它所包含的所有路由中的最小度量值，而放入到EIGRP拓扑表和IP路由表中（Based on this configuration, the summary route is placed into the EIGRP topology table and the IP routing table with a metric equal to the lowest metric of all routes that it encompasses）。而根据前面所展示的 show ip eigrp topology 命令的输出，该汇总地址将获得到与 10.3.3.0/24 前缀相同的度量值，如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.0.0.0/14, 1 successors, FD is 1377792
    via Summary (1377792/0), Null0
P 10.0.0.0/24, 1 successors, FD is 10127872
    via Rconnected (10127872/0)
P 10.1.1.0/24, 1 successors, FD is 3461120
    via Rconnected (3461120/0)
P 10.2.2.0/24, 1 successors, FD is 2627840
    via Rconnected (2627840/0)
P 10.3.3.0/24, 1 successors, FD is 1377792
    via Rconnected (1377792/0)
P 150.1.1.0/24, 1 successors, FD is 2169856
    via Connected, Serial0/0
```

## 什么是被动接口

### Understanding Passive Interface

如同本课程模块前面所指出的，在对某个网络开启了EIGRP后，路由器就开始在其位于某个特定网络范围内的所有接口上，发出Hello数据包。这样做可令到EIGRP能够动态地发现邻居，并建立各种网络关系。这对于那些切实有着到物理介质连接的接口上，比如以太网及串行接口等，是需要的。但这种默认行为在那些绝不会有其它设备连接的逻辑接口上（logical interfaces），比如环回接口，却是不必要的，因为路由器绝不会经由这些逻辑接口建立EIGRP的邻居关系，相反还会造成不必要的路由器资源浪费。

思科IOS软件允许管理员使用**路由器配置命令** `passive-interface [name] default`，将命名的接口（the named interface）或把所有接口，指定为被动模式。从而不会在这些被动接口上发出EIGRP数据包；在这些被动接口之间，就绝对不会建立邻居关系了。下面的输出演示了在路由器上如何将两个开启了EIGRP的接口，配置为被动模式：

```
R1(config)#interface Loopback0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback1
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0/0
R1(config-if)#ip address 150.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#router eigrp 150
R1(config-router)#no auto-summary
R1(config-router)#network 150.1.1.0 0.0.0.255
R1(config-router)#network 10.0.0.0 0.0.0.255
R1(config-router)#network 10.1.1.0 0.0.0.255
R1(config-router)#passive-interface Loopback0
R1(config-router)#passive-interface Loopback1
R1(config-router)#exit
```

基于此种配置，逻辑接口 `Loopback0` 与 `Loopback1` 是开启了EIGRP路由的，它们上面的直连网络将被通告给EIGRP邻居。但路由器 `R1` 是不会将EIGRP数据包从这两个接口发出去的。另一方面，接口 `Serial0/0` 上也配置上了EIGRP路由，不过这里是允许EIGRP在此接口上发出数据包的，因为它不是一个被动接口。所有三个网络都是安装到EIGRP路由表中的，如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 10.1.1.0/24, 1 successors, FD is 128256
      via Connected, Loopback1
P 10.0.0.0/24, 1 successors, FD is 128256
      via Connected, Loopback0
P 150.1.1.0/24, 1 successors, FD is 2169856
      via Connected, Serial0/0
```

但 `show ip eigrp interfaces` 显示EIGRP路由只是对 `Serial0/0` 接口是开启的，如下所示：

```
R1#show ip eigrp interfaces
IP-EIGRP interfaces for process 150
          Xmit Queue   Mean    Pacing Time    Multicast    Pending
Interface    Peers Un/Reliable SRTT    Un/Reliable    Flow Timer  Routes
Se0/0         1        0/0        0        0/15           0           0
```

也可在 `show ip protocols` 命令的输出中查看到那些配置为被动模式的接口，如下所示：

```
R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0/24
    10.1.1.0/24
    150.1.1.0/24
  Passive Interface(s):
    Loopback0
    Loopback1
  Routing Information Sources:
    Gateway          Distance      Last Update
    Distance: internal 90 external 170
```

路由器配置命令 `passive-interface [name|default]` 中的 `[default]` 关键字令到所有接口，都成为被动模式。先假设路由器上配置了50个的环回接口。如打算将每个环回接口都配置为被动模式，那么就需要50行的代码。而这时就可以使用 `passive-interface default` 命令，来令到所有接口都成为被动接口了。对于那些确实想要发出EIGRP数据包的接口，就可使用 `no passive-interface [name]` 命令进行配置。下面对 `passive-interface default` 命令的用法进行了演示：

```
R1(config)#interface Loopback0
R1(config-if)#ip address 10.0.0.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback1
R1(config-if)#ip address 10.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback3
R1(config-if)#ip address 10.3.3.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Loopback2
R1(config-if)#ip address 10.2.2.1 255.255.255.0
R1(config-if)#exit
R1(config)#interface Serial0/0
R1(config-if)#ip address 150.1.1.1 255.255.255.0
R1(config-if)#exit
R1(config-router)#network 10.0.0.1 255.255.255.0
R1(config-router)#network 10.1.1.1 255.255.255.0
R1(config-router)#network 10.3.3.1 255.255.255.0
R1(config-router)#network 10.2.2.1 255.255.255.0
R1(config-router)#network 150.1.1.1 255.255.255.0
R1(config-router)#passive-interface default
R1(config-router)#no passive-interface Serial0/0
R1(config-router)#exit
```

这里可以再度使用 `show ip protocols` 命令，来查看哪些接口是处于EIGRP下的被动模式的，如下所示：

```

R1#show ip protocols
Routing Protocol is "eigrp 150"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
  Redistributing: eigrp 150
  EIGRP NSF-aware route hold timer is 240s
  Automatic network summarization is not in effect
  Maximum path: 4
  Routing for Networks:
    10.0.0.0/24
    10.1.1.0/24
    10.2.2.0/24
    10.3.3.0/24
    150.1.1.0/24
  Passive Interface(s):
    Loopback1
    Loopback2
    Loopback3
    Loopback4
  Routing Information Sources:
    Gateway          Distance      Last Update
    (this router)      90          00:02:52
  Distance: internal 90 external 170

```

这里通过使用 `passive-interface default` 命令，令到多个被动接口的配置就得以简化，并减少了代码。而当其与 `no passive-interface Serial0/0` 一起使用时，EIGRP数据包仍旧在接口 `Serial0/0` 上发出，从而允许EIGRP邻居关系通过此接口建立起来，如下所示：

```

R1#show ip eigrp neighbors
IP-EIGRP neighbors for process 150
  H   Address       Interface   Hold   Uptime     SRTT    RTO     Q      Seq
                (sec)           (ms)        Cnt    Num
  0   150.1.1.2     Se0/0       12    00:02:47  1      3000    0      69

```

## 掌握EIGRP路由器ID的用法

### Understanding the Use of the EIGRP Router ID

与OSPF使用路由器ID（the router ID, RID）来识别OSPF邻居不同，**EIGRP的RID主要用途是阻止路由环回的形成。RID被用于识别那些外部路由的起源路由器**（the primary use of the EIGRP RID is to prevent routing loops. The RID is used to identify the originating router for external routes）。假如接收到一条有着与本地路由器一致的RID外部路由，那么就会将其丢弃。设计此特性的目的，就是降低那些其中有着多台自治系统边界路由器（AS Boundary Router, ASBR）正进行路由重分发的网络中，出现路由环回的可能性。

在确定RID时，**EIGRP将选取路由器上所配置的IP地址中最高的作为RID。但如果在路由器上配置了环回接口，那么将优先选取这些接口，因为环回接口是路由器上存在的最稳定接口**。除非将EIGRP进程移除，那么RID随后就绝不会变化了（也就是，假如RID是手动配置的情况）。RID始终会在EIGRP拓扑表中列出，如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
      r - reply Status, s - sia Status
P 10.2.2.0/24, 1 successors, FD is 128256
  via Connected, Loopback2
P 10.3.3.0/24, 1 successors, FD is 128256
  via Connected, Loopback3
P 10.1.1.0/24, 1 successors, FD is 128256
  via Connected, Loopback1
P 10.0.0.0/24, 1 successors, FD is 128256
  via Connected, Loopback0
P 150.1.1.0/24, 1 successors, FD is 2169856
  via Connected, Serial0/0
```

**注意：**这里重要的是掌握到RID与邻居ID通常是不同的，然而这对于那些比如只有一个接口的路由器可能不适用。（It is important to understand that the RID and the neighbour ID will typically be different, although this may not be the case in routers with a single interface, for example）。

EIGRP的路由器ID（RID）是通过路由器配置命令 `eigrp router-id [address]` 进行配置的。在输入了此命令后，RID就以这个新地址，在EIGRP的拓扑表中得以更新。为对此进行演示，这里就以查看路由器上的当前RID开始，如下面的拓扑表中所指出的：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(10.3.3.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
      r - reply Status, s - sia Status
...
[Truncated Output]
```

现在路由器上配置一个 1.1.1.1 的RID，如下所示：

```
R1(config)#router eigrp 150
R1(config-router)#eigrp router-id 1.1.1.1
R1(config-router)#
*Mar 1 05:50:13.642: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 150: Neighbor 150.1.1.2 (Serial0/0) is down: route
*Mar 1 05:50:16.014: %DUAL-5-NBRCHANGE: IP-EIGRP(0) 150: Neighbor 150.1.1.2 (Serial0/0) is up: new adj...
```

伴随这个改变，EIGRP邻居关系就被重置了，同时在EIGRP拓扑表中立即反映出了这个新的RID，如下所示：

```
R1#show ip eigrp topology
IP-EIGRP Topology Table for AS(150)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
      r - reply Status, s - sia Status
...
[Truncated Output]
```

在对EIGRP的路由器ID（RID）进行配置时，应记住下面两点：

- 不能（无法）将RID配置为 0.0.0.0
- 不能（无法）将RID配置为 255.255.255.255

现在，源自该路由器的所有外部路由，就都包含了这个EIGRP路由器ID了。在下面的邻居路由器 R2 输出中，就可对此进行验证：

```
R2#show ip eigrp topology 192.168.254.0/24
IP-EIGRP (AS 150): Topology entry for 192.168.254.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 7289856
Routing Descriptor Blocks:
150.1.1.1 (Serial0/0), from 150.1.1.1, Send flag is 0x0
    Composite metric is (7289856/6777856), Route is External
    Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 220000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
    External data:
        Originating router is 1.1.1.1
        AS number of route is 0
        External protocol is Connected, external metric is 0
        Administrator tag is 0 (0x00000000)
```

而对于内部EIGRP路由，则是不包含RID的，如下面的输出所示：

```
R2#show ip eigrp topology 10.3.3.0/24
IP-EIGRP (AS 150): Topology entry for 10.3.3.0/24
State is Passive, Query origin flag is 1, 1 Successor(s), FD is 2297856
Routing Descriptor Blocks:
150.1.1.1 (Serial0/0), from 150.1.1.1, Send flag is 0x0
    Composite metric is (2297856/128256), Route is Internal
    Vector metric:
        Minimum bandwidth is 1544 Kbit
        Total delay is 25000 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
```

## 第36天问题

1. You can see the ASN with the `show ip _____` command.
2. Every router you want to communicate with in your routing domain must have a different ASN. True or false?
3. What is the purpose of the EIGRP topology table?
4. By default, EIGRP uses the \_\_\_\_\_ bandwidth on the path to a destination network and the total \_\_\_\_\_ to compute routing metrics.
5. Dynamic neighbour discovery is performed by sending EIGRP Hello packets to the destination Multicast group address \_\_\_\_\_ .
6. EIGRP packets are sent directly over IP using protocol number \_\_\_\_\_ .
7. To populate the topology table, EIGRP runs the \_\_\_\_\_ algorithm.
8. The \_\_\_\_\_ includes both the metric of a network as advertised by the connected neighbour, plus the cost of reaching that particular neighbour.
9. Cisco IOS software supports equal cost load sharing for a default of up to four paths for all routing protocols. True or false?
10. What EIGRP command can be used to enable unequal cost load sharing?

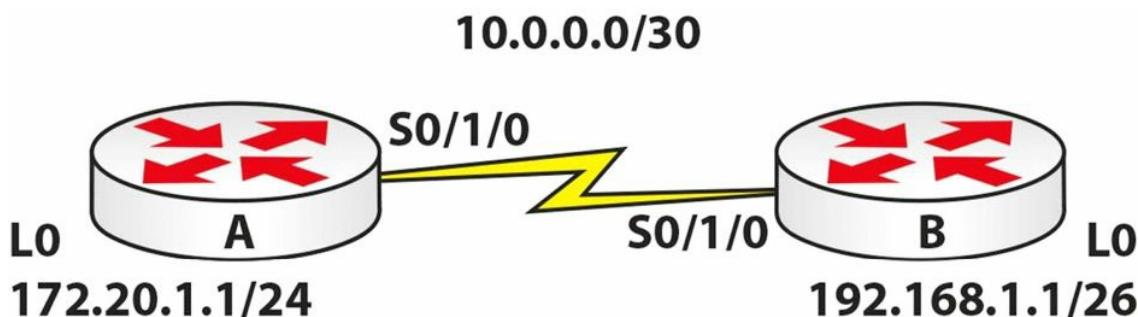
## 第36天问题答案

1. protocols .
2. False.
3. The topology table allows all EIGRP routers to have a consistent view of the entire network. All known destination networks and subnets that are advertised by neighbouring EIGRP routers are stored there.
4. Minimum, delay.
5. 224.0.0.10 .
6. 88.
7. DUAL.
8. Feasible Distance.
9. True.
10. The variance command.

## 第36天实验

### EIGRP的实验

#### 拓扑图



#### 实验目的

学习如何配置基本的EIGRP。

#### 实验步骤

1. 基于上面的拓扑，配置上所有IP地址。确保可以经由串行链路 ping 通。
2. 在两台路由器上以自治系统编号30，配置EIGRP。

```
RouterA(config)#router eigrp 30
RouterA(config-router)#net 172.20.0.0
RouterA(config-router)#net 10.0.0.0
RouterA(config-router)#^Z
RouterA#
RouterB#conf t
Enter configuration commands, one per line.
End with CNTL/Z.
RouterB(config)#router eigrp 30
RouterB(config-router)#net 10.0.0.0
%DUAL-5-NBRCHANGE: IP-EIGRP 30: Neighbor 10.0.0.1 (Serial0/1/0) is up: new adjacency
RouterB(config-router)#net 192.168.1.0
```

1. 对两台路由器上的路由表分别进行检查。

```

RouterA#sh ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
      i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
      * - candidate default, U - per-user static route, o - ODR
      P - periodic downloaded static route
Gateway of last resort is not set
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D        10.0.0.0/8 is a summary, 00:01:43, Null0
C        10.0.0.0/30 is directly connected, Serial0/1/0
      172.20.0.0/16 is variably subnetted, 2 subnets, 2 masks
D        172.20.0.0/16 is a summary, 00:01:43, Null0
C        172.20.1.0/24 is directly connected, Loopback0
D        192.168.1.0/24 [90/20640000] via 10.0.0.2, 00:00:49, Serial0/1/0
RouterA#

```

```

RouterB#show ip route
...
[Truncated Output]
...
      10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D        10.0.0.0/8 is a summary, 00:01:21, Null0
C        10.0.0.0/30 is directly connected, Serial0/1/0
D        172.20.0.0/16 [90/20640000] via 10.0.0.1, 00:01:27, Serial0/1/0
      192.168.1.0/24 is variably subnetted, 2 subnets, 2 masks
D        192.168.1.0/24 is a summary, 00:01:21, Null0
C        192.168.1.0/26 is directly connected, Loopback0
RouterB#

```

- 查明两台路由器都对各个网络进行着自动汇总。并于随后在路由器B上关闭自动汇总。

```

RouterB#show ip protocols
Routing Protocol is "eigrp 30"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Default networks flagged in outgoing updates
  Default networks accepted from incoming updates
  EIGRP metric weight K1=1, K2=0, K3=1, K4=0, K5=0
  EIGRP maximum hopcount 100
  EIGRP maximum metric variance 1
Redistributing: eigrp 30
  Automatic network summarization is in effect
  Automatic address summarization:
    192.168.1.0/24 for Serial0/1/0
      Summarizing with metric 128256
    10.0.0.0/8 for Loopback0
      Summarizing with metric 20512000
  Maximum path: 4
  Routing for Networks:
    10.0.0.0
    192.168.1.0
  Routing Information Sources:
    Gateway          Distance      Last Update
    10.0.0.1          90           496078
  Distance: internal 90 external 170
RouterB(config)#router eigrp 30
RouterB(config-router)#no auto-summary

```

- 对路由器A上的路由表进行检查。

```
RouterA#show ip route
...
[Truncated Output]
...
Gateway of last resort is not set
    10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
D      10.0.0.0/8 is a summary, 00:00:04, Null0
C      10.0.0.0/30 is directly connected, Serial0/1/0
    172.20.0.0/16 is variably subnetted, 2 subnets, 2 masks
D      172.20.0.0/16 is a summary, 00:00:04, Null0
C      172.20.1.0/24 is directly connected, Loopback0
    192.168.1.0/26 is subnetted, 1 subnets
D      192.168.1.0 [90/20640000] via 10.0.0.2, 00:00:04, Serial0/1/0
RouterA#
```

请访问[www.in60days.com](http://www.in60days.com), 免费观看作者完成此试验。

## 第38天 EIGRP对IPv6的支持

### EIGRP For IPv6

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第38天任务

- 阅读今天的课文（以下内容）
- 复习EIGRP模块
- 复习EIGRP故障排除模块

尽管针对IPv6的EIGRP内容，并没有在新的CCNA考试大纲中特别列出，但因为以下原因，此方面的内容将在本模块中加以涵盖。首先，CCNA题目对有关EIGRP与IPv6技术有着较高的关注，所以就算看起来不怎么可能，EIGRPv6方面的题目仍可能出现在考试中。其次，此方面内容相对容易而简单，因此掌握起来也不会花很多时间，尤其在考虑讲解并不会深入的情况下。

除了那些开放的标准协议外，思科专有的EIGRP也被修订到支持IPv6了。因为其支持IPv6，所以有时这个修订版的EIGRP被成为是EIGRPv6，而并不是因为它是EIGRP路由协议的第6版。类似地，IPv4的EIGRP有时也被称为EIGRPv4，以区别两个版本所支持的其所路由协议的不同 (In addition to open standard protocols, the Cisco-proprietary EIGRP has also been modified to support IPv6. This modified version of EIGRP is sometimes referred to as EIGRPv6 because of its support for IPv6, not because it is revision 6 of the EIGRP routing protocol. Similarly, EIGRP for IPv4 is also sometimes referred to as EIGRPv4 to differentiate between the routing protocol versions supported by either version)。

今天将学习以下内容：

- IPv6下的思科EIGRP概览与基础知识，Cisco EIGRP for IPv6 overview and fundamentals
- IPv6下的EIGRP的配置基础

本课程对应了以下CCNA大纲要求：

- 配置并验证EIGRP（单一自治系统，Configure and verify EIGRP(Single AS)）

EIGRPv6保留了EIGRPv4中的大部分相同基础的核心功能 (For the most part, EIGRPv6 retains the same basic core functions as EIGRPv4)。比如两个版本仍使用弥散更新算法来确保无环回的路径，同时两个版本都使用多播数据包来发送更新--尽管EIGRPv6使用的是IPv6的多播地址 `FF02::A`，而EIGRPv4使用的是组地址 `224.0.0.10`。在保留了一些相同核心基础的同时，版本之间有着一些不同之处。下表38.1列出了EIGRPv4与EIGRPv6之间，或简单且更通常地说是IPv4下的EIGRP与IPv6下的EIGRP之间的不同之处：

表 38.1 -- EIGRPv4与EIGRPv6的差异

协议特性 (Protocol Characteristic)	IPv4下的EIGRP	IPv6下的EIGRP
自动汇总特性	支持 (Yes)	不适用 (Not Applicable)
认证或安全特性	MD5	内建于IPv6中 (Built into IPv6)
要求对等点处于同一子网 (Common Subnet for Peers)	要求 (Yes)	不要求 (No)
通告内容 (Advertisement Contents)	子网/掩码 (Subnet/Mask)	前缀/长度 (Prefix/Length)
数据包的封装 (Packet Encapsulation)	IPv4封装 (IPv4)	IPv6封装 (IPv6)

**注意：**因为EIGRPv6使用邻居的链路本地地址作为下一跳地址，因此在位于同一自治系统及同一网段的两台路由器建立邻居关系时，就不需要其全局的IPv6单播子网一致了。这一点是要求邻居在同一子网的EIGRPv4，与使用链路本地地址建立邻居关系而消除了此要求的EIGRPv6之间，最为显著的不同之一 (Because EIGRPv6 uses the Link-Local address of the neighbour as the next-hop address, the global IPv6 Unicast subnets do not need to be the same for a neighbour relationship to be established between two routers that reside within the same autonomous system and are on a common network segment. This is one of the most significant differences between EIGRPv4, which requires neighbours to be on a common subnet, and EIGRPv6, which negates this need by using the Link-Local addresses for neighbour relationships instead)。

## 思科IOS软件在EIGRPv4与EIGRPv6配置上的差异

### Cisco IOS Software EIGRPv4 and EIGRPv6 Configuration Differences

思科IOS软件中对EIGRPv4与EIGRPv4的配置上，有着一些显著的差异。那么第一个显著差异就在于开启受路由的协议方式的不同。对于EIGRPv4来说，需要使用全局配置命令 `router eigrp [ASN]` 来开启EIGRPv4的路由，并指定该EIGRPv4自治系统编号。而在配置EIGRPv6时，则是使用 `ipv6 router eigrp [ASN]` 来开启EIGRPv6并指定出**本地路由器ASN**了 (There are some notable differences in the configuration of EIGRPv4 and EIGRPv6 in Cisco IOS software. The first notable difference is the way in which the routing protocol is enabled. For EIGRPv4, the `router eigrp [ASN]` global configuration command is required to enable EIGRPv4 routing and to specify the EIGRPv4 autonomous system number(ASN). When configuring EIGRPv6, the `ipv6 router eigrp [ASN]` global configuration command is used instead to enable EIGRPv6 and to specify the local router ASN)。

尽管EIGRPv4与EIGRPv6的开启有些类似，但在两个路由进程开启之后的协议状态中，是有着非常显著的不同的。默认在开启了EIGRPv4时，该协议就自动启动，并在其假定有桌正确配置的情况下，开始在所有指定的运作接口上发送Hello数据包。而当在思科IOS软件中启用EIGRPv6时，默认情况下在该协议被开启后，其将保持关闭状态。这就意味着就算在某些指定接口下得以开启，在执行路由器配置命令 `no shutdown` 之前，EIGRP进程仍不是运作中的 (While enabling EIGRPv4 and EIGRPv6 is somewhat similar, there is a very notable and significant difference in the protocol states once the routing process has been enabled. By default, when EIGRPv4 is enabled, the protocol automatically starts and, assuming correct configuration, begins sending Hello packets on all specified interfaces. When enabling EIGRPv6 in Cisco IOS software, by default, after the protocol has been enabled, it remains in the shutdown state. This means that even if enabled under specified interfaces, the EIGRP process will not be operational until the `no shutdown` router configuration command is issued)。

而EIGRPv4与EIGRPv6的另一配置差异，就是在EIGRPv6下，路由器ID是强制要求的，且必须以IPv4的点分十进制表示法进行指定。在分配RID时，要记住该地址不必是一个可路由或可达的地址（Yet another configuration difference between EIGRPv4 and EIGRPv6 is that with EIGRPv6, the router ID is mandatory and must be specified in IPv4 dotted-decimal notation. When assigning the RID, keep in mind that the address does not have to be a routable or reachable address）。

**注意：**如在本地路由器上有任何配置了IPv4地址的接口，那么该路由器将从这些接口选取路由器ID - 优先选取环回接口，在路由器上没有配置环回接口或环回接口不可运作时，就使用物理接口。在有环回接口运行时，将选取环回接口IP地址中最高的作为RID。在没有环回接口运行，而有物理接口运行时，就选择物理接口IP地址中最高的作为RID。在路由器上环回接口与物理接口都没有配置时，就必须使用 `eigrp router-id [IPv4 Addresses]` 命令，指定出一个RID（If there are any interfaces with IPv4 address configured on the local router, then the router will select the router ID from these interfaces -- preferring Loopback interfaces, and then using physical interfaces if no Loopback interfaces are configured or operational on the router. The highest IP address of the Loopback interface(s), if up, will be selected. If not, the RID will be selected from the highest IP address of the physical interfaces, if up. If neither is configured on the router, the `eigrp router-id [IPv4 Address]` command must be used）。

## 思科IOS软件中IPv6的配置与验证

### Configuring and Verifying EIGRPv6 in Cisco IOS Software

继续上一小节，其中突出了EIGRPv4与EIGRPv6之间的配置差异，本节对在思科IOS软件中开启并验证EIGRPv6功能与路由所需的步骤序列，加以贯穿，这些步骤如下：

1. 使用全局配置命令 `ipv6 unicast-routing`，来全局性地开启IPv6路由。在思科IOS软件中IPv6路由默认是关闭的。
2. 使用全局配置命令 `ipv6 router eigrp [ASN]` 来配置一或多个的EIGRPv6进程。
3. 如路由器上没有配置了IPv4地址的运行接口，就要使用路由器配置命令 `eigrp router-id [IPv4 Address]` 来手动配置EIGRPv6的RID。
4. 使用路由器配置命令 `no shutdown` 来开启EIGRPv6进程。
5. 在需要的接口上，使用接口配置命令 `ipv6 address` 与 `ipv6 enable`，开启其IPv6功能。
6. 使用接口配置命令 `ipv6 eigrp [ASN]`，来开启该接口下的一或多个EIGRPv6进程。

因为对于EIGRPv6来说自动汇总是不适用的，因此就没有关闭此行为的需要。为对EIGRPv6配置的掌握进行加强，请考虑下图38.1中所演示的拓扑，该图演示了一个由两台路由器所构成的网络。两台路由器都使用 AS 1 运行着EIGRPv6。路由器 R3 将通过EIGRPv6通告两个额外的前缀：

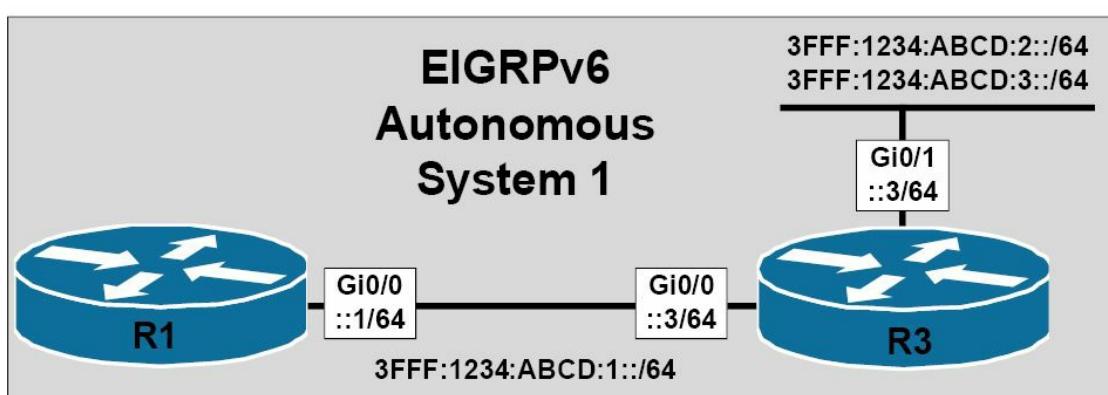


图38.1 -- 思科IOS软件中EIGRPv6的配置

根据上述配置步骤顺序，路由器 R1 上EIGRPv6将被如下配置上：

```
R1(config)#ipv6 unicast-routing
R1(config)#ipv6 router eigrp 1
R1(config-rtr)#eigrp router-id 1.1.1.1
R1(config-rtr)#no shutdown
R1(config-rtr)#exit
R1(config)#interface GigabitEthernet0/0
R1(config-if)#ipv6 address 3fff:1234:abcd:1::1/64
R1(config-if)#ipv6 enable
R1(config-if)#ipv6 eigrp 1
R1(config-if)#exit
```

而根据同样的步骤顺序，路由器 R3 上的EIGRPv6就被如下这样配置上：

```
R3(config)#ipv6 unicast-routing
R3(config)#ipv6 router eigrp 1
R3(config-rtr)#eigrp router-id 3.3.3.3
R3(config-rtr)#no shutdown
R3(config-rtr)#exit
R3(config)#interface GigabitEthernet0/0
R3(config-if)#ipv6 address 3fff:1234:abcd:1::3/64
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 eigrp 1
R3(config-if)#exit
R3(config)#interface GigabitEthernet0/1
R3(config-if)#ipv6 address 3fff:1234:abcd:2::3/64
R3(config-if)#ipv6 address 3fff:1234:abcd:3::3/64
R3(config-if)#ipv6 enable
R3(config-if)#ipv6 eigrp 1
R3(config-if)#exit
```

EIGRPv6的验证过程，将按照EIGRPv4的同样过程进行。首先要验证EIGRP的邻居关系已被成功建立。对于EIGRPv6，这是通过使用 `show ipv6 eigrp neighbours` 命令完成的，如下所示：

```
R1#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(1)
H   Address           Interface Hold Uptime      SRTT    RTO Q   Seq
               (sec)          (ms)          Cnt Num
0   Link-local address: Gi0/0     13  00:01:37    1200      0   3
FE80::1AEF:63FF:FE63:1B00
```

如同先前指出的那样，请注意这里的下一跳地址（也就是EIGRP的邻居地址）被指定为本地链路地址，而不是全局单播地址。此命令所打印出的所有其它信息，与 `show ip eigrp neighbors` 命令打印出是相同的。而要查看详细的邻居信息，可简单地在 `show ipv6 eigrp neighbours` 命令后面追加上 `[detail]` 关键字。使用此选项就打印出有关EIGRP版本、以及从那个特定EIGRP邻居处接收到的前缀数目等信息，如下所示：

```
R1#show ipv6 eigrp neighbors
EIGRP-IPv6 Neighbors for AS(1)
H   Address           Interface Hold Uptime      SRTT    RTO Q   Seq
               (sec)          (ms)          Cnt Num
0   Link-local address: Gi0/0     13  00:01:37    1200      0   3
FE80::1AEF:63FF:FE63:1B00
Version 5.0/3.0, Retrans: 1, Retries: 0, Prefixes: 3
Topology-ids from peer - 0
```

在对EIGRPv6的邻居关系进行验证之后，就可以对路由信息进行验证了。比如，要查看到从EIGRPv6邻居处接收到的那些IPv6前缀，就将使用 `show ipv6 route` 命令，如下面的输出所示：

```
R1#show ipv6 route eigrp
IPv6 Routing Table - default - 6 entries
Codes: C - Connected, L - Local, S - Static, U - Per-user Static route
      B - BGP, HA - Home Agent, MR - Mobile Router, R - RIP
      I1 - ISIS L1, I2 - ISIS L2, IA - ISIS inter area, IS - ISIS summary
      D - EIGRP, EX - EIGRP external, ND - Neighbor Discovery
D  3FFF:1234:ABCD:2::/64 [90/3072]
    via FE80::1AEF:63FF:FE63:1B00, GigabitEthernet0/0
D  3FFF:1234:ABCD:3::/64 [90/3072]
    via FE80::1AEF:63FF:FE63:1B00, GigabitEthernet0/0
```

请再次注意，这里所接收到的前缀，都包含着作为所有接收到的前缀的下一跳IPv6地址的本地链路地址。而要查看EIGRPv6的拓扑表，就应使用 `show ipv6 eigrp topology` 命令。该命令支持那些与用于查看EIGRPv4的拓扑表的 `show ip eigrp topology` 命令下可用的同样的参数。这里基于上面已部署的配置，`R1` 上的拓扑表显示出以下IPv6前缀信息：

```
R1#show ipv6 eigrp topology
EIGRP-IPv6 Topology Table for AS(1)/ID(1.1.1.1)
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - reply Status, s - sia Status
P 3FFF:1234:ABCD:2::/64, 1 successors, FD is 3072
    via FE80::1AEF:63FF:FE63:1B00 (3072/2816), GigabitEthernet0/0
P 3FFF:1234:ABCD:1::/64, 1 successors, FD is 2816
    via Connected, GigabitEthernet0/0
P 3FFF:1234:ABCD:3::/64, 1 successors, FD is 3072
    via FE80::1AEF:63FF:FE63:1B00 (3072/2816), GigabitEthernet0/0
```

与EIGRPv4中的情况一样，可在此命令的后面追加一个前缀，以查看到有关那个前缀或子网的详细信息。比如，要查看有关子网 `3FFF:1234:ABCD:2::/64` 的详细信息，就应简单的输入 `show ipv6 eigrp topology 3FFF:1234:ABCD:2::/64` 命令，如下所示：

```
R1#show ipv6 eigrp topology 3FFF:1234:ABCD:2::/64
EIGRP-IPv6 Topology Entry for AS(1)/ID(1.1.1.1) for 3FFF:1234:ABCD:2::/64
  State is Passive, Query origin flag is 1, 1 Successor(s), FD is 3072
  Descriptor Blocks:
    FE80::1AEF:63FF:FE63:1B00 (GigabitEthernet0/0), from FE80::1AEF:63FF:FE63:1B00, Send
    flag is 0x0
      Composite metric is (3072/2816), route is Internal
      Vector metric:
        Minimum bandwidth is 1000000 Kbit
        Total delay is 20 microseconds
        Reliability is 255/255
        Load is 1/255
        Minimum MTU is 1500
        Hop count is 1
        Originating router is 3.3.3.3
```

最后，一个简单的 `ping` 就可以且应该用于对子网之间的连通性加以验证。下面就是一个从 `R1` 到 `R3` 上的地址 `3FFF:1234:ABCD:2::3` 的 `ping` 操作：

```
R1#ping 3FFF:1234:ABCD:2::3 repeat 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 3FFF:1234:ABCD:2::3, timeout is 2 seconds:
!!!!!!
Success rate is 100 percent (10/10), round-trip min/avg/max = 0/0/4 ms
```

与EIGRPv4下的情况一样，也可使用 `show ipv6 protocols` 对EIGRPv6的一些默认协议数值进行检查，该命令的输出在下面有打印出来。该命令包含了那些开启了EIGRP实例的接口、路由重分发的信息（在适用时），以及手动配置指定或所配置的点分十进制的EIGRPv6路由器ID。

```
R1#show ipv6 protocols
IPv6 Routing Protocol is "eigrp 1"
EIGRP-IPv6 Protocol for AS(1)
    Metric weight K1=1, K2=0, K3=1, K4=0, K5=0
    NSF-aware route hold timer is 240
    Router-ID: 1.1.1.1
    Topology : 0 (base)
        Active Timer: 3 min
        Distance: internal 90 external 170
        Maximum path: 16
        Maximum hopcount 100
        Maximum metric variance 1
    Interfaces:
        GigabitEthernet0/0
Redistribution:
```

## 第38天问题

1. IPv6 security for EIGRPv6 is built-in. True or false?
2. Because EIGRPv6 uses the Link-Local address of the neighbour as the next-hop address, the global IPv6 Unicast subnets do not need to be the same in order for a neighbour relationship to be established between two routers that reside within the same autonomous system and are on a common network segment. True or false?
3. Which command do you use to enter EIGRP for IPv6 Router Configuration mode?
4. Which state is the EIGRP for IPv6 initially in (active or shutdown)?
5. How do you enable EIGRP for IPv6 on a router interface?

## 第38天答案

1. True.
2. True.
3. The `ipv6 router eigrp [ASN]` command.
4. The shutdown state.
5. Issue the `ipv6 eigrp [ASN]` command.

## 第38天实验

请重复第36天的EIGRP实验，不过这次要使用IPv6地址并激活IPv6下的EIGRP-IPV6：

- 在两台路由器上开启IPv6的单播路由
- 在接口上配置IPv6地址
- 使用 `ipv6 router eigrp 100` 命令配置EIGRP进程
- 使用命令 `eigrp router-id 10.10.10.10` 配置一个RID
- 使用 `no shutdown` 命令激活进程
- 使用 `ipv6 eigrp 10` 命令在IPv6接口上开启EIGRP
- 使用 `show ipv6 eigrp neighbors [detail]` 命令对邻居关系进行检查

- 使用命令 `show ipv6 route eigrp` 对所通告的路由进行检查
- 使用 `show ipv6 eigrp topology` 命令对EIGRP的拓扑进行检查

请访问[www.in60days.com](http://www.in60days.com)并免费观看作者如何完成的此实验。

## 第39天 开放最短路径优先协议

### Open Shortest Path First, OSPF

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第39天任务

- 阅读今天的课文（以下）
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2记诵指南
- 在网站[subnetting.org](http://subnetting.org)上学习15分钟

与EIGRP一样，对OSPF的讨论也可以花上几天时间，但这里需要针对那些考试要用到的知识点，进行着重学习。CCNA级别的OSPF知识，尚不足以在大多数网络上涉及与部署该路由技术。

今日将学习以下内容：

- OSPF原理
- DR与BDR
- OSPF的配置
- OSPF故障排除

这一课对应了以下CCNA大纲的要求：

- OSPF的配置与验证（单区，single area）
- 邻居邻接（Neighbour adjacencies, 或者叫邻居关系的形成）
- OSPF的各种状态
- 对多区的讨论（discuss multi-area）
- OSPFv2的配置
- 路由器ID
- LSA（链路状态通告）的类型

## 指定与后备指定路由器（Designated and Backup Designated Routers）

如同第12天模块中所指出的，OSPF会在广播与非广播网络类型上选举出指定路由器（DR）与/或非后备指定路由器（BDR）。对后备指定路由器并非这些网络类型上的强制性组件这一点的掌握，是重要的。事实上，仅选出指定路由器，而没有后备指定路由器时，OSPF同样能工作；只是在指定路由器失效时，没有冗余而已，同时网络中的OSPF路由器需要再度进行一遍选举流程，以选出新的指定路由器。

在网段上（广播或非广播网络类型），所有非指定/后备指定路由器，都将与指定路由器与选出的后备指定路由器（若有选出后备指定路由器）建立邻接关系，而不会与网段上的其它非指定/后备指定路由器形成邻接关系。这些非指定/后备指定路由器，将往全指定路由器多播组地址（the AllDRRouters Multicast group address）`224.0.0.6`发出报文与更新。只有指定路由器与后备指定路由器才会收听发到此组地址的多播报文。随后指定路由器将通告报文给全SPF路由器多播组地址（the AllSPFRouters Multicast group address）`224.0.0.5`。这就令到网段上的所有其它OSPF路由器接收到更新。

对于已选出指定路由器与/或后备指定路由器时报文交换顺序的掌握尤为重要。下面作为一个示例，请设想一个有着4台路由器，分别为 R1、R2、R3 与 R4，的广播网络。假定 R4 被选作指定路由器，R3 被选作后备指定路由器。那么 R1 与 R2 就既不是指定也不是后备指定路由器了，因此在思科OSPF命名法中就被称为 DROther 路由器。此时 R1 上有一个配置改变，随后 R1 就发出一个更新报文到 AllDRRouters 多播组地址 `224.0.0.6`。指定路由器 R4 接收到该更新，并发出一个确认报文（acknowledgement），回送到 AllSPFRouters 多播组地址 `224.0.0.5` 上，R4 随后使用 AllSPFRouters 多播组地址，将该更新发送给所有其它非指定/后备指定路由器。该更新被其它的 DROther 路由器，也就是 R2 接收到，同时 R2 发出一个确认报文到 AllDRRouters 多播组地址 `224.0.0.6`。该过程在下图39.1中进行了演示：

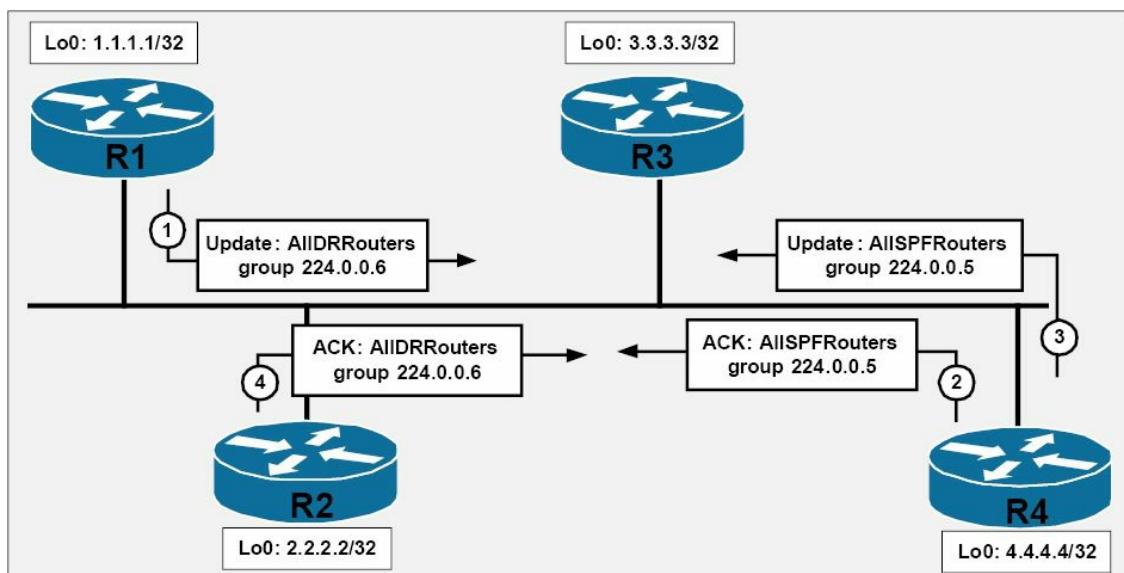


图39.1 -- OSPF的指定与后备指定通告

**注意：** 后备指定路由器仅简单地收听发送到 `224.0.0.5`（AllSPFRouters 多播组地址）与 `224.0.0.6`（AllDRRouters 多播组地址）两个多播组地址的数据包

某台路由器要成为网段的指定路由器或后备指定路由器，其就必须被选中。选举基于以下两点：

- 有着最高的路由器优先级值（the highest router priority value）
- 有着最高的路由器ID（the highest router ID）

默认下所有路由器都有着默认的优先级值 `1`。可使用接口配置命令 `ip ospf priority <0-255>` 对该值进行调整。路由器的优先级值越高，那么其就越有可能被选为该网段的指定路由器。有着第二高优先级的路由器，则将被选为后备指定路由器。如配置了优先级值 `0`，则该路由器将不参加到DR/BDR的选举过程。最

高路由器优先级与路由器ID原则，仅在参与指定/后备指定路由器选举过程的所有路由器同时加载OSPF进程时才起作用。如这些路由器没有同时加载OSPF进程，则最先完成OSPF进程加载的路由器，将成为网段上的指定路由器。

在确定OSPF路由器ID时，思科IOS将选取已配置的环回接口的最高IP地址。在没有配置环回接口时，IOS软件将使用所有已配置的物理接口的最高IP地址，作为OSPF路由器ID。思科IOS软件还允许管理员使用路由器配置命令 `router-id [address]`，来手动指定路由器ID。

重要的是记住在OSPF下，一旦指定路由器与后备指定路由器被选出，那么在进行另一次选举之前，它们都将始终作为指定/后备指定路由器。比如，在某个多路访问网络上已存在一台指定路由器与一台后备指定路由器的情况下，一台有着更高优先级或IP地址的路由器被加入到该同一网段时，既有的指定与后备指定路由器将不会变化。在指定路由器失效时，接过指定路由器的角色的是后备指定路由器，而不是新加入的有着更高优先级或IP地址的路由器。此外，一次新的选举将举行，而那台路由器极有可能被选举为后备指定路由器（Instead, a new election will be held and that router will most likely be elected BDR）。而为了让那台路由器成为指定路由器，就必须将后备路由器移除，或使用 `clear ip ospf` 命令对OSPF进程进行重置，以强制进行一次新的指定/后备指定路由器选举。一旦完成选举，OSPF像下面这样来使用指定与后备指定路由器：

- 用于减少网段上所要求的临接关系（To reduce the number of adjacencies required on the segment）
- 用于对多路访问网段上的路由器进行通告（To advertise the routers on the Multi-Access segment）
- 用于确保所有更新送达网段上的所有路由器

为了更好地掌握这些基础概念，这里参考下图39.2中的基本OSPF网络拓扑：

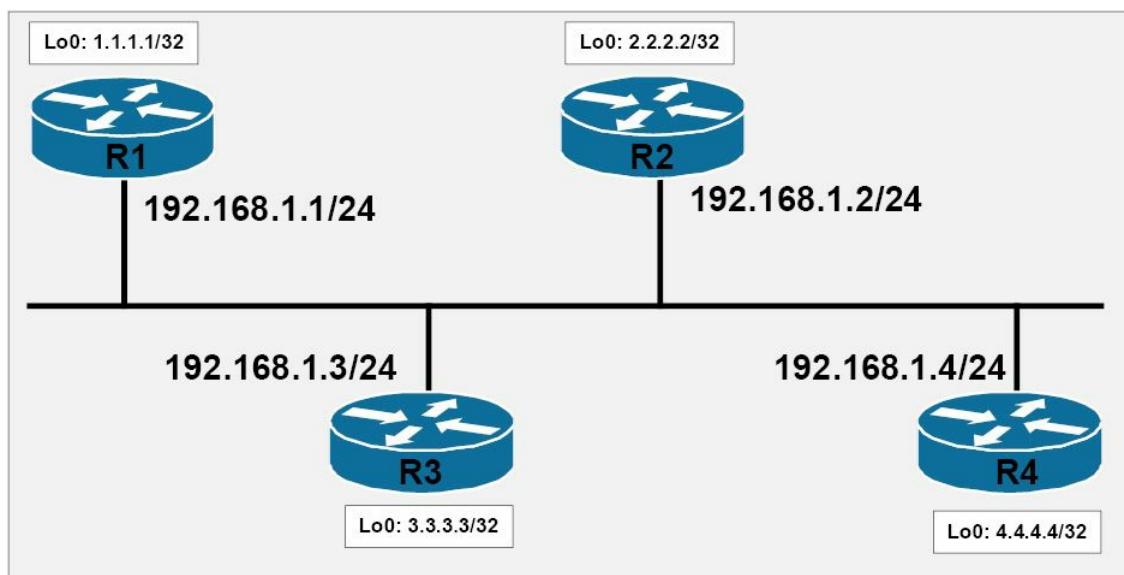


图 39.2 -- OSPF 指定与后备指定路由器基础

在图39.2中，网段上的各台路由器与指定及后备指定路由器之间建立临接关系，但相互之间并不建立临接关系（Referencing Figure 39.2, each router on the segment establishes an adjacency with the DR and the BDR but not with each other）。也就是说，非指定/后备指定路由器之间不会建立临接关系。这一特性阻止网段上的路由器形成相互之间的  $N(N-1)$  个临接关系，从而降低过多的OSPF数据包在网段上泛滥。

比如在没有网段上的指定/后备指定路由器概念时，各台路由器都需要与网段上的其它路由器建立临接关系。对于图39.2的情形，这将导致网段上的  $4(4-1)$ ，也就是 12 个临接关系。但有了指定/后备指定路由器后，每台路由器只需与这两台路由器，而无需与其它非指定与后备指定路由器，建立临接关系。指定路由器与后备指定路由器之间也会建立临接关系。此特性降低了网段及各台路由器上的临接关系数目，进而降低各台路由器上资源消耗（比如内存与处理器使用）。

对于第二点，OSPF将一条链路，视为两台路由器或两个节点之间的连接。在多路访问网络，比如以太网中，多台路由器可处于同一网段上，就如同图39.2中所演示的那样。在这样的网络中，OSPF使用网络链路状态通告（Network Link State Advertisement, Type 2 LSA, 类型2的链路状态通告），来对多路访问网段上的路由器进行通告。这种链路状态通告是由指定路由器生成，并仅在该区域传播。因为其它非指定/后备指定路由器并不在各自之间建立邻接关系，所以此类链路状态通告就令到那些路由器知悉在该多路访问网段上的其它路由器了。

为进一步说明这一点，这里参考图39.2，假定该网段上的所有路由器都具有默认的OSPF优先级 1（并同时加载OSPF进程），因为 R4 有着最高的路由器ID而被选为指定路由器。R3 因为有着第二高的路由器ID而被选为后备指定路由器。因为 R2 与 R1 既不是指定也不是后备指定路由器，因此它们被称为思科命名法中的 DROther 路由器。可在所有路由器上使用 show ip ospf neighbour 命令对此进行验证，如下所示：

```
R1#show ip ospf neighbor
Neighbor ID      Pri      State            Dead Time    Address          Interface
2.2.2.2          1        2WAY/DROTHER    00:00:38     192.168.1.2   Ethernet0/0
3.3.3.3          1        FULL/BDR       00:00:39     192.168.1.3   Ethernet0/0
4.4.4.4          1        FULL/DR        00:00:38     192.168.1.4   Ethernet0/0

R2#show ip ospf neighbor
Neighbor ID      Pri      State            Dead Time    Address          Interface
1.1.1.1          1        2WAY/DROTHER    00:00:32     192.168.1.1   FastEthernet0/0
3.3.3.3          1        FULL/BDR       00:00:33     192.168.1.3   FastEthernet0/0
4.4.4.4          1        FULL/DR        00:00:32     192.168.1.4   FastEthernet0/0

R3#show ip ospf neighbor
Neighbor ID      Pri      State            Dead Time    Address          Interface
1.1.1.1          1        FULL/DROTHER   00:00:36     192.168.1.1   FastEthernet0/0
2.2.2.2          1        FULL/DROTHER   00:00:36     192.168.1.2   FastEthernet0/0
4.4.4.4          1        FULL/DR        00:00:35     192.168.1.4   FastEthernet0/0

R4#show ip ospf neighbor
Neighbor ID      Pri      State            Dead Time    Address          Interface
1.1.1.1          1        FULL/DROTHER   00:00:39     192.168.1.1   FastEthernet0/0
2.2.2.2          1        FULL/DROTHER   00:00:39     192.168.1.2   FastEthernet0/0
3.3.3.3          1        FULL/BDR       00:00:30     192.168.1.3   FastEthernet0/0
```

**注意：**那些 DROther 路由器之所以处于 2WAY/DROTHER 状态，是因为它们仅与指定及后备指定路由器交换它们的数据库。那么就因为 DROther 路由器之间没有完整的数据库交换，所以它们绝不会达到 OSPF完整邻接状态（The DROther routers remain in the 2WAY/DROTHER state because they exchange their databases only with the DR and BDR routers. Therefore, because there is no full database exchange between the DROther routers, they will never reach the OSPF FULL adjacency state）。

因为 R4 已被选为指定路由器，它就生成网络链路状态通告（the Network LSA），这类链路状态通告，是就该多路访问网段上的其它路由器进行通告的。可在网段上的任意路由器上，使用 show ip ospf database network [link state ID] 命令，或在指定路由器上使用 show ip ospf database network self originate 命令，对此加以验证。下面演示了在指定路由器（R4）上命令 show ip ospf database network self originate 命令的输出：

```
R4#show ip ospf database network self-originate
    OSPF Router with ID (4.4.4.4) (Process ID 4)
        Net Link States (Area 0)
    Routing Bit Set on this LSA
    LS age: 429
    Options: (No TOS-capability, DC)
    LS Type: Network Links
    Link State ID: 192.168.1.4 (address of Designated Router)
    Advertising Router: 4.4.4.4
    LS Seq Number: 80000006
    Checksum: 0x7E08
    Length: 40
    Network Mask: /24
        Attached Router: 4.4.4.4
        Attached Router: 1.1.1.1
        Attached Router: 2.2.2.2
        Attached Router: 3.3.3.3
```

参考上面的输出，指定路由器（R4）发起了表示 192.168.1.0/24 子网的类型2（网络）链路状态通告（the Type 2(Network) LSA）。因为该子网上存在多台路由器，所以该 192.168.1.0/24 子网被称作OSPF命名法中的一条传输链路（a transit link in OSPF terminology）。输出中的通告路由器字段（the Advertising Router field）显示了生成此链路状态通告的那台路由器（4.4.4.4，R4）。网络掩码字段（the Network Mask field）则显示了该传输网络的子网掩码，也就是24位，或 255.255.255.0。

注：OSPF中链路类型（link type）有4种：P2P、Stub、Transit与Virtual link; 网络类型（network type）有两种：传输网络（Transit network）与末梢网络（Stub network）；链路状态通告有六种：Router LSA（一类）、Network LSA（二类）、Network summary LSA（三类）、ASBR summary LSA（四类）、AS external LSA（五类）与 NSSA LSA（七类）。[参考链接](#)。

所连接路由器字段（the Attached Router field）列出了在该网络网段上所有路由器的路由器ID。这样就令到该网段上的所有路由器，知悉有哪些其它路由器也同样位处该网段上。下面的输出，演示了在 R1、R2 与 R3 上的 show ip ospf database network [link state ID] 命令的输出，反映出同样的信息：

```
R2#show ip ospf database network
    OSPF Router with ID (2.2.2.2) (Process ID 2)
        Net Link States (Area 0)
    Routing Bit Set on this LSA
    LS age: 923
    Options: (No TOS-capability, DC)
    LS Type: Network Links
    Link State ID: 192.168.1.4 (address of Designated Router)
    Advertising Router: 4.4.4.4
    LS Seq Number: 80000006
    Checksum: 0x7E08
    Length: 40
    Network Mask: /24
        Attached Router: 4.4.4.4
        Attached Router: 1.1.1.1
        Attached Router: 2.2.2.2
        Attached Router: 3.3.3.3

R1#show ip ospf database network
    OSPF Router with ID (1.1.1.1) (Process ID 1)
        Net Link States (Area 0)
    Routing Bit Set on this LSA
    LS age: 951
    Options: (No TOS-capability, DC)
    LS Type: Network Links
    Link State ID: 192.168.1.4 (address of Designated Router)
    Advertising Router: 4.4.4.4
    LS Seq Number: 80000006
    Checksum: 0x7E08
    Length: 40
    Network Mask: /24
        Attached Router: 4.4.4.4
        Attached Router: 1.1.1.1
        Attached Router: 2.2.2.2
        Attached Router: 3.3.3.3
    OSPF Router with ID (4.4.4.4) (Process ID 4)

R3#show ip ospf database network
    OSPF Router with ID (3.3.3.3) (Process ID 3)
        Net Link States (Area 0)
    Routing Bit Set on this LSA
    LS age: 988
    Options: (No TOS-capability, DC)
    LS Type: Network Links
    Link State ID: 192.168.1.4 (address of Designated Router)
    Advertising Router: 4.4.4.4
    LS Seq Number: 80000006
    Checksum: 0x7E08
    Length: 40
    Network Mask: /24
        Attached Router: 4.4.4.4
        Attached Router: 1.1.1.1
        Attached Router: 2.2.2.2
        Attached Router: 3.3.3.3
```

网络链路状态通告的功能及其与其它类型的链路状态通告，特别是与路由器链路通告（类型一，the Router LSA(Type 1)）的关系，将在本模块稍后进行详细介绍。本小节的重点，应放在对指定路由器就多路访问网段上的网络链路状态通告的生成与通告，从而完成对位处该同一网段上的其它路由器的通告，这一过程的

理解上。这是因为网段上的路由器仅与指定及后备指定路由器建立邻接关系，而相互之间并不建立邻接关系。在没有相互之间的邻接关系下，路由器就绝不会知道该多路访问网段上的其它非指定/后备指定路由器。

最后，有关指定/后备指定路由器上的第三点，指定/后备指定路由器确保了网段上的所有路由器都有着完整的数据库。非指定/后备指定路由器将更新发送到多播组地址 224.0.0.6（AllDRRouter）。那么指定路由器就通过将该更新发送到多播组地址 224.0.0.5（AllSPFRouters），将这些更新通告给其它非指定/后备指定路由器。下图39.3演示了从 R1（一台 DROther）发往指定路由器组地址，涉及图39.2中的那些路由器的一个更新：

```
* Internet Protocol, Src: 192.168.1.1 (192.168.1.1), Dst: 224.0.0.6 (224.0.0.6)
  Open Shortest Path First
  OSPF Header
  LS Update Packet
    Number of LSAs: 1
    LS Type: Router-LSA
      LS Age: 1 seconds
      Do Not Age: False
    Options: 0x22 (DC, E)
      Link-State Advertisement Type: Router-LSA (1)
      Link State ID: 1.1.1.1
      Advertising Router: 1.1.1.1 (1.1.1.1)
      LS Sequence Number: 0x80000006
      LS Checksum: 0x5f95
      Length: 60
    Flags: 0x00 ()
    Number of Links: 3
    Type: Stub     ID: 10.10.10.10      Data: 255.255.255.255 Metric: 1
    Type: Stub     ID: 1.1.1.1        Data: 255.255.255.255 Metric: 1
    Type: Transit  ID: 192.168.1.1      Data: 192.168.1.1      Metric: 10
```

图 39.3 -- 发到指定/后备指定路由器组地址的一个 DROther 更新

R4（指定路由器）收到该更新，并接着将相同更新发送到多播组地址 224.0.0.5（AllSPFRouters）。该组地址是由所有OSPF路由器使用的，以确保网段上的所有其它路由器都收到此更新。下图39.4对来自 R4（指定路由器）的该更新进行了演示：

```
* Internet Protocol, Src: 192.168.1.4 (192.168.1.4), Dst: 224.0.0.5 (224.0.0.5)
  Open Shortest Path First
  OSPF Header
  LS Update Packet
    Number of LSAs: 1
    LS Type: Router-LSA
      LS Age: 2 seconds
      Do Not Age: False
    Options: 0x22 (DC, E)
      Link-State Advertisement Type: Router-LSA (1)
      Link State ID: 1.1.1.1
      Advertising Router: 1.1.1.1 (1.1.1.1)
      LS Sequence Number: 0x80000006
      LS Checksum: 0x5f95
      Length: 60
    Flags: 0x00 ()
    Number of Links: 3
    Type: Stub     ID: 10.10.10.10      Data: 255.255.255.255 Metric: 1
    Type: Stub     ID: 1.1.1.1        Data: 255.255.255.255 Metric: 1
    Type: Transit  ID: 192.168.1.4      Data: 192.168.1.1      Metric: 10
```

图 39.4 -- 到OSPF组地址的指定路由器更新

**注意：**可以看出这就是来自 R1 的更新，因为图39.3与图39.4中的通告路由器字段（the Advertising Router field）都包含了 R1 的路由器ID（the router ID, RID），也就是 1.1.1.1。

**注意：**OSPF使用到的其它LSA类型，将在本模块的后面详细介绍。

## 额外的路由器类型 (Additional Router Types)

除了多路访问网段上的指定与后备指定路由器外，对OSPF路由器的描述方式，还包括根据它们的位置与在OSPF网络中的作用。在OSPF网络中，通常会发现以下额外的路由器类型：

- 区域边界路由器（Area Border Routers）
- 自治系统边界路由器（Autonomous System Boundary Routers）
- 内部路由器（Internal Routers）
- 骨干路由器（Backbone Routers）

下图39.5演示了一个后两个区域--一个OSPF骨干区域（the OSPF backbone area( Area 0 )）与一个额外的一般OSPF区域（a additional normal OSPF area( Area 2 )），构成的基本OSPF网络。R2 有着一个与 R1 的外部边界网关协议邻居关系（an external BGP neighbour relationship）。该图例将用于描述此网络中不同的OSPF路由器类型。

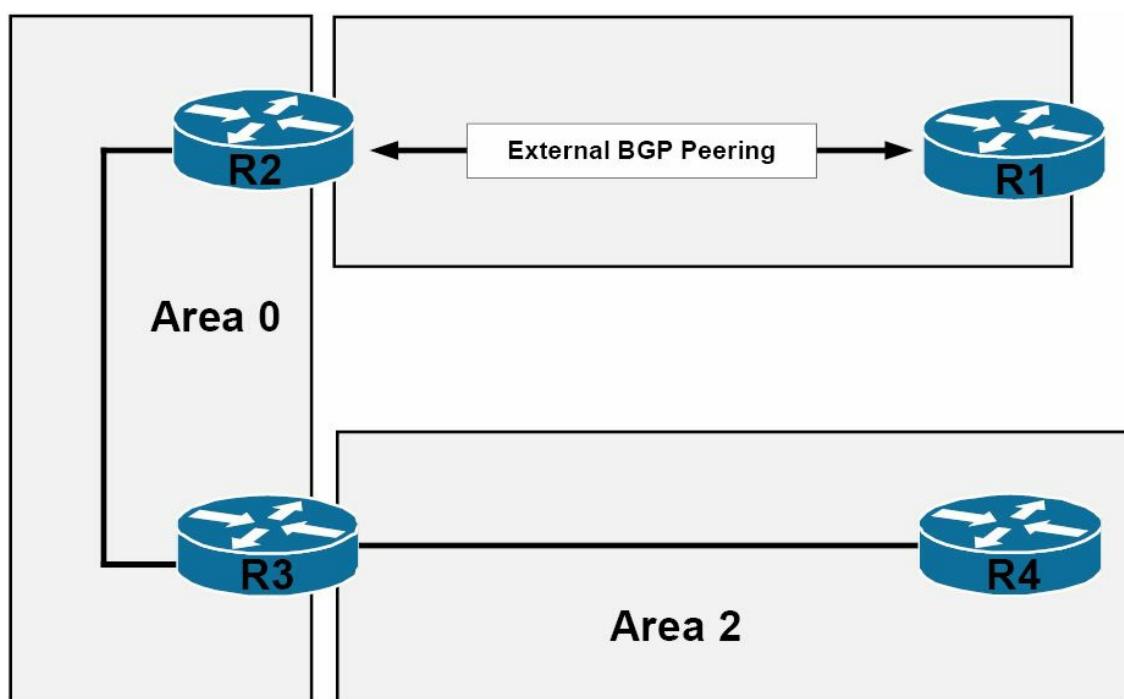


图 39.5 -- 额外的OSPF路由器类型

区域边界路由器（An Area Border Router, ABR），是一台将一个或多个OSPF区域，连接到OSPF骨干的OSPF路由器。这就意味着其必须有一个接口在 Area 0 中，同时有其它接口在某个不同的OSPF区域中。区域边界路由器是所有其归属区域的成员，且它们保有着每个其所归属区域的一个单独链路状态数据库

(ABRs are members of all areas to which they belong, and they keep a separate Link State Database for every area to which they belong)。参考图39.5，R3 就应被认为是一台区域边界路由器，因为它将 Area 2 连接到了OSPF骨干 Area 0。

而传统意义上的自治系统边界路由器，则是位处路由域的边沿，且定义了内部与外部网络的边界（An Autonomous System Boundary Router(ASBR), in the traditional sense, resides at the edge of the routing domain and defines the boundary between the internal and the external networks）。参考图39.5，R2 将被认为是一台自治系统边界路由器。除了注入来自其它协议（比如BGP）的路由信息外，在某台路由器将静态路由或是所连接的子网，注入到OSPF网络时，也可将其划分为自治系统边界路由器。

内部路由器的所有运作接口，都保持在单个的OSPF区域中。基于图39.5中演示的网络拓扑，R4 将被视为一台内部路由器，因为其仅有的接口，处于单个的OSPF区域中。

骨干路由器是那些有一个接口在OSPF骨干中的路由器。骨干路由器可以包括那些有着仅在OSPF骨干区域的接口的路由器，或者有一个接口在OSPF骨干区域，也有接口在其它区域的路由器（也就是区域边界路由器）。基于图39.5中演示的拓扑，路由器 R2 与 R3 都可被视为骨干路由器。

**注意：** OSPF的路由器可有多个角色。比如上面的 R2 就同时是一台自治系统边界路由器及骨干路由器， R3 又同时是一台骨干路由器与区域边界路由器。贯穿本模块，将详细审视这些类型的路由器与其在OSPF域中的角色与功能。

## OSPF数据包类型

OSPF路由器发出的不同类型数据包，包含在这些数据包共有的、24字节的OSPF头部（The different types of packets sent by OSPF routers are contained in the common 24-byte OSPF header）。尽管对该OSPF头部细节的深入，超出了CCNA考试要求的范围，但对该头部中所包含的各个字段，以及它们各自用途的基本掌握，仍然重要。下图39.6对此各种数据包共有的24个八位OSPF头部，进行了演示：

Version	Type	Packet Length
<b>Router ID</b>		
<b>Area ID</b>		
Checksum		Authentication Type
<b>Authentication Data</b>		

图 39.6 - OSPF协议数据包头部

其中的8位版本字段，指出了OSPF的版本。该字段的默认值是 2 。但在开启了OSPFv3时，该字段就被设置为 3 。在第13天时，对OSPFv3进行了详细介绍。

接着的8位类型字段，用于指明该OSPF数据包的类型。五种主要的OSPF数据包类型，将在本课程模块接下来进行介绍，它们是：

- 类型1 = Hello 数据包
- 类型2 = 数据库描述数据包 (Database Description packet)
- 类型3 = 链路状态请求数据包 (Link State Request packet)
- 类型4 = 链路状态更新数据包 (Link State Update packet)
- 类型5 = 链路状态确认数据包 (Link State Acknowledgement packet)

随后的16位数据包长度字段，是用于指明该协议数据包的长度。此长度包括了标准的OSPF头部。

下面的32位路由器ID自动，用于指明发出数据包的路由器的IP地址。在思科IOS设备上，该字段将包含运行OSPF的设备上配置的所有物理接口的最高的IP地址。如在设备上配置了环回接口 (Loopback interfaces)，那么该字段将包含所有配置的环回接口的最高IP地址。或者在显式地有管理员配置或指定了路由器ID时，该字段也可包含那个手动配置的路由器ID。

**注意:** 除非重启了路由器，或者获取IP地址的那个接口被关闭或移除，抑或在路由器上使用了提权的 EXEC 命令 `clear ip ospf process` 命令重置了OSPF进程，否则在路由器ID被选出后，该路由器ID都不会发生改变。

接下来的32位区域ID（Area ID），用于区分该数据包的OSPF区域（the OSPF area）。数据包只能属于单个OSPF区域。在数据包是通过虚拟链路（a virtual link）接收到的时，那么区域ID就会是OSPF的骨干区域，也就是 Area 0。本课程模块后面会对虚拟链路进行介绍。

校验和字段是16位长的，它指出了该数据包完整内容，从OSPF头部开始，但排除了64位的认证数据字段的标准IP校验和。如该数据包的长度不是正数个的16位字（16-bit words）长时，则会在进行检验和检查钱，以全 0 字节加以补充。

其后的16位认证类型字段（The 16-bit Authentication(Auth) Type field）指出所使用的认证的类型。该字段仅对OSPFv2有效，且可能包含以下3个代码之一：

- `Code 0` - 意思是空（0）认证，也就是没有认证；这是默认选项
- `Code 1` - 表明认证类型是普通文本（the authentication type is plain text）
- `Code 2` - 意思是认证类型为消息摘要算法（MD5, Message Digest Algorithm）

OSPF头部最后的64位认证数据字段，则是在开启了认证时，用于具体的认证信息或数据。重要的是记住该字段仅对OSPFv2有效。在使用的是普通文本认证时，该自动包含了认证密钥（the authentication key）。但在使用的是MD5认证时，该自动就被重新定义为几个其它字段，不过这超出了CCNA考试要求范围。下图39.7显示了线路上捕获到的OSPF数据包的不同字段：

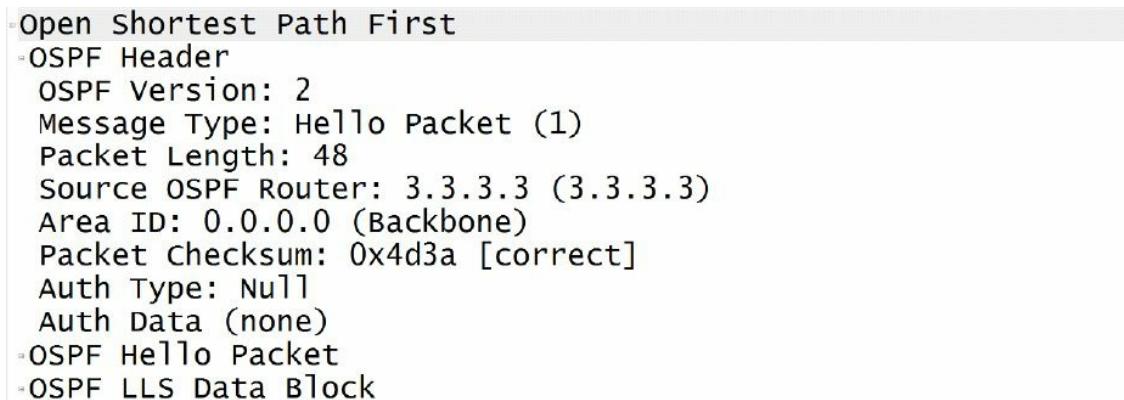


图 39.7 - OSPF 数据包头部的线上捕获

在OSPF数据包头部里头，8位的类型字段用于指明OSPF数据包的类型。这里再度说明一下，如下所示的5种OSPF数据包类型：

- 类型1 = Hello 数据包
- 类型2 = 数据库描述数据包（Database Description packet）
- 类型3 = 链路状态请求数据包（Link State Request packet）
- 类型4 = 链路状态更新数据包（Link State Update packet）
- 类型5 = 链路状态确认数据包（Link State Acknowledgement packet）

## OSPF Hello 数据包

Hello 数据包用于发现其它直接相连的OSPF路由器，以及在OSPF路由器之间建立OSPF邻接关系（OSPF adjacencies between OSPF routers）。对于广播及点对点网络，OSPF使用多播来发送 Hello 数据包。这些数据包被投送到 AllSPFRouters 多播组地址 224.0.0.5。对于非广播链路（比如帧中继），OSPF使用单播（Unicast）来将 Hello 数据包直接发送给那些静态配置的邻居。

**注意：**默认情况下，所有OSPF数据包（也就是包括多播与单播），都是以IP存活时间（TTL, Time To Live）`1`发送的。这就将这些数据包限制到本地链路。也就是说，无法与距离远于一跳的另一台路由器建立OSPF邻接关系。这一点也适用于EIGRP。

OSPF的Hello数据包，还在广播链路上用于指定路由器与后备指定路由器的选举。指定路由器仅侦听多播地址`224.0.0.6`（AllDRouters）。本课程模块前面已经介绍了指定与后备指定路由器。下图39.8演示了OSPF Hello数据包中所包含的字段：

```

- Open Shortest Path First
- OSPF Header
- OSPF Hello Packet
  Network Mask: 255.255.255.0
  Hello Interval: 10 seconds
- Options: 0x12 (L, E)
  0... .... = DN: DN-bit is NOT set
  .0.. .... = O: O-bit is NOT set
  ..0. .... = DC: Demand circuits are NOT supported
  ...1 .... = L: The packet contains LLS data block
  ....0... = NP: Nssa is NOT supported
  ....0.. = MC: NOT multicast capable
  ....1.. = E: ExternalRoutingCapability
  Router Priority: 1
  Router Dead Interval: 40 seconds
  Designated Router: 192.168.1.3
  Backup Designated Router: 192.168.1.2
  Active Neighbor: 20.2.2.2
- OSPF LLS Data Block

```

图 39.8 - OSPF 的 Hello 数据包

其中**4字节的网络掩码字段**包含了通告OSPF接口的子网掩码（The 4-byte Network Mask field contains the subnet mask of the advertising OSPF interface）。只有在广播介质上，才会检查子网掩码。对于本模块后面会介绍的未编号的点对点接口及虚拟链路，该字段将被设置为`0.0.0.0`。

后面两字节的Hello字段，显示Hello时间间隔，也就是两个Hello数据包之间的秒数，通告路由器要求此字段。其取值范围为`1`到`255`。在广播与点对点介质上的默认值为`10`，在所有其它介质上的默认值为`30`。

随后1字节的选项字段（The 1-byte Options field）是由本地路由器用于通告可选功能（optional capabilities）。选项字段中的每一位，都表示了不同的功能。深入了解这些位所对应的功能，超出了CCNA考试要求范围。

后面的1字节路由器优先级字段（The 1-byte Router Priority field）包含了本地路由器的优先级。默认该字段的值为`1`。该值用于指定与后备指定路由器的选举。可能的取值范围为`0`到`255`。优先级越高，那么该本地路由器就越有机会成为指定路由器。优先级值`0`就意味着该本地路由器不会参与指定或后备指定路由器的选举。

接下来的4字节指定路由器字段，列出指定路由器的IP地址。在比如某个点对点链路上，或当某台路由器已被显式地配置为不参与此选举时，于尚无指定路由器选出时，就使用值`0.0.0.0`。

其后的4字节后备指定路由器字段，标识出后备路由器，并列出当前后备指定路由器的接口地址。在没有选出后备指定路由器时，就使用值`0.0.0.0`。

最后的（活动）邻居字段（the (Active) Neighbour field）是一个可变长度的字段，显示网段上的所有已接收到Hello数据包的OSPF路由器。

## 数据库描述数据包（Database Description Packets）

在各台OSPF路由器对其本地数据库信息进行通告时，于数据库交换期间，就要用到数据库描述数据包。这些数据包通常被称作DBD数据包或DD数据包。第一个DBD数据包用于数据库交换过程的主从选举。DBD数据包还包含了由主路由器选定的初始序列编号（The first DBD packet is used for the Master and Slave

election for database exchange. The DBD packet also contains the initial sequence number selected by the Master)。有着较高路由器ID的路由器，成为主路由器并发起数据库同步过程。只有主路由器才能增加DBD数据包中的序列编号。主路由器开启数据库交换，并对从路由器进行信息轮询。数据库交换中的主从选举，是在邻居对的基础上进行的。

明白主从选举过程不同于指定与后备指定路由器的选举过程，尤为重要。通常后错误地假定它们一致（This is commonly incorrectly assumed）。主从选举过程只基于有着最高IP地址的路由器（两台邻居路由器之间）一条原则；但指定与后备指定路由器选举过程，则由IP地址或优先级值两个因素决定。

比如这里假设，两台名为 R1 与 R2 的路由器开始了邻接关系建立过程。R1 有着路由器ID 1.1.1.1，同时 R2 有着路由器ID 2.2.2.2。网络管理员将 R1 的OSPF优先级值配置为 255 以确保该路由器被选举为指定路由器。在主从关系确定过程中，R2 因为有着较高的路由器ID优势，而将被选举为主路由器。但在 R1 上配置的优先级值，导致 R1 被选举为指定路由器。而实际上，在主从选举过程中，作为指定路由器的 R1 就可作为从路由器。

在选出了主从路由器后，本地路由器就通过往对方路由器发送LSA头部，而使用DBD数据包对本地数据库进行概括（After the Master and Slave have been elected, DBD packets are used to summarise the local database by sending LSA headers to the remote router. LSA, Link-State Advertisement, 链路状态通告）。远端路由器分析这些头部，以判断在其自己的LSDB拷贝中是否缺少什么信息。下图39.9中对数据库描述数据包进行了演示：

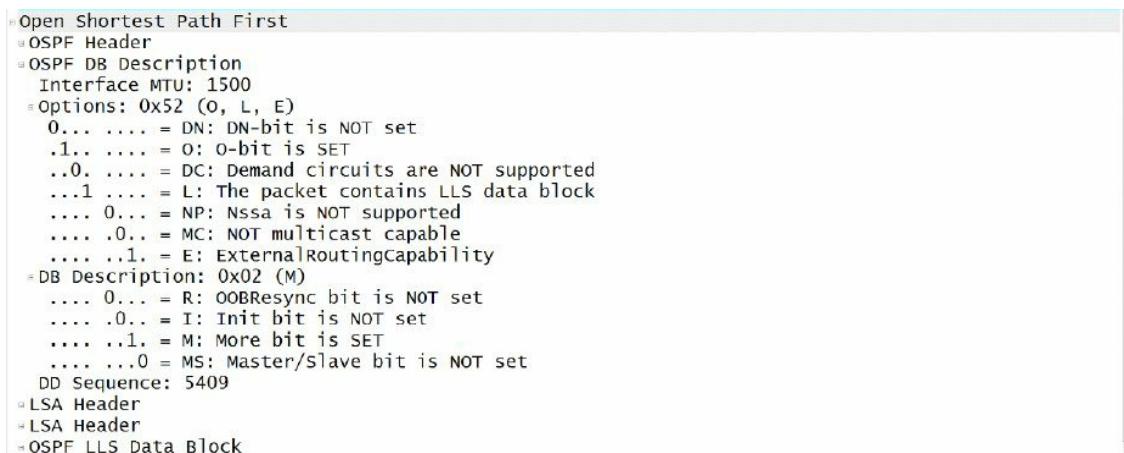


图 39.9 - OSPF 的数据库描述数据包

在DBD数据包中，两字节的接口MTU字段包含了发出接口的8位二进制的MTU值（the 2-byte Interface MTU field contains the MTU value, in octets, of the outgoing interface）。也就是说，该字段包含了通过相关接口所能发送的最大数据大小（以字节计）。当在虚拟链路上使用接口时，该字段就被设置为值 0x0000。有了成功建立OSPF的邻居邻接关系，所有路由器上的MTU必须一致。如在一台路由器上修改了这个值，就必须在相同子网的所有其它路由器上配置同样的值（或使用 ip ospf mtu-ignore 命令）。

**注意：**对于EIGRP来说，不必为了成功建立EIGRP的邻居关系，而要求接口MTU一致。

随后的1字节选项字段，包含的是与OSPF Hello 数据包相同的选项。为简明起见，不再对这些选项进行描述。

其后的数据库描述或标志字段，是一个1字节的、在邻接关系形成过程中，提供某台OSPF路由器可否就多个DBD数据包，与邻居进行交换的能力的字段（The Database Description or Flags field is a 1-byte field that provides an OSPF router with the capability to exchange multiple DBD packets with a neighbour during an adjacency formation）。

接着的4字节DBD序列号字段（The 4-byte Sequence Number field），通过使用一个序列号，而用于确保所有DBD数据包在同步过程中，得以接收与处理。主路由器在第一个DBD数据包中，将该字段初始化为一个独特值，其后的每个数据包的序列号都增加 1。序列号的增加，仅由主路由器进行。

最后的可变长度LSA头部字段（the variable length LSA Header），运送的是描述本地路由器信息的多个LSA头部。每个头部长度为20个8位二进制数，并对数据库中的各个LSA进行唯一地区别。每个DBD数据包可包含多个LSA头部。

## 链路状态请求数据包（Link State Request Packets）

链路状态请求数据包，是由OSPF路由器发送的，用以请求缺失的或过期的数据库信息。这些数据包包含了对所请求的链路状态通告进行独特描述的标识符。单个的链路状态请求数据包可能包含了请求多条链路状态通告的单个的标识符集或多个的标识符集。链路状态请求数据包还用于在数据库交换之后的，对数据库交换期间本地路由器不曾有的那些链路状态通告的请求。下图39.10对OSPF的链路状态请求数据包格式的演示：

```

- Open Shortest Path First
  - OSPF Header
  - Link State Request
    Link-State Advertisement Type: Router-LSA (1)
    Link State ID: 3.3.3.3
    Advertising Router: 3.3.3.3 (3.3.3.3)
  - Link State Request
    Link-State Advertisement Type: Network-LSA (2)
    Link State ID: 192.168.1.3
    Advertising Router: 3.3.3.3 (3.3.3.3)

```

图 39.10 - OSPF 链路状态请求数据包

其中的4字节链路状态通告类型字段（The 4-byte Link State Advertisement Type field）包含了所请求的链路状态通告类型。其可包含下列字段之一：

- 类型1 = 路由器链路状态通告
- 类型2 = 网络链路状态通告
- 类型3 = 网络汇总链路状态通告
- 类型4 = 自治系统边界路由器链路状态通告
- 类型5 = 自治系统外部链路状态通告
- 类型6 = 多播链路状态通告
- 类型7 = 次末梢区域外部链路状态通告（Not-So-Stubby Area, NSSA）
- 类型8 = 外部属性链路状态通告（External Attributes Link State Advertisement）
- 类型9 = 本地链路的不透明链路状态通告（Opaque LSA - Link Local, \*目前主要用于MPLS多协议标签交换协议）
- 类型10 = 区域的不透明链路状态通告（Opaque LSA - Area, \*目前主要用于MPLS多协议标签交换协议）
- 类型11 = 自治系统的不透明链路状态通告（Opaque LSA - Autonomous System, \*目前主要用于MPLS多协议标签交换协议）

**注意：**一些上面列出的链路状态通告将在后面的小节进行讲解。

4字节的链路状态ID字段，编码了特定于LSA的信息。包含在该字段的信息根据LSA的种类而有所不同。最后的4字节通告路由器字段，包含的是最先发起LSA的路由器的路由器ID。

## 链路状态更新数据包（Link State Update Packets）

链路状态更新 (LSU) 数据包是由路由器用于对链路状态通告进行通告的数据包 (advertise Link State Advertisements)。链路状态更新数据包可以是到OSPF邻居的单播，作为从邻居处接收到链路状态请求的回应。然而最为常见的是，它们被可靠地在整个网络中泛洪到 AllSPFRouters 多播组地址 224.0.0.5，直到所有路由器都有一份数据库的拷贝位置。所泛洪的更新，于随后在链路状态通告的确认数据包中加以确认。如链路状态通告未被确认，就会默认每隔5秒加以重传。下图39.11展示了一个发送给某个邻居的、作为LSR响应的链路状态更新数据包：

```
Internet Protocol, Src: 192.168.1.3 (192.168.1.3), Dst: 192.168.1.2 (192.168.1.2)
  Open Shortest Path First
    OSPF Header
      LS Update Packet
        Number of LSAs: 1
        LS Type: Summary-LSA (IP network)
          LS Age: 3600 seconds
          Do Not Age: False
          Options: 0x22 (DC, E)
            Link-State Advertisement Type: summary-LSA (IP network) (3)
            Link State ID: 150.1.1.0
            Advertising Router: 20.2.2.2 (20.2.2.2)
            LS Sequence Number: 0x80000001
            LS Checksum: 0x70d9
            Length: 28
            Netmask: 255.255.255.0
            Metric: 64
```

图 39.11 - 单播的LSU数据包

下图 39.12 演示了一个可靠地泛洪到多播组地址 224.0.0.5 的LSU：

```
Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 224.0.0.5 (224.0.0.5)
  Open Shortest Path First
    OSPF Header
      LS Update Packet
        Number of LSAs: 1
        LS Type: Summary-LSA (IP network)
          LS Age: 1 seconds
          Do Not Age: False
          Options: 0x22 (DC, E)
            Link-State Advertisement Type: summary-LSA (IP network) (3)
            Link State ID: 150.1.1.0
            Advertising Router: 20.2.2.2 (20.2.2.2)
            LS Sequence Number: 0x80000002
            LS Checksum: 0x6eda
            Length: 28
            Netmask: 255.255.255.0
            Metric: 64
```

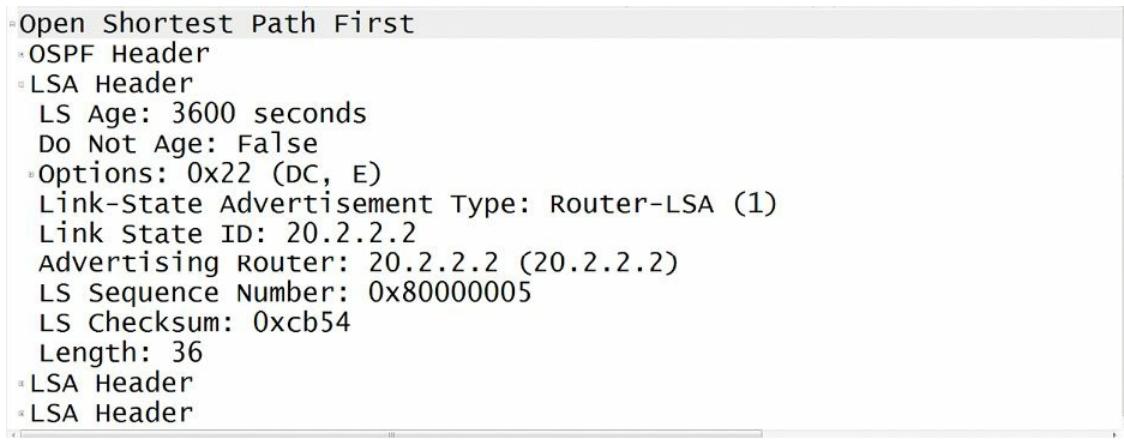
图 39.12 - 多播LSU数据包

链路状态更新数据包由两部分构成。第一部分是4字节的链路状态通告数目字段 (the 4-byte Number of LSAs field)。该字段显式了LSU数据包中所运送的LSA条数。第二部分则是一条或多条的链路状态通告。此可变长度字段包含了完整的LSA。每种类型的LSA都有共同的头部格式，与其各自特定的用来描述各自信息的数据字段。一个LSU数据包可包含单一的LSA或多条的LSA。

## 链路状态确认数据包 (Link State Acknowledgement Packets)

链路状态确认数据包 (LSAck) 用于对各条LSA进行确认及作为对LSU数据包的响应。通过显式地使用链路状态确认数据包来对泛洪的数据包加以确认，OSPF所使用的泛洪机制被认为是可靠的。

链路状态确认数据包包含了一般的OSPF头部，以及随后的一个LSA头部清单。此可变长度字段允许本地路由器以单个数据包对多条LSA进行确认。链路状态确认数据包是以多播发送的。在多路访问网络上，如果发送LSAck的是指定或后备指定路由器，那么这些LSAck就被发送到多播组地址 224.0.0.5 (AllSPFRouters)。而如果发送LSAck的不是指定或后备指定路由器，那么这些LSAck数据包就被发送到多播组地址 224.0.0.6 (AllDRRouters)。下图39.13对LSAck的格式进行了演示：



The screenshot shows a terminal window with the following output:

```
Open Shortest Path First
  OSPF Header
    LSA Header
      LS Age: 3600 seconds
      Do Not Age: False
    Options: 0x22 (DC, E)
    Link-State Advertisement Type: Router-LSA (1)
    Link State ID: 20.2.2.2
    Advertising Router: 20.2.2.2 (20.2.2.2)
    LS Sequence Number: 0x80000005
    LS Checksum: 0xcb54
    Length: 36
  LSA Header
  LSA Header
```

图 39.13 - 链路状态确认数据包

总之，重要的是记住不同的OSPF数据包类型及它们所包含的信息。这将不仅有助于考试，也可在将OSPF作为一个协议的整个运作进行掌握的过程中有所裨益。

在思科IOS软件中，可使用 `show ip ospf traffic` 命令来查看OSPF数据包的统计信息。该命令展示了发送及接收的OSPF数据包的总数，并将这些OSPF数据包细分到单独的OSPF进程，最终又细分到具体进程下开启了OSPF进程的各个接口上。该命令也可用于对OSPF邻接关系建立的故障排除，其作为调试用途时，不是处理器占用密集的方式。下面的输出中演示了该命令所打印的信息：

```

R4#show ip ospf traffic
OSPF statistics:
  Rcvd: 702 total, 0 checksum errors
    682 hello, 3 database desc, 0 link state req
    12 link state updates, 5 link state acks
  Sent: 1378 total
    1364 hello, 2 database desc, 1 link state req
    5 link state updates, 6 link state acks

    OSPF Router with ID (4.4.4.4) (Process ID 4)
  OSPF queue statistics for process ID 4:
    InputQ      UpdateQ      OutputQ
  Limit          0          200          0
  Drops          0           0           0
  Max delay [msec] 4           0           0
  Max size        2           2           2
    Invalid        0           0           0
    Hello          0           0           1
    DB des         2           2           1
    LS req         0           0           0
    LS upd         0           0           0
    LS ack         0           0           0
  Current size    0           0           0
    Invalid        0           0           0
    Hello          0           0           0
    DB des         0           0           0
    LS req         0           0           0
    LS upd         0           0           0
    LS ack         0           0           0

  Interface statistics:
    Interface Serial0/0
  OSPF packets received/sent
    Invalid  Hellos  DB-des  LS-req  LS-upd  LS-ack  Total
  Rx: 0       683      3        0       12       5      703
  Tx: 0       684      2        1       5       6      698
  OSPF header errors
    Length 0, Auth Type 0, Checksum 0, Version 0,
    Bad Source 0, No Virtual Link 0, Area Mismatch 0,
    No Sham Link 0, Self Originated 0, Duplicate ID 0,
    Hello 0, MTU Mismatch 0, Nbr Ignored 0,
    LLS 0, Unknown Neighbor 0, Authentication 0,
    TTL Check Fail 0,
  OSPF LSA errors
    Type 0, Length 0, Data 0, Checksum 0,
    Interface FastEthernet0/0
  OSPF packets received/sent
    Invalid  Hellos  DB-des  LS-req  LS-upd  LS-ack  Total
  Rx: 0       0        0        0       0       0       0
  Tx: 0       682      0        0       0       0      682
  OSPF header errors
    Length 0, Auth Type 0, Checksum 0, Version 0,
    Bad Source 0, No Virtual Link 0, Area Mismatch 0,
    No Sham Link 0, Self Originated 0, Duplicate ID 0,
    Hello 0, MTU Mismatch 0, Nbr Ignored 0,
    LLS 0, Unknown Neighbor 0, Authentication 0,
    TTL Check Fail 0,
  OSPF LSA errors
    Type 0, Length 0, Data 0, Checksum 0,
  Summary traffic statistics for process ID 4:
  Rcvd: 703 total, 0 errors
    683 hello, 3 database desc, 0 link state req
    12 link state upds, 5 link state acks, 0 invalid

```

```
Sent: 1380 total
      1366 hello, 2 database desc, 1 link state req
      5 link state upds, 6 link state acks, 0 invalid
```

## 临接关系的建立 (Establishing Adjacencies)

运行OSPF的路由器在建立临接关系之前，会经历几种状态。在这些状态期间，路由器要交换不同类型的数报包。这些报文交换令到所有路由器建立起临接关系，以具备网络的持久视图。随后对当前网络的变更，就增量更新发送出去。这些状态分别是：`Down`、`Attempt`、`Init`、`2-way`、`Exstart`、`Exchange`、`Loading` 以及 `Full states`，如下所示：

- `Down` 状态就是所有OSPF路由器的开始状态。然而，即便在所指定的路由器死亡间隔，那个接口尚未接收到 `Hello` 数据包，本地路由器仍可在此状态中显示出一个邻居（However, the local router may also show a neighbour in this state when no Hello packets have been received within the specified router dead interval for that interface）。
- `Attempt` 状态仅对那些非广播多路访问网络上的OSPF邻居有效。在该状态中，已发出了一个 `Hello` 数据包，但尚未在死亡间隔中接收到来自静态配置的邻居的信息；但会尽力与该邻居建立临接关系。
- 在OSPF路由器接收到来自邻居的 `Hello` 数据包，而本地路由器ID并未在接收到的邻居字段（the received Neighbor field）中列出时，就到了 `Init` 状态。如OSPF `Hello` 数据包的参数不匹配，比如各种计时器值等，那么OSPF路由器就再也不会进到此状态之后的状态了。
- `2-way` 状态表明OSPF邻居之间的双向通信（各台路由器已看到其它路由器的 `Hello` 数据包）。在该状态中，本地路由器已接收到一个在邻居字段中有着其自己的路由器ID的 `Hello` 数据包，同时两台路由器上的 `Hello` 数据包参数也是一致的。在此状态时，路由器就确定是否与这个邻居成为临接。在多路访问网络上，指定与后备指定路由器在此阶段得以选举出来。
- `Exstart` 状态用于数据库同步过程的初始化。本地路由器与其邻居在这个阶段确立何者负责数据库同步过程。在该状态中主从路由器被选举出来，同时在该阶段DBD交换的首个顺序编号有主路由器确定下来。
- `Exchange` 状态就是路由器使用DBD数据包对它们的数据库内容进行描述的地方。各个DBD序列被显式的确认，同时一次只允许一个突出的DBD。在此期间，LSR数据包已被发出，以请求LSA的一个新的实例（Each DBD sequence is explicitly acknowledged, and only one outstanding DBD is allowed at a time. During this phase, LSR packets are also sent to request a new instance of the LSA）。在此阶段，`M`（更多）位被用于请求缺失的信息（The `M` (More) bit is used to request missing information during this stage）。在两台路由器都完成了其完整数据库的交换后，它们将把该 `M` 位设置为 `0`。
- 在 `Loading` 状态，OSPF构造出一个LSR与链路状态重传清单。LSR数据包被发出，以请求某个LSA的较近期的、尚未在 `Exchange` 过程中接收到的实例。在此阶段，发出的更新被置于链路状态重传清单之上，直到本地路由器接收到确认为止。如本地路由器在阶段又接收到LSR，那么它将以包含了所请求信息的链路状态更新予以响应。
- `Full` 状态表明OSPF的邻居们已经完成了它们整个数据库的交换，且都达成一致（也就是它们有着网络的同样视图）。处于该状态的两台路由器就将该临接关系加入到它们的本地数据库，并就此关系在链路状态更新数据包中加以通告。到这里，路由表被计算出来，或在临接关系被重置后被重新计算出来。`Full` 正是一台OSPF路由器的正常状态。如果某台路由器被卡在了另一状态，那么就表明临接关系的形成中存在故障。对此的唯一例外就是 `2-way` 状态，该状态对于其中路由器仅到达指定或后备指定的 `Full` 状态的广播与非广播多路访问网络，就是所谓的正常状态。其它邻居总是将各自视为 `2-way` 的。

为了成功建立临接关系，两台路由器上的一些参数必须匹配。包括以下这些参数：

- 接口的MTU值（可被配置为忽略）
- `Hello` 与死亡计时器
- 区域ID

- 认证类型与口令
- 末梢区域标志 (The Stub Area flag)
- 兼容的网络类型 (Compatible network types)

本课程模块将陆续对这些参数进行介绍。如这些参数不匹配，那么OSPF的临接关系将绝不会完整建立。

**注意：** 处理不匹配的参数，还要记住在多路访问网络上，如两台路由器都配置了优先级值 0，那么临接关系也不会建立。在这类网络上，必须要有指定路由器 (The DR must be present on such network types)。

## OSPF的链路状态通告与链路状态数据库

### OSPF LSAs and the Link State Database(LSDB)

如同前面的小节中指出的，OSPF用到好几种类型的链路状态通告。每种链路状态通告都以标准的20字节链路状态通告头部开始。该标准LSA头部包括下面这些字段：

- 链路状态的老化时间 (Link State Age)
- 选项 (Options)
- 链路状态的类型
- 链路状态的ID
- 通告的路由器
- 链路状态的顺序编号
- 链路状态的校验和
- 长度

两字节的链路状态老化时间字段，指出自该LSA生成开始所历经的时间（以秒计）。LSA的最大老化时间是3600秒（1小时），这就意味着LSA的老化时间达到3600秒时，其就被移除数据库。为避免被移除，每隔1800秒对LSA进行更新。

一字节的选项字段包含了与OSPF Hello 数据包同样的选项。

一字节的链路状态类型字段，表示LSA的类型。LSA数据包不同的类型，在后面的小节中介绍。

四字节的链路状态ID字段，标识出由该LSA所描述的网络的一部分。该字段的内容，取决于通告的链路状态类型。

四字节的通告路由器字段，表示了产生该LSA的路由器的路由器ID。

四字节的链路状态顺序编号字段，对旧的或重复的链路状态通告进行探测。第一个顺序编号 0x80000000 是保留的；因此实际的第一个顺序编号总是 0x80000001。该值随着数据包的不断发出而增加。最大的顺序编号为 0x7FFFFFFF。

**注意：** 这里使用了补码表示有正负的整数，因此 0x80000000 就是整数 0， 0x80000001 就是整数 1。

两字节的链路状态校验和字段，对LSA的包括LSA头部的全部内容，执行弗莱彻校验和运算 (the Fletcher checksum, 参见[wikipedia:Fletcher's checksum](#))。链路状态老化时间字段未包含在校验和中。进行校验和计算的原因，是因为在LSA存储于内存中期间，可能由于路由器软件或硬件问题，或在LSA泛洪期间，由于物理层错误等原因，而造成LSA的失准。

**注意：** 在LSA被生成或接收到时，就会进行校验和的计算。此外，每个 CheckAge 间隔，也就是10分钟，也会进行校验和计算。如该字段的值为 0，那就是说没有进行校验和计算。

两字节的长度字段，是头部最后的字段，包含了该LSA的长度值（以字节计）。长度值包含了20字节的LSA头部。下图39.13对LSA头部进行了演示：

```

Open Shortest Path First
  OSPF Header
    LSA Header
      LS Age: 3600 seconds
      Do Not Age: False
    Options: 0x22 (DC, E)
    Link-State Advertisement Type: Router-LSA (1)
    Link State ID: 20.2.2.2
    Advertising Router: 20.2.2.2 (20.2.2.2)
    LS Sequence Number: 0x80000005
    LS Checksum: 0xcb54
    Length: 36
  LSA Header
  LSA Header

```

图 39.13 - 链路状态通告的头部

尽管OSPF支持11中不同类型的链路状态通告，但仅有LSA类型 1、2 与 3 用于计算内部路由，而LSA类型 4、5 及 7，则是用于计算外部路由，从而超出了CCNA考试要求范围。因为出于CCNA考试目的没有必要深入其它类型LSA的细节，所以这些LSA不会在本手册中进行介绍。但可在[in60days.com](http://in60days.com)上找到有关它们的一个简要提纲与可打印手册。

在思科IOS软件中，要查看链路状态数据库的内容，就使用 `show ip ospf database` 命令。在不带关键字使用此命令时，将打印出路由器连接的所有区域的LSA汇总。该命令支持几个有着更高的粒度的关键字，从而允许管理员将输出限制到仅特定类型LSA、仅由本地路由器通告的LSA，甚至OSPF中其它路由器通告的LSA。

尽管对每个关键字用法的输出进行演示是不现实的，但下面的小节仍对不同类型的LSA，以及与 `show ip ospf database` 命令结合使用从而查看到这些LSA的详细信息的一些常见关键字，进行了介绍。该命令所支持的关键字，在下面的输出中进行了演示：

```

R3#show ip ospf database ?
adv-router          Advertising Router link states
asbr-summary        ASBR Summary link states
database-summary    Summary of database
external            External link states
network             Network link states
nssa-external       NSSA External link states
opaque-area         Opaque Area link states
opaque-as           Opaque AS link states
opaque-link         Opaque Link-Local link states
router              Router link states
self-originated     Self-originated link states
summary             Network Summary link states
|                  Output modifiers
<cr>

```

## 路由器链路状态通告（类型1）

### Router Links State Advertisements(Type 1)

类型1的LSA，是由各台路由器为其所属的各个区域所生成的。路由器LSA列出了始发路由器的路由器ID（The router LSA lists the originating router's router ID）。每台单个的路由器都将为其所处的区域，生成一条类型1的LSA。路由器LSA是 `show ip ospf database` 命令输出中最先打印出的LSA类型。

## 网络链路状态通告（类型2）

### Network Link State Advertisements(Type 2)

OSPF使用网络链路状态通告（类型2的LSA），来在多路访问网段上对路由器进行通告（OSPF uses the Network Link State Advertisement(Type 2 LSA) to advertise the routers on the Multi-Access segment）。此类LSA是由指定路由器生成的，且仅在区域中传播（flooded）。因为其它非指定/后备指定路由器并不在相互之间建立邻接关系，所以网络LSA就令到这些路由器对该多路访问网络上的其它路由器有所知悉。

## 网络汇总链路状态通告（类型3）

### Network Summary Link State Advertisement(Type 3)

网络汇总LSA是一条本地区域之外，但仍出于OSPF域中的目的（网络）的汇总。也就是说，此类LSA同时对区域间及区域内的路由信息进行通告（The Network Summary(Type 3) LSA is a summary of destinations outside of the local area but within the OSPF domain. In other words, this LSA advertises both inter-area and intra-area routing information）。网络汇总LSA没有携带任何的拓扑信息。而是在该类型的LSA中唯一包含的信息，就是一个IP前缀（an IP prefix）。类型3的LSA是由区域边界路由器生成的，并被泛洪到所有邻接区域（adjacent areas）。默认情况下，每条类型3的LSA都与一条单独的路由器或网络LSA，一一对应的形式相匹配（By default, each Type 3 LSA matches a single Router or Network LSA on a one-for-one basis）。也就是说，对于每条单独的类型1及类型2的LSA，都存在着一条类型3的LSA。特别要留意这些LSA是如何在与OSPF骨干（区域）的联系下被传播的。此种传播或泛洪，按照下面这样进行（Special attention must be paid to how these LSAs are propagated in relation to the OSPF backbone. This propagation or flooding is performed as follows）：

- 对于区域内的路由（也就是对于类型1及类型2的LSAs），网络汇总（类型3）的LSA自非骨干区域被通告至OSPF骨干（区域，Network Summary(Type 3) LSAs are advertised from a non-backbone area to the OSPF backbone for intra-area routes(i.e., for Type 1 and Type 2 LSAs)）
- 对于区域内（也就是区域0的类型1与类型2 LSAs）及区域间路由（也就是由其它区域边界路由器泛洪到骨干区域的类型3 LSAs）的网络汇总（类型3）LSAs，被同时从OSPF骨干区域，通告到其它非骨干区域。

后面的三种链路状态通告，类型4、类型5与类型7，用于外部路由器计算。类型4与类型5将在接着的小节介绍，类型7将在本课程模块后面，于对不同的OSPF区域进行讨论时介绍。

## 自治系统边界汇总链路状态通告（类型4）

### ASBR Summary Link State Advertisements(Type 4)

类型4的LSA对有关自治系统边界路由器的信息进行描述（The Type 4 LSA describes information regarding the Autonomous System Boundary Router(ASBR)）。此类LSA包含了与类型3 LSA的相同数据包格式，并以一些显著的差异，完成同样的基本功能。与类型3的LSA类似，类型4的LSA是由区域边界路由器生成的。两种LSAs的通告路由器字段（the Advertising Router field）都包含着生成该汇总LSA的区域边界路由器的路由器ID。但是，类型4的LSA使用区域边界路由器，为仅有某条路由器LSA可达的各台自治系统边界路由器所创建的。随后该区域边界路由器将该类型4的LSA注入到相应区域。此类LSA提供到有关该自治系统边界路由器本身的可靠性信息。你应熟知的类型3与类型4 LSAs的关键不同，在下表39.2中有列出：

表 39.2 - 类型3与类型4汇总LSAs

类型3的汇总LSA	类型4的汇总LSA
提供有关网络链路的信息。	提供有关自治系统边界路由器的信息。
网络掩码字段 (The Network Mask field) 包含了该网络的子网掩码。	网络掩码字段将总是包含值 0.0.0.0 或简单的就是 0。
链路状态ID字段 (The Link State ID field) 包含了真实的网络编号。	链路状态ID字段包含了自治系统边界路由器的路由器ID。

## 自治系统外部链路状态通告 (类型5)

### AS External Link State Advertisements(Type 5)

外部链路状态通过用于对那些该自治系统的外部目的网络进行描述 (The External Link State Advertisement is used to describe destinations that are external to the autonomous system)。也就是说，类型5的LSAs提供了要抵达外部网络的必要信息。除了外部路由外，某个OSPF路由域 (an OSPF routing domain) 的默认路由，也可作为类型5的链路状态通告，而加以注入。

## OSPF的各种区域 (OSPF Areas)

除了在本课程模块之前的小节中描述并用到的骨干区域 (Area 0) 及其它非骨干区域外，OSPF规格还定义了记住“特殊”类型的区域。这些区域的配置，主要是为了通过阻止不同类型的LSAs (主要是类型5的LSAs) 诸如到确切区域，而减小出于这些区域中的路由器上的链路状态数据库的大小，这些其它区域包括：

- 次末梢区域 (Not-So-Stubby Areas, NSSAs)
- 完全的次末梢区域 (Totally Not-So-Stubby Areas, TNSSAs)
- 末梢区域 (Stub Areas, SAs)
- 完全末梢区域 (Totally Stubby Areas, TSAs)

### 次末梢区域 (Not-So-Stubby Areas, NSSAs)

次末梢区域是OSPF末梢区域的一种，其允许自治系统边界路由器使用NSSA外部LSA (类型7)，注入外部路由信息。如同在前面的小节中所指出的，类型4、类型5与类型7的LSAs是用于外部路由的计算。这里不会就类型7的LSAs的细节，或它们在NSSAs中的使用方式，进行检视。

### 完全次末梢区域 (Totally Not-So-Stubby Areas, TNSSAs)

完全次末梢区域是次末梢区域的一个扩展。与次末梢区域类似，类型5的LSAs不被允许进入TNSSAs；与NSSAs不同的是，汇总LSAs也不允许进入到TNSSAs中。此外，在配置了某个TNSSA时，默认路由就作为类型7的LSA注入到该区域。TNSSAs有着以下特性：

- 类型7的LSAs在该NSSA的区域边界路由器处被转换为类型5的LSAs
- 它们不允许网络汇总LSAs (They do not allow Network Summary LSAs)
- 它们不允许外部LSAs
- 默认路由是以一条汇总LSA被注入的

### 末梢区域 (Stub Areas)

末梢区域与NSSAs有些类似，主要的例外就是不允许外部路由（类型5或类型7）进入到末梢区域（Stub areas are somewhat similar to NSSAs, with the major exception being that external routes(Type 5 or Type 7) are not allowed into Stub Areas）。重要的是对末梢在OSPF何EIGRP中的功能是完全不同的。在OSPF中，某个区域作为末梢区域的配置，通过阻止外部LSAs被通告到这些区域，在无需额外配置下，就可减小这些区域中路由器的路由表及OSPF数据库的大小。末梢区域有着以下特性：

- 默认路由是通过区域边界路由器，以一条类型3的LSA注入到末梢区域的
- 来自其它区域的类型3的LSAs允许进入到这些区域
- 外部路由的LSAs（也就是类型4及类型5的LSAs）不被允许

## 完全末梢区域（Totally Stubby Areas）

完全末梢区域是末梢区域的一个扩展。但与末梢区域不同的是，完全末梢区域通过限制外部LSAs外，还限制了类型3的LSAs，从而进一步地减小了完全末梢区域中路由器上的链路状态数据库（Link State Database, LSDB）的大小。通常将TSA配置在那些有着到网络，比如在传统的分支网络，的单个入口及出口点的路由器上（TSA are typically configured on routers that have a single ingress and egress point into the network, for example in a traditional hub-and-spoke network）。该区域的路由器将所有外部流量转发到区域边界路由器。同时该区域边界路由器也是所有骨干区域及区域间流量到完全末梢区域的出口点（The ABR is also the exit point for all backbone and inter-area traffic to the TSA），其有着以下特性：

- 默认路由是作为类型3的网络汇总LSA注入到末梢区域的
- 自其它区域的类型3、类型4及类型5 LSAs不被允许进入到这些区域

## 路由度量值与最优路由选取

### Route Metrics and Best Route Selection

在以下小节中，将学到有关OSPF度量值及其运算的知识。

## OSPF度量值的计算（Calculating the OSPF Metric）

OSPF度量值通常被成为开销（The OSPF metric is commonly referred to as the cost）。开销是从链路的带宽，使用公式  $10^8 / \text{带宽} (\text{bps})$  （其中“带宽”以 bps 计）得到的。这就意味着依据不同链路的带宽，而赋予了它们不同的开销值。使用此公式，一个 10Mbps 的以太网接口的OSPF开销，将像下面这样计算出来：

- 开销 =  $10^8 / \text{带宽} (\text{bps})$
- 开销 =  $100\ 000\ 000 / 10\ 000\ 000$
- 开销 = 10

使用同样的公式，一条 T1 链路的OSPF开销，将像下面这样计算出来：

- 开销 =  $10^8 / \text{带宽} (\text{bps})$
- 开销 =  $100\ 000\ 000 / 1\ 544\ 000$
- 开销 = 64.77

**注意：** 在计算OSPF的度量值时，不会用到小数。因此这样的小数总是会向下取整到最接近的整数。

那么对于上一示例，一条 T1 链路的实际开销将向下取整到 64。

如先前所演示的那样，可使用 `show ip ospf interface [name]` 来查看到某个接口的OSPF开销。在度量值计算中用到的默认参考带宽，可在 `show ip protocols` 命令的输出中查看到，如下面的输出中所演示的那样：

```
R4#show ip protocols
Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 4.4.4.4
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    0.0.0.0 255.255.255.255 Area 2
  Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway          Distance      Last Update
    3.3.3.3           110          00:00:03
  Distance: (default is 110)
```

而在OSPF中使用的默认参考带宽，则可使用路由器配置命令 `auto-cost reference-bandwidth <1-4294967>`，并指定出以 Mbps 计的参考带宽值，而进行调整。这样做在那些有着具有超过 100Mbps 带宽值的链路，比如 GigabitEthernet 链路的网络中尤为重要。在这些网络中，赋予给 GigabitEthernet 的默认开销值将与 FastEthernet 链路的开销值一样。大多数情况下，这样的结果当然是不可取的，尤其是在OSPF尝试在这些链路上进行负载均衡时。

要阻止这种开销值计算偏差，就应在路由器上执行该路由器配置命令 `auto-cost reference-bandwidth 1000` 命令。这会引发使用新的参考带宽值，对路由器上的个开销值的重新计算。比如，依据该配置，某条 T1 链路的开具将如下进行重新计算：

- 开销 =  $10^9 / \text{带宽 (bps)}$
- 开销 =  $1\ 000\ 000\ 000 / 1\ 544\ 000$
- 开销 = 647.7

**注意：**再次，因为OSPF度量值不支持小数，该值将被向下取整到简单的 647 的度量值，如下面的输出所示：

```
R4#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 10.0.2.4/24, Area 2
  Process ID 4, Router ID 4.4.4.4, Network Type POINT_TO_POINT, Cost: 647
  Transmit Delay is 1 sec, State POINT_TO_POINT
  Timer intervals configured, Hello 10, Dead 60, Wait 60, Retransmit 5
    oob-resync timeout 60
    Hello due in 00:00:01
  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 0, Adjacent neighbor count is 0
  Suppress Hello for 0 neighbor(s)
```

在执行了路由器配置命令 `auto-cost reference-bandwidth 1000` 后，思科IOS软件就打印出下面的消息，表明应将此同样的值，应用该OSPF域中的所有路由器上。这在下面的输出中进行了演示：

```
R4(config)#router ospf 4
R4(config-router)#auto-cost reference-bandwidth 1000
% OSPF: Reference bandwidth is changed.
  Please ensure reference bandwidth is consistent across all routers.
```

尽管这一点可能看起来像一条重要的警告消息，但请记住该命令的使用，仅影响到本地路由器。在所有路由器上配置这条命令并不是强制性的；但为考试目的，应确保在所有路由器上应用了一个一致的配置。

## 对OSPF的度量值计算施加影响 (Influencing OSPF Metric Calculation)

可通过执行下面的操作，来对OSPF度量值的计算，施加直接的影响：

- 使用 `bandwidth` 命令，对接口带宽进行调整
- 使用 `ip ospf cost` 命令，手动指定开销

在对EIGRP的度量值计算进行讨论时的先前课程模块中，对 `bandwidth` 命令的使用进行了介绍。如先前指出的那样，默认OSPF的开销，是通过以参考带宽  $10^{18}$ ，也就是  $100\text{Mbps}$  除以链路带宽计算出来的。那么不论是提升还是降低链路带宽，都直接影响到该特定链路的OSPF开销。这是一种典型的用于确保某条路径优先于另一路径而被选用的 **路径控制机制** (a path control mechanism)。

但是，如同在先前的课程模块中所描述的那样，`bandwidth` 命令的影响，不仅限于路由协议。正是由于这个原因，作为第二种办法的手动指定开销值，就是推荐的对OSPF度量值计算施加影响的做法。

接口配置命令 `ip ospf cost <1-65535>`，被用于手动指定某条链路的开销。链路的开销值越低，其就比到相同目的网络的、有着更高开销值的其它链路，越有可能被优先选用。下面的示例演示了如何为某条串行

(T1) 链路配置上一个OSPF开销 5：

```
R1(config)#interface Serial0/0
R1(config-if)#ip ospf cost 5
R1(config-if)#exit
```

可使用 `show ip ospf interface [name]` 命令对此配置进行验证，如下面的输出所示：

```
R1#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 10.0.0.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type POINT_TO_POINT, Cost: 5
  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 00:00:04
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 1, maximum is 4
  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)
```

## OSPF的默认路由 (OSPF Default Routing)

与EIGRP支持好几种生成与通告默认路由的方式不同，OSPF仅使用路由器配置命令 `default-information originate [always] [metric <value>] [metric-type <1|2>] [route-map <name>]`，来动态地通告默认路由。

其所使用的 `default-information originate` 命令，将把该路由器配置为仅在路由表中已出现一条默认路由的情况下，通告一条默认路由 (The `default-information originate` command used by itself will configure the router to advertise a default route only if a default route is already present in the routing table)。但可

将 [always] 关键字追加到该命令，从而强制该路由器在路由表中尚不存在默认路由的情况下，生成一条默认路由。应小心使用这个关键字，因为它可能导致OSPF域中的流量黑洞，或者导致将所有位置目的地的数据包，转发到所配置的路由器。

关键字 [metric <value>] 用于指定所生成的默认路由的路由度量值。而关键字 [metric-type <1|2>] 可用于修改默认路由的度量值类型 (the metric type for the default route)。最后，[route-map <name>] 关键字将路由器配置为仅在该命名的路由器地图中所指定的条件满足时，生成一条默认路由。

下面的配置示例，演示了如何将一台开启OSPF的路由器，配置为在路由表中存在一条默认路由时，生成一条默认路由并对其进行通告。既有的默认路由可以是一条静态路由，甚至为在该路由器上配置了多种路由协议时，从另一种路由协议产生的一条默认路由。下面的输出演示的是基于一条配置的静态默认路由的此种配置：

```
R4(config)#ip route 0.0.0.0 0.0.0.0 FastEthernet0/0 172.16.4.254
R4(config)#router ospf 4
R4(config-router)#network 172.16.4.0 0.0.0.255 Area 2
R4(config-router)#default-information originate
R4(config-router)#exit
```

默认情况下，默认路由是作为类型5的LSA进行通告的。

## OSPF的配置 (Configuring OSPF)

以一行配置，就可以在路由器上开启基本的OSPF，并于随后通过添加 网络语句，来指明希望在哪些接口上运行OSPF，对于那些不打算通告的网络，则不予添加 (Basic OSPF can be enabled on the router with one line of configuration, and then by adding the network statement that specifies on which interfaces you want to run OSPF, not necessarily networks you wish to advertise) :

1. router ospf 9，其中 9 是本地有意义的编号
2. network 10.0.0.0 0.255.255.255 area 0

在至少一个接口处于 up/up 状态之前，OSPF都不会成为活动状态，并请记住要至少有一个区域必须为 Area 0。下图39.14演示了一个示例性的OSPF网络：



图 39.14 - 一个示例性OSPF网络

其中路由器A的配置为：

```
router ospf 20
network 4.4.4.4 0.0.0.0 area 0
network 192.168.1.0 0.0.0.255 area 0
router-id 4.4.4.4
```

路由器B的配置为：

```
router ospf 22
network 172.16.1.0 0.0.0.255 area 0
network 192.168.1.0 0.0.0.255 area 0
router-id 192.168.1.2
```

路由器C的配置为：

```
router ospf 44
network 1.1.1.1 0.0.0.0 area 1
network 172.16.1.0 0.0.0.255 area 0
router-id 1.1.1.1
router-id 1.1.1.1
RouterC#show ip route
Gateway of last resort is not set
    1.0.0.0/32 is subnetted, 1 subnets
C        1.1.1.1 is directly connected, Loopback0
        4.0.0.0/32 is subnetted, 1 subnets
O        4.4.4.4 [110/129] via 172.16.1.1, 00:10:39, Serial0/0/0
        172.16.0.0/24 is subnetted, 1 subnets
C        172.16.1.0 is directly connected, Serial0/0/0
O        192.168.1.0/24 [110/128] via 172.16.1.1, 00:10:39, Serial0/0/0
```

## OSPF的故障排除 (Troubleshooting OSPF)

这里再度说明一下，开放路径优先协议，是一种就其链路状态进行通告的，开放标准的链路状态路由协议。在一台链路状态路由器于某条网络链路上开始运作时，那个逻辑网络的相关信息，就被添加到该路由器的本地的链路状态数据库中。随后该本地路由器就在其可用的那些链路上，发送 Hello 报文，来判断是否其它链路状态路由器也在接口上运行。OSPF使用IP编号 89，直接允许在互联网协议上。

尽管要深入到所有潜在的OSPF故障场景是不可能的，不过接下来的小节，仍就在将OSPF部署为IGP的选择 (the IGP of choice, [Interior Gateway Protocol](#)) 时，一些最为常见的故障场景进行了讨论。

### 邻居关系的故障排除

#### Troubleshooting Neighbour Relationships

运行OSPF的路由器在建立邻接关系前，会度过好几种状态。这些不同状态分别是 Down、Attempt、Init、2-way、Exstart、Exchange、Loading 以及 Full 状态。OSPF邻接关系的首选状态是 Full 状态。该状态表明邻居已经完成了各自完整数据库的交换，并有着对网络的相同视图。但尽管 Full 状态是首选的邻接状态，在邻接关系建立的过程中，邻居们可能会“卡在”其它的某种状态中。由于这个原因，那么为了排除故障，就有必要掌握需要查找什么。

### 邻居表为空的情况 (The Neighbour Table Is Empty)

对于邻居表可能为空的原因（也就是为何 `show ip ospf neighbor` 命令可能不产生任何输出），有好几种。常见的原因如下所示：

- 基础的OSPF错误配置 (misconfigurations)
- 1层与2层故障
- 访问控制清单过滤掉了 (ACL filtering)
- 接口的错误配置

基本的OSPF错误配置，涵盖了很多东西。其可以包括比如不匹配的计时器、区域IDs、认证参数及末梢配置等。思科IOS中有大量的工具，可用于对基本的OSPF错误配置进行故障排除。比如，可使用 `show ip protocols` 命令来判断信息（比如有关那些开启了OSPF的网络）；可使用 `show ip ospf` 命令，来判断区域配置及各区域的接口；以及使用 `show ip ospf interface brief` 命令来判断哪些接口位处哪些区域中，以及在假定接口已开启了OSPF时，判断出这些接口已对哪些OSPF进程开启了。

另一个常见的错误配置就是将接口指定为了被动接口（Another common misconfiguration is specifying the interface as passive）。如果真这样做了，那么该接口就不会发出 Hello 数据包，同时使用那个接口就不会建立邻居关系。既可使用 `show ip protocols`，也可使用 `show ip ospf interface` 命令，来检查哪些接口被配置或指定为了被动接口。下面是在某个被动接口上的后一个命令的示例输出：

```
R1#show ip ospf interface Serial0/0
Serial0/0 is up, line protocol is up
  Internet Address 172.16.0.1/30, Area 0
  Process ID 1, Router ID 10.1.0.1, 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
    No Hellos (Passive interface)
  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)
```

最后，当在帧中继这样的非广播多路访问技术上开启OSPF时，请记住必须静态地定义出邻居，因为对于默认的非广播网络类型的邻居发现，OSPF不使用多播传输。在部署OSPF，这是一种常见的邻居表为空的原因。

1层与2层故障，也能导致OSPF邻接关系的不形成。在先前的课程模块中，就曾详细介绍了1层与2层的故障排除。使用诸如 `show interfaces` 这样的命令来对接口状态（即线路协议），以及接口上接收到的任何错误进行检查。在开启OSPF的路由器处于跨越多台交换机的VLAN中时，比如应就该VLAN中有着端到端的连通性（end-to-end connectivity），以及所有端口或接口都处于正确的生成树状态进行检查。

访问控制清单过滤，是另一种常见的造成邻接关系建立失败的原因。为排除此类故障，重要的是熟悉网络拓扑。比如，在建立某个邻接关系失败的路由器是通过不同物理交换机进行连接的时，就可能为ACL过滤是以先前为安全目的，而已配置在交换机上的VACL（VLAN ACL）的形式部署的。`show ip ospf traffic` 命令，就是一个可找出OSPF数据包是被阻塞了还是被丢弃了的有用工具，其会打印出如下输出所演示的，有关发出的OSPF数据包的信息：

```
R1#show ip ospf traffic Serial0/0
  Interface Serial0/0
  OSPF packets received/sent
    Invalid   Hellos    DB-des    LS-req    LS-upd    LS-ack    Total
  Rx: 0       0         0         0         0         0         0
  Tx: 0       6         0         0         0         0         6
  OSPF header errors
    Length 0, Auth Type 0, Checksum 0, Version 0,
    Bad Source 0, No Virtual Link 0, Area Mismatch 0,
    No Sham Link 0, Self Originated 0, Duplicate ID 0,
    Hello 0, MTU Mismatch 0, Nbr Ignored 0,
    LLS 0, Unknown Neighbor 0, Authentication 0,
    TTL Check Fail 0,
  OSPF LSA errors
    Type 0, Length 0, Data 0, Checksum 0,
```

在上面的输出中，留意到本地路由器在发送OSPF Hello 数据包但没有接收到任何东西。在路由器上的配置正确的情况下，就要对路由器或中间设备进行检查，以确保OSPF数据包未被过滤或丢弃。

空白邻居表的另一个常见原因，就是接口的不当配置。与EIGRP类似，OSPF不会使用从接口地址建立邻居关系。但与EIGRP不同，在接口子网掩码不一致时，OSPF也不会建立邻居关系。

就是接口子网掩码不同，开启了EIGRP的路由器也会建立邻居关系。比如有这样的两台路由器，其一有着使用地址 10.1.1.1/24 的一个接口，而另一台有着一个使用地址 10.1.1.2/30 的接口，它们被配置为背靠背的EIGRP实现（back-to-back EIGRP implementation），那么它们将成功地建立邻居关系。但应注意此类实现可能导致路由器之间的路由环回。处理不匹配的子网掩码，开启EIGRP的路由器也忽略最大传输单元（MTU）配置，而甚至在接口最大传输单元不同的情况下，建立邻居关系。使用 `show ip interfaces` 与 `show interfaces` 命令，就可对IP地址与掩码配置进行检查。

## 路由通告的故障排除（Troubleshooting Route Advertisement）

就像EIGRP的情况一样，有的时候可能会注意到OSPF没有对某些路由进行通告。大多数情况下，这都是由于一些错误配置，而非协议故障造成的（For the most part, this is typically due to some misconfigurations versus a protocol failure）。此类故障的一些常见原因包括下面这些：

- 接口上没有开启OSPF
- 接口宕掉了
- 接口地址出于不同的区域
- OSPF的错误配置

OSPF之所以不对路由器进行通告的一个常见原因，就是该网络未通过OSPF进行通告。在当前的思科IOS软件中，使用路由器配置命令 `network` 或接口配置命令 `ip ospf`，就可使网络得以通告。不管使用哪种方式，都可以使用 `show ip protocols` 命令，来查看将OSPF配置为对哪些网络进行通告，就如同下面的输出中所看到的：

```
R2#show ip protocols
Routing Protocol is "ospf 1"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 2.2.2.2
  Number of areas in this router is 1. 1 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.2.2.0 0.0.0.128 Area 1
    20.2.2.0 0.0.0.255 Area 1
  Routing on Interfaces Configured Explicitly (Area 1):
    Loopback0
    Reference bandwidth unit is 100 mbps
  Routing Information Sources:
    Gateway          Distance      Last Update
    1.1.1.1          110          00:00:17
  Distance: (default is 110)
```

此外，请记住还可以使用 `show ip ospf interfaces` 命令来找出那些接口开启了OSPF，及其它一些信息。除了网络配置，若接口宕掉，OSPF也不会对路由器进行通告。可使用 `show ip ospf interfaces` 命令，来确定接口状态，如下所示：

```
R1#show ip ospf interface brief
Interface    PID   Area     IP Address/Mask      Cost     State    Nbrs F/C
Lo100        1     0         100.1.1.1/24        1        DOWN    0/0
Fa0/0        1     0         10.0.0.1/24         1        BDR     1/1
```

参考上面的输出，可看到 Loopback100 出于 DOWN 状态。细看就可以发现该故障是由于该接口已被管理性关闭，如下面的输出所示：

```
R1#show ip ospf interface Loopback100
Loopback100 is administratively down, line protocol is down
  Internet Address 100.1.1.1/24, Area 0
  Process ID 1, Router ID 1.1.1.1, Network Type LOOPBACK, Cost: 1
  Enabled by interface config, including secondary ip addresses
  Loopback interface is treated as a stub Host
```

如使用 debug ip routing 命令对IP路由事件（IP routing events）进行调试，并于随后在 Loopback100 接口下执行 no shutdown 命令，那么就可以看到下面的输出：

```
R1#debug ip routing
IP routing debugging is on
R1#conf t
Enter configuration commands, one per line.
R1(config)#interface Loopback100
R1(config-if)#no shutdown
R1(config-if)#end
R1#
*Mar 18 20:03:34.687: RT: is_up: Loopback100 1 state: 4 sub state: 1 line: 0 has_route: False
*Mar 18 20:03:34.687: RT: SET_LAST_RDB for 100.1.1.0/24
  NEW rdb: is directly connected
*Mar 18 20:03:34.687: RT: add 100.1.1.0/24 via 0.0.0.0, connected metric [0/0]
*Mar 18 20:03:34.687: RT: NET-RED 100.1.1.0/24
*Mar 18 20:03:34.687: RT: interface Loopback100 added to routing table
...
[Truncated Output]
```

当有多个地址配置在某个接口下时，所有次要地址都必须位处与主要地址相同的区域中；否则OSPF不会对这些网络进行通告。比如，考虑下图39.15中所演示的网络拓扑：



图 39.15 - OSPF 的次要子网通告

参考图39.15，路由器 R1 与 R2 通过一条背靠背的连接（a back-to-back connection）相连。这两台路由器共享了 10.0.0.0/24 子网。不过 R1 还配置了一些在其 FastEthernet0/0 接口下的额外（次要）子网，因此 R1 上该接口的配置就如下打印出来：

```
R1#show running-config interface FastEthernet0/0
Building configuration...
Current configuration : 183 bytes
!
interface FastEthernet0/0
ip address 10.0.1.1 255.255.255.0 secondary
ip address 10.0.2.1 255.255.255.0 secondary
ip address 10.0.0.1 255.255.255.0
duplex auto
speed auto
end
```

在 R1 与 R2 上都开启了OSPF。 R1 上部署的配置如下所示：

```
R1#show running-config | section ospf
router ospf 1
router-id 1.1.1.1
log adjacency-changes
network 10.0.0.1 0.0.0.0 Area 0
network 10.0.1.1 0.0.0.0 Area 1
network 10.0.2.1 0.0.0.0 Area 1
```

R2 上部署的配置如下所示：

```
R2#show running-config | section ospf
router ospf 2
router-id 2.2.2.2
log adjacency-changes
network 10.0.0.2 0.0.0.0 Area 0
```

默认情况下，因为 R1 上的次要子网已被放入到一个不同的OSPF区域，所以它们不会被该路由器通告。这一点在 R2 上可以看到，在执行了 show ip route 命令时，就显示下面的输出：

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
      10.0.0.0/24 is subnetted, 1 subnets
C          10.0.0.0 is directly connected, FastEthernet0/0
```

为解决这个问题，就必须将那些次要子网，指派到 Area 0，如下所示：

```
R1(config)#router ospf 1
R1(config-router)#network 10.0.1.1 0.0.0.0 Area 0
*Mar 18 20:20:37.491: %OSPF-6-AREACHG: 10.0.1.1/32 changed from Area 1 to Area 0
R1(config-router)#network 10.0.2.1 0.0.0.0 Area 0
*Mar 18 20:20:42.211: %OSPF-6-AREACHG: 10.0.2.1/32 changed from Area 1 to Area 0
R1(config-router)#end
```

在此配置改变之后，那些网络就被通告给路由器 R2 了，如下所示：

```
R2#show ip route
Codes: C - connected, S - static, R - RIP, M - mobile, B - BGP
      D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
      N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
      E1 - OSPF external type 1, E2 - OSPF external type 2
      i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
      ia - IS-IS inter area, * - candidate default, U - per-user static route
      o - ODR, P - periodic downloaded static route
Gateway of last resort is not set
      10.0.0.0/24 is subnetted, 3 subnets
O          10.0.2.0 [110/2] via 10.0.0.1, 00:01:08, FastEthernet0/0
C          10.0.0.0 is directly connected, FastEthernet0/0
O          10.0.1.0 [110/2] via 10.0.0.1, 00:01:08, FastEthernet0/0
```

除了上书三种常见原因，不良的设计、实现，以及错误配置，也是导致OSPF不如预期的那样对网络进行通告的一个原因。导致此类故障常见的设计问题，包括一个不连续或分区的骨干区域（a discontiguous or partitioned backbone）以及区域类型的错误配置，比如将区域配置为完全末梢的区域。对于这种原因，就要对OSPF的工作原理及其在自己的环境中如何部署有扎实掌握。这样的掌握将极大地简化故障排除过程，因为在故障排除之前，就已经赢得了战斗的一半了。

## OSPF路由故障的调试 (Debugging OSPF Routing Issues)

在本课程模块的最后一节，将看看一些较为常用的OSPF调试命令。OSPF的调试，是通过使用 `debug ip ospf` 命令来开启的。该命令可结合下面这些额外关键字一起使用：

```
R1#debug ip ospf ?
adj          OSPF adjacency events
database-timer OSPF database timer
events        OSPF events
flood         OSPF flooding
hello         OSPF hello events
lsa-generation OSPF lsa generation
mpls          OSPF MPLS
nsf           OSPF non-stop forwarding events
packet        OSPF packets
retransmission OSPF retransmission events
spf           OSPF spf
tree          OSPF database tree
```

命令 `debug ip ospf adj` 将打印有关邻接事件的实时信息。在对OSPF的邻居邻接故障进行故障排除时，这是一个有用的故障排除工具。下面是一个由该命令打印的信息示例。下面的示例演示了如何使用该命令，来判断MTU不匹配而导致的无法到达 `Full` 状态，从而阻止了邻居邻接的建立：

```
R1#debug ip ospf adj
OSPF adjacency events debugging is on
R1#
*Mar 18 23:13:21.279: OSPF: DR/BDR election on FastEthernet0/0
*Mar 18 23:13:21.279: OSPF: Elect BDR 2.2.2.2
*Mar 18 23:13:21.279: OSPF: Elect DR 1.1.1.1
*Mar 18 23:13:21.279:      DR: 1.1.1.1 (Id)    BDR: 2.2.2.2 (Id)
*Mar 18 23:13:21.283: OSPF: Neighbor change Event on interface FastEthernet0/0
*Mar 18 23:13:21.283: OSPF: DR/BDR election on FastEthernet0/0
*Mar 18 23:13:21.283: OSPF: Elect BDR 2.2.2.2
*Mar 18 23:13:21.283: OSPF: Elect DR 1.1.1.1
*Mar 18 23:13:21.283:      DR: 1.1.1.1 (Id)    BDR: 2.2.2.2 (Id)
*Mar 18 23:13:21.283: OSPF: Rcv DBD from 2.2.2.2 on FastEthernet0/0 seq 0xA65 opt 0x52 flag 0x7 len 32
*Mar 18 23:13:21.283: OSPF: Nbr 2.2.2.2 has smaller interface MTU
*Mar 18 23:13:21.283: OSPF: NBR Negotiation Done. We are the SLAVE
*Mar 18 23:13:21.287: OSPF: Send DBD to 2.2.2.2 on FastEthernet0/0 seq 0xA65 opt 0x52 flag 0x2 len 192
*Mar 18 23:13:26.275: OSPF: Rcv DBD from 2.2.2.2 on FastEthernet0/0 seq 0xA65 opt 0x52 flag 0x7 len 32
*Mar 18 23:13:26.279: OSPF: Nbr 2.2.2.2 has smaller interface MTU
*Mar 18 23:13:26.279: OSPF: Send DBD to 2.2.2.2 on FastEthernet0/0 seq 0xA65 opt 0x52 flag 0x2 len 192
...
[Truncated Output]
```

从上面的输出，可以推断出本地路由器上的MTU高于 1480 字节，因为该调试输出显示邻居有着较低的MTU值。推荐的解决方案将是调整该较低的MTU值，以令到两个邻居有着同样的接口MTU值。这就可以允许该邻接达到 `Full` 状态。

命令 `debug ip ospf lsa-generation` 将打印出有关OSPF链路状态通告的信息。该命令可用于在使用OSPF时对路由通告的故障排除。下面是由该命令所打印的输出信息的一个示例：

```
R1#debug ip ospf lsa-generation
OSPF summary lsa generation debugging is on
R1#
R1#
*Mar 18 23:25:59.447: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0 from FULL to DOWN, Nei:
*Mar 18 23:25:59.511: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0 from LOADING to FULL,
*Mar 18 23:26:00.491: OSPF: Start redist-scanning
*Mar 18 23:26:00.491: OSPF: Scan the RIB for both redistribution and translation
*Mar 18 23:26:00.499: OSPF: max-aged external LSA for summary 150.0.0.0 255.255.0.0, scope: Translatio
*Mar 18 23:26:00.499: OSPF: End scanning, Elapsed time 8ms
*Mar 18 23:26:00.499: OSPF: Generate external LSA 192.168.4.0, mask 255.255.255.0, type5, age 0, metri
*Mar 18 23:26:00.503: OSPF: Generate external LSA 192.168.5.0, mask 255.255.255.0, type 5, age 0, metri
*Mar 18 23:26:00.503: OSPF: Generate external LSA 192.168.1.0, mask 255.255.255.0, type 5, age 0, metri
*Mar 18 23:26:00.503: OSPF: Generate external LSA 192.168.2.0, mask 255.255.255.0, type 5, age 0, metri
*Mar 18 23:26:00.507: OSPF: Generate external LSA 192.168.3.0, mask 255.255.255.0, type 5, age 0, metri
*Mar 18 23:26:05.507: OSPF: Generate external LSA 192.168.4.0, mask 255.255.255.0, type 5, age 0, metri
*Mar 18 23:26:05.535: OSPF: Generate external LSA 192.168.5.0, mask 255.255.255.0, type 5, age 0, metri
```

命令 `debug ip ospf spf` 提供有关最短路径优先算法事件的实时信息。该命令可以下面的关键字结合使用：

```
R1#debug ip ospf spf ?
  external  OSPF spf external-route
  inter    OSPF spf inter-route
  intra    OSPF spf intra-route
  statistic OSPF spf statistics
<cr>
```

与所有 `debug` 命令一样，在对 SPF 事件进行调试之前，都应对诸如网络大小及路由器上资源占用等因素加以考虑。下面是自 `debug ip ospf spf statistic` 命令的输出示例：

```
R1#debug ip ospf spf statistic
OSPF spf statistic debugging is on
R1#clear ip ospf process
Reset ALL OSPF processes? [no]: y
R1#
*Mar 18 23:37:27.795: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0 from FULL to DOWN, Nei:
*Mar 18 23:37:27.859: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on FastEthernet0/0 from LOADING to FULL,
*Mar 18 23:37:32.859: OSPF: Begin SPF at 28081.328ms, process time 608ms
*Mar 18 23:37:32.859:           spf_time 07:47:56.328, wait_interval 5000ms
*Mar 18 23:37:32.859: OSPF: End SPF at 28081.328ms, Total elapsed time 0ms
*Mar 18 23:37:32.859: Schedule time 07:48:01.328, Next wait_interval 10000ms
*Mar 18 23:37:32.859: Intra: 0ms, Inter: 0ms, External: 0ms
*Mar 18 23:37:32.859: R: 2, N: 1, Stubs: 2
*Mar 18 23:37:32.859: SN: 0, SA: 0, X5: 0, X7: 0
*Mar 18 23:37:32.863: SPF suspends: 0 intra, 0 total
```

**注意：**在开始故障排除流程时，在开启 SPF 的 `debug` 命令之前，请优先考虑使用 `show` 命令，比如 `show ip ospf statistics` 与 `show ip ospf` 命令。

## 第39天问题

1. OSPF operates over IP number \_\_\_\_\_.
2. OSPF does NOT support VLSM. True or false?

3. Any router which connects to Area 0 and another area is referred to as an \_\_\_\_\_ or \_\_\_\_\_.
4. If you have a DR, you must always have a BDR. True or false?
5. The DR/BDR election is based on which two factors?
6. By default, all routers have a default priority value of \_\_\_\_\_. This value can be adjusted using the \_\_\_\_\_ <0-255> interface configuration command.
7. When determining the OSPF router ID, Cisco IOS selects the highest IP address of configured Loopback interfaces. True or false?
8. What roles do the DR and the BDR carry out?
9. Which command would put network 10.0.0.0/8 into Area 0 on a router?
10. Which command would set the router ID to 1.1.1.1?
11. Name the common troubleshooting issues for OSPF.

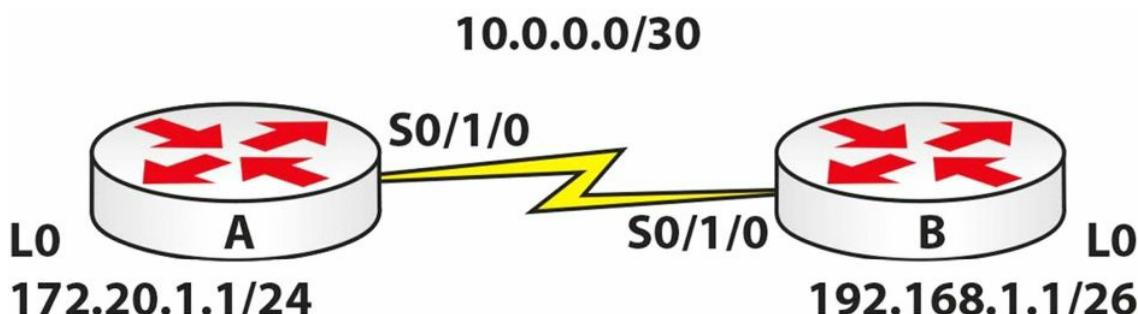
## 第39天答案

1. 89.
2. False.
3. Area Border Router or ABR.
4. False.
5. The highest router priority and the highest router ID.
6. 1, ip ospf priority .
7. True.
8. To reduce the number of adjacencies required on the segment; to advertise the routers on the Multi-Access segment; and to ensure that updates are sent to all routers on the segment.
9. The network 10.0.0.0 0.255.255.255 area 0 command.
10. The router-id 1.1.1.1 command.
11. Neighbour relationships and route advertisement.

## 第39天实验

### OSPF实验

拓扑



实验目的

学习如何配置基本的OSPF。

实验步骤

1. 基于上面的拓扑，配置上所有的IP地址。确保可经由那个串行链路进行Ping操作。
2. 将OSPF添加到路由器 A。将 Loopback0 上的网络放入到 Area 1，将那个 10 网络放入到 Area 0。

```

RouterA(config)#router ospf 4
RouterA(config-router)#network 172.20.1.0 0.0.0.255 area 1
RouterA(config-router)#network 10.0.0.0 0.0.0.3 area 0
RouterA(config-router)#^Z
RouterA#
%SYS-5-CONFIG_I: Configured from console by console
RouterA#show ip protocols
Routing Protocol is "ospf 4"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 172.20.1.1
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    172.20.1.0 0.0.0.255 area 1
    10.0.0.0 0.0.0.3 area 0
  Routing Information Sources:
    Gateway          Distance      Last Update
    172.20.1.1      110          00:00:09
  Distance: (default is 110)

```

1. 将OSPF添加到路由器 B。将该环回网络放入到OSPF的 Area 40。

```

RouterB(config)#router ospf 2
RouterB(config-router)#net 10.0.0.0 0.0.0.3 area 0
RouterB(config-router)#
00:22:35: %OSPF-5-ADJCHG: Process 2, Nbr 172.20.1.1 on Serial0/1/0 from LOADING to FULL, Loading Done
RouterB(config-router)#net 192.168.1.0 0.0.0.63 area 40
RouterB(config-router)# ^Z
RouterB#show ip protocols
Routing Protocol is "ospf 2"
  Outgoing update filter list for all interfaces is not set
  Incoming update filter list for all interfaces is not set
  Router ID 192.168.1.1
  Number of areas in this router is 2. 2 normal 0 stub 0 nssa
  Maximum path: 4
  Routing for Networks:
    10.0.0.0 0.0.0.3 area 0
    192.168.1.0 0.0.0.63 area 40
  Routing Information Sources:
    Gateway          Distance      Last Update
    172.20.1.1      110          00:01:18
    192.168.1.1     110          00:00:44
  Distance: (default is 110)

```

1. 对两台路由器上的路由表进行检查。查找那些OSPF通告的网络。将见到一个 IA，也就是OSPF的区域间 (inter-area)。还将见到OSPF的 AD，也就是管理距离 (Administrative Distance) 110。

```
RouterA#sh ip route
...
[Truncated Output]
    10.0.0.0/30 is subnetted, 1 subnets
C      10.0.0.0 is directly connected, Serial0/1/0
    172.20.0.0/24 is subnetted, 1 subnets
C      172.20.1.0 is directly connected, Loopback0
    192.168.1.0/32 is subnetted, 1 subnets
O IA    192.168.1.1 [110/65] via 10.0.0.2, 00:01:36, Serial0/1/0
RouterA#
```

1. 在两台路由器上分别执行一些可用的OSPF命令。

```
RouterA#sh ip ospf ?
<1-65535>      Process ID numberborder-routers Border and Boundary Router Information
database        Database summary
interface       Interface information
neighbor        Neighbor list
```

请访问[www.in60days.com](http://www.in60days.com)并观看作者是如何完成该实验的。

# 第40天 Syslog、SNMP与Netflow

## Syslog, SNMP and Netflow

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第40天任务

- 阅读今天的课文（下面）
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2补充指南
- 在[subnetting.org](http://subnetting.org)网站上花15分钟

将（系统）消息或事件在本地或某台syslog服务器上进行记录，是一项核心的维护任务。Syslog是一种允许主机将事件通知消息，经由IP网络发送到事件消息收集器（event message collectors），也叫做syslog服务器或syslog守候程序（daemon），的协议。也就是说，某台主机或设备可被配置为，生成一条syslog消息，并将该消息转发到某个特定的syslog守候程序（服务器）的方式。

简单网络管理协议，则是一种广泛使用的管理协议，其定义了一套用于连接到IP网络设备通信的标准。SNMP提供了一种用于对网络设备进行监视与控制的方法。与思科IOS的IP SLA操作（IP Service Level Agreement, IP 网络服务等级协议，该特性通过使用活动流量监视来测量网络性能，而允许客户对IP服务等級进行分析）一样，SNMP可用于收集统计数据、监测设备性能，以及提供到网络的一个基线（a baseline of the network），且SNMP是使用最为广泛的网络维护与监测工具之一。

尽管SNMP可以提供流量统计，但其无法区分各种数据流（While SNMP can provide traffic statistics, SNMP cannot differentiate between individual flows）。不过思科IOS的Netflow就可以做到。数据流（A flow）简单地就是一系列的、有着同样源与目的地址、源与目的端口、协议接口，以及同样的服务参数类（Class of Service parameters）的数据包。

今天将学到有关以下方面的知识：

- Syslog
- SNMP
- Netflow

此课程对应了以下CCNA大纲要求：

- 配置并验证Syslog
  - syslog 输出的使用
- 对SNMP版本2与版本3进行描述

- netflow 数据的使用

## 日志记录 (Logging)

所谓某个 `syslog` 守候程序或某台 `syslog` 服务器，就是一个对发送给它的 `syslog` 消息进行监听的实体。将某个 `syslog` 守候程序配置为请求某台特定设备向其发送 `syslog` 报文，是不可行的。也就是说，在某台特定设备无法生成 `syslog` 报文的情况下，那么 `syslog` 守候程序就什么也不能做。真实世界中，集团公司通常采用 SolarWinds（或类似）软件来做 `syslog` 的捕获。此外，诸如 Kiwi Syslog daemon 这类自由软件，也可用于 `syslog` 的捕获。

`Syslog` 使用用户数据报协议（User Datagram Protocol, UDP），作为所采用的传输机制，因此数据包没有被排序与确认。因为 UDP 没有包含在 TCP 中的额外开销（the overhead included in TCP），这意味着在某些重度使用的网络中，一些数据包可能被丢弃，而因此导致记录的信息丢失。不过思科 IOS 软件允许管理员出于冗余目的，配置多台 `syslog` 服务器。`syslog` 方案由两个主要元素构成：`syslog` 服务器与 `syslog` 客户端。

`syslog` 客户端将 `syslog` 报文，使用 UDP 作为传输层协议，指定一个 513 目的端口，发送给 `syslog` 服务器。这些报文的大小不能超过 1024 字节；不过没有最小长度的限制。所有 `syslog` 报文，都包含三个不同的部分：优先级、头部与报文。

`syslog` 报文的优先级，同时表示了设施，与报文的严重程度（The priority of a `syslog` message represents both the facility and the severity of the message）。此数字是一个 8 位的数字。前 3 个最低有效位（The first 3 least significant bits），表示报文的严重程度（在 3 位的情况下，可表示 8 中不同的严重程度），其它 5 位表示了某项设施。可使用这些值，来对 `syslog` 守候程序中的事件进行过滤。

**注意：**请注意这些值是由那些生成事件的应用产生的，而不是由 `syslog` 服务器本身产生的。

下表 40.1 中列出并介绍了思科 IOS 设备所设置的值（请记住这些严查程度级别与它们的名称）：

表 40.1 - 思科 IOS 软件 `syslog` 的优先级分级与定义

严重程度级别	严重程度级别名称	syslog 的定义	介绍
0	紧急 (Emergencies)	LOG_EMERG	此级别用于那些将导致系统不可用的最严重的错误情景。
1	告警 (Alerts)	LOG_ALERT	此级别用于表示那些需要管理员立即注意的情况。
2	严重 (Critical)	LOG_CRIT	此级别用于表明那些严重性低于告警，但仍需管理员介入的严重情况。
3	错误 (Errors)	LOG_ERR	此级别用于表明系统中有错误发生，但这些错误并不会导致系统不可用。
4	警告 (Warnings)	LOG_WARNING	此级别用于表示有关系统操作未能成功完成的警告情况。
5	通知 (Notifications)	LOG_NOTICE	此级别用于表示系统中的状态改变（比如路由协议邻接关系过渡到 down 状态）。
6	消息 (Informational)	LOG_INFO	此级别用于表示有关系统正常运行的消息。
7	调试 (Debugging)	LOG_DEBUG	此级别用于表示通常用于故障排除目的的实时的（调试的）信息。

在 syslog 中，设施 (the facility) 用于表示生成消息的源。源可以是某个本地设备上的进程、应用，或者甚至操作系统本身。设施是以数字（整数）表示的。在思科IOS软件中，有八个本地使用设施可由进程及应用（以及设备本身）用于发送 syslog 消息。默认思科IOS设备使用设施 local7 来发送 syslog 报文。但要注意大多数思科设备提供了改变默认设施级别的选项。在思科IOS软件中，可使用全局配置命令 `logging facility [facility]` 来指定 syslog 的设施。该命令可用的选项如下所示：

```
R1(config)#logging facility ?
auth    Authorization system
cron   Cron/at facility
daemon System daemons
kern   Kernel
local0 Local use
local1 Local use
local2 Local use
local3 Local use
local4 Local use
local5 Local use
local6 Local use
local7 Local use
lpr     Line printer system
mail   Mail system
news   USENET news
sys10  System use
sys11  System use
sys12  System use
sys13  System use
sys14  System use
sys9   System use
syslog Syslog itself
user   User process
uucp   Unix-to-Unix copy system
```

要通过 `syslog` 来发送消息，就必须在设备上执行以下顺序的步骤：

1. 使用 `logging on` 配置命令在路由器或交换机上全局性开启日志记录功能。默认在思科IOS软件中，日志记录是开启的；但仅开启了将消息发送到控制台。对于要将消息发到除控制台外的其它任何目的地，`logging on` 命令都是强制要求的。
2. 使用全局配置命令 `logging trap [severity]`，指定出要发送到 `syslog` 服务器的消息的严重程度。可使用数字或使用等价的严重性名称，来指定发送消息的严重程度。
3. 使用全局配置命令 `logging [address]` 或 `logging host [address]`，指定一个或多个的 `syslog` 服务器目的地址。
4. 作为可选项，使用 `logging source-interface [name]` 指定在 `syslog` 报文中的源IP地址。在有着配置的多个接口的设备上，这是普遍的做法。若未指定此命令，则 `syslog` 报文将包含路由器或交换机用于抵达服务器的接口的IP地址。而在出于冗余目的有着多个接口时，该地址就会在主要路径（接口）宕掉时发生改变。因此，通常将其设置为某个环回接口。

下面的配置实例，演示了如何将所有信息（informational(level6)）及以下的报文，发送到一台有着IP地址 192.168.1.254 的 `syslog` 服务器：

```
R2(config)#logging on
R2(config)#logging trap informational
R2(config)#logging 192.168.1.254
```

此配置可使用 `show syslog` 命令进行验证，如下所示：

```
R2#show logging
Syslog logging: enabled (11 messages dropped, 1 messages rate-limited, 0 flushes, 0 overruns, xml disabled)
Console logging: disabled
Monitor logging: level debugging, 0 messages logged, xml disabled, filtering disabled
Buffer logging: disabled, xml disabled, filtering disabled
Logging Exception size (4096 bytes)
Count and timestamp logging messages: disabled
No active filter modules.
Trap logging: level informational, 33 message lines logged
    Logging to 192.168.1.254(global) (udp port 514, audit disabled, link up), 2 message lines logged
```

一般在配置日志记录时，重要的是要确保路由器或交换机的时钟反映的是真实的当前时间，这可实现与错误数据的关联。日志消息上的不准确或不正确时间戳，会令到使用过滤或关联流程，来做错误与问题隔离十分困难，并十分耗时。在思科IOS软件中，系统时钟可手动配置，或者将设备配置为自动将其时钟与网络时间协议服务器进行同步。在后面的小节将对这两种方法进行讨论。在网络中仅有少数互联网络设备时，手动的时钟或时间配置没有问题。在思科IOS软件中，系统时间是通过使用 `clock set hh:mm:ss [day & month | month & day] [year]` 特权 EXEC 命令进行配置的。其不是在全局配置模式下配置或指定的。下面的配置示例，演示了如何将系统时钟设置为 2010 年 10 月 20 日上午 12:15：

```
R2#clock set 12:15:00 20 october 2010
```

也可以向下面这样在路由器上应用同样的配置：

```
R2#clock set 12:15:00 october 20 2010
```

在此配置下，可使用 `show clock` 命令来查看到系统时间：

```
r2#show clock
12:15:19.419 utc wed oct 20 2010
```

观察到一个有趣现象，就是在使用 `clock set` 命令手动配置或设置了系统时间是，其默认到GMT (UTC) 时区，如上面所看到的。为了确保系统始终反映对于不在GMT时区的那些正确时区，就必须使用全局配置命令 `clock timezone [time zone name] [GMT offset]`。比如，美国有六个不同的时区，每个时区都有不同的GMT偏移量。这些时区分别是东部时间（Eastern Time），中部时间（Central Time），山区时间（Mountain Time），太平洋时间（Pacific Time）、夏威夷时间（Hawaii Time）以及阿拉斯加时间（Alaska Time）。

此外，一些地方使用标准时间（Standard Time）与夏令时间（Daylight Saving Time）。考虑这个因素，那么在手动配置系统时钟时，确保于所有设备上正确设置系统时间（标准还是夏令时）就很重要了。下面的配置实例，演示了如何将系统时钟，设置为比GMT晚6个小时的中部标准时间（Central Standard Time, CST）时区的2010年10月20日上午12点40分：

```
R2#config t
Enter configuration commands, one per line.
End with CNTL/Z.
R2(config)#clock timezone CST -6
R2(config)#end
R2#clock set 12:40:00 october 20 2010
```

依据此配置，本地路由器上的系统时钟现在显示为下面这样：

```
R2#show clock
12:40:17.921 CST Wed Oct 20 2010
```

**注意：** 如在 `clock timezone` 命令之前使用 `clock set` 命令，那么使用 `clock set` 命令所指定的时间，将被 `clock timezone` 命令的使用进行偏移。比如假定上面示例中使用的配置命令是像下面这样输入的时：

```
R2#clock set 12:40:00 october 20 2010
R2#config t
Enter configuration commands, one per line.
End with CNTL/Z.
R2(config)#clock timezone CST -6
R2(config)#end
```

因为这里 `clock set` 命令先使用，所以路由器上的 `show clock` 命令将显示偏移了6小时的系统时钟，就如使用 `clock timezone` 命令所指定的那样。在同样的路由器的以下输出对此行为进行了演示：

```
R2#show clock
06:40:52.181 CST Wed Oct 20 2010
```

**注意：** 使用全局配置命令 `clock summer-time zone recurring [week day month hh:mm week day month hh:mm [offset]]`，可将思科IOS的路由器与交换机可配置为自动切换到夏令时间（summertime, Daylight Saving Time）。这样做可消除标准时间与夏令时期间，在所有手动配置的设备上，手动调整系统时钟的需要。

第二种设置或同步系统时钟的方法，就是使用网络时间协议服务器作为参考时间源了。在那些有着多余几台设备的较大网络中，这是首选方法。NTP是一个专用于机器网络时间同步的协议。在[RFC 1305](#)中对NTP进行了文档说明，其运行在UDP上。

NTP网络通常是从权威的时间源处，比如无线电时钟或连接某台时间服务器的原子钟，获取它的时间。NTP随后经由网络对此时间进行分发。NTP是相当高效的；每分钟不多于一个数据包，就可以将两台机器同步到一毫秒之内。

NTP使用层的概念（a concept of a stratum），来描述某台机器距离权威时间源有多少跳。请记住这不是路由或交换的条数，而是NTP跳数，这是一个完全不同的概念。一台层 1 的时间服务器（A stratum 1 time server），通常具有一个直接安装的无线电或原子钟，同时一台层 2 的时间服务器（a stratum 2 time server），则是通过NTP从层 1 的时间服务器接收其时间，如此等等。在某台设备被配置了多台NTP参考服务器时，它将自动选择有着配置为通过NTP进行通信的最低层编号的机器，作为其时间源（When a device is configured with multiple NTP reference servers, it will automatically choose as its time source the machine with the lowest stratum number that it is configured to communicate with via NTP）。

在思科IOS软件中，使用全局配置命令 `ntp server [address]`，来将某台设备配置带有一台或多台NTP服务器的IP地址。如先前指出的那样，可通过重复使用同样的命令，指定多个NTP参考地址。此外，该命令还可用于配置服务器与客户端之间的安全及其它特性。下面的配置实例，演示了如何将某台设备配置为将其时间与一台有着IP地址 10.0.0.1 的NTP进行同步：

```
R1(config)#ntp server 10.0.0.1
```

根据此配置，可使用 `show ntp associations` 命令来对NTP设备之间的通信进行检查，如下面的输出所示：

```
R2#show ntp associations
address      ref clock      st  when   poll   reach   delay   offset   disp
*~10.0.0.1  127.127.7.1  5   44     64     377    3.2     2.39    1.2
*           master (synced), # master (unsynced), + selected, - candidate, ~ configured
```

`address` 字段表示NTP服务器的IP地址，如同该字段下所指出的值 10.0.0.1 所确认的那样。而 `ref clock` 字段则表示了那台NTP服务器所使用的参考时钟。在此实例中，IP地址 127.127.7.1 表示该设备使用的是一个内部时钟（127.0.0.0/8 子网）作为其参考时间源。如该字段包含了另一个值，比如 192.168.1.254，那么那将是该服务器用作其参考的IP地址。

接着的 `st` 字段表示该参考的层（the stratum of the reference）。从上面的打印输出，可以看到 10.0.0.1 的NTP设备有着 5 的层数。本地设备的层数，将增加 1 到值 6，如下所示，因为其是从有着层 5 的服务器处接收到的时间源。如有另一台设备被同步到该本地路由器，那么它将反应出一个 7 的层数，如此等等。用于检查NTP配置的第二个命令，就是 `show ntp status` 命令了，其输出如下面所示：

```
R2#show ntp status
Clock is synchronized, stratum 6, reference is 10.0.0.1
nominal freq is 249.5901 Hz, actual freq is 249.5900 Hz, precision is 2**18
reference time is C02C38D2.950DA968 (05:53:22.582 UTC Sun Mar 3 2002)
clock offset is 4.6267 msec, root delay is 3.16 msec
root dispersion is 4.88 msec, peer dispersion is 0.23 msec
```

该 `show ntp status` 命令的输出表示时钟是被同步到所配置的NTP服务器（10.0.0.1）。此服务器有着层数 5，因此本地设备反应了一个层数 6。在配置了NTP是一个观察到的一个有意思的事情，就是本地时间将默认到GMT，如在上面的输出中所看到的那样。为确保该设备显示正确的时区，就必须在该设备上执行 `clock time-zone` 命令。

在不论是通过手动还是NTP设置好系统时钟之后，都要确保发送给服务器的日志包含正确的时间戳。这是通过使用全局配置命令 `service timestamp log [datetime | uptime]` 执行的。关键字 `[datetime]` 支持下面这些字面的额外子关键字：

```
R2(config)#service timestamps log datetime ?
  localtime      Use local time zone for timestamps
  msec           Include milliseconds in timestamp
  show-timezone  Add time zone information to timestamp
  year           Include year in timestamp
<cr>
```

而 [uptime] 关键字则没有额外关键字，而将本地路由器配置为仅包含系统运行时间（the system uptime）作为发送的消息的时间戳。下面的配置实例，演示了如何将本地路由器配置为所有消息都包含本地时间、毫秒信息，以及时区：

```
R2#configure terminal
Enter configuration commands, one per line.
End with CNTL/Z.
R2(config)#logging on
R2(config)#logging console informational
R2(config)#logging host 150.1.1.254
R2(config)#logging trap informational
R2(config)#service timestamps log datetime localtime msec show-timezone
```

根据此配置，本地路由器的控制台将打印以下消息：

```
Oct 20 02:14:10.519 CST: %SYS-5-CONFIG_I: Configured from console by console
Oct 20 02:14:11.521 CST: %SYS-6-LOGGINGHOST_STARTSTOP: Logging to host 150.1.1.254 started - CLI initi:
```

此外，在服务器 150.1.1.254 上的 syslog 守候程序，将反映出同样内容，如下图40.1中的Kiwi Syslog Manager屏幕截图中所示：

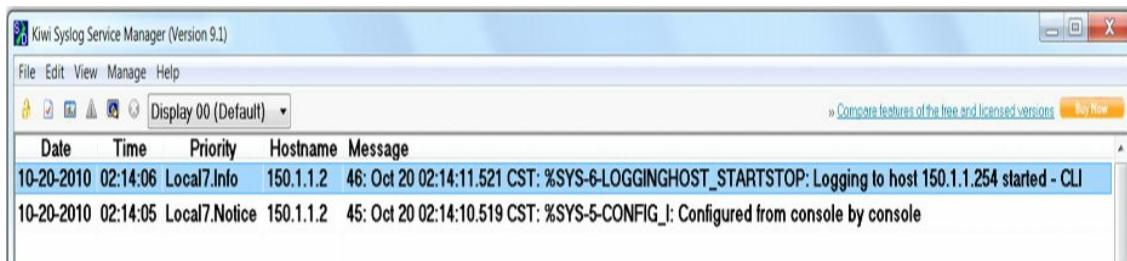


图 40.1 - 日志时间戳的配置

## 简单网络管理协议（Simple Network Management Protocol）

简单网络管理协议，是一个应用层（Layer 7）协议，使用UDP端口 161 与 162，促进网络设备之间管理信息的交换。一个SNMP管理的网络，由管理系统、代理程序及所管理的设备构成（An SNMP-managed network consists of a management system, agents, and managed devices）。其中，管理系统执行监控应用及对所管理的设备进行控制。其也执行大多数的管理流程，并提供了用于网络管理的大部分存储资源。某个网络科恩该是由一套或多套的管理系统所管理。

代理程序，则是出于各个受管理的设备之上，而对本地管理信息数据，比如性能信息或事件与软件装置中捕获到的错误信息（error information caught in software traps），转换为管理系统可读取的形式。SNMP代理程序使用将数据传输到网络管理软件的SNMP get-request 指令。SNMP代理程序从管理信息库

(Management Information Bases, MIBs) 或从错误或修改陷阱设置处捕获数据，而管理信息库则是设备参数与网络数据存放的地方 (SNMP agents capture data from Management Information Bases(MIBs), which are device parameters and network data repositories, or from error or change traps)。

而受管理元素，比如路由器、交换机、计算机或防火墙，是通过SNMP代理程序进行管理的。受管理设备对管理信息进行收集与存储，从而令到这些信息通过SNMP对其他有着相同协议兼容性的管理系统可用。下图40.2演示了SNMP管理的网络的三个主要部件之间的交互：

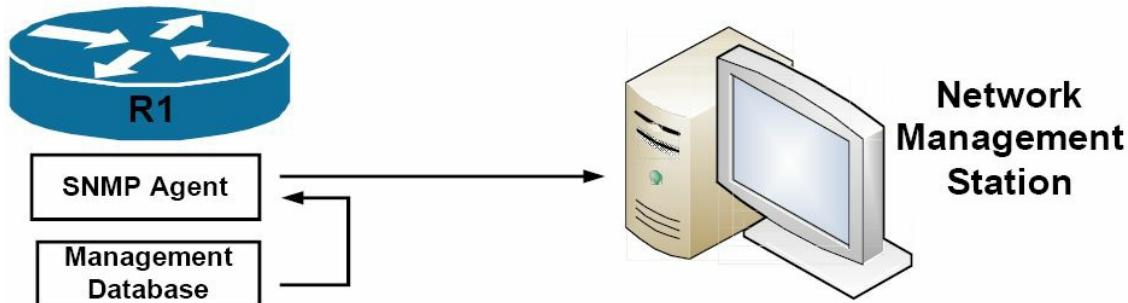


图 40.2 - SNMP 网络组件的交互

参考图40.2, R1 就是SNMP管理的设备。逻辑上出于该设备上的，就是SNMP代理程序。SNMP代理程序，将存储在受管理设备的管理数据库中的本地管理信息数据，转化为这里称为网络管理站（Network Management Station, NMS）的管理系统可读取的形式。

在使用SNMP时，使用三种常见的SNMP命令： `read`、`write` 与 `trap`，使得受管理的设备得以被监视与控制。网络管理站所使用的 `read` 命令，用于监视受管理的设备。这是通过NMS对由受管理设备所维护的不同变量进行检查完成的。而 `write` 命令，则是由NMS用于对受管理设备进行控制的。NMS使用该命令可对存储在受管理设备上的变量的值，进行修改。最后，SNMP的 `trap` 命令，是由受管理设备，用来将事件报告给NMS的。设备可配置为将SNMP陷阱或通知，发送给NMS。所发送的陷阱或通知，取决于设备上所运行的思科IOS软件版本，以及设备的平台。

SNMP陷阱简单地就是就网络上的某个状况，通知SNMP管理器的消息（SNMP traps are simply messages that alert the SNMP manager of a condition on the network）。一个SNMP陷阱的实例，可能包含了某个接口从 `up` 状态过渡到了 `down` 状态。SNMP的主要问题在于它们是无确认的。这就意味着发出设备无法确定该陷阱是否被NMS接收到。

而SNMP通知命令，则是包含了来自SNMP管理器的接收确认的SNMP陷阱。这些消息可用于表示诸如失败的认证尝试，或失去到邻居路由器的连接等消息。管理器在没有接收到通知请求的情况下，它就不发送响应。而发送者在从没有接收到响应的情况下，通知请求可被再度发送。因此，SNMP的通知，更可能抵达其想要的目的（Thus, informs are more likely to reach their intended destination）。

尽管通知比陷阱更为可靠，但不利支出在于它们在路由器上与网络中消耗了更多的资源。与发出后就丢弃的陷阱不同，在接收到一个响应或请求超时之前，通知请求（an inform request）必须要驻留在内存中。此外陷阱仅发送一次，而通知在没有接收到一个来自SNMP服务器的响应之前，必须多次发送。

下图40.3演示了SNMP管理器与SNMP代理程序之间，发送陷阱与通知的通信：

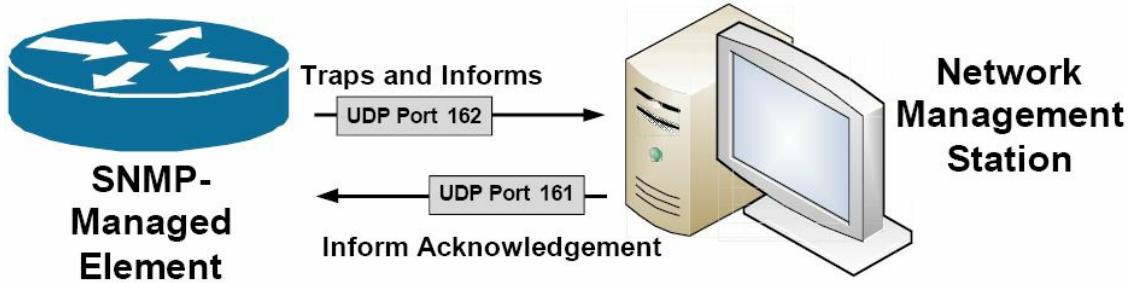


图 40.3 - 由网络管理站与SNMP管理的元素所使用的UDP端口

SNMP的三个版本分别是版本 1、2 与 3。版本 1，或 SNMPv1，是SNMP协议的最初实现。SNMPv1 运行在诸如用户数据报协议（UDP）、互联网协议（IP），以及开放系统互联的无连接网络服务（OSI Connectionless Network Service, CLNS）之上。

SNMPv1 是广泛使用的，且是互联网社区中使用的事上的网络管理协议。

SNMPv2 对 SNMPv1 进行了修订，包含了在性能、安全性、保密性及管理器到管理器通信等方面的提升。SNMPv2 还定义了两种新的操作（命令、operations）：GetBulk 与 Inform。GetBulk 用于有效地获取大块的数据（large blocks of data）。Inform 操作允许一个网络管理站发送陷阱信息到另一网络管理站，并于随后接收一个响应。在 SNMPv2 中，如某个对 GetBulk 操作进行响应的代理程序无法在一个清单中提供所有变量的值，那么它就提供部分结果。

SNMPv3 提供了先前版本的SNMP所不具备的以下三项额外安全服务：消息完整性、认证及加密。SNMPv3 使用消息完整性来确保数据包在传输过程中不被篡改。SNMPv3 还使用了用于判断消息是否是来自有效的源。最后 SNMPv3 提供了用于打乱（scramble）数据包内容，以防止其被未授权的源看到的加密机制。

在思科IOS软件中，使用 `snmp-server host [hostname | address]` 命令，来指定本地设备将发送陷阱或通知的目的主机名或IP地址。为实现网络管理站对本地设备的轮询，SNMPv1 与 SNMPv2 要求使用全局配置命令 `snmp-server community <name> [ro | rw]`，为只读或读写访问，指定一个共有字符串（a community string）。

SNMPv3 蜜柑有使用这种同样的基于共有的安全形式（the same community-based form of security），而是使用了用户与组的安全（user and group security）。下面的配置实例，演示了如何配置带有两个共有字符串的本地设备，其一用于只读访问，另一个用于读写访问。此外，该本地设备还配置了为思科IOS的SLA（Service Level Agreement, 服务级别协议）操作/命令与 `syslog`，而使用只读共有字符串，将SNMP陷阱发送到 1.1.1.1：

```
R2#config t
Enter configuration commands, one per line.
End with CNTL/Z.
R2(config)#snmp-server community unsafe RO
R2(config)#snmp-server community safe RW
R2(config)#snmp-server host 1.1.1.1 traps readonlypassword rtr syslog
```

下图40.4演示了一个基于SNMP轮询（SNMP polling）的、使用ManageEngine OpManager网络监控软件的，设备资源使用情况与可用性的示例报告：

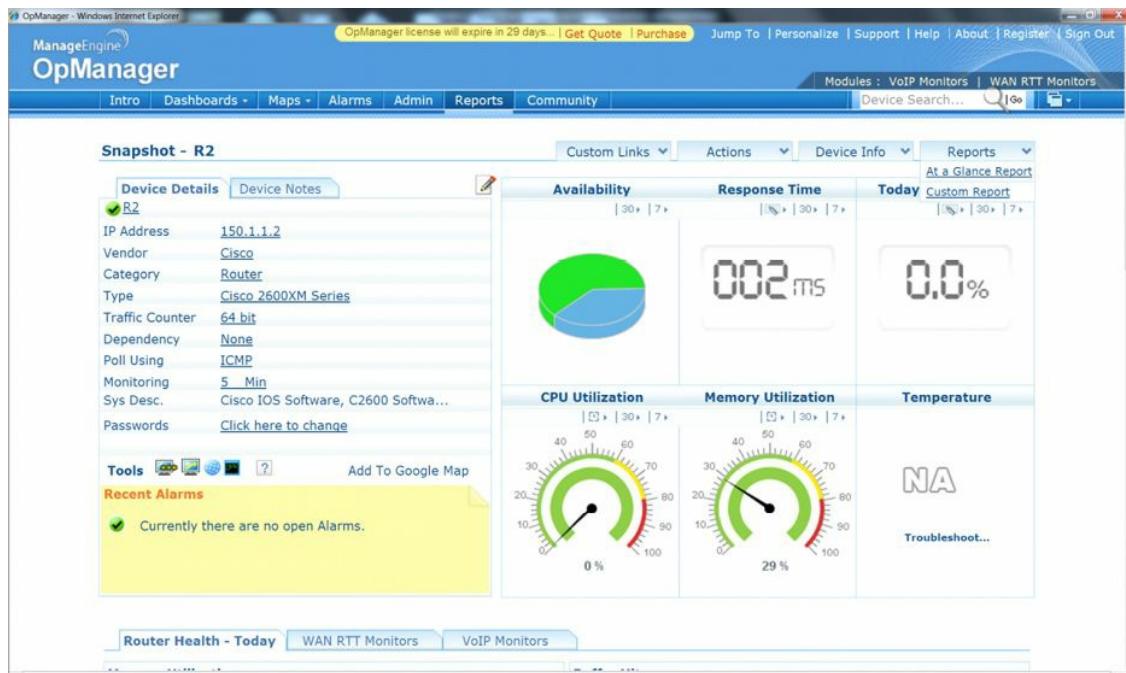


图 40.4 - 有关设备资源使用情况的示例SNMP报告

## 思科IOS的NetFlow (Cisco IOS NetFlow)

与SNMP一样，思科IOS的NetFlow是一个强大的维护与监控工具，可用于对网络性能进行基准测量及辅助故障排除。但其与SNMP之间有着一些显著的区别。第一个不同就是SNMP主要报告的是有关设备统计数据（比如资源使用情况等），而思科IOS的NetFlow则是就流量统计数据进行报告（比如数据包与字节）。

这两个工具之间的第二个不同，就是SNMP是一种基于轮询的协议（a poll-based protocol），意味着受管理设备被轮询信息。在那些SNMP设备发送陷阱（甚至报告，even report）到管理站的实例中，也可认为它是基于推送的（push-based）。而思科IOS的NetFlow，则是基于推送的技术，意味着配置了NetFlow的设备，是将其收集的信息发送出来，到某个中心存储库的。由于这个原因，NetFlow与SNMP互为补充，可作为标准网络维护与监控工具套件（the standard network maintenance and monitoring toolkit）的组成部分。但它们并非各自的替代；这是一个常被误解的概念，重要的是记住这一点。

IP（数据）流基于五个，上至七个的一套IP数据包属性，它们可能包含下面这些：

- 目的IP地址
- 源IP地址
- 源端口
- 目的端口
- Layer 3 的协议类型
- 服务类（Class of Service）
- 路由器或交换机的接口

除了这些IP属性外，（数据）流还包含了其它一些额外信息。这些额外信息包括对于计算每秒数据包与字节数有用的时间戳。时间戳还提供了有关某个数据流生命周期（持续时间）的信息。数据流还包括下一跳IP地址的信息，其包含了边界网关协议的路由器自治系统信息。除了TCP流量的标志外，数据流源与目的地址的子网掩码信息也有包含，而TCP流量的诸多标志，则可用于对TCP握手进行检查。

**译者注：**总的来说，思科IOS的NetFlow中的数据流，包含了数据包属性（七种）、时间戳、包含BGP路由自治系统的下一跳IP地址信息、TCP流量的诸多标志，以及源与目的地址的子网掩码信息。

简要地讲，思科IOS的NetFlow特性，除了可用于提供有关网络用户与网络应用、峰值用量时间，与流量路由之外，还可用于有关的信息网络流量记账、基于用量的网络计费、网络规划、安全、拒绝服务攻击的监视能力，以及网络监控。所有的这些用途，令到其成为一个非常强大的维护、监控与故障排除工具。

思科IOS的NetFlow软件，对数据流数据进行收集，并将其存储在一个名为“NetFlow缓存”或简单地说就是“数据流缓存”的数据库中。数据流信息会留存到该数据流终止或停止、超时或缓存溢出为止。有两种方式来访问存储在数据流中的数据：使用命令行界面（也就是使用 `show` 命令），或导出该数据，并通过使用某种类型的报告工具对导出的数据进行查看。下图40.5演示了在思科IOS路由器上的NetWork操作，以及数据流缓存的生成方式：

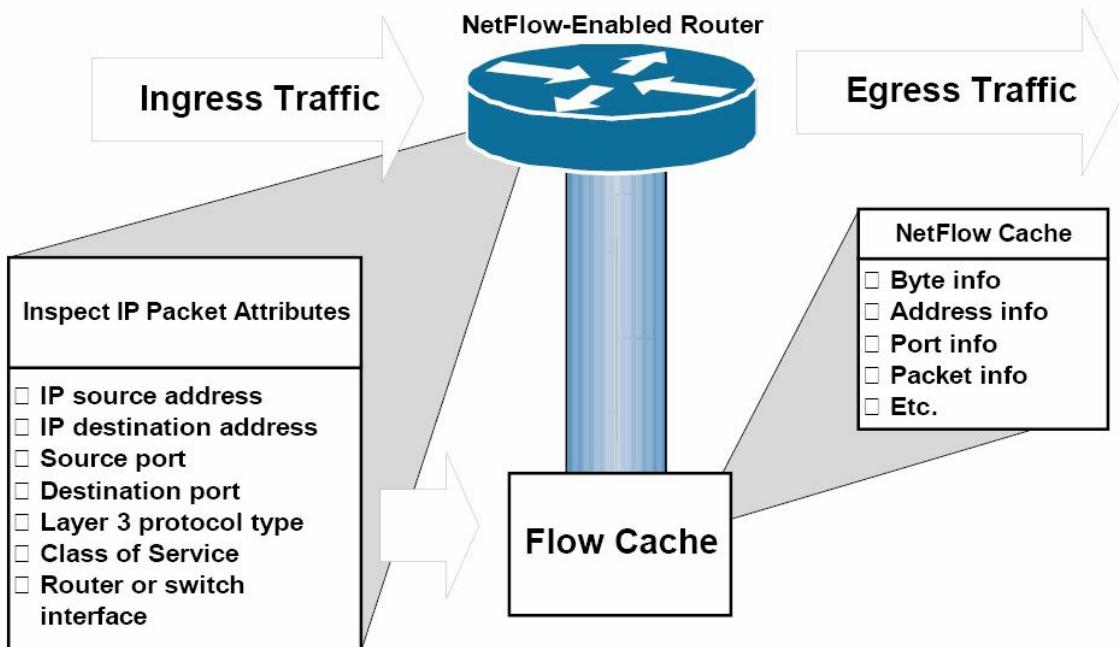


图 40.5 - 基本的NetFlow操作与数据流缓存的生成

参考图40.5，在本地路由器上入口流量被接收到。该流量被路由器加以探测，且IP属性信息被用于创建一个数据流。随后该数据流信息被存储在流缓存中。该信息可使用命令行界面进行查看，或被导出到某个称为NetFlow收集器的外部目的地，随后在NetFlow收集器出，该同样的信息可使用某种应用报告工具（an application reporting tool）进行查看。要实现将NetFlow数据报告给NetFlow收集器，需要使用以下步骤：

1. 在设备上要配置思科IOS的NetFlow特性，以将数据流捕获到NetFlow缓存。
2. 要配置好NetFlow导出功能，以将数据流发送到收集器。
3. 就那些已经有一段时间不活动的、以被终止的，或者仍活动但超出了活动计时器的数据流，对NetFlow进行搜索（The NetFlow cache is searched for flows that have been inactive for a certain period of time, have been terminated, or, for active flows, that last greater than the active timer）。
4. 将这些已标识出的数据流导出至NetFlow收集器服务器（Those identified flows are exported to the NetFlow Collector server）。
5. 将接近30到50个数据流打包在一起，并通常经由UPD进行传送。
6. NetFlow收集器软件从数据创建出实时或历史性的报告。

在配置思科IOS的NetFlow特性时，需要三个主要步骤，如下所示：

1. 在那些希望对信息进行捕获并在流缓存中存储的所有接口上，使用接口配置命令 `ip flow ingress`，把接口配置为将数据流捕获进入NetFlow缓存。重要的是记住NetFlow仅在每个接口的基础上配置的

(Configure the interface to capture flows into the NetFlow cache using the `ip flow ingress` interface configuration command on all interfaces for which you want information to be captured and stored in the flow cache. It is important to remember that NetFlow is configured on a per-interface basis only)。

**Dario先生的提醒：**命令 `ip route-cache flow` 可在物理接口及其下的所有子接口上，开启 (NetFlow) 数据流 (the `ip route-cache flow` command will enable flows on the physical interface and all subinterfaces associated with it)。而 `ip flow ingress` 命令则将开同一接口上的单个子接口、而非所有子接口上，开启 (NetFlow) 数据流。这在对观看某个接口的子接口 x、y 及 z 上的数据流不感兴趣，而真正想要观看同一接口上的子接口 A、B 与 C 子接口上的数据流时，此命令就很好用。此外，在NetFlow版本5下，唯一选项是使用 `ip flow ingress` 命令来监视上传统计数据 (with NetFlow v5, the only option was to monitor inbound statistics using the `ip flow ingress` command)。不过随着NetFlow版本9的发布，现在就了使用 `ip flow egress` 命令，来对离开各个接口的流量进行监控的选择了。

**注意：**从思科IOS版本 12.4(2)T 及 12.2(18)SXD 起，已将命令 `ip flow ingress` 替换为 `ip route-cache flow` 命令。而从思科IOS版本 12.2(25)S 起，命令 `show running configuration` 的输出已被修改，因此命令 `ip route-cache flow` 命令，以及 `ip flow ingress` 命令，将在二者之一被配置后，出现在 `show running-configuration` 的输出中。

随后NetFlow信息就存储在本地路由器上，同时可在本地设备上，使用 `show ip cache flow` 查看到。

在打算将数据导出到NetFlow收集器的情况下，将需要两个额外任务，如下：

1. 使用全局配置命令 `ip flow-export version [1 | 5 | 9]`，配置思科IOS NetFlow的版本或格式。  
NetFlow版本 1 (v1) 是在首个NetFlow发布中所支持的最初格式。在用于分析导出的NetFlow数据的应用仅支持该版本时，才应使用此版本。相比版本 1，版本 5 导出更多的字段，同时也是应用最广泛的版本。而版本 9 则是最新的思科IOS NetFlow版本，也是一个新的IETF标准的基础。版本 9 是一个灵活的导出格式版本。
2. 使用全局配置命令 `ip flow-export destination [hostname | address] <port> [udp]`，配置并指定 NetFlow收集器的IP地址，并于随后指定NetFlow收集器用于接收来自思科设备的UDP输出的UDP端口。其中的 `[udp]` 关键字是可选的，且在使用该命令是不需要指定，因为在将数据发送到NetFlow收集器时，用户数据报协议是默认使用的传输协议。

以下实例演示了如何为某个指定的路由器接口开启NetFlow：

```
R1#config t
Enter configuration commands, one per line.
End with CNTL/Z.
R1(config)#interface Serial0/0
R1(config-if)#ip flow ingress
R1(config-if)#end
```

根据此配置，可使用 `show ip cache flow` 命令来查看在数据流缓存中所收集的统计数据，如下面的输出所示：

```
R1#show ip cache flow
IP packet size distribution (721 total packets):
 1-32   64   96   128   160   192   224   256   288   320   352   384   416   448   480
 .000 .980 .016 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000
 512   544   576  1024  1536  2048  2560  3072  3584  4096  4608
 .002 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000 .000

IP Flow Switching Cache, 278544 bytes
 4 active, 4092 inactive, 56 added
 1195 ager polls, 0 flow alloc failures
 Active flows timeout in 30 minutes
 Inactive flows timeout in 15 seconds

IP Sub Flow Cache, 21640 bytes
 4 active, 1020 inactive, 56 added, 56 added to flow
 0 alloc failures, 0 force free
 1 chunk, 1 chunk added
 last clearing of statistics never

Protocol      Total    Flows   Packets Bytes  Packets Active(Sec)  Idle(Sec)
-----        Flows     /Sec    /Flow   /Pkt   /Sec     /Flow
TCP-Telnet      2       0.0      34     40      0.0     10.5    15.7
TCP-WWW         2       0.0      9      93      0.0     0.1     1.5
UDP-NTP         1       0.0      1      76      0.0     0.0     15.4
UDP-other       42      0.0      5      59      0.0     0.0     15.7
ICMP            5       0.0      10     64      0.0     0.0     15.1
Total:          52      0.0      7      58      0.0     0.4     15.1

SrcIf      SrcIPaddress  DstIf      DstIPaddress  Pr SrcP DstP  Pkts
Se0/0      150.1.1.254  Local      10.0.0.1    01 0000 0800  339
Se0/0      10.0.0.2     Local      1.1.1.1    06 C0B3 0017   7
Se0/0      10.0.0.2     Local      10.0.0.1   11 07AF D0F1   1
Se0/0      10.0.0.2     Local      10.0.0.1   11 8000 D0F1  10
Se0/0      150.1.1.254  Local      10.0.0.1   01 0000 0800  271
Se0/0      10.0.0.2     Local      1.1.1.1    06 C0B3 0017  59
```

下面的示例演示了如何配置并开启指定路由器接口的NetFlow数据收集，并于随后使用NetFlow版本 5 的数据格式，将数据导出到某台有着IP地址 150.1.1.254 的NetFlow收集器：

```
R1(config)#interface Serial0/0
R1(config-if)#ip flow ingress
R1(config-if)#exit
R1(config)#interface FastEthernet0/0
R1(config-if)#ip flow ingress
R1(config-if)#exit
R1(config)#interface Serial0/1
R1(config-if)#exit
R1(config)#ip flow-export version 5
R1(config)#ip flow-export destination 150.1.1.254 5000
R1(config)#exit
```

根据此配置，就可在那台NetFlow Collector上，使用某种应用报告工具，查看到收集的信息。而尽管有数据的导出，仍然可以在本地设备上，使用 `show ip cache flow` 命令来查看统计数据，在对网络故障进行排除或报告问题时，此命令可作为一个有用的工具。

## 使用NetFlow的数据进行故障排除（Troubleshooting Utilising NetFlow Data）

典型的企业网络，有着成千上万的、仅在很短时间内就生成海量NetFlow数据的连接。NetFlow数据可转换为帮助管理员弄清楚网络中正在发生什么的，有用图形与表格。NetFlow数据可辅助于以下方面：

- 提升整体网络性能
- 对一些诸如网络电话（VoIP）的应用提供支持
- 更好地对峰值流量进行管理（Better manage traffic spikes）
- 加强网络规定的执行（Enforce network policies）
- 揭示出那些指向恶意行为的流量模式（Expose traffic patterns that point to malicious activities）

NetFlow信息还可帮助管理员掌握到任何时候，各种数据类型所消耗的网络资源百分比。一眼就可以发现由电邮、计费与ERP系统及其它应用等使用了多少带宽，以及工作日期间有多少用户在观看YouTube视频，或在打互联网电话。

NetFlow数据可以易于理解的形式进行呈现，这就使得管理员能够轻易地对更多细节信息进行研究。他们可以就用户、应用、部门、对话、接口与协议等所产生的流量进行检查。使用NetFlow数据可以解决的一些情况示例，包括：

- 网络容量问题（Capacity issues）：NetFlow可清楚地显示什么应用使用了最多的带宽，及它们在何时使用了最多的带宽。此信息有助于改变应用流量模式，从而提升网络性能。通用的做法对用户进行应用。
- 安全问题（Security issues）：NetFlow数据可对网络上的未授权流量模式进行探测，并可在对网络造成任何危害之前阻止威胁。
- 网络语音故障（比如低质量，VoIP problems(poor quality, for example)）：在使用NetFlow分析识别出原因后，这方面的问题可被矫正。NetFlow报告可给出对网络语音通话造成影响的带宽不足（insufficient bandwidth）、延迟或网络抖动等因素。

## 第40天问题

1. What underlying protocol does syslog use?
2. The syslog client sends syslog messages to the syslog sever using UDP as the Transport Layer protocol, specifying a destination port of \_\_\_\_\_.
3. The priority of a syslog message represents both the facility and the severity of the message. This number is an \_\_\_\_\_ -bit number.
4. Name the eight Cisco IOS syslog priority levels.
5. In Cisco IOS software, the \_\_\_\_\_ global configuration command can be used to specify the syslog facility.
6. Which command do you use to globally enable logging on a router?
7. Name the command used to specify the syslog server destination.
8. Name the command used to set the clock on a Cisco IOS router.
9. On which ports does SNMP operate?
10. Name the command you can use to change the NetFlow version.

## 第40天答案

1. UDP.
2. 514 .
3. 8 .
4. Emergencies, alerts, critical, errors, warnings, notifications, informational, and debugging.
5. The `logging facility [facility]` command.

6. The `logging on` command.
7. The `logging [address]` OR `logging host [address]` command.
8. The `clock set` command.
9. UDP 161 and 162 .
10. The `ip flow-export version x` global configuration command.

## 第40天实验

### 日志记录实验

在思科路由器上配置日志记录：

- 选择日志记录设施 `local3` : `logging facility local2`
- 执行全局的 `logging on` 命令
- 选择日志记录的严重程度 `informational`
- 在一台PC机上配置一个自由的 `syslog` 服务器并将其连接到路由器
- 执行 `logging [address]` 命令来指定该 `syslog` 服务器
- 指定 `logging source-interface` 命令
- 验证命令 `show logging`
- 配置 `service timestamp log datetime localtime msec show-timezone` 命令
- 在PC机上检查 `syslog` 消息

### SNMP实验

在思科路由器上配置SNMP：

- 使用 `snmp-server host` 命令配置SNMP服务器
- 使用 `snmp-server community` 命令，配置SNMP的只读（RO）与读写（RW）共有字符串（Configure SNMP RO and RW communities using the `snmp-server community` command）

### NetFlow实验

在思科路由器上配置NetFlow：

- 在某个路由器接口上开启IP数据流的入口与出口（Enable IP flow ingress and egress on a router interface）
- 在有流量通过路由器后，对 `show ip cache flow` 命令进行检查
- 使用 `ip flow-export` 命令对NetFlow的版本进行配置
- 使用 `ip flow-export` 命令配置一台外部NetFlow服务器

请访问[www.in60days.com](http://www.in60days.com)网站，免费观看作者完成此实验。

## 第41天 广域组网

### Wide Area Networking

---

Gitbook: [ccna60d.xfoss.com](https://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第41天任务

- 约定今天的课文（下面）
- 复习昨天的课文
- 完成自主选择的实验
- 阅读ICND2记诵指南
- 在[subnetting.org](http://subnetting.org)网站上花15分钟

思科将WAN有关的概念拆分到了ICND1与ICND2考试中，后者关注的是帧中继及PPP协议（Frame Relay and PPP protocols）。因此，本手册将看看基本的WAN概念、技术及协议。

今天，将学习以下知识：

- 关于WAN的组件（WAN components）
- 关于WAN的协议（WAN protocols）
- 基本的串行线路的配置
- WAN连接的故障排除

此课程模块对应了以下ICND2考试大纲的有关有求：

- 认识不同的广域网技术
  - 城域以太网（Metro Ethernet）
  - 甚小孔径终端（Very Small Aperture Terminal, VSAT，参考[维基百科](#)）
  - T1/E1（参考[维基百科T-carrier](#), [维基百科E-carrier](#)）
  - T3/E3
  - 综合业务数字网（Integrated Services Digital Network, ISDN）
  - 数字用户线路（Digital Subscriber Line，缩写：DSL）
  - 同轴线组网
  - 第3代/第4代蜂窝网络（Celluar 3G/4G, 基站蜂窝网络）
  - 虚拟私人/专用网络（Virtual Private Network, VPN）
  - 多协议标签交换（Multi-Protocol Label Switching, MPLS）
- 配置并验证一条基本的WAN串行连接
- 对PPPoE进行部署与故障排除

## 广域网概述 (WAN Overview)

为了提供网络设施不同部分的连通性，广域网跨越极大的地理范围。与局域网环境不同，并非所有的WAN组件都是由其所服务的特定企业保有的。相反，WAN设备或连通性，可从服务提供商处进行短期或长期租用（rended, 短期、口头、临时的租用，leased, 长期、书面、固定期限的租用）。

大多数服务提供商都有良好培训，以确保它们可同时在极大的地理范围上，适当地支持传统数据流量，以及语音与视频业务（这些对延迟都更为敏感）。

有关WANs的另一个有趣的地方，即与LANs不同，这里通常有某种初期固定的投入，以及某种周期性的经常业务费用（Another interesting thing about WANs is that, unlike LANs, there is typically some initial fixed cost and some periodic recurring fees for the services）。在广域组网下，用户既不会拥有连接与某些设备，还将必须持续付费给服务提供商。这就是应避免高配（即购买仅需的带宽）的原因之一。这就带来对部署有效的服务质量机制（implementing effective Quality of Service mechanisms），以避免购买额外WAN带宽的需求。靠开销通常与带宽高配中出现的经常性开支有关（The high costs are usually associated with the recurring fees that might appear in the case of over-provisioning the bandwidth）。

有关WAN技术设计方面的要求，通常派生自以下这些方面：

- 应用的类型（Application type）
- 应用的可用性（Application availability）
- 应用的可靠性（Application reliability）
- 与某种特定WAN技术有关的成本情况（Costs associatedd with a particular WAN technology）
- 应用的使用级别（Usage levels for the application）

## 广域网的类别 (WAN Categories)

WAN分类中的一个必要概念就是电路交换技术，该技术最为相关的实例，就是公众交换电话网络了（An essential concept in WAN categorisation is circuit-switched technology, the most relevant example of this technology being the Public Switched Telephone Network, PSTN）。而归入此类别的一种技术，就是综合业务数字网。电路交换WAN连接的工作方式，是在需要连接时变为连接建立状态，并在连接不需要时连接终止。反映这种电路交换行为的另一个实例，就是老式的拨号连接（仅有PSTN的拨号调制解调器的模拟信号访问）。

**注意：**就在不久之前，拨号技术都还是访问互联网资源的唯一方式，这种方式提供到平均 40 kbps 的可用带宽。如今这种技术几乎绝迹了。

电路交换选择的反面，就是长期租用线路技术了（leased-line technology）。这种技术是一条完全专属的连接，持续可用并由租户公司拥有。长租线路的实例，包括基于时分复用的长租线路（Time Division Multiplexing(TDM)-based leased lines）。这类接入方式通常都很昂贵，因为单个客户具有连接的整个使用权。

WAN技术的另一种流行分类，涉及包交换网络（packet-switched networks）。在包交换设施中，共享带宽利用了虚拟电路技术（in a packet-switched infrastructure, shared bandwidth utilises virtual circuits）。客户可通过服务提供商的设施云，创建出一条虚拟路径（与长租线路类似）。此虚拟电路有着专属的带宽，但技术上将虚拟电路并非一条真实的长租线路。帧中继就是此种技术类型的一个实例。

包括作为帧中继前身的 X.25 在内的一些古早WAN技术。这种技术在某些实现中仍有出现，但已经很罕见了（如今帧中继也很少见了）。

另一种可能听过的WAN类别，就是单元交换技术（cell-switched technology）了。这种WAN类型通常包含在包交换技术中，因为它们非常类似。一种单元交换技术的实例，就是异步传输模式（Asynchronous Transfer Mode, ATM，这种技术如今也相当罕见了）。ATM是以固定大小的数据单元运作的，而不是使用数据包（如同在帧中继网络中所用的）。单元交换技术构成一个共享带宽的环境，因此服务提供商可确保客户有着通过其设施的固定水平的带宽。

宽带（Broadband）接入是另一种正在增长中的WAN类别，这种WAN接入方式包含了诸如以下这些技术：

- 数字订户线路（Digital Subscriber Line, DSL）
- 同轴线网络（Cable）
- 无线接入（Wireless）

宽带接入有着此种能力：采用如老式传输电视信号的同轴线的某种连接，并解决如何充分使用该既有带宽的不同方面的能力。比如通过将一个额外的、可与原先的电视信号一同传输的数据信号进行复用。

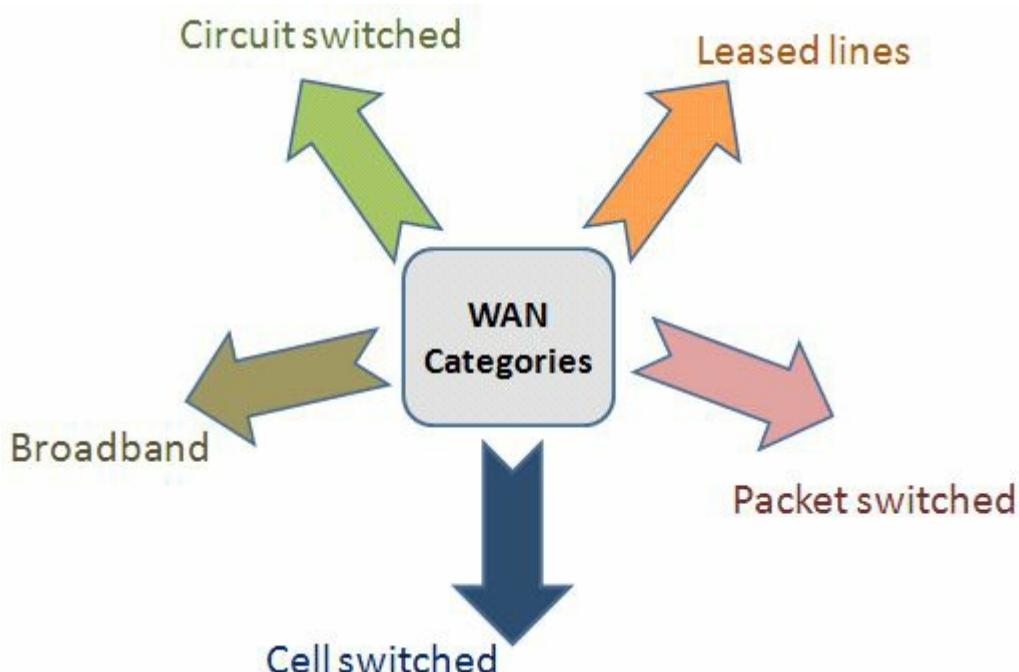


图 41.1 - 广域网的类别

如同在上面的图41.1中详细展示的那样，在讨论广域网的类别时有许多选项，同时这里只是对它们的简单介绍。所有这些技术都可以支持运行在 20/80 设计原则下的现代网络，所谓 20/80 设计原则，指的是 80% 的网络流量使用了某种广域网技术，来访问远端资源。

## 非广播多路复用技术（NBMA Technologies）

出现在广域组网中的一种特殊技术，就是非广播多路复用了。所谓非广播多路复用技术，表示某些在传统广播组网中没有的挑战。当某个需要经由同一网络进行通信系统分组不支持原生的广播时，就出现了非广播多路访问的需求（The need for NBMA arises when there is no native Broadcast support for a group of systems that want to communicate over the same network）。在设备无法原生地发送以多路访问网段上的所有设备为目的的数据包时，问题也就出现了。帧中继、ATM与ISDN默认就是非广播多路访问技术的实例。

所有这些技术都不具备支持广播的任何能力。这一点阻止了它们在其运作中运行那些用到广播的路由协议。在非广播网络中，原生的多播支持也是没有的。在某种路由协议情景下，参与的所有节点都必须接收到多播更新 (In the case of a routing protocol, all the nodes that participate must receive Multicast updates)。对于使用非广播多路访问网络的这个问题，一种办法就是作为重复的单播数据包，来发送多播或广播数据包。在这种方式下，广播/多播帧，是单独地发送到拓扑中的各节点的。此场景中的变通部分，就是设备必须想办法找到一种解决 Layer 3 到 Layer 2 解析的办法。特定数据包必须投送到需要接收到它们的特定机器上。

接着这个 Layer 3 到 Layer 2 的解析问题的方法必须存在才行。Layer 3 的地址通常是IP地址，而 Layer 2 的地址有通常根据所使用的技术而有所不同。在帧中继的情况下，Layer 2 地址将由数据链路连接标识符 (Data Link Connection Identifier, DLCI) 编号构成，那么就必须找到一种将DLCI解析到IP地址的方法。

在广播网络的情况下，Layer 3 的解析，使用MAC地址作为 Layer 2 的地址，且MAC地址也必须被解析到 IPv4地址。这是通过地址解析协议 (Address Resolution Protocol, ARP) 完成的。在基于广播的网络中，设备通过指定其想要与其进行通信的设备（通常经由DNS学习到），及询问特定于那些设备的MAC地址，而广播出解析请求。对地址解析请求的响应，是经由单播并包含了所请求的MAC地址 (In a Broadcast-based network, the devices broadcast the requests by specifying the devices it wants to communicate with(typically learned via DNS) and asking for the MAC addresses specific to those devices. The reply is via Unicast and includes the requested MAC addresses)。

在非广播多路访问环境中，仍需要将 Layer 3 地址 (IP地址) 绑定到 Layer 2 地址 (数据链路连接标识符)。这可通过使用反向ARP的一种自动化方式完成 (This can be done in an automated fashion using Inverse ARP)。此操作用于将远端的 Layer 3 地址解析到 Layer 2 地址，并仅用于本地。反向ARP可用在帧中继环境中。反向ARP作为一种在非广播多路复用环境中 Layer 3 到 Layer 2 解析的方案，问题在于其受限于直接连接的设备。这就造成在部分网状网络 (partial-mesh networks，其中并非所有设备都是直接连接的) 中的问题。

非广播多路服务的接口有两种 -- 多点与点对点接口，如下图41.2所示。多点接口要求某种 Layer 3 到 Layer 2 的解析方法。顾名思义，多点接口可作为多个 Layer 2 电路的端节点 (As its name implies, it can be the termination point of multiple Layer 2 circuits)。

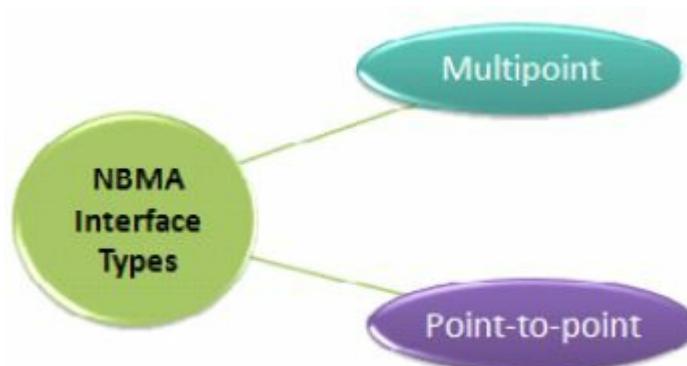


图 41.2 - 非广播多路访问接口类型

在设备的主要物理接口上配置了帧中继时，那个接口将默认成为多点的。而在某个帧中继物理接口上创建了一个子接口时，创建它的选项，就意味着多点存在了 (If a subinterface is created on a Frame Relay physical interface, the option of creating it as Multipoint exists)。对于物理接口与子接口（逻辑接口？），都必须配置上 Layer 3 到 Layer 2 的解析。在帧中继中，有两个选项来完成此解析：

- 反向ARP (Inverse ARP)
- 静态映射 (Statically map)

Layer 3 到 Layer 2 的解析，并不总是NBMA接口上的问题，因为可创建出点对点广域网接口（Point-to-Point WAN interfaces）。点对点接口仅能端接单个的 Layer 2 电路，因此在接口仅与单个设备通信时，Layer 3 到 Layer 2 的解析就无必要。在只有一条电路上，进行通信的就只有一个 Layer 2 地址。在比如运行一个帧中继的点对点子接口，或一个ATM的点对点子接口时，Layer 3 到 Layer 2 的解析问题会消失。

## 广域网组件 (WAN Components)

广域网需要一些物理组件，来建立连接 (WAN requires a number of physical components to enable a connection)。依据所使用的连接类型 (比如ISDN、ADSL、帧中继、长租线路等) 与其它因素，诸如后备连接与传入网络数目等，这些组件会有所不同。

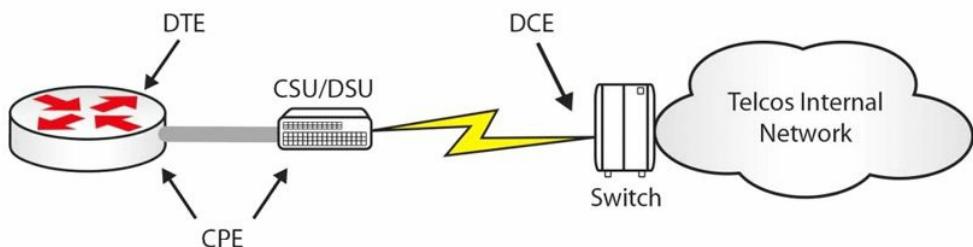


图 41.3 - 基本的广域网组件

上图41.3展示了一个基本的接到ISP的串行连接 (Figure 41.3 above shows a basic serial connection going out to an ISP)。作为用户，要负责数据终端设备 (Data Terminal Equipment, DTE)，也就是接受传入链路的用户路由器接口。用户还将负责连接到用户的信道服务单元/数据业务单元 (Channel Service Unit/DSU) 的电线，CSU/DSU将用户数据转换成ISP可传输的格式。CSU/DSU通常已被内建到用户路由器的WAN接口卡 (WAN interface card, WIC) 中。上图中的CPE为客户驻地设备 (Customer Premise Equipment, CPE)，由用户负责。

从这一点开始，通常就是ISP或电信公司来负责连接了。它们铺设电缆并提供将数据在其网络上传输的交换站 (From this point on, your ISP or Telco is usually responsible for the connection. They lay the cables and provide switching stations, which transport the data across their network)。ISP保有作为 (连接) 末端、提供时钟的数据通信设备 (Data Communication Equipment, DCE)，所谓时钟，指的是在线路上数据可以何种速率进行传递。

常见的广域网连接类型包含下面这些：

- 长租线路 - 7x24小时可用的专用连接 (Leased-line - a dedicated connection available 24/7)
- 电路交换 - 在需要时建立连接 (Circuit-switching - set up when required)
- 包交换 - 共享链路/虚拟电路 (Packet-switching - shared link/virtual circuit)

用户所购买的连接类型，取决于其需求与预算。在可承担专线费用时，就将有着带宽的独占性使用，同时安全问题也较少。而共享连接则意味着高峰时段连接速度较慢 (A shared connection can mean a slower connection during peak times)。

## 广域网的协议 (WAN Protocols)

常见的广域网协议包括点对点协议 (Point-to-Point Protocol, PPP)、高级别数据链路控制协议 (High-level Data Link Control, HDLC) 与帧中继协议 (Frame Relay)。当然也有许多其它协议，只是这里需要重点关注CCNA大纲中所包含的这三个协议。

点对点协议 (PPP) 可用于思科设备连接到一台非思科设备时。PPP同样具备包含认证的优势。其可在多种连接类型，包括数字订户线路 (DSL) 连接、电路交换连接以及异步/同步连接等上使用。

思科的高级数据链路控制 (High-level Data Link Control, HDLC) 是思科对开放标准HDLC的实现。HDLC需要数据终端设备 (DTE) 与数据通信设备 (DCE) ， 并是思科路由器 (串行接口) 上默认的封装类型。为对链路状态进行检查，会从DCE发出保持活动报文 (注：保持活动报文是由一台设备往另一台设备发送的，用于检查二者之间运作，或阻止链路破坏的报文) 。

如同早前所讨论的，帧中继是一种近年来以日渐式微的包交换技术，因为数字订户链路接入已成为相较帧中继更为经济及更可行的WAN连接方式。帧中继工作在从 56kbps 到 2Mbps 的速度上，并在每次连接需要时，建立起虚拟电路。没有将安全考量构建到帧中继中 (不过请参阅下面Farai的补充内容) 。后面将详细介绍到帧中继。

Farai先生谈到 -- “虽然帧中继可以使用按需创建的交换虚拟电路 (Switched Virtual Circuits, SVCs) ， 但其一般使用总是存在的永久虚拟电路 (Permanent Virtual Circuits, PVCs) 。永久虚拟电路是虚拟专用网络的一种 (a type of Virtual Private Network(VPN)) 。不过有人在帧中继上运行 PPP，从而实现帧中继连接的PPP安全性。”

## 城域以太网 (Metro Ethernet)

城域以太网 (Metro Ethernet) 技术涉及在城域网上运营商以太网的运用 (Metro Ethernet technologies involve the use of carrier Ethernet in Metropolitan Area Networks(MANs)) 。城域以太网可将公司局域网或个人终端用户连接到广域网或互联网。公司通常使用城域以太网将其分支机构连接到内部网 (Companies often use Metro Ethernet to connect branch offices to an intranet) 。

典型的城域以太网部署，通常采用铜缆或光缆，使用以互联的网络节点的星形或网状拓扑 (A typical Metro Ethernet deployment uses a star or a mesh topology with interconnected network nodes using copper or fibre optic cables) 。在城域以太网部署中采用标准及广泛应用的以太网技术，与同步光网络 (Synchronous Optical Networking, SONET) /同步数字体系 (Synchronous Digital Hierarchy, SDH) ， 或多协议标签交换 (Multi-Protocol Label Switching, MPLS) 相比，可提供到诸多优势：

- 较少的成本 (Less expensive)
- 更容易部署 (Easier to implement)
- 更易于管理 (Easier to manage)
- 因为其使用了标准的以太网方法，故易于连接客户设备 (Easy to connect customer equipment because it uses the standard Ethernet approach)

典型的城域网，可在接入/聚合/核心标准设计 (一种思科的设计模式，the access/aggregation/core standard design,) 下进行结构化，如下所示：

- 接入层 - 通常位于客户驻地处。这一层可能包含一台办公室路由器或家用网关 (Access Layer -- usually at the customer's premises. This may include an office router or residential gateway) 。
- 聚合层 - 通常由微波、数字订户线路 (DSL) 技术或点对点以太网链路等构成 (Aggregation Layer -- usually comprises microwave, DSL technologies, or Point-to-Point Ethernet links) 。
- 核心层 - 可能使用多协议标签交换技术来对不同城域网进行互联 (Core Layer -- may use MPLS to interconnect different MANs) 。

在城域网中，以使用实现数据包区分的以太网VLAN标签的方式，客户流量隔离通常得以确保 (Customer traffic separation is usually ensured in a MAN by using Ethernet VLAN tags that allow the differentiation of packets) 。

## 甚小口径终端（VSAT）

甚小口径终端（Very Small Aperture Terminal, VSAT）技术，是一种基于无线卫星技术的电讯系统。甚小口径终端是由小型卫星地球站与一个典型的天线构成，如下图41.4所示：

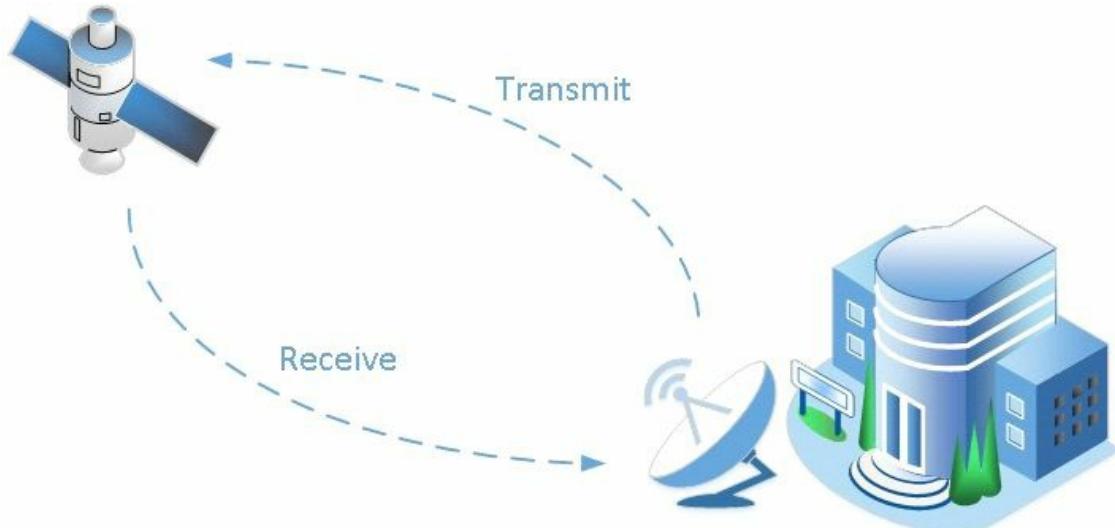


图 41.4 - 卫星通信

典型的甚小口径终端，其组件包括下面这些：

- 主地球站（Master earth station）
- 远端地球站（Remote earth station）
- 卫星

主地球站是整个VSAT网络的网络控制中心。于主地球控制站于完成整个网络的配置、管理与监测。

远端地球站则是安装在客户驻地的硬件设备，包含以下这些：

- 室外单元（outdoor unit, ODU）
- 室内单元（indoor unite, IDU）
- 连接电缆（interfacility link, IFL）

VSAT卫星环绕全球，从地球站接收信号并将信号发送给地球站（The VSAT satellite orbits round the globe and receives and transmits signals from and to the earth stations）。

VSAT网络可以下面的拓扑之一进行配置：

- 星形拓扑（Star topology）
- 网状网络（Mesh topology）
- 星形网状拓扑（Star-mesh topology）

使用卫星技术来确保WAN的连通性，一般要比使用传统的地面网络连接要昂贵（Using satellite technology to ensure WAN connectivity is generally more expensive than using a traditional terrestrial network connection）。此类连接所提供的速度可达 5Mbps 的下载与 1Mbps 的上传，对于远端站点这通常是足够的。

使用卫星连通性的一个显著不足，就是流量延迟的增加，延迟可达单向（天线到卫星或卫星到天线） 250ms，这是由于在极远距离上无线电信号的使用造成的。在规划安装卫星广域网连接时，延迟就应予以仔细分析，因为延迟增加可能导致那些对延迟敏感的应用停摆，当然对其它应用并没有什么影响。

使用卫星连通性的另一挑战，就是卫星碟形天线必须要有到卫星的视线（Another challenge of using satellite connectivity is that the satellite has to have line of sight to the satellite）。这就意味着必须使用高频范围（ $2\text{GHz}$ ），同时任何的干扰（如像是下雨或暴风云等自然现象），都将对连接的吞吐能力与可用性造成影响。

## T1 / E1

$\text{T1}$  与  $\text{E1}$  的广域组网标准已存在相当长时间了。 $\text{T1}$  代表 T-载波级别1（T-carrier Level 1,  $\text{T1}$ ），其为一条使用了基于时间的、与不同信道相关的数字信号的市分复用的线路（a line that uses Time Division Multiplexing with digital signals associated with different channels based on time）。 $\text{T1}$  使用 24 个分立信道、运行在  $1.544\text{Mbps}$  的线路传输速率，那么每个单独信道分配的就是  $64\text{Kbps}$ （ $\text{T1}$  operates using 24 separate channels at a  $1.544\text{Mbps}$  line rate, thus allocating  $64\text{Kbps}$  per individual channel）。这 24 个信道可以想怎么用就怎么用，设置可根据需要从服务提供商那里只购买其中的几个信道。笼而统之，可将一条  $\text{T1}$  连接，看作是一个有着 24 条分立线路的中继/捆绑（In general terms, consider a  $\text{T1}$  connection as a trunk/bundle carrying 24 separate lines）。在以下地区， $\text{T1}$  作为一项经常使用的标准：

- 北美
- 日本
- 韩国

$\text{E1}$ （E-载波级别1）是一种与  $\text{T1}$  类似的标准，不过仅在欧洲使用。 $\text{E1}$  与  $\text{T1}$  的主要区别在于， $\text{E1}$  使用了 32 个信道，而不是 24 个，这些信道仍然运行在  $64\text{Kbps}$ ，因此提供到共计  $2.048\text{Mbps}$  的线路速率。与  $\text{T1}$  一样， $\text{E1}$  也是基于时分复用的，因此二者之间的所有其它功能都一样。

## T3 / E3

$\text{T3} / \text{E3}$  标准提供到相较它们的  $\text{T1}$  与  $\text{E1}$  前辈更高的带宽。 $\text{T3}$  表示 T-载波级别3（T-carrier Level 3），且是一种通常基于同轴电缆与BNC连接器（英语：Bayonet Neill-Concelman，直译为“尼尔-康塞曼卡口”）的连接类型。这一点与通过双绞线介质进行提供的  $\text{T1}$  有所不同。

$\text{T3}$  连接通常被称为数字信号3（Digital Signal 3,  $\text{DS3}$ ，有贝尔实验室所涉及的T载波信号发送方案）连接，而  $\text{DS3}$  连接则与在  $\text{T3}$  线路上所传递的数据有关。因为  $\text{T3}$  使用相当于 28 条的  $\text{T1}$  电路，也就是 672 个  $\text{T1}$  信道，从而提供到额外的吞吐量。这就提供了总共  $44.736\text{Mbps}$  的线路速率。

$\text{E3}$  除了等价于 16 条的  $\text{E1}$  电路，也就是 512 个  $\text{E1}$  信道，及总计  $33.368\text{Mbps}$  的线路速率外， $\text{E3}$  连接与那些  $\text{T3}$  类似。

因为  $\text{T3} / \text{E3}$  提供了在需要时增加吞吐总量的能力，因此  $\text{T3} / \text{E3}$  连接通常用在大型数据中心里。

## 数字综合业务网（ISDN）

数字综合业务网（Integrated Services Digital Network, ISDN），是一种在传统模拟电话线路上，实现数字通信的技术，从而语音与数据都可在公众交换电话网（Public Switched Telephone Network, PSTN）上进行数字传输。因为其生不逢时，诞生之时恰逢其它替代技术也在开发，所以ISDN从来也没有如预期的那样得到广泛应用。

数字综合业务网有两种流派（There are two flavours of ISDN）：

- 数字综合业务网的基本速率接口（ISDN Basic Rate Interface）

- 主速率接口 (ISDN Primary Rate Interface)

采用ISDN协议设备被成为 **终端仿真设备**，而这类设备又可分类为原生ISDN与非原生ISDN设备（The ISDN-speaking devices are called terminal emulation equipment and the devices can be categorised into native ISDN and non-native ISDN equipment）。原生ISDN设备又制作作为ISDN就绪的装置构成，且这些设备被称为 **TE1**（终端设备一，Terminal Equipment 1）装置。非原生ISDN设备，则是由 **TE2** 装置构成。非原生ISDN设备，可使用特别的终端适配器（Terminal Adapters, **TA**s）与原生ISDN设备进行集成，也就是说只有 **TE2** 的装置，才需要终端适配器模块。

移步到ISDN服务提供商处，将找到网络端接二（Network Termination 2, **NT2**）设备，及网络端接一（Network Termination 1, **NT1**）设备。这些设备是传输介质的转换设备，将五线连接，转换成两线连接（本地环回）。本地环回就是用户连接线路，且它是一条两条线的链路。

网络端接装置（the network termination devices）的一个有趣的地方在于，在北美，是由客户负责 **NT1** 设备，而在世界上的其它地方，则是由服务提供商负责 **NT1** 设备的。因为这个问题，一些思科路由器提供了内建的 **NT1** 功能，而这些路由器将在端口编号下标注一个可见的 **u** 字符，这样用户就可以很快注意到路由器的此项能力。**u** 这个记号，是来自于ISDN的参考点命名法（the ISDN reference points terminology），该命名法对ISDN设施中的何处可能有故障进行了描述，如下图41.5中所示：

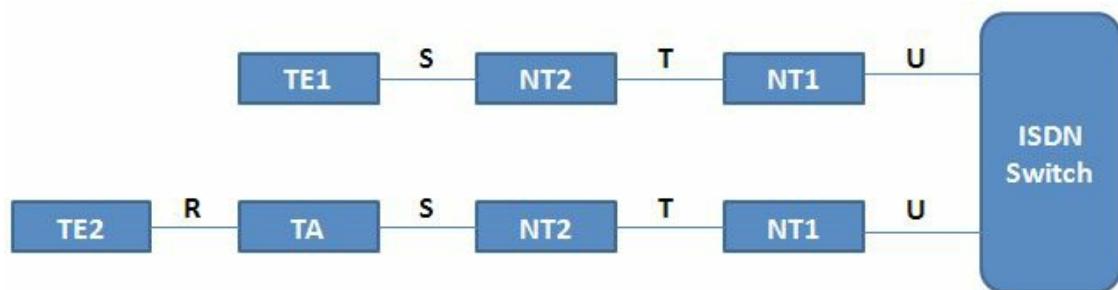


图 41.5 - ISDN 的参考点

在某个ISDN网络的故障排除或维护过程中，这些参考点尤为重要。ISDN交换机通常位于服务提供商处。这些不同的ISDN参考点就是下面这些：

- **u** 参考点 - 位于ISDN交换机与 **NT1** 装置之间
- **T** 参考点 - 位于 **NT2** 与 **NT1** 装置之间
- **S** 参考点 - 位于终端（**TE1** 或 **TA**）与 **NT2** 之间
- **R** 参考点 - 位于非ISDN原生设备（**TE2**）与 **TA**s 之间

**注：** 1、**TE2 + TA == TE1**；2、大多数 **NT1** 设备都包含了 **NT2** 设备的功能，因此 **S** 与 **T** 一般合并为 **S/T**；  
3、在北美，**NT1** 设备属于用户自备设备，用户必须自己来维护，因此电话公司提供给用户 **U** 接口。在其他国家，**NT1** 设备由电信公司维护，他们提供给客户 **S/T** 接口。

ISDN的基本速率接口（Basic Rate Interface, BRI）连通性，包含两个用于传递数据的 **B**（Bearer, 承载？）信道，与一个用于信号与控制（也能用于数据）的 **D**（Delta）信道。基本速率接口被缩写为 **2B+D**，以对每个接口所提供的信道数目进行提示。ISDN中的每个承载信道都将用 **64Kbps** 进行运作。可在这些接口上配置多链路的点对点协议（Multilink Point-to-Point Protocol），以允许用户达到 **128Kbps** 的带宽。与当今网络需求相比，此带宽被认为是相当低的。

BRI ISDN中的 **D** 信道，专用于 **16Kbps** 的控制流量。在ISDN环境中还有全部用于组帧控制及其它额外开销的 **48Kbps**（带宽），那么BRI的总ISDN带宽就是 **192Kbps**（来自 **B** 信道的 **128Kbps** + **D** 信道的 **16Kbps** + 额外开销 **48Kbps**），the **delta channel** in BRI ISDN is dedicated to **16Kbps** for control traffic. There are

also 48Kbps overall for framing control and other overhead in the ISDN environment, meaning the total ISDN bandwidth for BRI is 192Kbps ( 128Kbps from the B channels + 16Kbps from the D channel + 48Kbps Overhead)。

在美国和日本，ISDN主速率接口（Primary Rate Interface, PRI）有着 23 个 B 承载信道及一个 D 控制（delta）信道。所有承载与控制信道都支持 64Kbps。在包含了额外开销后，总的PRI带宽为 1.544Mbps。在世界上的其它地方（即欧洲与澳大利亚），PRI包含 30 个承载信道与一个控制信道。

ISDN PRI连接通常用作从公众交换电话网到大型电话系统（<电话>专用交换分机，专用电话总机，private branch (telephone) exchange, PBX）的连通性。23 或 32 个 B 信道中的每个，都可用作单独电话线路，因此可将整个的PRI连接，看作是传递多条线路的中继线路（a trunk that carries multiple lines）。使用PRI连接而不是多条独立线路的优势在于，其更易于管理且提供了伸缩性。

上面所介绍的技术，叫做时分复用技术。时分复用市值可将多个信道，经由单一完整的传输介质进行结合，并将这些不同信道分别用于语音、视频及数据。时分指的是将连接分切为不同通信信道的、小的时间窗口（TDM refers to being able to combine multiple channels over a single overall transmission medium and using these different channels for voice, video, and data. Time Division refers to splitting the connection into small windows of time for the various communication channels）。

在某个公众交换电话网中，就需要沿同一传输介质，传输多个通话，因此就使用时分复用来达到此目标。实际上在电报时期，时分复用就开始了，并于随后在传真机及其它使用到时分复用的设备上广泛使用。

在拥有长租线路（购买的专用带宽）时，所售卖的电路是以带宽的名义进行计量的。北美的一条数字信号1（Digital Signal 1, DS1）或T载波级别1（T-carrier Level 1, T1）电路提供了 24 个、每个 64Kbps 的时间槽（24 time slots of 64Kbps each）。以及一个 9Kbps 的控制时间槽（a 9Kbps control time slot, 如前所提到的，总共 1.544Mbps）。时分复用的术语，与长租线路采购流程紧密的联系在一起（TDM terminology is tightly connected with the leased-line purchasing process）。

## 数字订户线路（Digital Subscriber Line, DSL）

数字订户线路用作家庭用户的ISDN的替代方案（DSL is used as an alternative to ISDN for home users）。DSL连接的类型有数种，但最重要的几种包括下面这些：

- ADSL(Asymmetric DSL, 非对称DSL)
- HDSL (High-bit-rate DSL, 高速DSL)
- VDSL (Very-high-bit-rate DSL, 甚高速DSL)
- SDSL (Symmetric DSL, 对称DSL)

非对称DSL是经由标准电话线路运作的DSL连接的最常见形式。其被成为非对称的原因，就是其提供了不相等的下载和上传吞吐量，其中下载速率要高于上传速率。一条标准的非对称DSL连接，通常提供到在最远3公里的，最大 24Mbps 的下载吞吐量与最大 3.5Mbps 的上传吞吐量。

在非对称DSL下，客户是连接到服务提供商处的DSL访问服用器（DSL Access Multiplexer, DSLAM）上的。DSL访问服用器是一台对来自多个用户的连接，进行聚合的DSL汇聚设备（DSLAM is a DSL concentrator device that aggregates connections from multiple users）。

**注意：**ADSL的一个问题在于订户与DSLAM的距离受限。

高速DSL（High Bitrate DSL, HDSL）与甚高速DSL（Very High Bitrate DSL, VDSL），是其它大规模使用的DSL技术，提供了与ADSL相比增加了的吞吐量。甚高速率DSL可运行在最高 100Mbps 的速率。

对称DSL提供同样的下载与上传吞吐量，但其从未被标准化，抑或得到大规模使用。

## 同轴线 (Cable)

家庭用户也可经由标准电视同轴线连接，接收到数字信号。通过使用有线电视数据服务接口规范（the Data Over Cable Service Interface Specification, DOCSIS）标准，可经由同轴线提供互联网访问。此方式通常是一种廉价的服务，因为服务提供商不需要为数据服务安装新的设施。对既有网络的唯一升级，就是在客户驻地安装一台廉价的同轴线调制解调器，该调制解调器通常为用户设备提供了 RJ45 的数据连通性。

经由同轴线的数据流量传输速率可高达 100Mbps，这对于家庭用户甚至小型公司来说，都是绰绰有余的。

**注意：**除开电视和数据信号，同轴线连接还可传递语音流量。

可与同轴线结合使用的另一种技术，就是以太网上的点对点协议（Point-to-Point Protocol over Ethernet, PPPoE）。这种连接方式可在同轴线调制解调器与端点设备（the endpoint devices）之间使用，以为同轴线调制解调器设施添加安全性。这种连接方式允许用户登录，并提供为了获取要使用的同轴线业务，而必须加以认证的用户名与口令。（登录）凭据是通过到同轴线调制解调器的以太网连接，并晚于对运行在以太网上的PPP的使用之前，进行传递（The credentials are carried across the Ethernet connection to the cable modem and beyond by using the PPP running over the Ethernet）。后面将简短地对PPPoE进行介绍。

## 蜂窝网络 (Cellular Networks)

蜂窝网络是与移动设备（比如手机、平板电脑、个人数字助理等）结合，用于与经典语音业务一起的数据的发送与接收。这些网络通过划分为不同单元，而覆盖到极大的地理范围（These networks cover large geographical areas by splitting them into cells）。天线的布设经过详细考虑，从而确保对蜂窝单元的优化覆盖，及在用户从一个地方走到另一个地方时的单元间的无缝漫游。传统连通性类型也叫做 2G，2G 包括下面两种：

- 全球移动通信系统（Global System for Mobile Communications, GSM）
- 码分复用（Code Division Multiple Access, CDMA）

虽然从功能上，它们都被称作 2G 网络，但依据所使用的运营商及所居住的国家，可能用到 GSM 或 CDMA 类型的通信。这些网络被设计为使用电路交换的模拟连接，且最初并不是设计用于发送数据的。因为数据连接使用包交换技术，所以 2G 类型的连接，提供的是有限数据传输支持。

经由蜂窝网络、实现了全功能包交换与专属数据传输的新近连接类型，包括以下这些：

- 高速包访问（High Speed Packet Access, HSPA+）
- 长期演进（方案）（Long Term Evolution, LTE）

LTE 与 HSPA+ 是由第三代合作伙伴计划（3rd Generation Partnership Project, 3GPP）所创建的标准，该计划是数个判断它们需要某种在蜂窝网络上发送数据的标准化方式的电讯公司，之间的协作。

HSPA+ 是一个基于 CDMA 的、提供高达 84Mbps 下载速率与 22Mbps 上传速率的标准。LTE 则是基于 GSM/EDGE 的，提供高达 300Mbps 的下载与 75Mbps 上传速率的标准。

**注意：**这些标准都在持续开发中，因此它们的吞吐速率可能在将来会有所提升。

GSM 3G（第三代，third generation），是一个对具备提供可达数兆（several Mbps）传输速率的网络的泛称。可通过提升信道的分配带宽（the channels' allocated bandwidth, 请参考[这里](#)，及[这里](#)），并同时使用包交换技术，来达到这种传输速率。

GSM 4G (第四代, fourth generation), 则是GSM相关标准的最新补充 (the latest addition to the GSM portfolio) , 同时在大多数国家, 其仍处于部署阶段。4G 提供到超过 100Mbps 的、适合高速宽带互联网访问的传输速率。GSM 4G 完全基于IP通信, 且原先 3G 中所使用的扩频视频技术 (the spread spectrum radio technology) , 在 4G 中被正交频分多路复用多载波技术 (Orthogonal Frequency Division Multiplex Access multi-carrier, OFDMA) 所取代, 从而可确保更高的传输速率。

## 虚拟专用网技术 (VPN Technologies)

虚拟专用网是一种覆盖于通信网络之上, 给予到这些通信网络业务所需的安全性与可管理性的技术。在 VPN技术下, 在享受到低成本与互联网可用的同时, 还可建立起安全关系、自动连接、认证及加密等特性 (VPN is a technology that overlays communications networks and gives them the security and manageability required by businesses. With VPN technology, you can set up secure relationships, automated connections, authorizations, and encryption, while still enjoying the low cost and availability of the Internet) 。

虚拟专用网对跨越互联网传输的, 或公司内部范围的数据进行保护 (VPNs protect data while in transit across the Internet, or within a company's enclave) 。虚拟专用网有着多种能力, 但其主要功能包括这些:

- 保持数据机密 (经由加密实现, Keep data confidential(encryption))
- 确保通信双方身份可靠 (经由认证实现, Ensure the identities of two parties communicating(authentication))
- 保护通信各方的身份信息 (经由隧道化实现, Safeguard the identities of communicating parties(tunnelling))
- 确保数据是准确的, 且以其最初形式呈现 (具有不可抵赖性, Ensure data is accurate and in its original form(non-reudiation))
- 防止数据包被反复发送 (可防止回放, Guard against packets being sent over and over(replay prevention))

虽然虚拟专用网概念大多数时间默认就带有了安全性, 但仍存在不安全的虚拟专用网 (Even though the VPN concept implies security most of the time, unsecured VPNs still exists) 。帧中继就是不安全虚拟专用网的一个实例, 因为它提供了两个地点之间的专用通信, 但却可能在其上没有任何的安全特性。是否应将安全性添加到VPN连接, 取决于该连接的特定需求。

而因为在服务提供商设施中缺乏可见性, VPN的故障排除难于进行 (VPN troubleshooting is difficult to manage because of the lack of visibility in the service provider infrastructure) 。通常将服务提供商视为聚合了全部网络地点的连接的云。在执行VPN的故障排除时, 应首先确定故障不在自己的设备上, 随后才联系服务提供商。

虚拟专业网的类型有很多, 包括下面这些:

- 站点到站点的VPNs, 或内部网VPNs (Site-to-Site VPNs, or Intranet VPNs) , 比如覆盖式VPN (如帧中继, overlay VPN (like Frame Relay)) 或对等点到对等点VPN (如同多协议标签交换, Peer-to-Peer VPNs(like MPLS)) 。在将不同地点经由公共设施进行连接时, 必须使用这些类型。在使用对等点到对等点设施时, 可在站点之间无缝通信, 而不必担心IP地址分配的重复。
- 远程访问VPNs (Remote Access VPNs) , 比如虚拟专用的拨号网络 (Virtual Private Dial-up Network, VPDN) , 是一种通常考虑到安全性的VPN的拨号方式 (which is a dial-up approach for the VPN that is usually done with security in mind) 。
- 外部网VPNs (Extranet VPNs) , 在要连接到业务伙伴或客户的网络时, 需要使用此种VPN。

在使用VPNs时，就通常是将流量进行隧道化处理，以将其经由某项设施加以发送（When you use VPNs, you are often tunnelling traffic in order to send it over an infrastructure）。一种 Layer 3 的隧道化方法，被叫做通用路由封装（Generic Routing Encapsulation, GRE）。通用路由封装实现了流量的隧道传输，但其并不提供安全性。为了在对流量进行隧道化传输的同时提供到安全性，可使用一种名为IP安全（IP Security, IPsec）的技术。IPsec 是 IPv6 的一项强制实现的组件，但对 IPv4 来说却不是。IPsec 同时与认证、授权与计费（Authentication, Authorisation and Accounting, AAA）服务一同使用，实现对用户行为的追踪。

VPNs带来的主要好处如下：

- 可伸缩性（可将更多站点持续加入到VPN，Scalability(you can contiously add more sites to the VPN))
- 灵活性（可使用如MPLS这样的非常灵活的技术，Flexibility(you can use very flexible technologies like MPLS))
- 成本低（可以较低代价，经由互联网实现流量的隧道化传送，Cost(You can tunnel traffic through the Internet without much expense))

## 多协议标签交换技术（Multiple Protocol Label Switching, MPLS）

多协议标签交换，是通过将一个标签追加到任意类型的数据包上，而运作的（Multiprotocol Label Switching(MPLS) functions by appending a label to any type of packet）。随后数据包就根据该标签的值，而非任何 Layer 3 信息，经由网络设施得以转发。给数据包打上标签，提供了非常高效的转发，且令到 MPLS 可工作在极大范围的现有技术上。通过简单地将一个标签添加到数据包头部中，MPLS 就可在许多物理与数据链路层的广域网实现中使用（The labeling of the packet provides very efficient forwarding and allows MPLS to work with a wide range of undelying technologies. By simply adding a label in the packet header, MPLS can be used in many Physical and Data Link Layer WAN implementations）。

MPLS的标签，是放在 Layer 2 头部与 Layer 3 头部之间的。使用MPLS技术，仅在数据包进入服务提供商云时，才会加入额外开销。在进入MPLS网络后，相比传统的 Layer 3 网络，数据包交换的完成要快得多，因为MPLS的包交换只是基于MPLS标签的交换，而不是要拆封整个的 Layer 3 头部（By using MPLS, overhead is added only when the packet enters the service provider cloud. After entering the MPLS network, packet switching is done much faster than in traditional Layer 3 networks because it is based only on swapping the MPLS label, instead of stripping the entire Layer 3 header）。

MPLS有两种不同样式（MPLS comes in two different flavours）：

- 帧模式的MPLS（Frame Mode MPLS）
- （数据）单元模式的MPLS（Cell Mode MPLS）

帧模式的MPLS是最为流行的MPLS类型，而在此场景中，标签是放在 Layer 2 头部与 Layer 3 头部之间的（因此MPLS通常被视为一种 Layer 2.5 的技术）。单元模式的MPLS用在 ATM 网络中，并使用 ATM 头部中的一些字段，作为标签。

兼容MPLS的路由器（MPLS-capable routers），也被叫做标签交换路由器（Label Switched Routers, LSRs），同时这些路由器也有两种样式：

- 边沿标签交换路由器（服务提供商边沿路由器，Edge LSR(PE routers)）
- 服务提供商标签交换路由器（P(Provider) LSR）

**PE routers**（服务提供商边沿路由器），是那些关注标签分布的服务提供商边沿设备（**PE routers are Provider Edge devices that take care of label distribution**）；它们根据标签对数据包进行转发，并负责标签的插入与移除。**P routers** 就是服务提供商路由器，它们的职责包括 标签式转发，以及基于标签的高效率包转发（**P routers are Provider routers and their responsibility consists of label forwarding and efficient packet forwarding based labels**）。

注：请参考[这里](#)。

## 基本的串行线路配置 (Basic Serial Line Configuration)

在不打算改变默认的 **HDLC**（High-level Data Link Control，高级数据链路控制，思科专有）封装时，那么为建立WAN连接，仅需完成下面的步骤：

1. 给接口添加一个IP地址
2. 开启接口（以 `no shutdown` 命令）
3. 确保在数据通信设备侧有一个时钟速率（*Ensure there is a clock rate on the DCE side*）

在连接了数据通信设备电缆时的配置如下：

```
Router#config t
Router(config)#interface Serial0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#clock rate 64000
Router(config-if)#no shutdown
Router(config-if)#^Z
Router#
```

## 以太网上的点对点协议 (Point-to-Point over Ethernet, PPPoE)

以太网上的点对点协议，是一个用于在以太网帧内部，封装点对点协议帧的网络协议（*Point-to-Point Protocol over Ethernet(PPPoE) is a network protocol used to encapsulate PPP frames inside Ethernet frames*）。

要实现客户部署非对称数字订户线路，他们就必须支持在极大安装基数的老旧桥接的客户处设备上的点对点样式的认证与授权。PPPoE 技术提供了将主机网络经由简单的桥接访问设备，连接到远端访问集中器，或聚合集中器的能力（*As customers deploy ADSL, they must support PPP-style authentication and authorisation over a large installed base of legacy bridging customer premises equipment(CPE)*）。PPPoE provides the ability to connect a network of hosts over a simple bridging access device to a remote access concentrator or aggregation concentrator）。在此模型下，每台主机都使用其自身的点对点协议栈，因此呈现给用户的是一个熟悉的用户界面。访问控制、计费与服务类型（*type of service*），可基于每名用户，而不是基于每个地点完成。

如同在[RFC 2516](#)中所指明的那样，PPPoE有两个不同阶段：发现阶段与会话阶段（*As specified in RFC 2516, PPPoE has two distinct stages: a discovery stage and a session stage*）。在主机发起一个PPPoE会话时，其必须首先进行发现，以找到可满足客户端请求的服务器，并找到对等点的以太网MAC地址而建立一个PPPoE会话ID。在PPP定义一个对等点到对等点的关系时，发现本质上就是一个客户端服务器的关系（*While PPP defines a peer-to-peer relationship, discovery is inherently a client-server relationship*）。

## PPPoE的配置

下面的小节涵盖了服务器（互联网服务提供商处）与客户端PPPoE的配置。之所以包含此内容，是因为现在CCNA大纲强制要求考生知道如何配置PPPoE。

### 服务器的配置

创建PPPoE服务器配置的第一步，是定义一个将对传入连接进行管理的宽带聚合组（broadband aggregation group, BBA group）。该宽带聚合组必须关联到某个虚拟模板：

```
Router(config)#bba-group pppoe GROUP
Router(config-bba-group)#virtual-template 1
```

下一步为面向客户端的接口，创建出一个虚拟模板。在虚拟模板上，需要配置一个IP地址以及一个可从中为客户端分配到协商地址的地址池（The next step is to create a virtual template for the customer-facing interface. On the virtual template you need to configure an IP address and a pool of address from which clients are assigned a negotiated address）：

```
Router(config)#interface virtual-template 1
Router(config-if)#ip address 10.10.10.1 255.255.255.0
Router(config-if)#peer default ip address pool POOL
```

该IP地址池是在全局配置模式中定义的。这与DHCP地址池的配置类似：

```
Router(config)#ip local pool POOL 10.10.10.2 10.10.10.254
```

最后一步就是在面向客户端的接口上开启该PPPoE分组：

```
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#pppoe enable group GROUP
Router(config-if)#no shutdown
```

## 客户端的配置 (Client Configuration)

在客户端侧上，必须创建出一个拨号器接口（On the client side a dialer interface has to be created）。拨号器接口将对PPPoE连接进行管理。可将手动IP地址分配给拨号器接口，或将其设置为从服务器请求一个IP地址（使用 `ip address negotiated` 命令）：

```
Router(config)#interface dialer1
Router(config-if)#dialer pool 1
Router(config-if)#encapsulation ppp
Router(config-if)#ip address negotiated
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#pppoe-client dial-pool-number 1
Router(config-if)#no shutdown
```

## 关于认证 (Authentication)

为了令到PPPoE连接安全，可使用两种方法：

- 口令认证协议 (Password Authentication Protocol, PAP) - 不安全的、以明文方式发送凭据 (包含用户名与口令)
- 询问握手协议 (Challenge Handshake Authentication Protocol, CHAP) - 安全的 (明文的用户名与经 MD5 散列化的口令)，是首选方式

可如下配置 PAP：

服务器侧：

```
Server(config)#username Client password Password
Server(config)#interface virtual-template 1
Server(config-if)#ppp authentication pap
Server(config-if)#ppp pap sent-username Server password Password
```

客户端：

```
Client(config)#username Server password Password
Client(config)#interface dialer 1
Client(config-if)#ppp authentication pap
Client(config-if)#ppp pap sent-username Client password Password
```

CHAP 可如下进行配置：

服务器侧：

```
Server(config)#username Client password Password
Server(config)#interface virtual-template 1
Server(config-if)#ppp authentication chap
```

客户端：

```
Client(config)#username Server password Password
Client(config)#interface dialer 1
Client(config-if)#ppp authentication chap
```

## PPPoE的验证与故障排除 (PPPoE Verification and Troubleshooting)

在PPPoE会话成功形成后，客户端控制台上将出现下面的消息：

```
%DIALER-6-BIND: Interface Vi1 bound to profile Di1
%LINK-3-UPDOWN: Interface Virtual-Access1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access1, changed state to up
```

在客户端路由器上使用下面的命令，可对拨号器接口，以及从PPPoE服务器处获取到的（协商到的）IP地址进行检查：

```
Router#show ip interface brief
Interface                  IP-Address      OK? Method Status          Protocol
Virtual-Access1            unassigned     YES unset  up/up
Dialer1                   10.10.10.2   YES IPCP   up/up
```

在客户端路由器上可使用下面的命令，显示出PPPoE会话的状态：

```
Router#show pppoe session
1 client session
  Uniq ID  PPPoE  RemMAC          Port      Source   VA        State
                SID  LocMAC
                VA-st
  N/A       16  ca00.4843.0008  Fa0/0      Di1     Vi1      UP
                           ca01.4843.0008
```

一些对于PPPoE连接进行故障排除有用的命令如下：

```
Router#debug ppp ?
 authentication  CHAP and PAP authentication
 bap              BAP protocol transactions
 cbcpc           Callback Control Protocol negotiation
 elog             PPP ELOGS
 error            Protocol errors and error statistics
 forwarding       PPP layer 2 forwarding
 mppe             MPPE Events
 multilink        Multilink activity
 negotiation      Protocol parameter negotiation
 packet           Low-level PPP packet dump
```

## WAN连接的故障排除（Troubleshooting WAN Connections）

在试图启动一条广域网连接（现在先不管PPP与帧中继连接）时，可运用开放系统互联模型：

**Layer 1** -- 对线缆进行检查，以确保其连接正确。其外还要检查一下有没有执行 no shutdown 命令，以及在数据通信设备侧有没有应用一个时钟速率。

```
RouterA#show controllers serial 0
HD unit 0, idb = 0x1AE828, driver structure at 0x1B4BA0
buffer size 1524 HD unit 0, V.35 DTE cable

RouterA#show ip interface brief
Interface      IP-Address      OK? Method Status          Protocol
Serial0        11.0.0.1        YES unset  administratively down down
Ethernet0      10.0.0.1        YES unset  up                  up
```

**Layer 2** -- 检查以确保对接口应用了正确的封装。确保链路的另一侧有着同样的封装类型。

```
RouterB#show interface Serial0
Serial1 is down, line protocol is down
Hardware is HD64570
Internet address is 12.0.0.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
```

**Layer 3** -- IP地址与子网掩码对不对，子网掩码与另一侧是不是匹配。

```
RouterB#show interface Serial0
Serial1 is down, line protocol is down
Hardware is HD64570
Internet address is 12.0.0.1/24
MTU 1500 bytes, BW 1544 Kbit, DLY 1000 usec, rely 255/255, load 1/255
Encapsulation HDLC, loopback not set, keepalive set (10 sec)
```

## 第41天问题

1. Name at least three WAN categories.
2. The need for NBMA appears when there is no native \_\_\_\_\_ support for a group of systems that want to communicate over the same network.
3. In NBMA environments you still need to bind the Layer 3 address (IP address) to the Layer 2 address (DLCI). This can be done in an automated fashion, using a technology called Inverse ARP. True or false?
4. Name 2 NBMA interface types.
5. \_\_\_\_\_ requires DTE and DCE and is the default encapsulation type on Cisco routers.
6. \_\_\_\_\_ technologies involve the use of carrier Ethernet in Metropolitan Area Networks (MANs).
7. T1 is a standard often used in what geographical regions?
8. What are the two flavours of ISDN?
9. \_\_\_\_\_ is the most common form of DSL connection that functions over standard telephone lines. It offers unequal download and upload throughput, with the download rate being higher than the upload rate.
10. \_\_\_\_\_ functions by appending a label to any type of packet.

## 第41天答案

1. Circuit-switched, cell-switched, broadband, leased-line, and packet-switched.
2. Broadcast.
3. True.
4. Multipoint and Point-to-Point.
5. HDLC.
6. Metro Ethernet.
7. North America, Japan, and South Korea.
8. BRI and PRI.
9. ADSL.
10. MPLS.

## 第41天实验

### PPPoE实验

在两台路由器之间，以本课程模块中所给出的信息，配置带有CHAP的PPPoE：

#### 服务器配置：

```
Router(config)#bba-group pppoe GROUP
Router(config-bba-group)#virtual-template 1
Router(config)#interface virtual-template 1
Router(config-if)#ip address 10.10.10.1 255.255.255.0
Router(config-if)#peer default ip address pool POOL
Router(config)#ip local pool POOL 10.10.10.2 10.10.10.254
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#pppoe enable group GROUP
Router(config-if)#no shutdown
```

**客户端配置:**

```
Router(config)#interface dialer1
Router(config-if)#dialer pool 1
Router(config-if)#encapsulation ppp
Router(config-if)#ip address negotiated
Router(config)#interface FastEthernet0/0
Router(config-if)#no ip address
Router(config-if)#pppoe-client dial-pool-number 1
Router(config-if)#no shutdown
```

**询问握手认证协议 (CHAP) 配置:**

```
Server(config)#username Client password Password
Server(config)#interface virtual-template 1
Server(config-if)#ppp authentication chap
Client(config)#username Server password Password
Client(config)#interface dialer 1
Client(config-if)#ppp authentication chap
```

**对配置进行验证:**

```
Router#show pppoe session
1 client session
  Uniq ID  PPPoE  RemMAC      Port      Source   VA       State
                SID  LocMAC
                           VA-st
  N/A        16  ca00.4843.0008  Fa0/0      Di1     Vi1      UP
                           ca01.4843.0008
```

请访问[www.in60days.com](http://www.in60days.com)并自由观看作者完成该实验。

# 第42天 帧中继与点对点协议

## Frame Relay and PPP

---

Gitbook: [ccna60d.xfoss.com](http://ccna60d.xfoss.com)

你可以在 <https://github.com/gnu4cn/ccna60d> 上 fork 本项目，并提交你的修正。

本书结合了学习技巧，包括阅读、复习、背书、测试以及 hands-on 实验。

本书译者用其业余时间完成本书的翻译工作，并将其公布到网上，以方便你对网络技术的学习掌握，为使译者更有动力改进翻译及完成剩下章节，你可以 [捐赠译者](#)。

---

## 第 42 天任务

- 阅读今天的课文（下面）
- 复习昨天的课文
- 完成今天的实验
- 阅读ICND2的记诵指南
- 在[subnetting.org](http://subnetting.org)网站上花15分钟

多年来，帧中继都是CCNA甚至CCIE大纲的重要部分；但由于公司数字订户线路的广泛可用与长租专线的价格越来越亲民，从而导致帧中继技术的流行度近来日渐式微。这里之所以要涉及，是因为其包含在CCNA大纲中。点对点协议仍有广泛使用。

今天将学到以下内容：

- 帧中继的运作 (Frame Relay operations)
- 帧中继的配置
- 帧中继的故障排除
- 点对点协议的运作
- 点对点协议的配置
- 点对点协议的故障排除

本课程对应了以下CCNA大纲要求：

- 识别不同的广域网技术
  - 帧中继技术
- 配置并验证思科路由器之间的点对点协议

## 帧中继的运作

### Frame Relay Operations

帧中继是基于较早的名为 x.25 协议的一个 Layer 2 广域网协议，x.25 协议因为其全面的错误检查能力，也仍被ATM技术所使用（which is still used by ATMs due to its extensive error-checking capabilities）。帧中继由一条其上可形成许多逻辑电路物理电路构成。帧中继的连接是按需建立的。下图演示了一个帧中

继网络的实例：

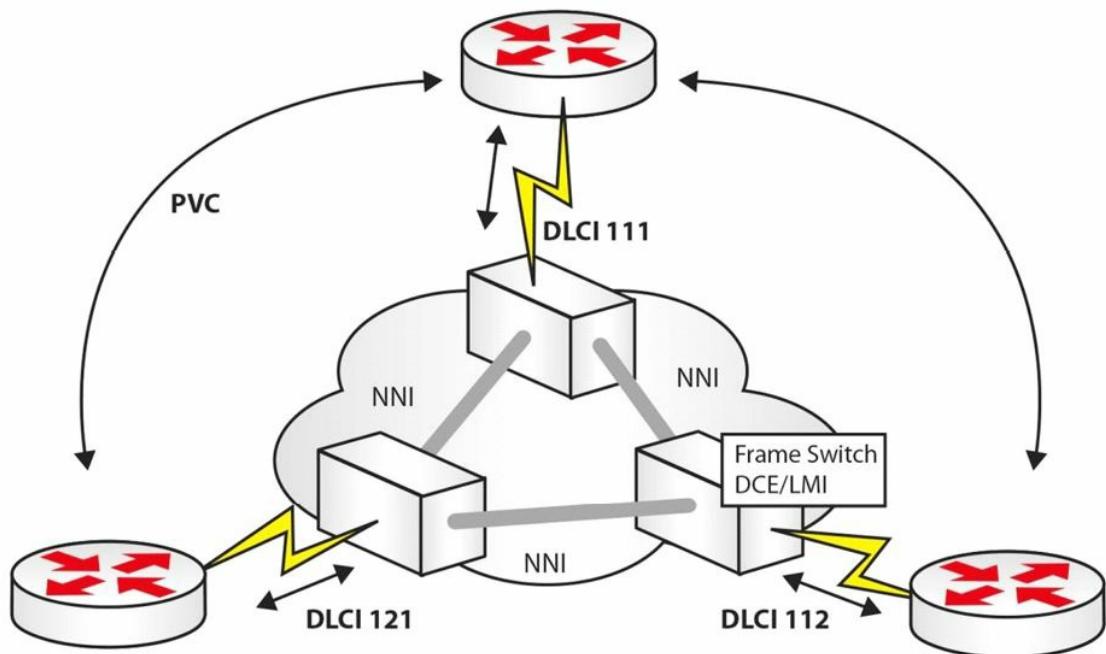


图 42.1 - 一个帧中继网络

## 常见的帧中继术语

Common Frame Relay Terms

## 本地管理接口（Local Management Interface, LMI）

本地管理接口是运行在帧中继交换机上的一个保活（机制）（Local Management Interface (LMI) is a keepalive which runs from the Frame Relay switch）。帧中继交换机属于服务提供商，位于服务提供商处。如未使用思科默认的类型，那么就要在自己的路由器上需要指定本地管理接口的类型。本地管理接口有三种可用的类型，如下所示：

- 思科（默认）类型
- ANSI (America National Standard Institution, 美国国家标准学会) 类型
- Q933a类型 (ITU Telecommunication Standardization Sector, 简写ITU-T, 国际电信联盟电信标准化部门, Q.933 Annex A standard, [wikipedia: Local Management Interface](#) )

下图42.2演示了这些本地管理接口：

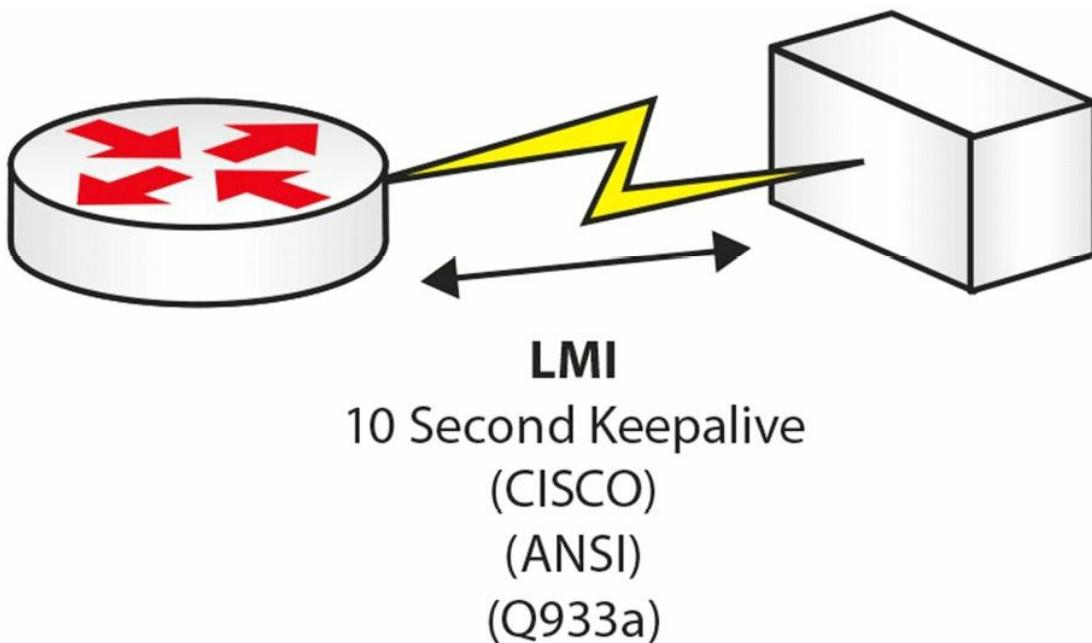


图 42.2 - 本地管理接口的类型

在帧中继连接出现错误时，那么对本地管理接口消息的调试，就将是故障排除步骤的其中一步，如下面的输出所示：

```
RouterA#debug frame-relay lmi

00:46:58: Serial0(in): Status, myseq 55
00:46:58: RT IE 1, length 1, type 0
00:46:58: KA IE 3, length 2, yourseq 55, myseq 55
00:46:58: PVC IE 0x7 , length 0x6 , dlci 100, status 0x2 , bw 0
```

本地管理接口每 10 秒发出，且所有第六个报文为一个完整状态更新。如上所示，希望本地管理接口报告 `status 0x2`，表示这是一条活动的链路（An LMI is sent every 10 seconds, and every sixth message is a full status update. As above, you want it to report `status 0x2`, which is an active link）。

## 永久虚拟电路 (Permanent Virtual Circuit, PVC)

永久虚拟电路，是自帧中继网络的一端，到另一端所形成的逻辑端对端连接，如下图42.3所示（A Permanent Virtual Circuit(PVC) is the logical end-to-end connection formed from one end of your Frame Relay network to the other, as illustrated in Figure 42.3 below）。每个端点都被赋予到一个数据链路连接标识符编号（a Data Link Connection Identifier, DLCI, number, 请参阅下一小节），以对其进行标示。

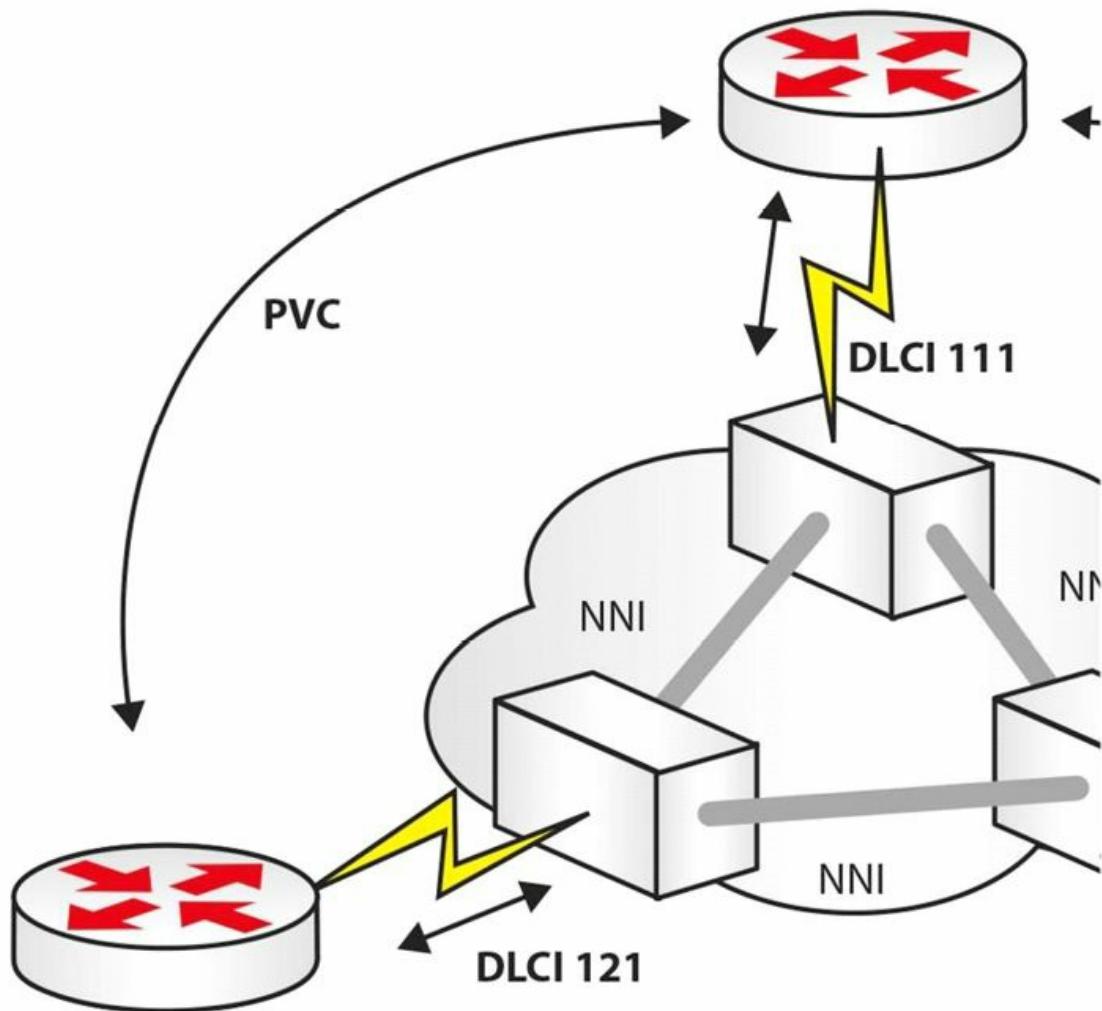


图 42.3 - 永久虚拟电路

注：NNI: Network-to-Network Interface, 网络到网络接口, 参考[wikipedia: NNI](#)。

## 数据链路连接标识符 (Data Link Connection Identifier, DLCI)

数据链路连接标识符，是一个本地有意义的编号，用于标识到帧中继交换机的连接，如下图42.4所示。该编号可为 10 到 1007 之间的任意数字，包括了 10 与 1007。

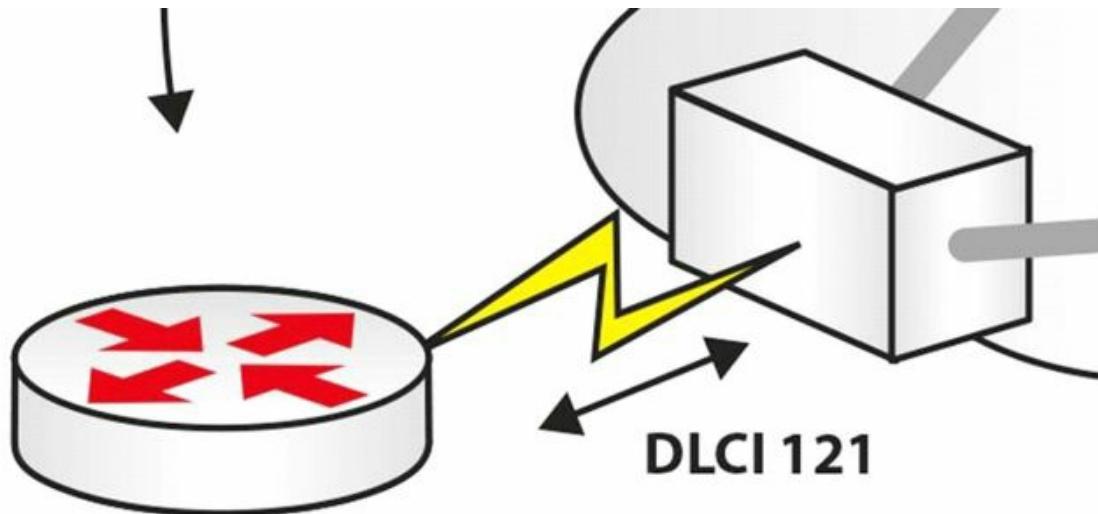


图 42.4 - 数据链路连接标识符将用户路由器标识给电信公司 (DLCI Identifies Your Router to the Telco)

通常在对帧中继链路进行故障排除时，故障在于客户或服务提供商，在它们的配置上使用了错误的数据链路连接标识符编号（Often, when troubleshooting Frame Relay links, the issue lies with either the customer or the service provider using the wrong DLCI number on their configuration）。

当数据链路连接标识符为活动状态时，那么端到端连接将按以下步骤形成（When your DLCI is active, an end-to-end connection forms in the following order）：

1. 活动的DLCI发出反向地址解析协议请求（Active DLCI sends Inverse ARP request）
2. DLCI等待带有网络地址的应答（DLCI waits for reply with network address）
3. 远端路由器地址的映射建立起来（Map created of remote router address）
4. DLCI经历 Active/Inactive/Deleted 状态（DLCI status of Active/Inactive/Deleted）

## 网络到网络接口 (Network-to-Network Interface, NNI)

网络到网络接口，是帧中继交换机之间的连接。

## 关于帧中继技术 (Frame Relay Technology)

帧中继是一种非广播多路访问（Non-Broadcast Multi-Access, NBMA）技术。这就意味着必须应付地址解析的问题，除非在使用点对点接口的情形下（Frame Relay is a Non-Broadcast Multi-Access(NBMA) technology. This means that you have to deal with address resolution issues, except for the situations in which you use Point-to-Point interfaces）。

帧中继中的 Layer 2 地址，被称作数据链路连接标识符（Data Link Connection Identifier, DLCI），而这是本地有意义的。比如在轴辐（hub-and-spoke, 中心分支）环境中，中心设备应有着与其各个分支进行通信的唯一DLCI（For example, in a hub-and-spoke environment, the hub device should have a unique DLCI to communicate to each of its spokes）。

在对思科设备上的帧中继永久虚拟电路状态进行检查时，将看到一个由本地管理接口所定义的状态代码，该代码可以是下列的任意一种：

- 活动状态（Active，全都没有问题）
- 不活动状态（Inactive，本地节点上没有问题，但在远端节点上可能有故障）
- 已被删除（Deleted，服务提供商网络中存在问题）

举例来说，思科设备提供了三种口味本地管理接口（As an example, Cisco device offer three flavours of LMI）：

- CISCO，思科默认的LMI
- ANSI，美国国家标准学会LMI
- Q.933a，国际电信联盟电信标准委员会LMI

思科路由器已被配置为自动尝试所有这三种的LMI类型（从 cisco LMI开始），并使用与服务提供商匹配的那个类型，因此在设计阶段，有关LMI类型方面无需过多考虑（Cisco routers are configured to automatically try all three of these LMI types(starting with cisco LMI) and use the one that matches whatever the service provider is using, so this should not be of much concern in the design phase）。

在设计阶段需要考虑的最重要方面之一，就是要用到的地址解析方法。如设计中用到多点接口（也就是可端接多个 Layer 2 电路的接口），那么就需要找到某种提供 Layer 3 到 Layer 2 解析的方式（One of the most important aspects that need to be considered in the design phase is the address resolution methodology used. If you are utilising Multipoint interfaces in your design(i.e., interfaces that can terminate multiple Layer 2 circuits), you need to find a way to provide the Layer 3 to Layer 2 resolution）。如同先前所讨论的，有两个选项可帮助实现三层到二层的解析：

- 动态地，使用反向地址解析协议（Dynamically, utilising Inverse ARP）
- 静态地，通过在思科设备上的 frame-relay map 静态配置命令（Statically, via the frame-relay map static configuration command on Cisco devices）

**注意：**为检查 Layer 3 到 Layer 2 的成功解析，可使用 show frame-relay map 命令。

在多点接口（a Multipoint interface）上，反向ARP将自动发生。此功能将于给配置为帧中继的接口添加IP地址后，立即启用。在给配置为帧中继的接口添加IP地址那一刻，所有该接口所运行的、被支持的协议的反向ARP请求，就开始从分配到那个特定接口的所有电路上发出（On a Multipoint interface, Inverse ARP would happen automatically. This functionality is enabled right after adding an IP address on an interface configured for Frame Relay. At that moment, requests start being sent out all of the circuits assigned to that specific interface for any supported protocol the interface is running）。

该自动请求过程可通过 no frame-relay inverse-arp 命令关闭，但不能设计一个停止对请求进行响应的网络。经由设计，是无法关闭反向ARP应答的，因此帧中继设备总是会通过帧中继反向ARP，尝试协助其它尝试进行 Layer 3 到 Layer 2 解析的设备（The request process can be disabled with the no frame-relay inverse-arp command, but you can never design a network that will stop responding to requests. By design, Inverse ARP replies cannot be disabled, so the Frame Relay speaker will always attempt to assist somebody that attempts to do a Layer 3 to Layer 2 resolution via Frame Relay Inverse ARP）。

帧中继设计中的反向地址解析协议行为，将自动协助先前讨论过的经由重复单播方法的广播（The Inverse ARP behaviour in the Frame Relay design will automatically assist with Broadcast through the replicated Unicast approach discussed before）。在使用反向ARP时，广播支持默认就有。

在将两台路由器经由物理接口连接到帧中继云时，就意味着从帧中继角度讲，那些特定接口就是多点的了，因为默认物理帧中继接口就是多点结构。就算两台路由器之间的连接可能看起来是点对点的，但该连接仍是帧中继的多点连接（If you connect two routers to the Frame Relay cloud using physical interfaces, this means that the specific interfaces are Multipoint from a Frame Relay perspective, because a physical Frame Relay interface by default is a Multipoint structure. Even though the connection between the two routers may appear to be Point-to-Point, it is a Frame Relay Multipoint connection）。

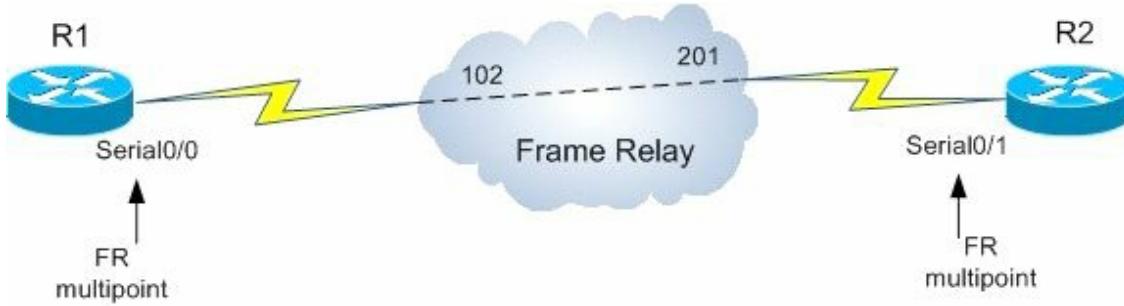


图 42.5 - 帧中继多点实例

因为两台路由器使用多点接口，所以默认这两台设备将通过使用反向ARP动态处理三层到二层的解析。如打算设计不使用反向ARP的方案，就可在各台设备上关闭动态映射行为，并于随后配置上静态的帧中继映射。

对于三层到二层的解析，点对点配置就是理想的选择，因为解析过程在使用这类接口时不会发生。在配置点对点帧中继时，可使用点对点子接口，而这些子接口不会从LMI获取到DLCI编号分配，这就与多点情况一样（Point-to-point configurations are the ideal choice when it comes to Layer 2 to Layer 2 resolution because this process does not occur when using such interface types. When configuring Point-to-Point Frame Relay, you would use Point-to-Point subinterfaces and these subinterfaces would not get the DLCI assignments from LMI, like in the Multipoint situation）。

另一选项将是创建子接口，并将这些创建出来的子接口声明为多点。这类接口将与物理多点接口一样运作，因此需确定要使用的解析方法，也就是反向ARP或静态映射。也可使用两种解析方法的结合，比如在连接的一端部署反向ARP，并在另一端定义静态映射（Another option would be to create subinterfaces and declaring them as Multipoint. These types of interfaces behave exactly like the physical Multipoint interfaces, so you need to decide on the resolution method to be used, either Inverse ARP or static mappings. A combination of these methods can be used, for example, by implementing Inverse ARP on one end of the connection and defining static maps on the other end）。

接口类型设置与所选的三层到二层解析方法仅本地有意义。这意味着在帧中继设计中可以有各种变化（The interface type settings and the selected Layer 3 to Layer 2 resolution method is only locally significant. This means that you can have all kinds of variations in your Frame Relay design），比如下图表42.1中所列出的这些：

表 42.1 - 帧中继设计中的各种组合

本地接口	连接到	远端接口
主接口 (Main interface)		主接口 (Main interface)
主接口 (Main interface)		点对点接口 (Point-to-Point interface)
主接口 (Main interface)		多点接口 (Multipoint interface)
多点接口 (Multipoint interface)		点对点接口 (Point-to-Point interface)
多点接口 (Multipoint interface)		多点接口 (Multipoint interface)
点对点接口 (Point-to-Point interface)		点对点接口 (Point-to-Point interface)

部分网状网络的设计与配置，将是极具挑战性的。部分网状网络就意味着在所有涉及帧中继环境的端点之间，并不会全都提供二层电路（Partial-mesh designs and configurations will be the most challenging. This implies that Layer 2 circuits will not be provisioned between all endpoints involved in the Frame Relay

environment)。

在轴辐（中心-分支， hub-and-spoke）环境中，分支之间没有直接相连，因此就意味着它们无法通过反向 ARP进行彼此解析。为解决这些问题，可执行以下措施：

- 提供额外的静态映射（Provide additional static mappings）
- 配置点对点的子接口（Configure Point-to-Point subinterfaces）
- 对轴辐设施加以设计，使得三层路由的设计可解决解析的问题（比如通过使用OSPF的点对多点网络类型， Design the hub-and-spoke infrastructure so that the Layer 3 routing design can solve the resolution problems(e.g., by using the OSPF Point-to-Multipoint network type)）

帧中继支持可对服务质量（Quality of Service, QoS）施加影响的标记。比如，帧中继头部就包含了一个丢弃资质为（a DE(Discard Eligible) bit）。对于QoS的帧中继环境，数据包可藉由丢弃资质位加以标记，而这就告诉服务提供商那些特定数据包不是非常重要，在壅塞时可被丢弃。这样做将令到那些没有设置丢弃资质位的数据包优先。

在帧中继环境中可配置其它参数，就是向前显式壅塞通知与向后显式壅塞通知（Forward Explicit Congestion Notifications(FECNs) and Backward Explicit Congestion Notifications(BECNs)），这通常会是一个突如其来的考试问题（which commonly crops up as an exam question）。帧中继设备在配置了 FECNs或BECNs时，就可通知壅塞设备，并可导致发送速率的下降（The Frame Relay equipment, if configured to do so, can notify devices of congestion and can cause the slowing down of the sending rates）。

## 配置帧中继（Configuring Frame Relay）

不幸的是，配置帧中继较为棘手，这是因为不同的网络类型，要求不同的命令（Unfortunately, it can be somewhat tricky to configure Frame Relay, and this is because different network types require different commands）。这一点的原因，在于要解决WAN上网络地址解析方法，以及路由协议如何运作的问题。配置帧中继的步骤如下所示：

1. 设置封装方式（Set encapsulation）
2. 设置本地管理接口类型（可选的， Set LMI type(optional)）
3. 配置静态/动态地址映射（Configure static/dynamic address mapping）
4. 解决特定于协议的一些问题（Address protocol-specific problems）

CCNA考试不要求考生知道如何配置电信级帧中继交换机。只有在对家里或远程实验室中自己的帧中继连接进行配置时，才想要知道如何完成帧中继交换机的配置。

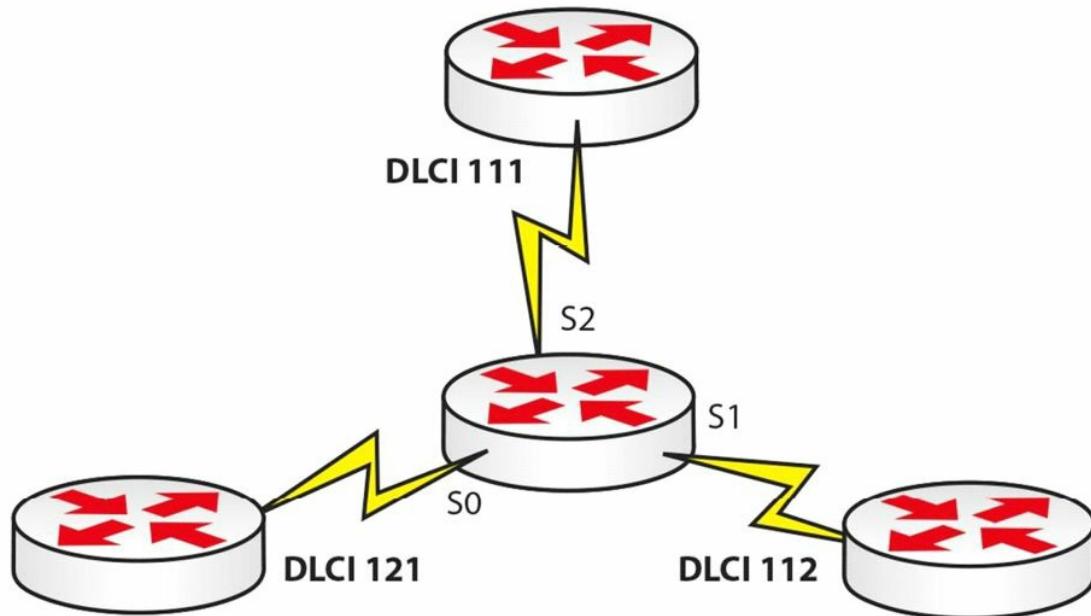


图 42.6 - 帧中继网络

对于上面的网络拓扑，将在中间的帧中继交换机上进行下面的配置。请仅将此信息用作参考，因为考试并不要求：

```

Router#conf t
Router(config)#frame-relay switching
Router(config)#int s0
Router(config-if)#clock rate 64000
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay intf-type dce
Router(config-if)#frame-relay route 121 interface s1 112
Router(config-if)#frame-relay route 121 interface s2 111
Router(config-if)#no shut
Router(config-if)#int s1
Router(config-if)#clock rate 64000
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay intf-type dce
Router(config-if)#frame-relay route 112 interface s0 121
Router(config-if)#frame-relay route 112 interface s2 111
Router(config-if)#int s2
Router(config-if)#clock rate 64000
Router(config-if)#encapsulation frame-relay
Router(config-if)#frame-relay intf-type dce
Router(config-if)#frame-relay route 111 interface s0 121
Router(config-if)#frame-relay route 111 interface s1 112
Router(config-if)#no shut
Router#show frame-relay route

```

## 帧中继故障排除

如早先指出的那样，在电信公司将你的DLCI映射到错误端口，或其得到的编号是错误的时候，他们常常会搞错映射信息。此时就需要在给电信公司打电话或保修之前，证实他们搞错了，那么就要用到下面的命令：

- `show frame-relay pvc`
- `show frame-relay lmi`

- show frame-relay map
- debug frame-relay pvc
- debug frame-relay lmi

## 帧中继的各种错误

令人费解的是，考试中有的时候会问一些有关帧中继链路上所报出错误的问题，因此需要知道：

- BECN -- 在帧传输相反方向中的帧经历了拥塞
- FECN -- 在帧传输方向上经历了拥塞

## 点对点协议的运作

由于下面这些因素，点对点协议（Point-to-Point Protocol，PPP）被认为是一种互联网友好的协议：

- 其支持数据压缩
- 内建了认证（PAP 及 CHAP）
- 网络层的地址协商（Network Layer address negotiation）
- 错误侦测能力

在包括下面这些多种连接类型上都可以使用点对点协议：

- DSL
- ISDN
- 各种同步与异步链路
- HSSI

点对点协议可拆分为以下的二层子层（Layer 2 sublayers）：

- NCP -- 建立网络层协议（为网络层服务，establishes Network Layer protocols(serves the Network Layer)）
- LCP -- 链路建立、链路认证以及对链路质量进行测试（服务物理层，estalishes, authenticates, and tests link quality）
- HDLC -- 对链路上的数据报进行封装

掌握上面这些知识，将在CCNA考试中大有裨益！

## 点对点协议的配置

如同下图 42.7 及下面的输出那样，点对点协议是非常容易配置的。稍后还将演示如何为点对点协议加上认证。



图 42.7 -- 一个点对点协议的连接

```
R1#conf t
R1(config)#interface s0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#clock rate 64000
R1(config-if)#encapsulation ppp
R1(config-if)#no shut
R2#conf t
R2(config)#interface s0
R2(config-if)#ip add 192.168.1.2 255.255.255.0
R2(config-if)#encapsulation ppp
R2(config-if)#no shut
```

## 点对点协议的认证

点对点协议有着内建的口令认证协议（PAP, Password Authentication Protocol）或询问握手认证协议（CHAP, Challenge Handshake Authentication Protocol）形式的认证。口令认证协议将口令以明文形式在链路上发送，这有着安全风险，而询问握手认证协议则是发送使用了MD5加密的散列值。下面是询问握手认证协议的配置：

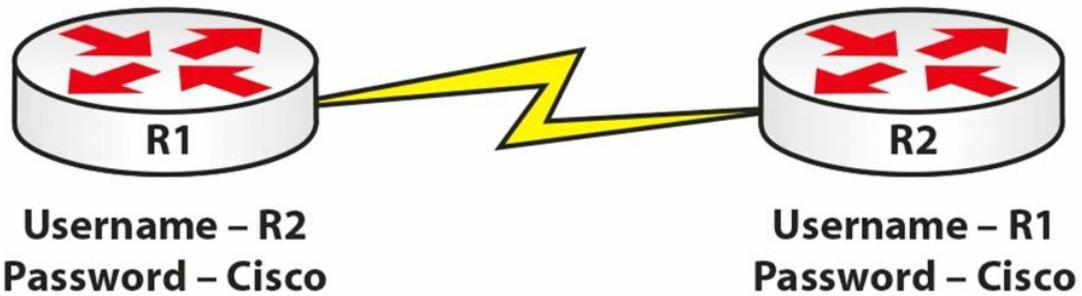


图 42.8 -- 带有询问握手认证协议的点对点协议

```
R1#conf t
R1(config)#username R2 password Cisco
R1(config)#interface s0
R1(config-if)#ip add 192.168.1.1 255.255.255.0
R1(config-if)#clock rate 64000
R1(config-if)#encapsulation ppp
R1(config-if)#ppp authentication chap
R1(config-if)#no shut

R2#conf t
R2(config)#username R1 password Cisco
R2(config)#interface s0
R2(config-if)#ip add 192.168.1.2 255.255.255.0
R2(config-if)#encapsulation ppp
R2(config-if)#ppp authentication chap
R2(config-if)#no shut
```

若要配置口令认证协议，可将上面配置中的 [chap] 关键字，替换为 [pap] 关键字。还可将点对点协议配置为尝试使用询问握手认证协议，在使用询问握手认证协议失败时，再尝试口令认证协议。这就是所谓的点对点协议回退特性（PPP fallback），下面就是配置此特性的命令：

```
R2(config-if)#ppp authentication chap ppp
```

## PPP的故障排除

执行一下 `show interface serial 0/0` 命令，或以其他相应的接口编号，以将IP地址、接口状态及封装类型等参数显示出来，如下面的输出所示：

```
RouterA#show interface Serial0/0

Serial0 is up, line protocol is up
  Hardware is HD64570
  Internet address is 192.168.1.1/30
  MTU 1500 bytes, BW 1544 Kbit, DLY 20000 usec,
    reliability 255/255, txload 1/255, rxload 1/255
  Encapsulation PPP, loopback not set
  Keepalive set (10 sec)
```

在使用了询问握手认证协议时，就要检查用户名是否与所呼叫的路由器是否匹配，同时记住这里的主机名是区分大小写的。使用 `debug ppp authentication` 与 `debug ppp negotiation` 两个命令来对点对点协议会话的建立进行故障排除。

## 第42天问题

1. Frame Relay is based on which older protocol?
2. What are the three types of LMIs available?
3. An LMI is sent every \_\_\_\_\_ seconds, and every \_\_\_\_\_ message is a full status update.
4. The DLCI number is only locally significant, so you could have a different one for the other end of your Frame Relay connection. True or false?
5. Explain the difference between BECNs and FECNs.
6. PPP does not include data compression or error detection. True or false?
7. Name the PPP sublayers.
8. Write out the command to configure CHAP with PPP.
9. Which command will show you the encapsulation type on your Serial interface?
10. \_\_\_\_\_ sends the passwords over the link in clear text, which poses a security risk, whereas \_\_\_\_\_ sends a hashed value using MD5 security.

## 第42天答案

1. The X.25 protocol.
2. CISCO, ANSI, and Q933a.
3. 10, sixth.
4. True.
5. Backward Explicit Congestion Notification (BECN): Frames in the direction opposite of the frame transmission experienced congestion; Forward Explicit Congestion Notification(FECN): Congestion was experienced in the direction of the frame transmission.
6. False.
7. NCP, LCP, and HDLC.
8. `ppp authentication chap`.
9. The `show interface serial [number]` command.
10. PAP, CHAP.

## 端口镜像操作（华为）

端口镜像可以是：本地端口镜像、远程端口镜像。

### 本地端口镜像

1. 设置观察端口：

```
SwitchA> sys  
SwitchA] observe-port 1 interface Ethernet 0/0/47 cr
```

上面的命令，将 `Ethernet 0/0/47` 设置为观察端口，其中的 `1` 是指观察索引号，只能设置 `4` 个的观察端口（Huawei Versatile Routing Platform Software(VRP) Version 5.70）。

1. 设置镜像端口

```
SwitchA> sys  
SwitchA] interface Ethernet 0/0/36  
SwitchA-Ethernet0/0/36] port-mirroring to observe-port 1 inbound/outbound/both cr
```

需要先进入接口模式，然后执行以上的 `port-mirroring` 命令，其中的 `1` 是第一步设置的观察索引号，`inbound/outbound/both` 是指：上传流量、下载流量与全部流量。

需要注意的是：在上面两个接口中，`Ethernet 0/0/47` 是接入监测设备的端口，而 `Ethernet 0/0/36` 是被监测的端口；检测设备网卡要打开混杂模式（`$sudo ifconfig enp2s0 promisc`，关闭混杂模式：`$sudo ifconfig enp2s0 -promisc`）；

### 远程的端口镜像

远程的端口镜像，是在本地端口镜像操作基础上，将观察端口（也就是上述本地端口镜像中的 `Ethernet 0/0/47`）放到一个特别的镜像用 `vlan` 中，并在本地交换机、核心交换机及观察端口所在交换机的中继端口上允许该 `vlan` 通过，并将远端交换机上的观察端口，置为 `access` 模式，放在镜像用 `vlan` 上即可。

## IPv6地址空间

在IPv6地址空间中，当前所用到的主要有4段（分别是全球单播地址、本地唯一单播地址、链路范围单播地址及多播地址），其它各段为IETF保留。

### 用到的4个地址段

- 2000::/3 -- 从 2000:: 到 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 全球单播地址空间, Global Unicast Address Space
- fc00::/7 -- 从 fc00:: 到 fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 本地唯一单播地址, Unique Local Unicast Address Space
- fe80::/10 -- 从 fe80:: 到 febf:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 链路上的单播地址空间, Link Scoped Unicast Address Space
- fec0::/10 -- 从 fec0:: 到 feff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF。 RFC38979 中弃用，先前的站点本地范围地址前缀。
- ff00::/8 -- 从 ff00:: 到 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 多播地址空间, Multicast, 该段中由IANA分配的地址在这里进行了登记： IPv6 Multicast Addresses

### 全部IPv6地址空间

- ::/8 -- 从 :: 到 00ff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 0100::/8 -- 从 0100:: 到 01ff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 0200::/7 -- 从 0200:: 到 03ff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 0400::/6 -- 从 0400:: 到 07ff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 0800::/5 -- 从 0800:: 到 0fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 1000::/4 -- 从 1000:: 到 1fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF
- 2000::/3 -- 从 2000:: 到 3fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
  - 为IETF所保留, Reserved by IETF

### 全球单播地址空间, Global Unicast Address Space

- 4000::/3 -- 从 4000:: 到 5fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- 6000::/3 -- 从 6000:: 到 7fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- 8000::/3 -- 从 8000:: 到 9fff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- a000::/3 -- 从 a000:: 到 bfff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- c000::/3 -- 从 c000:: 到 dfff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- e000::/4 -- 从 e000:: 到 efff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- f000::/5 -- 从 f000:: 到 f7ff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- f800::/6 -- 从 f800:: 到 fbff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- fc00::/7 -- 从 fc00:: 到 fdff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 本地唯一单播地址, Unique Local Unicast Address Space
- fe00::/9 -- 从 fe00:: 到 fe7f:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF
- fe80::/10 -- 从 fe80:: 到 febf:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 链路上的单播地址空间, Link Scoped Unicast Address Space
- fec0::/10 -- 从 fec0:: 到 feff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 为IETF所保留, Reserved by IETF。 RFC38979中弃用, 先前的站点本地范围地址前缀。
- ff00::/8 -- 从 ff00:: 到 ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff
- 多播地址空间, Multicast

来源: [IPv6 Address Space](#)

# GNS3 入门

Getting Started with GNS3

## 简介

世界范围内数十万网络工程师们，都在使用着GNS3，他们使用GNS3来对虚拟与真实网络进行模拟、配置、测试以及故障排除。GNS3允许在笔记本电脑上运行一个有着几台设备的小型拓扑，也可以在跨越多台服务器，以致云上运行有着众多设备的大型拓扑。

GNS3是开放源代码的自由软件，可从 <http://gns3.com> 下载到。

GNS3目前仍是活跃开发着，有着超过80万会员的持续增长的社区。加入到GNS3的社区，就意味着你加入了一个由众多学生、网络工程师、架构师以及其他那些已经下载了GNS3超过一百万次的这些人当中。

GNS3在世界上的众多公司中都有使用，包括很多财富500强的公司。

在诸如思科CCNA这样的认证考试中，GNS3可以用来备考，也可以对真实世界的网络部署进行测试和验证。GNS3最初的开发者，Jeremy Grossman，一开始就是为了帮助他的CCNP认证考试编写的这个软件。正是因为他的最初工作，今天我们才可以使用模拟器，而不需要去购买昂贵的硬件，来达到学习网络技术的目的。

10多年前，GNS3就已允许网络工程师将实体硬件设备进行虚拟。最初只能通过使用名为 Dynamips 的软件对思科设备进行虚拟，现在GNS3已经进化到支持多家网络厂商的众多设备了，包括思科虚拟交换机、思科ASAs、Brocade vRouters、Cumulus Linux交换机、Docker实例、HPE VSRs，以及多个Linux设备等等。在 <https://gns3.com/marketplace/appliances> 可以查看到这些支持的设备。

**注意** GNS3 已有超过10年历史。因此互联网上的一些信息已经过时或完全是错误的。希望这个文档能够回答一些疑问，并帮助你开始GNS3之旅。

**注意** GNS3 不止支持思科设备。虽然因为大多数网络工程师都对学习思科设备感兴趣而提到思科设备。在现今的GNS3中，许多其他的商业或是开源厂商都是支持的。如今，你可以对许多不同厂商设备的互操作性进行测试，甚至可以尝试那些采用了SDN、NFV、Linux以及Docker技术的深奥设置。

**建议：**如你使用的是旧版本的GNS3，那些建议你将其升级到当前的稳定版GNS3。

## 什么是 GNS3

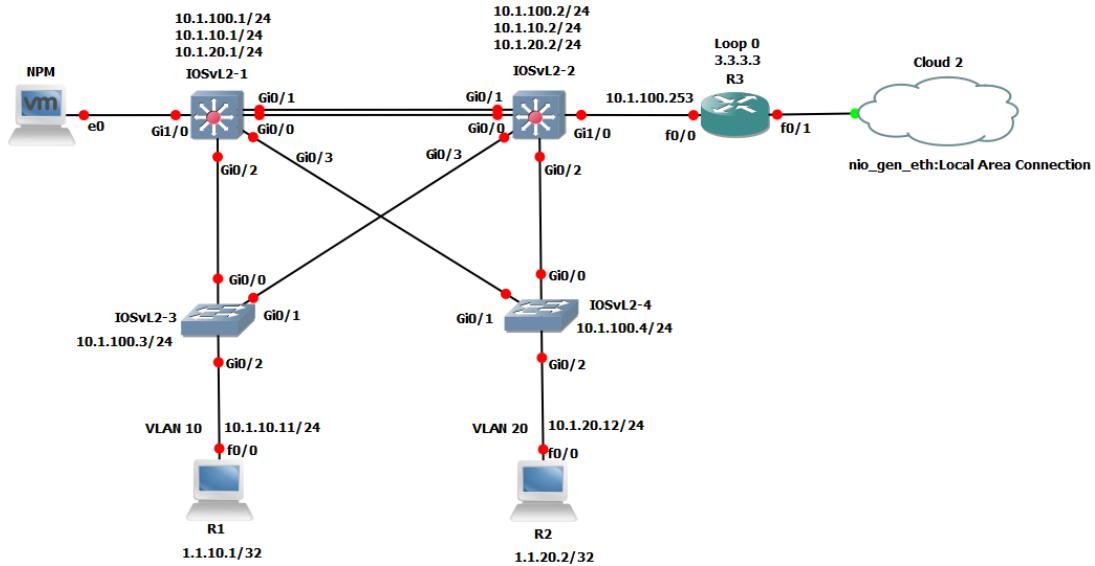
### 架构

GNS3是由两个软件组建构成的：

- GNS3一体软件（The GNS3-all-in-one software, GUI, 图形用户界面）
- GNS3的虚拟机（The GNS3 virtual machine, VM）

关于GNS3一体软件：

他是GNS3的客户端部分，同时是一个图形用户界面（graphical user interface, GUI）。在本地计算机（Windows, MAC, Linux）上安装这个一体软件，通过他创建出网络拓扑。所谓网络拓扑，就是我们常见的如下的屏幕截图：



## 服务器选项：

通过一体软件的图形界面客户端，在GNS3中创建网络拓扑时，所创建出的设备就需要经由一个服务器进程进行驻守并运行起来。关于GNS3软件的服务器部分，有少数几个选项：

1. 本地的GNS3服务器
2. 本地的GNS3虚拟机
3. 远程的GNS3虚拟机

本地GNS3服务器运行在安装GNS3一体软件的同一计算机上。比如在使用一台Windows PC时，那么GNS3 GUI与本地GNS3服务器，就都是以进程形式运行在Windows中的。那些诸如Dynamips这样的其他进程，也将运行在PC上：

若你决定运行GNS3虚拟机（推荐做法），那么既可以在你的PC上，使用诸如VMWare Workstation、Virtualbox 或 Hyper-V本地运行；或者也可以在运行了VMWare ESXi的服务器上远程运行 GNS3 VM，甚至还可以在云上远程运行。

**注意** 在不使用 GNS3 VM的情况下，同样可以使用GNS3。这在刚开始时是一个好办法，但这种设置带有局限性，而无法提供到与拓扑规模与设备支持相关的许多选项（You are able to use GNS3 without using the GNS3 VM. This is a good way to get started initially, but this setup is limited and does not provide as many choices with regards to topology size and devices supported）。在要创建更多先进GNS3拓扑，或者打算包含诸如思科VIRL设备（IOSvL2, IOSvL3, ASA v）, 或其他需要Qemu模拟器的设备时，就推荐使用GNS3 VM（而且通常是必要的）。

**注意** 仅使用GNS3一体软件来启动一个基本的GNS3拓扑，一旦那个拓扑可以运行，就可以参考附加文档来设置一个本地的GNS3虚拟机了。

## 关于仿真与模拟 (Emulation vs. Simulation)

GNS3同时支持仿真与模拟设备。

**仿真：** GNS3对某个设备的硬件进行模仿与仿真，同时你是在虚拟设备上运行着真实的镜像。比如可以从某台真实的物理思科路由器上拷出思科IOS镜像，进而再GNS3中对思科路由器进行仿真。

**模拟：**GNS3对某个设备的特性与功能进行模拟，比如交换机。你并未运行真正的操作系统（诸如思科IOS），而是运行的一台由GNS3所开发的模拟设备，就比如GNS3内建的2层交换机。

**注意** 模拟与仿真之间的界线，如今已不那么分明的。现在你可以运行真正的思科操作系统的思科VIRL镜像，这些镜像运行在标准的虚拟硬件上。GNS3则是仿真了这些VIRL镜像所需要的硬件。

**记住：**无需过多考虑模拟与仿真之间的区别，除非在下面几点情况下：

1. Dynamips 是一种较为陈旧的对思科硬件进行仿真的技术。它使用真实的思科IOS镜像。对于基本的CCNA类型拓扑，是可以的，但有着一些局限，比如只支持较旧的思科IOS版本（12.X），这些版本已不被思科支持或积极更新。
2. 推荐的在GNS3中使用的思科镜像，是思科VIRL（IOSv、IOSvL2、IOS-XRv、ASAv）的那些镜像。这些镜像是受思科支持并处于积极更新中的。所支持的这些镜像，是思科IOS（15.X）的当前发布，同时提供了最佳规模与用户体验（provide the best scale and user experience）。

## GNS3 对照

总会有人问到底那个软件是最好的。这个问题取决于个人喜好，而每种解决方案都有着同样的优势与不足（Questions often arise about which software is best. A lot of this is down to personal preference with all solutions providing some benefits and having some disadvantages）。

不堪的过往：如今网工的世界，远好于过去的日子！在以前，网络工程师们为了学习他们的CCNAs、CCNPs或CCIEs，就只有很少的选择：购买或租用物理的思科设备。

如今，在学习或测试网络时，就有了多种选择：

1. GNS3
2. 思科 Packet Tracer
3. 思科VIRL
4. 物理设备
5. 其他方案

## 关于GNS3

如同上面所提到的，GNS3是开放源代码的软件，可以自由下载使用。如想要看看GNS3的代码，那么其源代码在GitHub上可以获取到。GNS3团队希望你发现GNS3是有用、有益的，不过如果你不喜欢GNS3的某些东西，或者想要添加一些特性，就可以为GNS3贡献代码。加入到GNS3社区或志愿者，对代码进行检查或添加代码推荐。在一个有着超过80万成员的社区里，我们总能相互学习。

当然你可以使用其他选项。其中一些是免费的，一些则需要花钱。就使用那些适合于你的吧。在需要时，也可以结合使用多个方案。我们乐于见到今日百花齐放的局面，这有助于我们所有人提升与学习网络技能。

### 优势

- 自由软件
- 开放源码软件
- 没有月度或年度许可证费用
- 没有所支持设备数量上的限制（唯一的限制是你的硬件：CPU与内存）
- 支持多种交换选项（NM-ESW16以太网交换模块、IOU/IOL二层镜像、VIRL IOSvL2）
- 支持所有的VIRL镜像（IOSv、IOSvL2、IOS-XRv、CSR1000v、NX-OSv、ASAv）
- 支持多厂商环境

- 可带或不带管理程序运行 (Can be run with or without hypervisors)
- 支持免费或付费的管理程序 (Supports both free and paid hypervisors, Virtualbox, VMWare workstation, VMWare player, ESXi, Fusion)
- 提供了下载, 自由, 预先配置好的, 优化过的配置, 可简化部署
- Linux下的原生支持, 无需额外的虚拟软件
- 可免费使用多家厂商的软件
- 大型活跃的社区 (超过80万成员)

#### 不足

- 需要由用户提供思科的镜像 (从cisco.com下载, 或购买VIRL许可证, 或从物理设备上拷出来)
- 不是一个自包含的软件包, 而需要本地安装的软件 (GUI)
- 因为是本地安装, GNS3会受到你的PC设置和限制的影响 (防火墙、安装设置、公司笔记本电脑策略等等)

## Packet Tracer

思科 Packet Tracer 是一个思科给思科学院学员使用的官方产品, 对思科网络进行模拟。其并不对思科硬件进行仿真, 也不支持思科或其他厂商的真实镜像。

#### 优势

- 易于设置
- 支持思科路由器、交换机及PC的模拟
- 对于CCNA的学习是足够的
- 模拟多个设备与协议 (路由器、交换机、无线、RADIUS等等)
- 免费的 (需要在思科的 NetAcad网站上注册)

#### 不足

- 代码专有 --不是开源的
- 只模拟思科设备 (并未运行真正的思科镜像)
- 不是多厂商支持
- 无法与真实的物理设备进行集成
- 只能使用其开发者实现了的那些IOS命令。不是所有Packet Tracer中所模拟的平台上的命令, 都是可用的

## 思科VIRL

思科已经创建出另一个官方支持的网络模拟平台 - 思科虚拟互联网路由实验室 (Cisco Virtual Internet Routing Lab, VIRL)。与思科Packet Tracer相比, 这是一个强大得多的方案, 从而不仅能够在上面学习, 也可以对真实网络进行模拟。

**注意** 思科VIRL是一个与GNS3更为接近的产品, 除了学习思科技术外, 还允许网络工程师对真实世界的网络进行模拟。

#### 优势

- 支持思科路由器、交换机、防火墙及PC的模拟 (IOSv, IOSvL2, ASA v...)
- 对于CCNA、CCNP及CCIE的学习都不错
- 支持思科防火墙 (ASA v)
- 丰富的协议与特性支持: RP/CST+、Etherchannel、端口安区、MPLS、VRFs等等, 完整清单参考这里: <http://virl.cisco.com/work/>

- 支持最新版的思科IOS (15.X)

### 不足

- 不是自由软件。需要支付每年\$200的个人版VIRL订阅费
- 所支持的设备数量有限。使用个人版时，每个网络拓扑不能超过20个思科节点
- VIRL的配置较为复杂
- 资源密集（需要的内存与CPU很大很高）
- 需要虚拟软件（VMWare Workstation Player/Pro, Fusion 或 ESXi）
- 不支持VirtualBox
- 不具有多厂商支持 -- 只支持思科的网络设备

注意 GNS3支持所有的VIRL镜像。可将VIRL镜像倒入到GNS3中，并在每个拓扑下不加限制的进行使用（仅受限于你的硬件资源）。

## 设备支持

GNS3支持多厂商的众多设备，同时随时都有更多设备添加进来。GNS3 marketplace是查看当前支持设备清单的最佳位置：

<https://gns3.com/marketplace/appliances>

## 用例

GNS3最为著名的，是用作学习与教学的一个平台。为网络技术学生与网络工程师用于实践和准备诸如思科CCNA这样的厂商考试，GNS3被使用已有多年。

GNS3也可用于其他诸如概念验证与商业演示等用例。GNS3提供了一种容易的、极具成本优势的方式的新型软件，诸如网络管理或软件定义网络类软件。其实现了虚拟实验室环境下，而非必需特定物理设备，对多家厂商的互操作性进行测试。

在一台笔记本电脑上，就可以创建出整个的GNS3拓扑并加以运行。这就允许工程师把网络拓扑与软件，在路上给客户或其他人演示了。

使用GNS3的其他理由：

- 无需网络硬件，即可进行预部署测试的实时网络模拟：运行对网络硬件真实行为进行仿真的操作系统
- 在无风险的虚拟环境中对超过20家不同网络厂商进行测试：无需硬件就可以快速对多家硬件厂商进行运行和测试
- 为故障排除与概念验证（proof of concept, PoC）测试，创建动态网络地图：在构建网络之前，就对你的网络进行测试，以缩短获得生产网络运行起来的时间
- 可将GNS3与真实网络连接起来：通过将GNS3技术直接连接到真实网络，从而利用上既有硬件，并将当前实验室进行扩展
- 为网络认证考试训练目的，对GNS3中的网络拓扑与实验进行定制：GNS3是网络从业者寻求各种认证的最佳学习工具，而无需在家里搭建一个实验室

## 关于GNS3的版本

当前最新版的GNS3可在这里找到：<https://gns3.com/software>

GNS3的开发版本，可在这里找到：<https://github.com/GNS3/gns3-gui/releases>

**注意** 请使用最新的稳定版GNS3。只有在遇到问题或GNS3数据丢失时，才去使用开发中的发布版本。在准备考试或有某个最后期限的项目时，不要使用开发中的版本。

## GNS3帮助与支持

GNS3提供了多种获取帮助的途径，包括：

### 文档

可在这里访问到GNS3文档：<https://docs.gns3.net/>

### 社区

社区时获得帮助的最佳场所。加入到数以万计的GNS3用户与专家，进而互相帮助，让GNS3成为主流。

<http://gns3.com/community>

**什么可以做：** 在GNS3社区可以汇报程序缺陷（bugs）并提问

**什么不可以做：** 不要谈论违法的事，或违反GNS3用户政策的事。不要试图索取思科IOS镜像。不要分享思科IOS镜像。不要做任何试图盗版或违反法律的事。

### GNS3 Youtube 频道

可在David Bombal的GNS3频道上观看视频：

<https://www.youtube.com/playlist?list=PLhfrWILOoKPTPPv6ZiNHFM2FKAZ96f-r>

### GNS3课程

可在GNS3线上学院的GNS3课程注册，学习GNS3：

<http://academy.gns3.com/>

## 所支持的操作系统

GNS3支持以下操作系统：

- Windows 7 (64 bit)
- Windows 8 (64 bit)
- Windows 10 (64 bit)
- Windows Server 2012 (64 bit)
- Windows Server 2016 (64 bit)
- Mac OS X Marericks(version 10.9) and later
- Linux

其他可运行GNS3 VM的平台：

- ESXi
- 诸如packet.net这样的基于裸金属云提供商（Bare Metal Cloud based providers such as Packet.net）

## 支持的设备

### Supported Appliances

GNS3支持多种操作系统、设施及仿真器。[我该使用哪种仿真器](#)

## 需要使用GNS3虚拟机吗？

在使用Windows或Mac OS时，对于绝大多数模拟，都建议使用GNS3的虚拟机。GNS3开发团队花费了大量精力来创建出了一个轻量、可靠的，避开了在使用本地安装的GNS3时遇到过的多种常见问题的，创建GNS3拓扑的方法（The GNS3 development team have worked hard to create a lightweight, robust way of creating GNS3 topologies that avoids multiple common issues experienced when using a local install of GNS3）。这些问题就包括在Windows（不推荐）上原生运行VIRL时，缺少合适的Qemu支持的问题。

但如果只是要创建基本的，使用思科IOS路由器的GNS3拓扑，那么本地（Dynamips）安装就足够了。本地安装就是说，只安装GNS3的图形用户界面，而不使用GNS3的虚拟机。

本地安装是更简单的设置，但有许多限制，可作为GNS3之旅的起点。在逐渐适应了GNS3之后，就建议往GNS3的虚拟机设置过渡，获得GNS3的丰富选项与最佳优化。

**注意** 在Windows与Mac OS上使用GNS3的虚拟机。在Linux上原生运行GNS3时，GNS3虚拟机是可选的，而非必须的。

## 不支持或不推荐的

### ASA 8

**注意** ASA 8不被支持

在互联网上可以找到很多有关如何从物理设备提取ASA 8镜像，并在GNS3中进行使用的教程。这种办法在过去是唯一获取到ASA镜像的办法，但具体结果是随机的。在现代计算机和操作系统中这种情况变得更糟了。比如在Windows 10中运行ASA 8就有许多问题。

这样做的问题在于，所使用的镜像是为思科的特定设备制作的。Qemu可以对该硬件进行部分仿真，但特定于物理ASA的一些组建是缺失的。比如硬件ASA设备的硬件时钟就没有。ASA的内核有时可以将其你的计算机速率对其替代，但结果总是根据具体情况而定。

在同时运行多个ASAs会遇到各种问题。

## 带Qemu镜像的本地安装

### Local install with Qemu images

在Windows或 Mac OS上，GNS3不支持或不推荐本地GNS3安装下使用Qemu镜像。此时应该在GNS3虚拟机中使用Qemu镜像。

**Qemu 镜像示例** IOSvL2, IOSv, IOS-XRv, ASAv以及GNS3网站上所有可用的设备：

<https://gns3.com/marketplace/appliances>

## 关于复杂网络拓扑

在Windows或Mac OS上创建复杂网络拓扑时，推荐使用GNS3虚拟机。只有在创建简单GNS3拓扑时，才使用本地安装的GNS3。